

TDT4290 CUSTOMER DRIVEN PROJECT
DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE
NORWEGIAN UNIVERSITY OF SCIENCE AND TECHNOLOGY

ROCK CONCERT AUDIENCE AS A SCREEN



NTNU – Trondheim
Norwegian University of
Science and Technology

NOVEMBER 20, 2013

CUSTOMER

NETLIGHT AS

CUSTOMER REPRESENTATIVE

PEDER KONGELF

ADVISOR

ANH NGUYEN DUC

GROUP 5 MEMBERS

JAN BEDNARIK

TOMAS DOHNALEK

MILOS JOVAC

AGNETHE SORAA

Abstract

The aim of this report is to describe the 13 weeks long process of developing the mobile application called Digital Lighter as well as the management techniques used to control the four membered team. This project represents the main assignment for students of the course TDT4290 – Customer Driven Project held at Norwegian University of Science and Technology.

The main purpose of the resulting application is to entertain the audience of the music concerts by enabling them to actively participate the common experience by using their mobile devices. The users are instructed to hold their mobile devices and raise their hands while becoming "pixels" in the huge "human-screen" which can be used to display various imagery.

During planning the research focusing on similar products was done, the development methodology was chosen and the collaboration tools were established. The requirements that gave rise to the basic application concept were then elicited. The implementation itself is based on Google Android platform while Java SDK was used. Besides standard Android API the open source computer vision library OpenCV was also utilized for necessary image processing. Other important technologies this project builds on comprise of netlib networking library and NTP protocol used for synchronizing the devices.

The application was incrementally improved during development process and the resulting product achieved all the requirements the customer specified. The conclusion can also be drawn that even though this project represents proof of a concept task the final version of the application is working product which given certain limitations can be scaled up in order to be used for more real world purposes.

Preface

This report is one of the deliverables in the course TDT4290, Customer Driven Project. The course given by Department of Computer and Information Science at the The Norwegian University of Science and Technology, in the fall of 2013. Based on this project, the group's work will be evaluated and graded by the appropriate personnel in charge of the course.

We would like to thank our teaching supervisor, Anh Nguyen Duc, for regular inputs and guidance. We would also like to give a special thanks to our customer, Peder Kongelf from Netlight consulting, who has given us the opportunity to work on such an interesting project, and also for being enthusiastic and helpful throughout the whole course.

Contents

1	Introduction	1
1.1	Project purpose and concept	1
1.2	Project name	2
1.3	Goals	2
1.4	Stakeholders	4
1.5	Structure of report	5
1.6	Terminology	6
2	Preliminary studies	9
2.1	Similar projects	9
2.2	Software development methodology	12
2.3	Development technologies	14
2.4	Project management tools	16
2.5	Version control tool	17
2.6	Communication tools	19
2.7	Documentation tools	20
3	Planning	21
3.1	Work breakdown structure	21
3.2	Organization	23
3.3	Risk Management	24
3.4	Communication	26
3.5	Duration and workload	27
3.6	Project scope	28
3.7	Project plan	28
3.8	Test plan	35
4	Requirements	37
4.1	Description	37
4.2	Requirements	38
4.3	Summary	41
5	Software architecture	43
5.1	Logical view	43
5.2	Development view	45

5.3	Process view	47
5.4	Physical view	47
5.5	Scenarios	48
5.6	Used design patterns	48
6	Sprint 0	51
6.1	Sprint planning	51
6.2	Sprint goals	51
6.3	Testing	52
6.4	Occurring risks	52
6.5	Customer feedback	53
6.6	Retrospective	53
7	Sprint 1	55
7.1	Sprint planning	55
7.2	Preliminary studies	57
7.3	Sprint goals	59
7.4	Architecture	60
7.5	Occurring risks	64
7.6	Customer feedback	65
7.7	Retrospective	65
8	Sprint 2	69
8.1	Sprint planning	69
8.2	Sprint goals	70
8.3	Architecture	71
8.4	Implementation	72
8.5	Testing	72
8.6	Occurring risks	72
8.7	Customer feedback	73
8.8	Retrospective	73
9	Sprint 3	75
9.1	Sprint planning	75
9.2	Preliminary studies	76
9.3	Sprint goals	78
9.4	Architecture	79
9.5	Implementation	81
9.6	Testing	82
9.7	Occurring risks	82
9.8	Customer feedback	82
9.9	Retrospective	83
10	Sprint 4	85
10.1	Sprint planning	85
10.2	Sprint goals	87
10.3	Architecture	87

10.4 Testing	89
10.5 Occurring risks	90
10.6 Customer feedback	91
10.7 Retrospective	92
11 Sprint 5	95
11.1 Sprint planning	95
11.2 Sprint goals	96
11.3 Architecture	97
11.4 Implementation	100
11.5 Testing	104
11.6 Occurring risks	105
11.7 Customer feedback	107
11.8 Retrospective	108
12 Sprint 6	111
12.1 Sprint planning	111
12.2 Sprint goals	112
12.3 Implementation	113
12.4 Occurring risks	114
12.5 Customer feedback	114
12.6 Retrospective	115
13 Evaluation	117
13.1 Group evaluation	117
13.2 Project evaluation	125
13.3 Technologies and tools evaluation	128
14 Conclusion	133
14.1 Final product	133
14.2 Limitations	135
14.3 Further work	136
14.4 Summary	137
A Manager manual	139
A.1 Installation	139
A.2 Usage	139
B Client user manual	142
B.1 Installation	142
B.2 Usage	142
C Customer meetings	144
D Supervisor meetings	148
E Figures	151

List of Figures

1.1	Digital Lighter	2
1.2	Oganization chart	5
1.3	Grid example	7
2.1	Wham City Lights	10
2.2	Xylobands bracelet	11
2.3	Scrum	14
2.4	Trello	16
2.5	Gravity features	17
2.6	TargetProcess's board	18
3.1	The work breakdown structure	22
3.2	Scrum poker	31
3.3	Detailed sprint decomposition	33
3.4	Gantt Chart	34
4.1	Basic usage scenario	37
4.2	Use case diagram	38
4.3	Quality attributes.	41
5.1	Server class diagram	44
5.2	Client class diagram	45
5.3	Component diagram of server application	46
5.4	Component diagram of client application	46
5.5	Activity diagram of server application	47
5.6	Activity diagram of client application	47
5.7	Deployment diagram	48
7.1	Client-server architecture	60
7.2	Sprint 1 client class diagram	61
7.3	Sprint 1 server class diagram	62
7.4	Sprint 1 Communication	63
7.5	Deployment diagram	63
7.6	Android version distribution	65
7.7	Burn down chart for sprint 1	66

8.1	Command message with multiple instructions	72
8.2	Burn down chart for sprint 2	74
9.1	Goal for sprint 3	79
9.2	Sprint 3 device detection module class diagram	80
9.3	Deployment diagram	81
9.4	Sprint 3 activity diagram	81
9.5	Sprint 3 data flow diagram	82
9.6	Burn down chart for sprint 3	83
10.1	Traffic light	88
10.2	Sprint 4 device locating module	89
10.3	Sprint 4 media player module	90
10.4	Sprint 4 activity diagram	91
10.5	Burn down chart for sprint 4	92
11.1	Time sync class diagram	99
11.3	Deployment diagram	99
11.2	Sequence diagram	100
11.4	Tree algorithm detection example	102
11.5	New command message with multiple instructions.	103
11.6	DNS record inherit graph.	106
11.7	Illustration of communication modules specific libraries.	107
11.8	Burn down chart.	108
12.1	Marge 4x4 pixels	114
12.2	Burn down chart for sprint 6	115
13.1	Colors of screens in different light environments	131
14.1	Server UI	134
14.2	Client UI	135
A.1	Create server screen	139
A.2	Choose name of a server	140
A.3	Choosing media screen	140
A.4	Camera screen	141
B.1	Clients screen	142
E.1	Concept of application based on Digital Lighter	151

List of Tables

1.1	List of all stakeholders with their role and email contact	5
1.2	List of all chapters and short description	6
2.1	Confrontation between the agile and waterfall methodologies	13
3.1	Skills and previous experience	23
3.2	Assigned roles and their responsibilities	24
3.3	Handling risks	25
3.4	Initial time estimation table	27
3.5	Main Scrum features	29
6.1	User stories selected for Sprint 0	52
7.1	Implementation user stories selected for sprint 1	56
7.2	Documentation stories selected for sprint 1	57
7.3	Project management stories selected for sprint 1	58
8.1	Implementation user stories selected for sprint 2	70
8.2	Documentation stories selected for sprint 2	70
8.3	Project management stories selected for sprint 2	71
9.1	Implementation user stories selected for sprint 3	76
9.2	Documentation stories selected for sprint 3	76
9.3	Project management stories selected for sprint 3	77
10.1	Implementation user stories selected for sprint 4	86
10.2	Documentation stories selected for sprint 4	86
10.3	Project management stories selected for sprint 4	87
11.1	Implementation user stories selected for sprint 5	96
11.2	Documentation stories selected for sprint 5	96
11.3	Project management stories selected for sprint 5	97
11.4	Test devices in sprint 5	104
12.1	Implementation user stories selected for sprint 6	112
12.2	Documentation stories selected for sprint 6	112
12.3	Project management stories selected for sprint 6	113

1 | Introduction

This report describes the development of the innovative mobile system for entertaining the visitors of the common social events such as concerts or exhibitions. The system comprises of two mobile applications, and it is titled Digital Lighter. This project is the part of the course TDT4290 Customer Driven Project. Customer Driven Project is held at The Norwegian University of Science and Technology accounted for 15 credits. The course aims to give the students a real experience with customers in a relevant IT project, and shall give the students a feel of managing a project in a group. The course will provide realistic experience in both report writing and product development driven by a customer.

1.1 Project purpose and concept

The project was proposed by the consulting company Netlight represented by Peder Kongelf, who serves the role of the customer driving the project and communicating with the development team. Peder Kongelf together with company Netlight will be introduced in Section 1.4.

The underlying task of this project is to develop the mobile application. The application will mainly be used as the mean to entertain the visitors of the music concerts, and other social events. This is allowing them to participate exclusively and share an experience by simply holding their mobile devices in the air. Each single visitor would for a short while become a pixel in the giant human screen made of the audience.

The idea is based on the assumption that the overwhelming majority of people who would prospectively visit the concert possess a smartphone capable of running simple client application controlling the device's screen. At some point the whole audience would be instructed to raise their hands with the mobile devices' screens aimed towards the stage. The camera together with the big screen displaying the crowd would be placed on the stage so that the audience would be able to see their own reflection.

There would be another (server) application which would serve the purpose of controlling the camera, and detecting the audience devices. Once all of the devices are recognized, the server application would start sending the signals to each user device specifying which color it should display for the specified time frame. As a result it would be possible to display various imagery ranging from pictures to animations and videos on this big screen, which is consisting of visitors' smartphones aka pixels.

As the music concerts represent the events where people seeking for the enjoyment gather it is fairly easy to convince the visitors to join this shared experience. It can be widely seen now days that the concert managers, or even the musicians themselves are attempting to introduce specific added value to the concert by instructing the crowd to sing, dance or wave hands synchronously, in order to strengthen the whole musical experience. The satisfied visitors tend to visit the next event again, which is of course where the concert manager aims.

Therefore it can be said there is never enough of good and creative ideas about the crowd experience and joy catalyst. The Digital Lighter is one of them and as will be specified in Section 2.1 the market currently lacks such a product as this is truly innovative concept.

1.2 Project name

The customer wants a product to make the audience as a screen on a rock concert. The team has decided to name the product *Digital Lighter*. This name was settled, because this product digitalizes the concept from "the old days", where the crowd at a concert held up lighters, to create a special atmosphere.



Figure 1.1: Digital Lighter

1.3 Goals

Goals are statements that describe what the project will accomplish. The very first step in all projects is to define goals. It is important to spend sufficient time on this step, or else it can lead to unsuccessful project completion in many different ways.

1.3.1 Project goals

Finish the project within the scheduled timetable One of the goals is to finish the project within the given time frame. In this particular case it means to deliver both product and report until 21th of November, when the final presentation take place.

Finish the project within the specified requirements Another goal is to make sure customer's needs described in requirements Chapter 4.2 are met. This could be done simply by finishing the project with the specifics the customer really wanted. The best way to solidify this is to verify accomplishment by customer hand off and close down.

1.3.2 Group goals

Receive a good grade Getting a good grade is important because it says something about what you have learned in this course. This is relevant in terms of what job we might get when we have graduated. It also shows an employer that we can work hard.

Receive a good recommendation from customer/supervisor If we work really hard, and impress the supervisor and the customer it would be nice to get a recommendation. This would be a bonus for our CV.

1.3.3 Personal goals

Agnethe My personal goal is to learn about group dynamics and collaboration. Another goal is to learn more about writing reports, because it is important to get some experience before writing for the masters degree. One of my goals is to deliver a product the customer is pleased with. It is also a great chance that I might work as an IT-consultant, and I want to learn as much as possible about this profession. I also want to learn as much as possible about android and mobile development. My last goal is to get better at holding presentations, because there will always be another presentation to hold.

Jan This class was suggested to me by a few students who have already passed it. As from what I have learned it should be challenging and the students should receive a lot of experience while working on a project. From this point of view my personal goal is to improve my soft-skills regarding team collaboration as well as the technical skills, mostly the Java/Android programming. While I study at the Computer graphics department on my home university the image processing part of this project is also of the great value for me. I also expect to get the proper feedback from both the supervisor and the customer so I could learn from the mistakes or wrong approaches.

Milos I had the opportunity to work on actual projects before, but never as a part of a team. My primary goal is to determine how to cooperate and work with others in achieving mutual goals. I want to learn about agile software development in practice. To find out how to separate tasks, how to manage an ongoing project, and how to be a valuable part of the team. I didn't expect to gain any technical knowledge of this course, but after discovering the project assignment I definitely want to learn about image processing. I expected multicultural team and that report has to be written in English. Therefore I wanted to profound my knowledge of English Language.

Tomas I have chosen NTNU university for my exchange studies because of this class. I was expecting to improve my soft-skills, software engineering knowledge and also my written and spoken English and to be treated like in real project. As this class has already started, I can add to my personal goals also learning Java programming language, Android platform development and improving my skills concerning image processing.

1.4 Stakeholders

The stakeholders in this project is any person or organization, which has some interest in, or is affected by this project. Together they construct the different restrictions and goals for the project. It is possible see the organization chart in Figure 1.2.

Customer Peder Kongelf is the customer. He works as an consultant for Netlight AS, which is a consulting company with field of expertise within IT management, IT governance, IT-strategy, IT-organization and IT-research. He has experiences with working as a consultant, working in scrum, and also with mobile development. He is able to provide the team guidance and support with various issues during the project, and even though he is the customer, he teaches the team a lot as well.

Supervisor Anh Nguyen Duc is the supervisor assigned to this project. The main responsibility of the team's supervisor is to overview the whole project, discuss the work done, and also to discuss the future plans.

Of course the interest from the university in general is to provide some real experience to the students, and prepare them for the business world.

Development team The development team's role is to meet all requirements presented by the customer and by The Department of Computer Science and Information Technology. The team is responsible for the development of the project, and the team is also responsible for writing all the documentation for this course.

It is possible to see a list of stakeholders, with their contact information in Table 1.1.

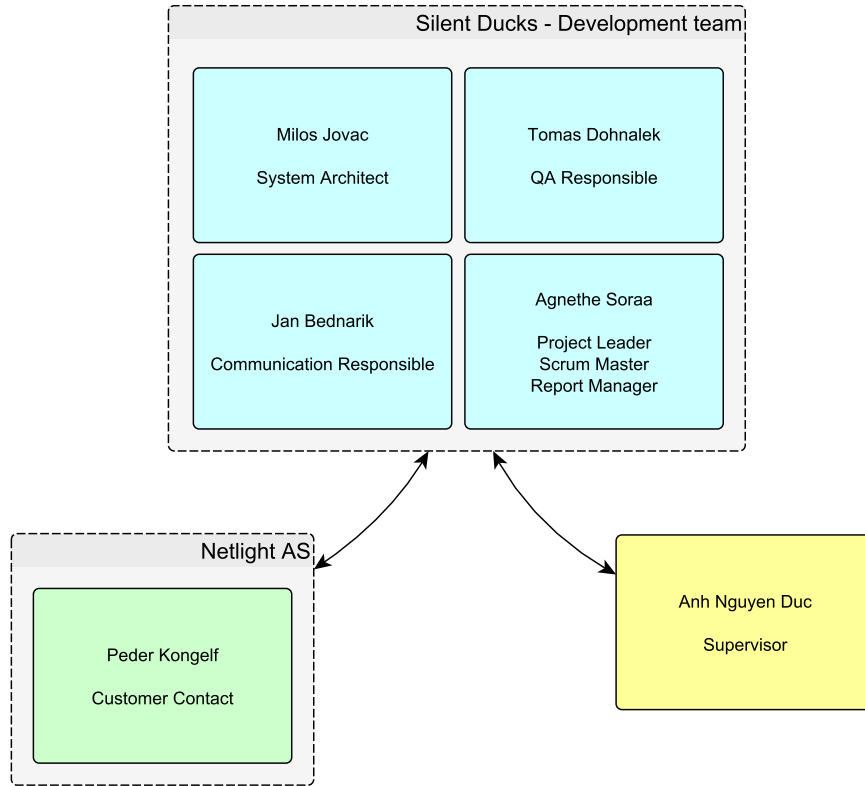


Figure 1.2: Chart depicting team organization and its stakeholders

Table 1.1: List of all stakeholders with their role and email contact

Person	Email	Role
Peder Kongelf	peder.kongelf@gmail.com	Customer
Anh Nguyen Duc	anhn@idi.ntnu.no	Supervisor
Milos Jovac	milosjovac@gmail.com	Team member
Jan Bednarik	ja.bedna1@gmail.com	Team member
Agnethe Soraa	agnethes0raa@gmail.com	Team member
Tomas Dohnalek	dohnto@gmail.com	Team member

1.5 Structure of report

It is possible to see the structure of this report in Table 1.2.

Table 1.2: List of all chapters and short description

Chapter	Description
Chapter 1	The <i>introduction</i> chapter introduces the problem, and introduces the members and stakeholders of this project. This chapter also explains the motivation and goals.
Chapter 2	The <i>preliminary studies</i> chapter describes the work and research done. This chapter describes alternative approaches and solutions, and an evaluation of these.
Chapter 3	The <i>planning</i> chapter describes the project plan, the organization, and planned workload.
Chapter 4	The <i>requirements</i> chapter describes the requirements, both functional and non-functional. It also describes use cases for the system.
Chapter 5	The <i>architecture</i> chapter explains the structure of the system, and how it is put together.
Chapter 6–12	In the sprint chapters the reader can see how the product has developed during the time of the project. This includes planning, architecture, the implementation, testing and evaluation of the sprint.
Chapter 13	<i>Evaluation</i> chapter includes the team dynamics and risk evaluation. Last but not least it describes an evaluation of the customer, supervisor and of course the task.
Chapter 14	<i>Conclusion</i> chapter sums up the project, and discusses the solution, and looks at further work.
Appendix	The appendix contains user manual, installation guide and examples of meeting minutes.

1.6 Terminology

After a few meetings with the supervisor and customer, it was settled that terminology must be presented in report to prevent misunderstandings, especially since both customer and supervisor used different terminology. Below there is listed product related terminology that will be used in further chapters of this report.

Client user is a participant of the concert who wants to actively take part in the show, and is able to download the client application.

Server user or simply **Manager** is a person who controls what media should be played on the screen made from mobile phone's of users. The Server user can also manage other general settings.

End user is either manager or end client user.

DigitalLighter is the final product. It is containing of two separate applications *client* and *server*.

Client or **Client side** is an application controlled by client user, and provides an opportunity to be part of the screen by displaying figures on his mobile.

Server or **Server side** is and application controlled by manager and provides an opportunity to change the media displayed on screen.

Pixel is a client user's mobile phone under control of the manager.

Grid is a matrix used as an abstraction for mapping mobile devices into a single pixel of an image. The image of specific size requires the grid of specific suitable size. For example imagine, that on the screen made of mobile devices, a single letter *D* will be played. This letter can be represented as an image displayed in Figure 1.3a with size 4x4 pixels. A grid is a 4x4 matrix created according to the image as shown in Figure 1.3b.

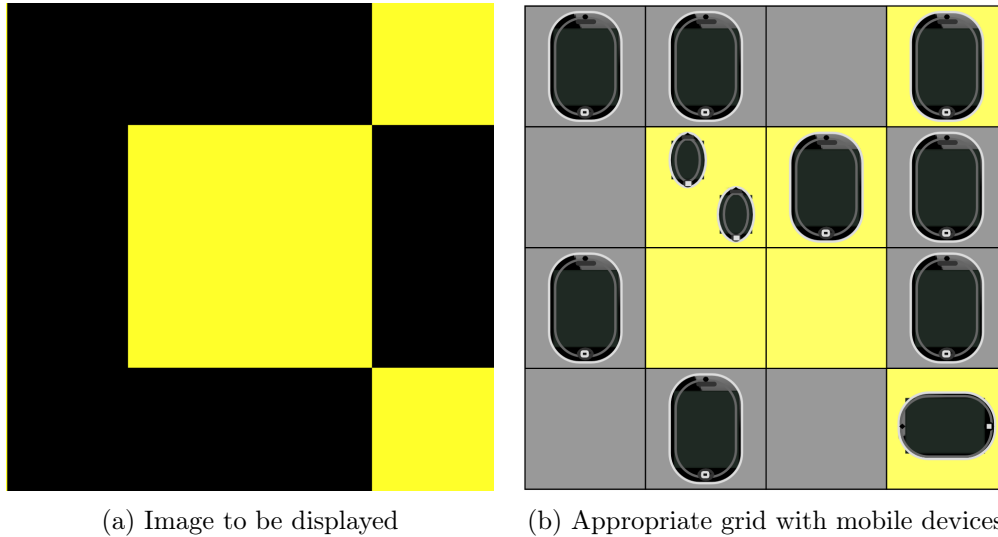


Figure 1.3: Grid example

2 | Preliminary studies

This chapter is devoted to describing the outcomes of the preliminary research focusing on the similar already existing projects and technologies that could be utilized. There is also mentioned choice of software development methodology and other related tools.

2.1 Similar projects

No other services using exactly the same technical solution as compared to our product was found. Nevertheless a few similar entertainment services that aim to amuse the music concert audience already exist. These services basically encourage the audience to use either their smartphones or other device as the source of light that helps to create impressive and colorful show. A description of similar projects can be found below.

2.1.1 Wham City Lights

Description

The Wham City Lights is the mobile application developed by the one year old Baltimore based start-up ¹. As far as the Digital Lighter is concerned the Wham City Light service seems to be the closest solution as it works with the users' smart-phones and uses them as a source of the light and even the sound.

Users attending the given concert only need to download and install the free application, start it and then hold their smart-phones in the air. The application then controls the color displayed on the smart-phone screen, the camera flashes and the sound going out of the speakers and it creates the spatial sounds capes and lighting designs. What is more the application does not require the network connection as the instructions are modulated into the ultrasonic inaudible signal that is being continuously transmitted during the performance.

Using this innovative technique all the smart-phones can be effectively synchronized so that the changing lights and imagery on the phone follows and complete the music performance. Nevertheless the so-called light show must be pre-programmed and this system cannot determine the location of the mobile devices.

¹<http://whamcitylights.com/>

The demonstration of the real world performance of the application and the concept for that matter can be seen in the official Wham City Lights marketing video² and in Figure 2.1.



Figure 2.1: Wham City Lights

Impact

The Wham City Lights application or its derivatives have been so far used on relatively many music events of greater importance where several thousands of people were present and used the application. To list a few:

- America's Got Talent 2013
- CMA Music Festival 2013
- Intel sales conference 2013
- Billboard Music Awards 2013

Availability

The application is available for the devices using Android or iOS operating systems and it can be downloaded from the Google Play³ and App Store⁴ respectively. The developer nevertheless offers the customers a concert specific applications with relevant user interface built-in. As for the light show itself, programming part can be done either by the customer or the developer.

Relation to our project

Like our project the Wham City Lights application builds upon users' smart-phones that are remotely controlled in order to display intended imagery.

²<http://www.youtube.com/watch?v=faJ1Av5kBCE>

³<https://play.google.com/store/apps/details?id=com.whamcitylights>

⁴<https://itunes.apple.com/us/app/wham-city-lights/id580034697?mt=8>

Advantages The devices attending the light show do not need the network connection and the displayed imagery is well synchronized with the music.

Disadvantages The location of the mobile devices cannot be determined and the different processing speed of the devices causes incorrect synchronization.

2.1.2 Xylobands

So called Xyloband⁵ is the invention of the company RB Concepts Limited. The device itself consists of the plastic bracelet that includes the LED diodes of the different colors and the micro-controller. The bracelets are controlled by the radio signal being broadcast from the stage and as a result they change their colors, flashes and in general create colorful imagery. Such a bracelet can be seen in Figure 2.2.



Figure 2.2: Xylobands bracelet

Impact

Xylobands received great fame thanks to the well known British rock band Coldplay which used them during their 2012 tour.

Availability

The bracelets can be ordered on the official Xylobands website.

⁵<http://xylobands.com/>

Relation to our project

The Xylobands product is somewhat different from the Digital Lighter as it does not utilize the users' mobile devices. The similarity can be found in the way the bracelets are synchronized using wireless signal.

Advantages The users do not need to bring their own mobile device and all of the audience has the possibility to become the part of the light show.

Disadvantages The location of the single participants cannot be determined.

2.2 Software development methodology

A software development methodology is a framework used for structuring, planning and controlling the process of developing an information systems [8]. In the real world ICT companies utilize quite a few various software development methodologies that differ mainly in the overall size of the project they are meant to manage, life cycle phases design and the level of certainty that initial requirements would not change.

The team was choosing between two main general approaches, the traditional software development process or iterative-development process [7]. The first one mentioned mainly comprise of so called waterfall based models. In theory the waterfall model thoroughly describes the software life cycle phases as a strict sequence that should result in a functional product. The real world experiences however show that this model is not suitable for the projects where the requirements are unclear in the preliminary phases and likely to change a lot during the development process [3].

It is impossible to have the crystal clear image of the resulting product in the beginning[7] which applies to Digital Lighter project very well. This fact prevents both the customer and the development team to specify all of the requirements beforehand. On the contrary, the team agreed with the customer it is more desirable to employ the methodology which enables the development team to add the new features in an iterative manner and to present functional pieces of code (prototypes) to the customer regularly so that the right course of the project can be maintained.

Therefore it is more reasonable to rely on the approaches from the latter group, the approaches based on iterative-development process. In general these models are designed in the way that they enable the development team to continuously verify and validate the product and detect the process problems early and as discussed with the customer this is something both development team and the customer will benefit from. It has been decided by the team to use the agile development models as described in Section 2.2.1 and specifically the Scrum model covered in Section 2.2.2.

2.2.1 Agile methodologies

Agile methodologies were created as the contrary to the traditional software development techniques based on the waterfall model. According to the Agile Manifesto [1] created by the well established non profit organization Agile Alliance⁶ the agile approach basically mainly focus on team-customer communication, working software, tight collaboration and ability to respond to changes. As stated in Table 2.1 these and other features agile methodologies provide suite the Digital Lighter the best as the customer stated he requires to be closely involved in the development process.

Table 2.1: Confrontation between the agile and waterfall methodologies and their applicability in Digital Lighter project

Digital Lighter properties	Agile methodologies	Waterfall
Small sized team	✓	✓
Quick development	✓	✗
Prototyping	✓	✗
Frequent product releases	✓	✗
Strong customer involvement	✓	✗
Focus on documentation	✗	✓

It can be noticed that the Digital Lighter project requires the team to make a great effort writing the documentation which does not particularly correspond with the agile approach. The issue is that this is still the school project and the extensive documentation is a part of the task. Still as far as the properties of the Digital Lighter project are concerned the advantages of the agile approach are prevailing.

2.2.2 Scrum

Scrum is one of the well established agile methodologies and it has been embraced by the team for several reasons. The customer had good experience using Scrum before and he strongly encouraged the team to use this methodology. Moreover one of the team members has the real world experience working in Scrum. Even though it requires "learning-by-doing" approach to learn Scrum in this case all of the team members were committed to learn new methodology as it seems to be widely used in the commercial field and therefore the team members would get the valuable experience.

It must be mentioned the team does not use the pure Scrum methodology but rather adjusts it to its specific needs as the team is not able to fulfill all of the procedures Scrum dictates. This approach is referred to as a *Scrumbut* [13] which comes from the self explanatory claim "We use scrum, but ...".

⁶<http://www.agilealliance.org/>

The Scrum framework consists of the roles, processes and artifacts [13]. The team’s approach to the role assigning and fulfilling the processes is described in Section 3.7.1. The overall concept of Scrum can be seen on Figure 2.3.

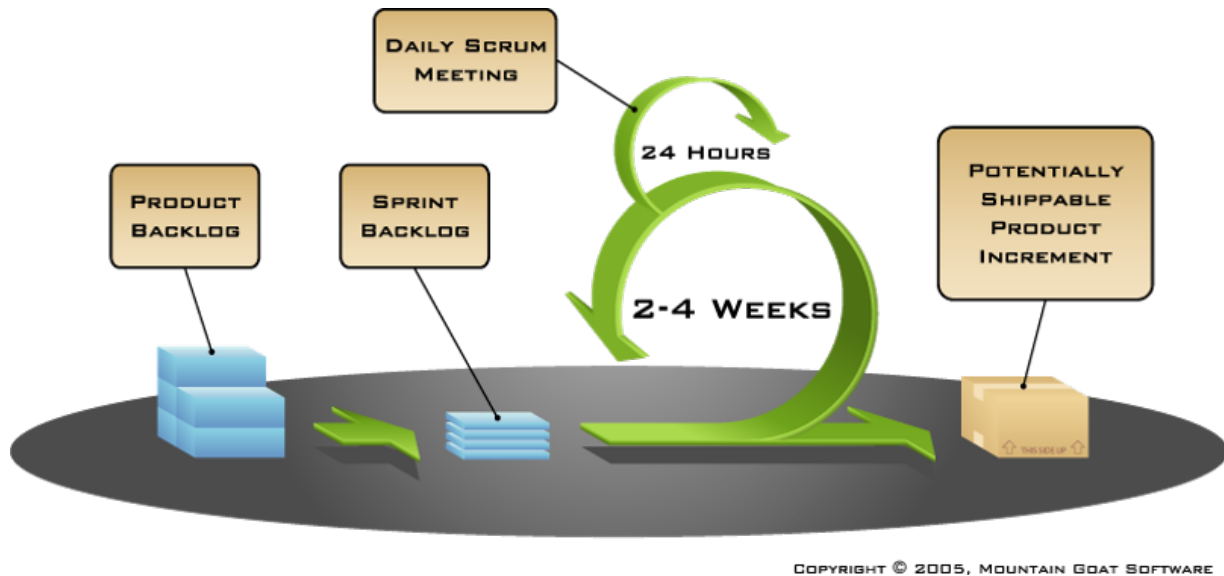


Figure 2.3: Scrum

Scrum related terms

User story is a short text of user’s language of a system that describes user’s interaction with the system. User stories are actually narrative texts that describe an interaction of the user and the system, focusing on the value a user gains from the system.

Epic is a large *user story* that will need to be broken down into smaller stories. It is too big to fit into a sprint, or it is unknown as to whether it is too big to fit into a sprint or not.

Task If user story is too complex, it can be divided into several tasks. Task is then basic unit of problem division.

Milestone is an event that receives special attention. It can add significant value to project scheduling by helping team to more accurately determine whether or not the project is on schedule.

2.3 Development technologies

According to the customer’s requirements the product must be executable on the mobile devices and it should utilize the image processing technologies as well. Therefore there was a need to make

two main decisions: which mobile platform to prefer and which existing image processing libraries to use. Both of the decisions are explained in greater detail in sections 2.3.1 and 2.3.2.

2.3.1 Android

The mobile devices market is currently flooded with smart-phones and tablets running on dozen of different platforms. Since Android platform has major position on the mobile market, the choice was relatively straightforward. Considering the survey carried on by the well established analyst, the company IDC, as of Q2 2013 Google's Android OS held almost 80 % of the market share⁷ leaving iOS, Windows Phone, BlackBerry and others well behind. What is more all of the members of our team posses the Android based smart-phone thus it has been decided by the team and customer to choose the Android platform.

Android SDK

Google provides the Android developers with all the necessary tools and API's through the open source and free SDK⁸. While developing under SDK the Java programming language together with the Android event-driven architecture is used.

Android NDK

Through the NDK⁹ Google provides the toolset allowing the programmers parts of the application using native-code languages such as C and C++. Use of NDK was discussed resulting in adopting it only for computationally demanding modules such as the image processing since the applications based on already compiled code tend to run faster. Nevertheless scaling down the problem is allowed by the customer and thus the use of Java perfectly suffices for our purposes. Provided the customer requires to scale the problem up, there is a possibility to switch to using NDK anytime.

2.3.2 OpenCV

As far as the image processing is considered the outcome of our preliminary research is as follows. Currently there exist a few open source and free libraries generally focusing on working with the multimedia. OpenCV library is probably the most well-known library providing a broad range of image processing functions. It is written and meant to be used in conjunction with C++ but the possibility to use Java already exists too. The support for using OpenCV on Android platform is provided through the OpenCV4Android SDK that can be obtained from the official OpenCV website¹⁰ and used for free. So far the toolkit is considered sufficient for the needs but

⁷<http://www.idc.com/getdoc.jsp?containerId=prUS24257413>

⁸Software Development Kit

⁹Native Development Kit

¹⁰<http://docs.opencv.org/>

if any problems occur, it might be necessary to switch to the combination of OpenCV, C++ and Android NDK.

2.3.3 TestFlight

TestFlight¹¹ is a free web service for mobile developers that provides the tools to easily deploy and distribute the mobile application. What is more, TestFlight offers the SDK for developers so that it can be integrated directly into the applications. The advantage of using the TestFlight is the fact that the developer is able to track the bugs, crashes and users' or testers' feedback. The customer suggested the team to use such a service. Quick research was made and two different services were found: TestFlight and HockeyApp. Unlike HockeyApp the TestFlight is a free service therefore it was decided by the team that this service will be adopted.

2.4 Project management tools

Since Scrum methodology was chosen, there was a need to find an appropriate management tool supporting Scrum. There exist many possible tools, but most of them are charged from certain number of team members. Here are presented tools that have been used by the team for a testing purposes or in real development process.

2.4.1 Trello

Trello is a collaboration tool¹² that can organize tasks into various boards, it shows what task has been assigned to whom and in what stage the task is, as shown on Figure 2.4. Even though Trello supports a lot of features, it is not originally designed for Scrum use and there is no support of *Epics*, *Burn-down chart* and time tracking. Therefore after short discussion it was decided by the team that Trello is not suitable for this purpose.

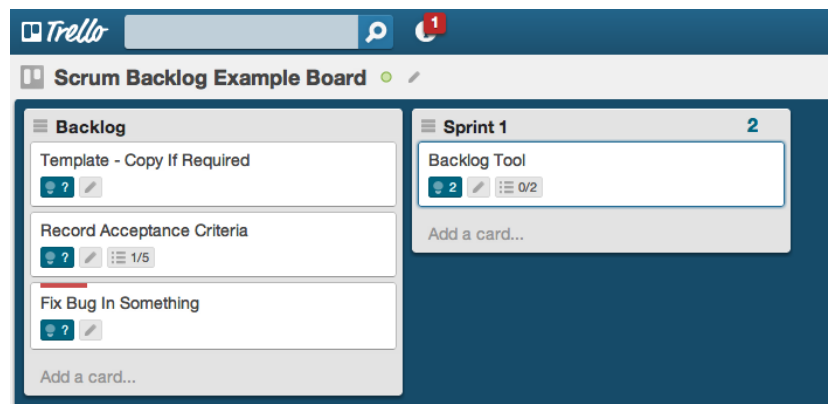


Figure 2.4: Trello

¹¹<https://testflightapp.com/>

¹²<https://trello.com/>

2.4.2 Gravity

Gravity is a simple project management tool¹³ currently in beta phase. It supports splitting stories into tasks and also automatically calculates *Burn-down chart*. Last but not least it is free of charge for 5 team members, it supports issue tracking and labels, which can be used as a compensation for *Epics*. On the other hand it does not support time tracking. You can see a picture of Gravity's features in Figure 2.5. After a while it showed up that Gravity has some bugs since it is in Beta version and it also lack a time tracking feature.

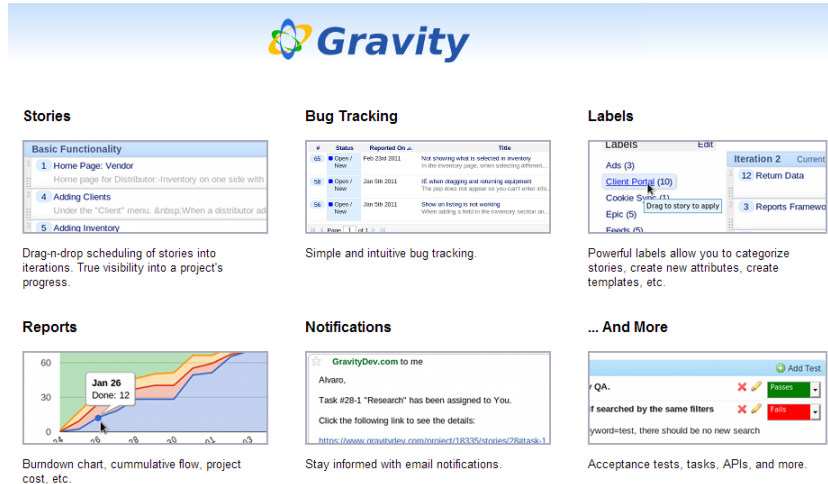


Figure 2.5: Gravity features

2.4.3 Target Process

Target Process is a project management tool¹⁴ that supports different processes such as Scrum, Kanban and own process. To its main features belongs role distinguishing, different graphs, such as *Burn-down chart*, time estimation, time tracking of both stories and its tasks, drag and drop prioritizing which makes it easy to use from customer side and different kinds of boards (diverse point of views on project). On the contrary it also does not support *Epics* and it is much more complicated to use than other tools introduced. You can see example of board in TargetProcess3 in Figure 2.6. Despite its complexity it has been decided by the team to use the TargetProcess as it suited the team's needs the best.

2.5 Version control tool

Version control systems (VCS) are usually stand-alone applications but they can be integrated into other applications. These systems usually allow users to browse previous versions of content. It has been decided to use VCS both for source code of applications and for the project report.

¹³www.gravitydev.com

¹⁴<http://www.targetprocess.com/>

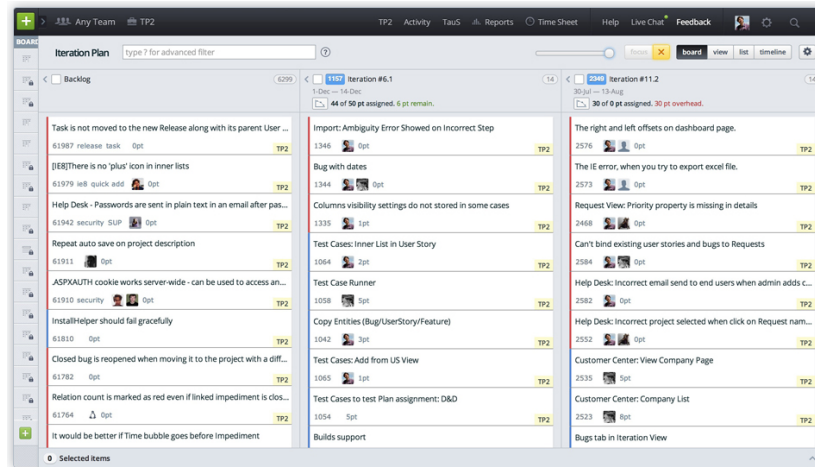


Figure 2.6: TargetProcess's board

As every member of the team had experience with different version control systems, they become the candidates. As members of the team work both on Microsoft Windows and GNU/Linux, it is necessary that chosen VCS supports clients in both platforms. Another criteria was support of online code repository free of charge.

2.5.1 Subversion

Subversion or SVN is an open source centralized VCS and has many clients across different platforms. SVN is supported by Google Code¹⁵ repository. SVN supports atomic commits, branching and more. Although this tool provides the user all of the necessary functionality it has been agreed by the team not to use it as the customer suggested to use the other tool.

2.5.2 Git

Git on the other hand is distributed VCS and offers immediate offline operations. Projects like Linux kernel¹⁶ and Glibc¹⁷ are using Git as a VCS. Although Git was created by Linus Torvalds, there are Windows clients available. Git is supported by GitHub¹⁸, what is web-based hosting service for VCS. Git has been adopted as team's version control system due to the customer suggestion, the distribution feature and general good experience with Github. You can find Digital Lighter's repository on <https://github.com/dohnto/CDP> page.

¹⁵<http://code.google.com/>

¹⁶<https://www.kernel.org/>

¹⁷<https://www.gnu.org/software/libc/>

¹⁸<https://github.com/>

2.6 Communication tools

Regular communication with stakeholders and creation of a positive understanding is one of the essential parts of the successful project that can help build effective long-term relationships all parties benefit from. Not only the development team should meet the customer and the supervisor on regular basis but there is also the need for easy distribution and sharing of all kinds of the documents. As the time resources of all parties are limited the right collaboration tools enabling seamless and fast communication must be established.

There are certain standard tools that are widely used as a mean of exchanging both textual information and data files. The mainstream well known and effective tools include standard e-mail for messaging, tool for video conferencing, cloud system for data sharing and/or calendar for planning the meetings. Certain platforms providing all necessary tools exist, one can choose among solutions made by Microsoft, Apple, Google and Facebook to list a few most popular.

2.6.1 Email

Standard email messages will serve the purpose of the communication mean.

2.6.2 Google Drive

In order to provide the customer with the convenient overview of the work done all relevant documents including the meeting minutes, private notes taken during the meetings and the project report are stored on the Google drive cloud storage service and shared with the customer.

2.6.3 Google Calendar

The shared Google Calendar was established to keep track of arranged meetings.

2.6.4 Skype

The Skype is used as a videoconferencing tool as it is considered to be a software of a good quality and the customer disposes of the Skype account.

2.6.5 Facebook group

For the purpose of internal communication and data exchange among the team members a Facebook group was chosen.

2.7 Documentation tools

There are basically two options for documentation writing. It is either possible to use WYSIWYG editor such as Microsoft Word, LibreOffice or document preparation system based on markup language such as $\text{\LaTeX}$.

2.7.1 $\text{\LaTeX}$

It was strongly suggested by the customer to use the latter one as the reports written using $\text{\LaTeX}$ tend to make more professional impression.

3 | Planning

In this chapter the project is planned in general. Currently the team is not in a position where it would be able to create a detailed plan for each sprint, so the following sections serve the purpose of a rough plan for the whole project. After finishing a preliminary studies some decisions about methodology tools, technology, collaboration, and a rough project plan were made. Planning increases the efficiency and it reduces the risks involved. It also facilitates proper coordination within a development team, and this will help us maintain a good control. Planning will help the team to achieve the goals by using the available time and resources, and of course planning helps in decision making

3.1 Work breakdown structure

The whole project was thoroughly examined considering the stated scope and the goals and it was decomposed into eight main distinctive modules:

1. *Pre-study* includes research of similar projects and tools and technologies, which can be used.
2. *Requirements elicitation* comprises its negotiation, creating use diagrams, etc.
3. *Project planning* includes various topics such as defining milestones, assigning of roles, risk management, etc.
4. *Application design* comprises not only designing the architecture of final product, but also designing GUI and communication protocol.
5. *Application implementation* contains all implementation of both server and client.
6. *Testing and evaluation* includes test data collection and actual miscellaneous testing and its evaluation.
7. *Project management* contains not only all meetings, but also attending lectures.
8. *Documentation* includes creating various documents concerning the product, meetings or this report.

The main modules can be further decomposed into smaller actions. The graphical representation of the whole work breakdown structure can be seen in Figure 3.1.

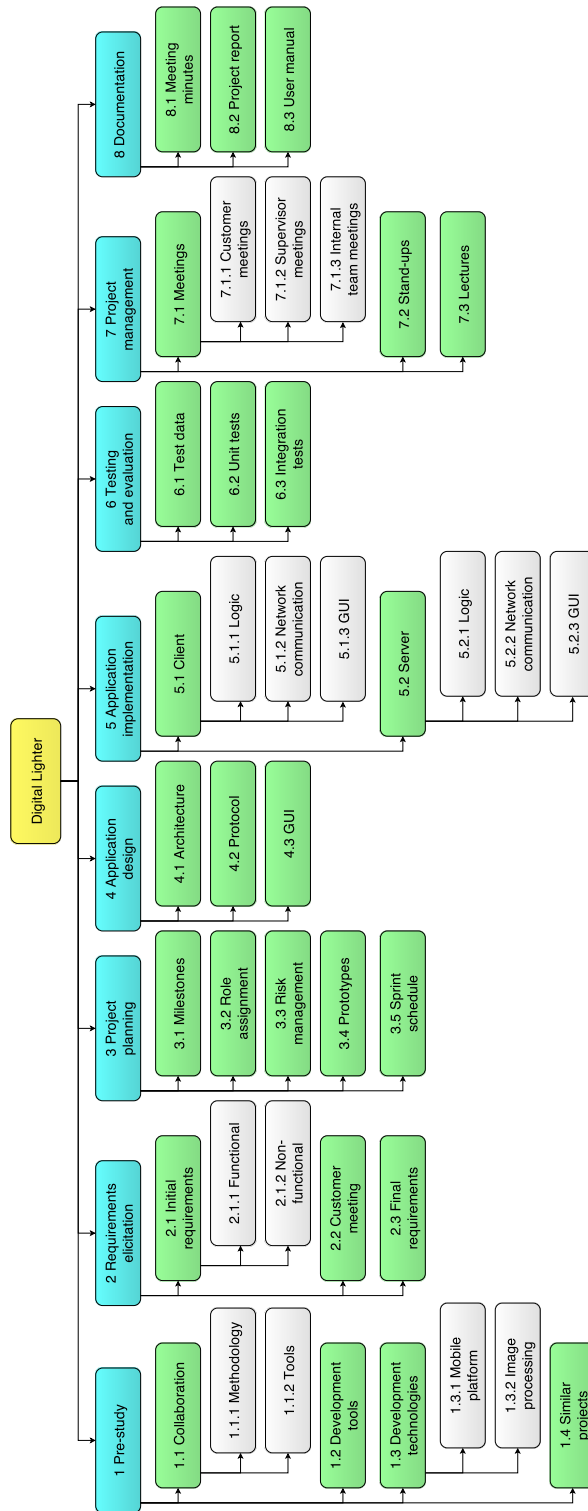


Figure 3.1: The work breakdown structure

Table 3.1: Skills and previous experience table. Coding: ● expert, ● experienced, ● neutral, ● little experience, ● no experience

	Agnethe	Tomas	Milos	Jan
Leadership	●	●	●	●
Scrum	●	●	●	●
Mobile software development	●	●	●	●
L^AT_EX	●	●	●	●
Network programming	●	●	●	●
Image processing	●	●	●	●
Java	●	●	●	●
C++	●	●	●	●
Testing	●	●	●	●

3.2 Organization

This chapter outlines how the project group is organized and why. To perform efficiently and effectively it is important to figure out what kind of organizational structure is suitable. Structures provides an overview of who makes decisions and is responsible, how information flows within and between organizational units and suggests what it takes to move between departments. This means that a good organizational structure provides the team with the clarity of the different responsibilities. Since one of the goals is to finish the project in the given time frame it is important to have a good organizational structure.

3.2.1 Role Assignment

Since team is using ScrumBut the roles were adapted a bit. We chose to rename the product owner to be communication responsible. The reason for this is not only a customer to continuously keep contact with, but also a supervisor. The team decided to be more specific for some of the role, and not just use "the team". An we also wanted to have a role for quality assurance.

When the team assign roles it was done according to members' skills and previous experience. It was decided to make a survey of relevant knowledge. Results of this survey can be seen in Table 3.1. This table was used as a base for the role assignment. In the beginning, during planning, not so many roles were establish, but during the development process itself it was found out that some of them are necessary. Nevertheless only few roles are used and the responsibility of unused roles are distributed to all members. The bottom line is that as a development team all of the members are equal. The assigned roles can be seen in Table 3.2.

Table 3.2: Assigned roles and their responsibilities

Role	Person	Responsibility
Project Leader	Agnethe	Responsible for progress of the project according to the plan. Distributes work to group members. Has final call in arguments.
System Architect	Milos	Check consistency and analyze all layers of the product.
Scrum Master	Agnethe	Leads the scrum stand-ups.
Communication Responsible	Jan	Responsible for communicating with customer and supervisor. Regularly send meeting minutes, agenda and other documents to customer and supervisor.
QA Responsible	Tomas	Ensure a quality of all documents and end-product.
Documentation Responsible	Agnethe	Responsible for delegating and supervising work on final report.

3.3 Risk Management

In the Table 3.3 the consequence and possibility in a number between 1–10. The risk, or the risk factor, is the consequence multiplied with the possibility. The reactive measures is possible solutions when something goes wrong. Proactive measures is what we can do to try to avoid the risks.

Other risks can be seen from analyzing the skill table. If for example the only two persons on the team who are familiar with image processing are not present, then this will be a risk. There is a great possibility that this kind of task will take longer time then planned for.

Also if some of the rows in the skill table all were painted red, which would imply that the team had no experience with this, then this would be a risk. In such a case, studying of that field of area would have to start immediately and a meeting with the customer should be planned about considering scaling down the task.

This project is developed under a few technical, resource, time and knowledge limitations.

The biggest limitation is the image processing part. Half of the team has no experience with this, and the other half has little experience. Their experience is mostly theoretical information about the subject, and practical experience is preferred. This limitation is known to the team, and the plan is to learn by doing. This approach is chosen so that the team would no spend more time than necessary doing research.

Another limitation is lack of experience with mobile development within the development team. All of the team members have Android phones, and to be able to test our application, an Android

Table 3.3: Handling risks

Event	Consequence	Possibility	Risk	Reactive Measures	Proactive Measures
Someone gets sick	4	5	20	Other people do more work.	Free weekends
Coding problems	4	7	28	Talk to supervisor and/or relevant persons	preparing for the task
Testing problems	4	4	16	Talk to customer about reformulating requirements	Double check requirements with customer
Implementing things we are not supposed to	7	6	42	Try to adopt functionality or start all over	Don't do anything that is not in backlog and keep good communication with customer
Dead end with technologies	8	8	64	Talk to supervisor & Guru office	Do thoroughly research
Unrealistic time estimate	7	8	56	Work overtime	Planing poker
Frequent changes in requirements specification	6	3	18	Renegotiate with a customer	Try no to change finished modules and keep weekly meetings with the customer
Customer too ambitious	9	5	45	Renegotiate with a customer	Keep customer informed about what to expect
Hardware problems	9	3	27	Obtain a new one	Keep your devices updated

application must be developed. Only one team member has experience with this.

If the project is not scaled down, the system could not be tested as the required resources are not available. Numerous audience, dozens of Android phones or big screen is a good example.

As this course last for a 13 weeks, it is normal that some trade-offs must be made.

3.4 Communication

Specific guidelines and practices were adopted so that the team and the customer could continuously verify that the project development keeps the right direction and that the requirements are being fulfilled.

3.4.1 Customer collaboration

Since the scope of the project and the actual requirements are determined by the customer it is essential to establish tight collaboration practices and even involve the customer in the collaboration tools the team uses. The description of our approach to fulfilling sufficient customer relations follows.

Meetings It has been agreed that weekly meetings taking approximately 60 minutes will be held. The standard agenda consists of the approval of the meeting minutes summarizing the last meeting, comments on the meeting minutes and the specific issues regarding the team-customer collaboration and technical issues. As the customer is located in Oslo the meetings will be carried out using the video-conference devices provided by the NTNU (accessible in the Accenture Lab). Contents of each meeting including the topics discussed, decisions made and the future plans are summarized in the meeting minutes. The team should provide the customer with the meeting minutes document before the next meeting is held.

Software tools As mentioned earlier the team decided to utilize the Target Process 3 as a collaboration tool suitable for handling Scrum-driven projects. The customer was invited to join the framework so that he could observe the development process more closely. Thanks to the software tool the customer is able to easily adjust the order of the user stories according to his preferences. the customer was also invited to the file sharing service and version control system that the team uses to save and distribute the documents and the source code.

3.4.2 Supervisor collaboration

It has been agreed on weekly meetings also with supervisor, at least from the beginning of the course. These meetings are for the team to ensure that the right track is followed and also for the supervisor to be updated with team's progress.

If urgent situation occurs, supervisor will be contacted via email with detail of problem or with invitation for premature meeting.

3.4.3 Team collaboration

Mostly during the pre-study phase the developers need to distribute and maintain the information considered important for the next work. As stated in Section 2.6 the Facebook group was chosen and set up to serve this purpose as it disposes of the tools suitable for exchanging website links, opinions and multimedia.

It has been also decided that the project report should be written continuously by all members as a part of our daily work flow. Through this practice the team will be able to keep track of the amount of work done and all members will make sure they understand well what direction the project is aiming.

3.5 Duration and workload

Compendium proposed week workload 25 person-hours per week. During our internal meeting it was decided that each member will spend 30 hours per week because the team consists only of 4 members. Fixed daily working hours were suggested so that the workload could be distributed through the whole semester. Regular stand-ups will be made according to Scrum methodology.

According to work break down structure, which can be seen in Figure 3.1, and presented week workload, initial time estimation of whole project was created. You can see the estimation in Table 3.4. As a part of a evaluation, spent hours will be filled, and then retrospective to the planning phase can be done.

Table 3.4: Initial time estimation table

Task	From date	To date	Hours	
			Estim.	Spent
Pre-study	26.08.2013	15.09.2013	70	30
Requirements elicitation	26.08.2013	29.09.2013	80	86
Project planning	26.08.2013	15.09.2013	75	59
Application design	26.08.2013	27.10.2013	90	52
Application implementation	02.09.2013	20.11.2013	450	439
Testing and evaluation	02.09.2013	20.11.2013	130	50
Project management	26.08.2013	20.11.2013	120	235
Documentation	26.08.2013	20.11.2013	425	412
Σ			1440	1363

3.6 Project scope

After third meeting with our customer it was decided to lay out the scope of the project that is tangible and doable under 13 weeks. Customer idea although good and innovative has one flaw – it is a huge undertaking.

As the team operates under certain limitations explained in Section 3.3, customer and the team agreed on next terms:

- Take project title as domain of work and not final product.
- Scale down problem but attack all main problems of the domain.
- When designing architecture disregard scaling of product.
- Disregard some of problems that are not important for final prototype, but with approval of a customer before making this kind of decision.

3.7 Project plan

This section contains a general plan regarding the Scrum process. For all the sprints, the general sprint duration, and all the milestones for this project. In this chapter there is also a Gantt diagram and a work breakdown structure to visualize the plan to reach the different milestones.

3.7.1 Utilizing Scrum

After fast research and consultation with the supervisor and customer who already has the experience with CDP student group from last years the team decided to use Scrum methodology for a few reasons. Requirements and scope of the undertaking were not that precisely defined at the time the methodology had to be chosen. The team's approach therefore could not be founded on the sequential methodology as is "waterfall".

One of the requirements was to have frequent meetings with the customer and involve him in development process. Sprints in Scrum enables the development team to do just that – to have meeting with customer at the end of every sprint, and plan next one together. Sprints do not have to be the same length so the developing process and risks might be managed better, and the team will be able to have as much sprints as possible under time limit of 13 weeks.

Scrum will provide the team with the derivatives that can be enhanced in increments and that allow the team to gradually reduce the risk and keep customer informed about the progress.

Even though the Scrum itself is not represented by the strict standard there can be found specific essential features typical for this framework [11] [13, p. 15, p. 97, p. 154]. Table 3.5 summarizes which aspects of Scrum were decided to be utilized and which were omitted.

Following chapters describe how the roles were assigned and how the Scrum processes are fulfilled.

Table 3.5: Table lists the main features of Scrum and shows which are fully used, partially used and omitted altogether by the team

Scrum features	Usage	Explanation
Sprints	✓	
Sprint artifacts (product backlog, sprint backlog)	✓	
Sprint planning	✓	
Daily Scrum (stand-ups)	✓	The team members are not working full hours and thus they do not meet in school every day.
Planning poker	✓	Some user stories are assigned time estimation according to the customer's suggestion.
Work boiled down to user stories	✓	Not all work being done is recorded by the user stories. Many tasks like time tracking, handling written communication with the stakeholders or discussing the occurring problems are not tracked.
Acceptance of changes in requirements	✓	
Close collaboration with customer	✓	
Team members commit to their goals for the sprint	✓	
Product owner (aka customer) commits to ordering the product backlog	✓	
After each sprint review and retrospective should be performed	✗	The team only reflects on the last sprint while writing the report. It does not hold specific meeting focusing on review and/or retrospective.
The role of Scrum master	✗	Even though the Scrum master role was assigned the team decided that the members should mostly perform equally so the Scrum master fulfills the duties only partially.
Bonus user stories	+	The customer agreed with specifying the <i>bonus user stories</i> for some sprints. These stories are optional to accomplish, they are set the lowest priority and the team only works on them if there is time left.

Roles

Scrum has only three main roles, the Product owner, Scrum master and the team.

Product owner Is responsible for the requirements of the product, and responsible for the product vision and improving return on investment. The product owner is supposed to manage the end user, and stake holder expectations. Therefore they are also responsible for prioritizing the product backlog, and release planning and providing clear and testable requirements to the team. They collaborate with the team, end users and stake holders ensuring that the goals are met and they accept the software at the end of each sprint.

Scrum Master Is responsible for coaching and guiding through the Scrum methodology. The scrum master is responsible for creating a trustful and inclusive environment. They keep the process moving forward ensuring that the values and principals of scrum along with its framework are followed. As Agnethe dispose of the previous experience working in Scrum team she has been assigned this role.

The Team The team are cross-functional, autonomous and self organizing. They are responsible for estimating the size of requirements driving from a tactical perspective making their own design and implementation decisions. They track the progress of their own work with the guidance of the scrum master and the team commit to delivering increments of software being accountable to the product owner for delivering as promised. You can see detailed sprint decomposition

Processes

The most important Scrum ceremonies include daily stand up meetings, sprint planning, sprint retrospective and sprint review.

Daily stand up meetings are meant to enable the team member to make their progress visible, summarize the obstacles and to commit to the next tasks. Basically all of the members should be able to answer these three questions:

- What have you done since yesterday?
- What are you planning to do today?
- What issues are you currently facing?

These meetings are especially useful for keeping it clear who works on what. Nevertheless as it is described in Chapter 3.5 the team members do not work full hours and thus do not meet every day. Given this fact it is impossible to make the daily stand up meetings and this is one of the reasons the team rather falls back on Scrumbut.

Sprint planning serves the purpose of discussion between the team and the customer focused on selecting the user stories to be moved to the next sprint's backlog and specifying their priority. Both team and customer agreed they will collaborate on each sprint planning using the online tool TargetProcess described in Section 2.4.3.

Planning poker or scrum poker is technique for estimating time effort or relative size of the user stories [2]. It gets performed by all team members with regular playing cards or specially designed deck. Team agreed on using planning poker for user stories time estimation, and using "Scrum Poker Cards (Agile)" android application¹ instead of playing cards. Choosing possibilities are presented on Figure 3.2.

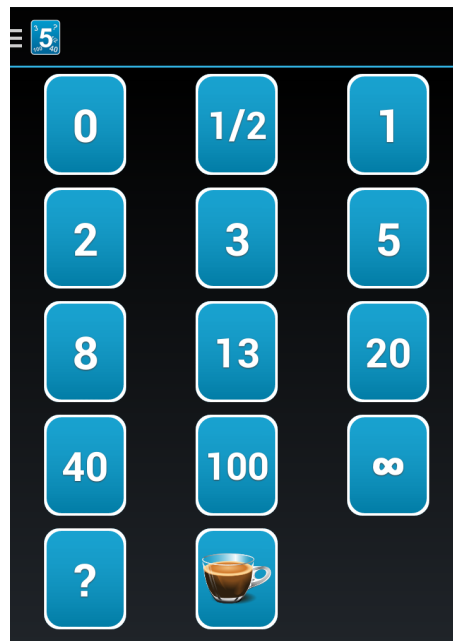


Figure 3.2: Scrum poker

Sprint retrospective is a discussion about what went well or wrong during the last sprint. It is a good practice that enables the team to keep overview of the work progress.

Product review provides the opportunity for the product owner to give feedback about the work done and suggest improvements and changes. In the case of DeigitalLighetr project it has been agreed on that the product review will contain the demonstration of the current working prototype.

¹<https://play.google.com/store/apps/details?id=artarmin.android.scrum.poker>

Artifacts

Product backlog and user stories are used to divide the complex problem into simpler subtasks. Backlog accumulates the user stories which must be completed during product development. For storing the stories and managing the backlogs the online tool TargetProcess is used. It enables the team to involve the customer and collaborate with him tightly.

Burndown chart graphically presents the resources available as compared to the work that must be done. TargetProcess handles generating the burndown charts well.

3.7.2 General plan for the sprints

In general the sprints will last for two weeks. The sprint length was proposed by customer (and accepted) to ensure frequent meetings, support higher number of deliverables during the implementation and to give the team an opportunity to experience as many sprints as possible. The only sprints that do not have length two weeks, is the first and the last. This is because these two are more related to getting started, and finishing the project. The other sprints will have a more general approach.

After each sprint, there is a sprint review with the customer. The sprint review takes place every other Thursday, which means that the Wednesday before, the team has to prepare for the review. In each sprint review there is both presentation and demonstration what have been done so far. After the presentation there is a retrospective with the customer. It is important to evaluate the sprint delivery with the customer. After the retrospective, the team with the customer will plan the next sprint, and the customer can prioritize which user stories he wants the team to work on next. The Friday after the sprint review, the team will do a detailed planning of the next sprint.

On the Thursdays in between we have weekly meetings with the customer. It is important to have good communication with the customer. These meetings give us opportunities to ask questions, and to make sure that the team is on the right track.

You can see detailed sprint decomposition in Figure 3.3.

Sprint duration: You can see the Gantt chart depicting sprint duration and allocation into weeks in Figure 3.4.

3.7.3 Product milestones

Milestones have been set in order to more accurately determine whether or not the project is going in the right direction. After completing some of the groups of milestones running prototypes will be delivered. There are 4 prototypes in total that team agreed on with the customer. After reaching point of 4 prototypes meeting with the customer regarding how the project is about to continue will be set. The last prototype implies that all milestones regarding product are reached. Set of milestones related to our school obligations have been set as well.

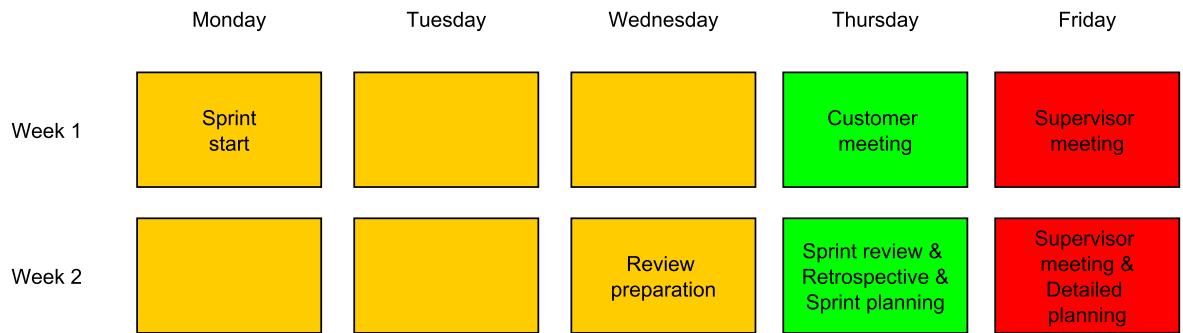


Figure 3.3: Detailed sprint decomposition

Overview of all milestones including 4 prototypes:

Obedient client – Prototype 1

- The server sends command to one client.
- The client "plays" received command (white light 10 seconds).

Obedient crowd – Prototype 2

- The server sends same signal to multiple clients.
- The server sends multiple different commands to the clients.
- The client plays received commands (Red light 2 seconds, Green light 2 seconds).

I see light – Prototype 3

- Server identifies one client from light.
- Server maps one client to grid (4 quads).

Traffic light control – Prototype 4

- Server identifies multiple clients from light.
- Server maps all devices to the grid.
- Server plays traffic light simulation to the grid.

Course related milestones

- Finish Project Plan document.

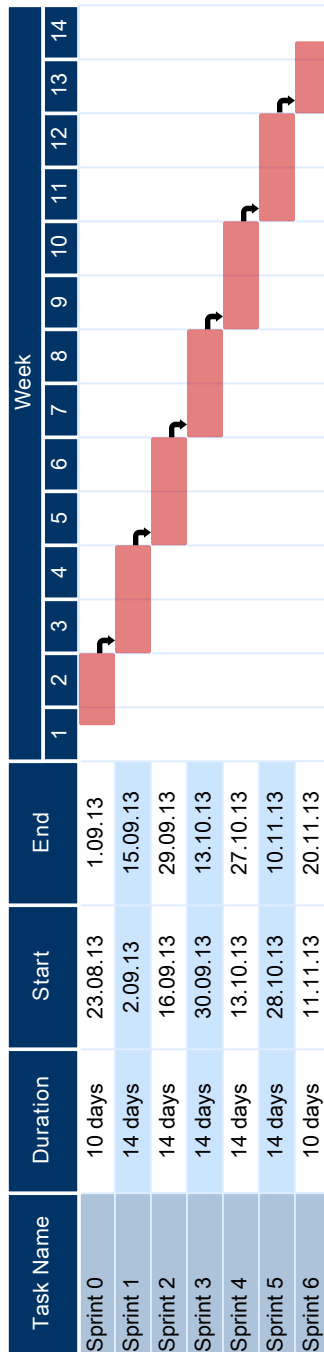


Figure 3.4: Gantt Chart

- Learn about team dynamics.
- Deliver Report draft.

3.8 Test plan

The purpose of this chapter is to introduce the overall plan for the testing of our product, and it describes the scope of the overall testing. In this section there will be explained how the testing of the product was planned, who will be responsible for it, and very short mention the test criteria. This is the strategy that will be used to verify and ensure that the product meets its requirements, and of course the quality attributes.

3.8.1 Approach

This is a proof of concept project. The customer did not want the team to focus on the testing part on this project, but in order to deliver a valid product it will be necessary to do some testing. In order to save time and spend most of the working hours on the implementation, the black box testing will be used.

The happy path is utilized. A happy path is a default scenario featuring no exceptional or error conditions, and comprises the sequence of activities executed if everything goes as expected. Happy path testing is a well-defined test case that uses known input, that executes without exception and that produces an expected output. This means that the team do not spend time actually writing tests, but still it must be manually tested that the product does what it is supposed to.

Verification and validation testing is also done. This is the process of checking that the product meets specifications, and that it fulfills its intended purpose. It may also be referred to as software quality control, and it is high-level checking. This will be done with test cases, or scenario testing. A test case is a set of conditions or variables under which a tester will determine whether an application, or one of its features, is working as it was originally established to do. The mechanism for determining whether a system has passed or failed such a test is known as a test oracle. In the case of Digital Lighter product these will be the requirements, or the use cases. Scenario testing is similar, but test cases tests single steps, and scenarios covers a number of steps. The scenarios should be easy to evaluate. The development team will not implement tests, but the tests must be performed manually. In each sprint test scenarios will be designed where the expected results with the actual result will be compared. The integration between the deliveries in the different sprints will also have to be manually tested.

For user testing the service TestFlight is used. TestFlight allows the customer to easily access the applications, and try out the features. The team can also get feedback from other end-users, and this way the product can be improved.

3.8.2 Responsibilities

The whole development team is responsible for testing the product. Each person is responsible for testing the features they have implemented. It is important to test continuously throughout the sprints, and it will be done when the relevant functionality is implemented. The test will be performed at a time that allows the developers to fix bugs before the sprint has ended. The

person who coded specific part should not test it, rather the different person who worked on something different should perform the tests. This is mostly for the usability of the product. The manpower and time is limited, which means the team cannot guarantee that this is always the case.

3.8.3 Test criteria

The product should function according to customer's requirements, described in Section 4.2, and approved testing mock data. Approval of tests and project validity will be done by customer's approval scenarios. Those can be images, a video, or sequence of activities that will lead to specific result. Test can be considered to be valid if expected result is associated with one of the mentioned approval tests.

In order to be considered successful, test execution needs to have the output as expected. Deviating from expected will help team in finding bugs and errors. Confirming the expected output, since black box method is used, will be enough proof of working code.

4 | Requirements

In this chapter expected behavior of product and its resulting requirements will be described. First there will be introduced simple life cycle of the product. Although project title was accepted as a domain and not a final product, following requirements could be applied also on project that is not scaled down.

4.1 Description

To clarify and properly explain life cycle of one run of server side application, its usage and possible requirements, structure of UML's activity diagram [5] was adopted. Therefore we show in Figure 4.1 interaction between manager and one client user. Notice that there are some actions of both manager and client user which do require direct manipulation with mobile phone or other actors such as "Notice users to raise their mobiles", "Raise mobile", "Put down mobile" and more.

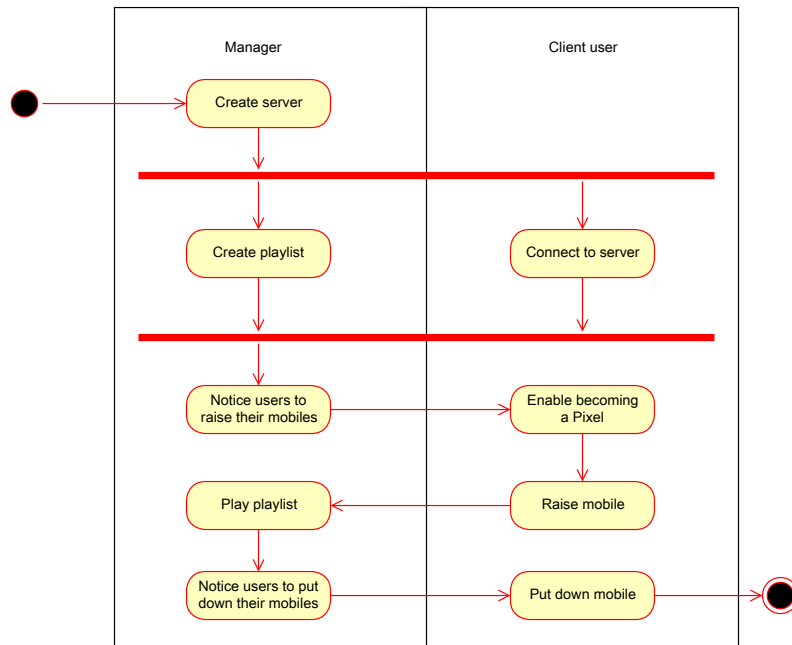


Figure 4.1: Basic usage scenario

4.2 Requirements

Requirements can be divided into two categories [6, p. 19]: functional and non-functional. Functional ones describe a function which a system or system component must be able to perform [12, p. 35]. On the other hand non-functional requirements cover what product is, they are any other requirements than functional [6, p. 20].

4.2.1 Functional

The use case diagram of whole system with all actors is shown in Figure 4.2 According to this use case diagram, list of functional requirements was created.

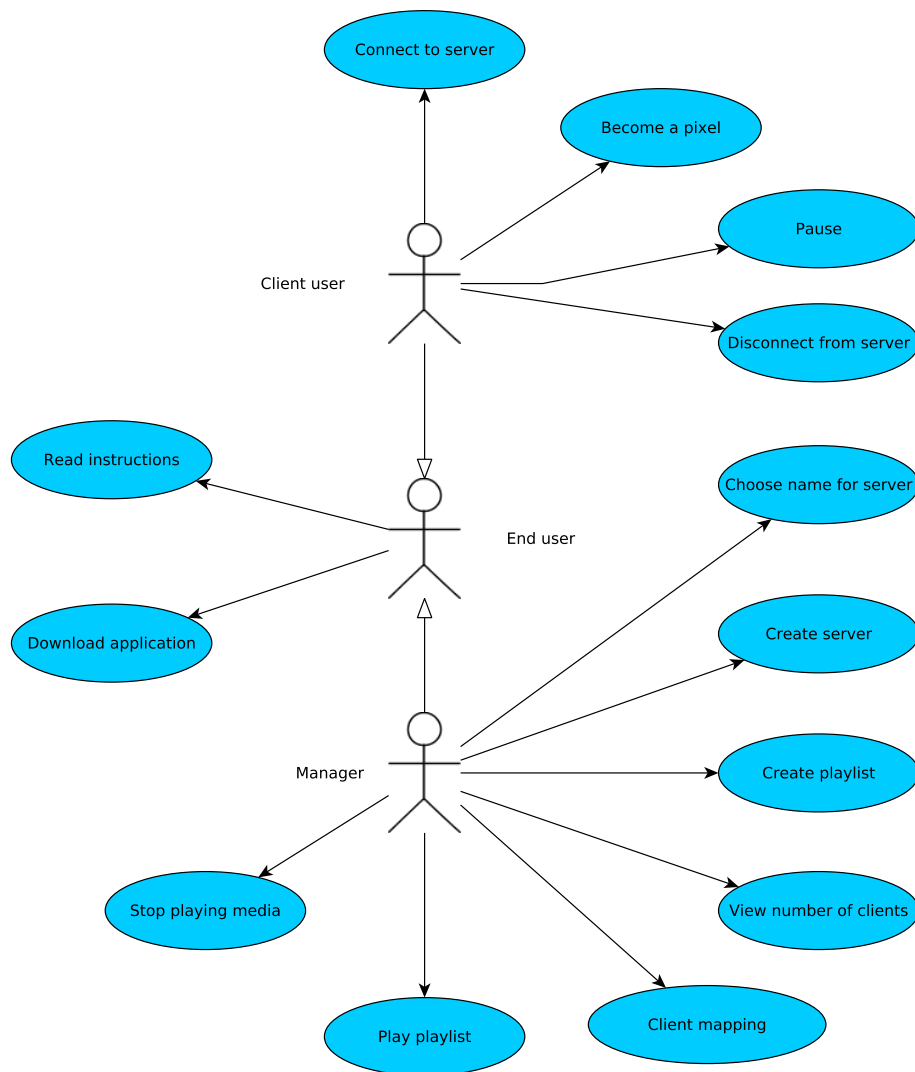


Figure 4.2: Use case diagram

End user You can see list of those requirements from end user's point of view below.

E1 I as a end user want to be able to read instructions.

E2 I as a end user want to be able to download relevant application from mobile application store.

Client user You can see list of functional requirements from client user's point of view below.

C1 I as a client user I want to easily choose to which server/concert stage I would like to connect.

C2 I as a client user I want to connect to chosen server.

C3 I as a client user I want to become a *Pixel*.

C4 I as a client user I want to be able to pause being a Pixel.

C5 I as a client user I want to be able to disconnect from server/stage.

Manager You can see list of functional requirements from manager's point of view below.

M1 I as a manager I want to be able to choose name for my server/stage.

M2 I as a manager I want to be able to create a server.

M3 I as a manager I want to be able view attendance.

M4 I as a manager I want to be able to start mobile phone localization.

M5 I as a manager I want to be able to choose playlist to be played.

M6 I as a manager I want to be able to start playing given media.

M8 I as a manager I want to be able to stop the media.

4.2.2 Non-functional

Non-functional requirements are the overall factors that affect run-time behavior, system design, and user experience. Some of these attributes are related to the overall system design, while others are specific to run time, design time, or user centric issues. The list and description of non-functional requirements for DigitalLighter application is below, and also shown on Figure 4.3.

Accessibility is the degree to which the product is available to as many users as possible. Accessibility can be viewed as the "ability to access" and benefit from the product. To achieve accessibility team plan to fulfill following requirements:

N1 Product must be developed as two separate applications, server and a client.

N2 Client side application must work on Android platform.

N3 Application must be deployed to Google Play application store.

Maintainability – the product needs to isolate defects or their cause, prevent unexpected breakdowns, correct defects and repair or replace faulty components. If one component need to be replaced, then the system should do this without having to replace still-working parts. The product also needs make future maintenance easy.

Performance – is about managing system resources in the face of particular types of demands to achieve acceptable system behavior. In particular performance objectives that have to be satisfied are:

N4 Client must be able to play command colors at least 1 frame per second.

N5 Client need to have minimal possible impact on device's battery.

N6 Server must be able to detect fake lights and cancel their influence on the system,

N7 Server must be responsive and without User Interface freezing.

N8 The product must support at least 16 client devices.

Modifiability – code must be prepared for changes, and this can only be done by increase cohesion, and reduce coupling. If the modules get built smaller, the risk of having to make a change will be reduced. Team should also make sure that dependencies are restricted. Increasing the cohesion, or interdependency within modules, can affect build time, load time, initialization time or run time, therefore some trade-offs have to be done. By all means system must satisfy following requirement:

N9 The product must be scalable – it must work with different count of client devices.

Interoperability – describes the extent to which systems and devices can exchange data, and interpret that shared data. Digital Lighter server and client application should be able to mutually share specific structure or format of data that is preserved and unaltered during the process. For this purpose:

N10 The communication protocol should be established.

Usability – DigitalLighter have to be easy to use, understandable, and easy to learn. This can be achieved with good User Interface and logical sequence of actions in order to achieve desired behavior. Therefore team should use only native widgets that all Android users are familiar with, making multiple Activities¹ composed of functionally related options, and providing only functionality given by requirements so visual noise can be reduced.

¹<http://developer.android.com/reference/android/app/Activity.html>

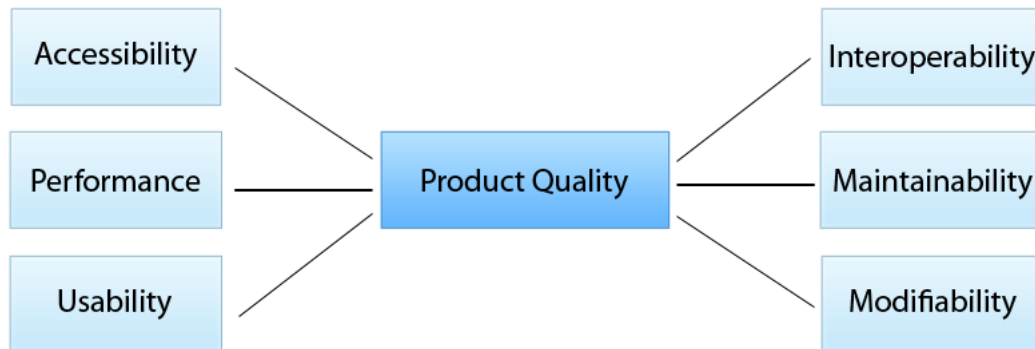


Figure 4.3: Quality attributes.

4.3 Summary

In this chapter it was explained how the application's basic life-cycle looks like and the use case diagram in Figure 4.2 was introduced. According to use case diagram there were created *Epics* which serve as a requirements. After that non-functional requirements were introduced following by quality attributes that Digital Lighter have to satisfy. These are summarized in Figure 4.3. In following chapters each user story which is connected to relevant requirement will be referenced with requirement's identification.

As you could see, client user's interaction with both applications is rather simple (and as the customer suggested the user interface is not core of the work) the main attention will be focused on image processing, network programming, synchronization and mapping devices into screen.

User stories will be chosen by customer after discussion with team during each planning meeting. Its priorities can be change anytime since the team is trying to be as agile as possible. During the planning meeting, if needed, new stories can be created or some stories can be discarded.

5 | Software architecture

In this chapter, the product architecture is described using the 4 + 1 architectural view model [10]. It is based on the multiple concurrent views and therefore better explanation of architecture can be provided. Only the high level design will be presented in this chapter and in each sprint. The specific and detailed architecture will be described. The design is made with emphasis on generality, therefore it is possible, that in particular sprints in the architecture section it will be adapted to specific situation.

Since some design patterns were used, the purpose of using this specific pattern and its advantages will be discussed in the end of this chapter.

5.1 Logical view

The logical view primarily describes functional requirements [10, p. 3]. The client and the server are decomposed into a set of key abstractions, taken from the problem domain, in the form of classes.

The logical view will be described by class diagrams. Only core classes will be discussed. Detailed architecture of particular parts will be presented in sprint chapters.

You can see server class diagram in Figure 5.1. As a starting point in this diagram can be considered instance of class **MainActivityServer**. This class is controller of whole application and handles inputs by manager. It also initialize one of the core classes – *ConnectionService*. Instance of this class has ability to send commands to clients using methods such as **broadcast** or **unicast**, and it is responsible for creating a service on local network. Its interface is used through whole system.

When inputs from the camera is needed, the instance of class **CameraActivity** is emerged. Since that moment, it takes responsibility for all handling inputs from manager and also for displaying feedback of other components on the application screen. Mentioned class serves only as intermediary between camera and instance of class which implements interface **DeviceLocatingStrategy**. This object can process incoming images and according to them it updates its inner data structure, which maps sockets into position on grid. Two classes implements that interface: **DeviceMapper** and **DeviceTracker**. First mentioned one is used initially, when there are no assumptions about devices positions. After **DeviceMapper** assigns position to each socket, his role is taken by

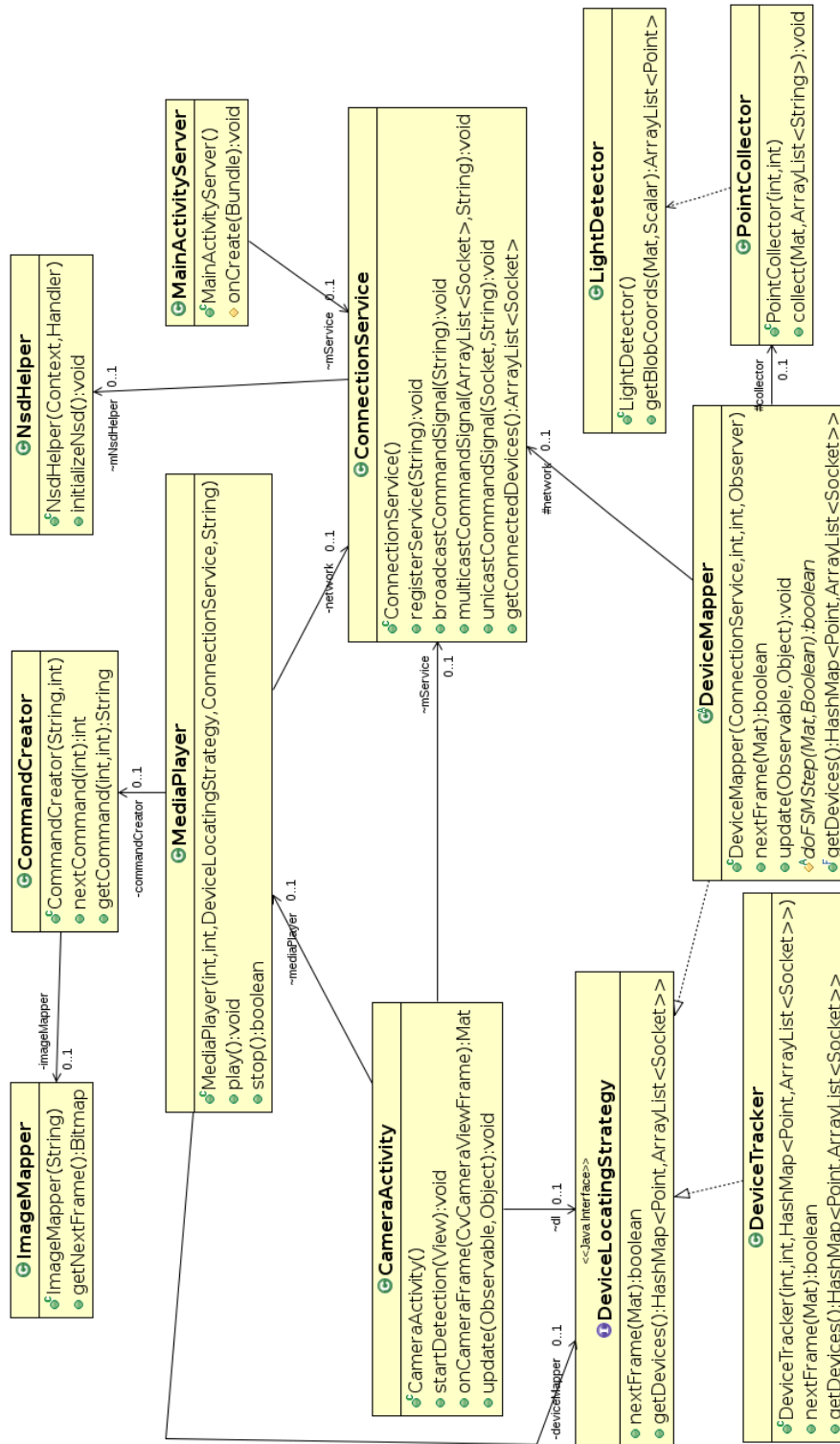


Figure 5.1: Server class diagram

DeviceTracker, which is designed¹ to be able to track movement of devices in time and track the changing position.

Instance of class **MediaPlayer** provides interface to load a media from a given path and start and stop playing that media.

You can see client class diagram in Figure 5.2. Instance of class **MainActivity** handles inputs from user and controls the flow of the whole application. **SNTPClient** is a class that main responsibility is to synchronize time through network with other devices. Instance of class **Connection** is responsible for connecting to an existing server and also for handling incoming control commands. These commands are interpreted by instance of class **ClientPlayer**.

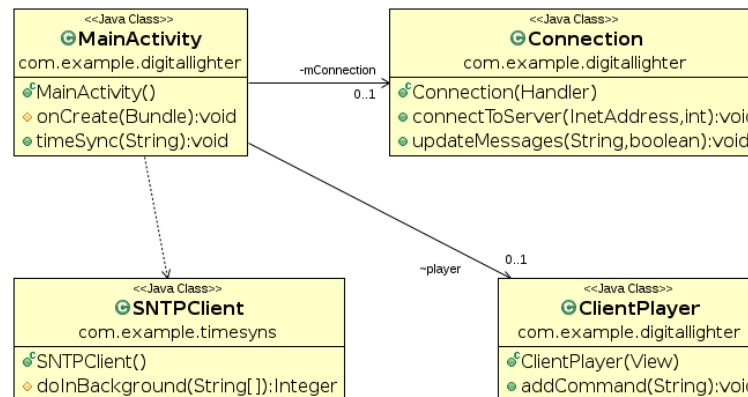


Figure 5.2: Client class diagram

5.2 Development view

The development view, is a view of the system's architecture. It can be described by a component diagram. It focuses on software module organization in development. Component diagram shows components in system and their provided and required interfaces.

Component diagram for server application can be seen in Figure 5.3. There are five components: *Network*, *Media player*, *Device locating*, *Device detection*, and *Core*.

Network is a component that provides interface which allows other components to communicate with clients. This includes broadcasting, unicasting and multicasting. It is also responsible for handling the new connections from clients and creating new service.

Media player is a component that provides simple interface consisting of start and stop command. It is responsible for playing actual media and timing for sending control commands to clients.

¹This class is actually not implemented due to time restriction after discussion with customer. It is only prepared for future extension.

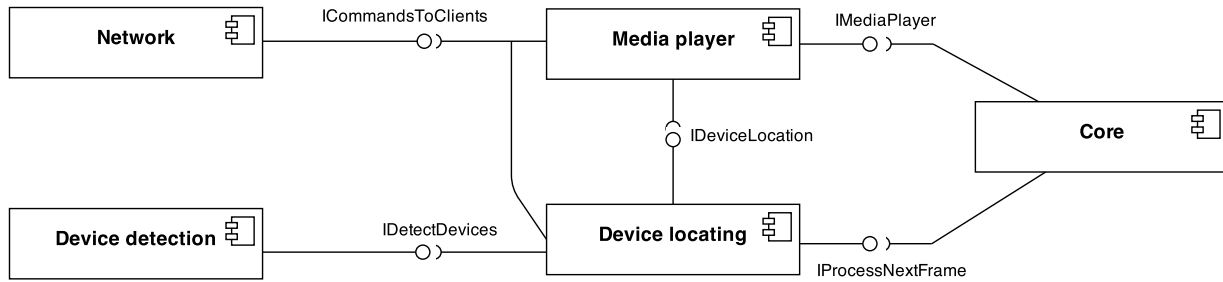


Figure 5.3: Component diagram of server application

Device locating is a component providing two interfaces. It must be able to handle incoming images and also it provides to other components locations of all connected clients.

Device detection is a component that provides single interface. It accepts image as an input and it returns positions of possible devices.

Core is a component that handles camera and manager's actions.

Component diagram for client application can be seen in Figure 5.4. There are four components: *Network*, *Command player*, *Time synchronization*, and *Core*.

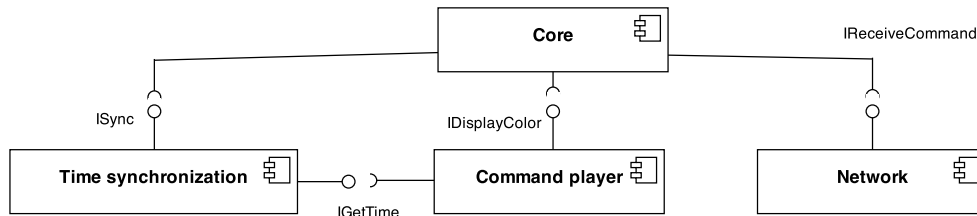


Figure 5.4: Component diagram of client application

Network is a component that provides interface for receiving control commands. It is also responsible for connecting to server.

Command player is a component that provides interface for displaying color at certain time in screen. Therefore it also requires access to synchronized current time.

Time synchronization as a component that provides two interfaces. *ISync* is an interface through time synchronization with remote server can be started. *IGetTime* interface provides current synchronized time.

Core is a component which handles client user's actions and requires several interfaces, mention above.

5.3 Process view

The process view shows concurrency and distribution of a system and system's processes. It can also show communication between these units. Process view can be described by activity diagrams.

You can see activity diagrams in Figures 5.5 and 5.6. Note, that client application is sending signal action *connect to server*, which is connected to server's event action *new client connected*. All clients connections are stored in data store *client list* and particular actions require all client connections to be able to send any information to clients.

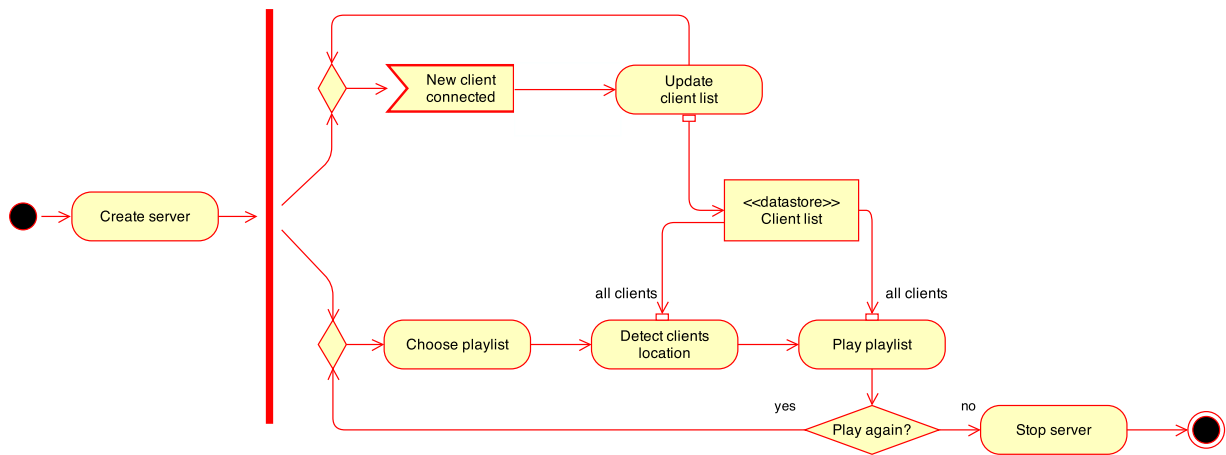


Figure 5.5: Activity diagram of server application

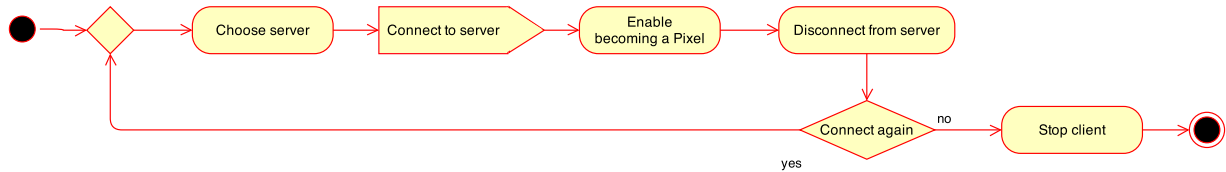


Figure 5.6: Activity diagram of client application

5.4 Physical view

The physical view [10] describes a mapping of software onto hardware. It takes into account the non-functional requirements of the system, and it can be described by deployment diagrams.

Deployment diagram depicting physical view can be seen in Figure 5.7. According to requirement N1, there are two executable files deployed; one deployed to client device and second to server

device. There is also need for NTP² server, which will maintain all connected devices synchronized. To generalize the architecture design, the *camera* device is treated as a separate device, which will only send its output to server device.

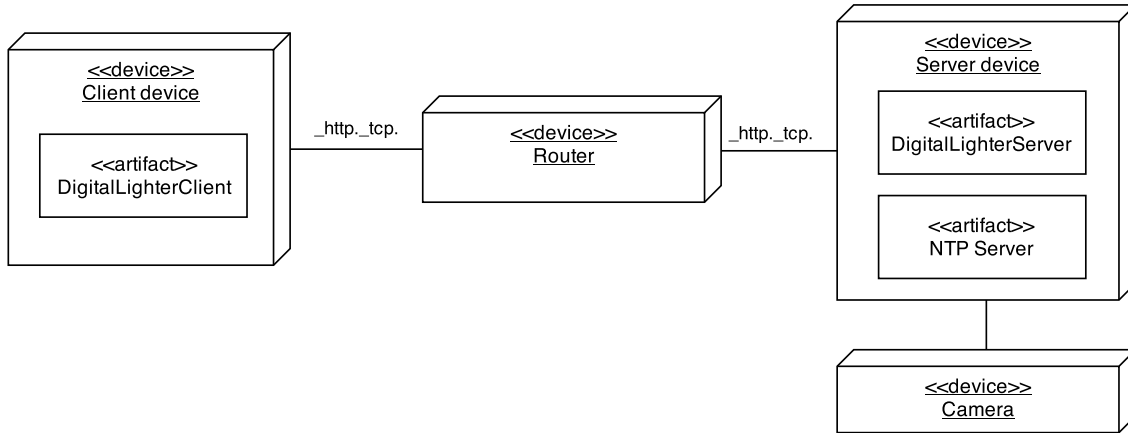


Figure 5.7: Deployment diagram

5.5 Scenarios

Use case diagrams can be considered as scenarios, and is therefore introduced in Figure 4.2.

5.6 Used design patterns

Design patterns solve specific object oriented design problems and make the design itself more flexible, elegant and reusable [9, p. 11]. They are proven techniques and they also improve documentation – due to well-known terminology it is easier to understand an architecture.

Below, you will find description of used design patterns and their utilization.

5.6.1 Strategy

Strategy pattern defines a family of algorithms, encapsulated them and make them interchangeable[9, p. 349]. This pattern is applicable for example when related classes differs in a behavior or when different variants of algorithms are available.

Strategy pattern was utilized in few cases. First, classes `DeviceMapper` and `DeviceTracker` are related classes and they differ in the behavior, and therefore there was created an strategy interface `DeviceLocatingStrategy`. Both concrete strategies implements the behavior in their own way.

²<http://ntp.org/>

Since class `DeviceMapper` is an abstract class, it can be also considered as another strategy. In this case, concrete strategies will implement different algorithm of detection of devices' position. These concrete strategies will be introduced in Chapters 10 and 11.

5.6.2 Observer

Observer pattern defines a dependency one-to-many between objects. When the object changes its state, all the dependent objects are automatically notified. This pattern is often used when change in state of one object will cause changes in others, but there is unknown the count of these dependent objects. It also does not force the developer to tight objects, the design can therefore stay general. [9, p. 327]

There was a need for displaying a feedback from device detection component, which is performed by the instance of class `PointCollector`. Therefore the `PointCollector` implements interface `Observable`. One of the dependency objects is instance of class `CameraActivity`, which draws detected devices into real time camera record. Another dependent object is instance of `DeviceMapper`, which is connecting clients' sockets with positions, and therefore it is also dependent on results of device detection.

6 | Sprint 0

In this chapter Sprint 0 will be described. It is written about what the team did during the first sprint, which also was a planning period for the team. At the end there will be evaluation of the whole sprint, and answers to following questions: What went well? What could be improved?

6.1 Sprint planning

The Sprint 0 is embraced as a preliminary sprint, where team can set up all necessary collaboration tools, equipment, prepare templates for meetings and mainly to acquaint team members with Scrum methodology. The original plan was to finish sprint 0 on 8th of September, but it have been decided to terminate it prematurely due to finishing sprint goals in shorter time than expected. Other reason for terminating the sprint was desire to start actual work on the product itself.

The user stories are listed in Table 6.1. Since software collaboration tool was used when sprint was already in progress, team did not manage to estimate the time needed to complete each story beforehand and thus the column **Est.** is left empty.

6.2 Sprint goals

- Read compendium
- Find and agree suitable collaboration tools
- Start working on the Project plan
- Fill backlog with stories
- Assign roles to members of the team
- Assign a name to our project

Table 6.1: User stories selected for Sprint 0

ID	Ref.	Description	Hours	
			Est.	Sp.
0.1	WBS 7.3	As a student I need to read the course compendium.	6	8
0.2	WBS 2.1	As a student I need to read, and understand our project.	6	8
0.3	WBS 7.1	As a student I need to agree on customer and supervisor meetings.	2	2
0.4	WBS 8.1	As a student I need to make a template for the meeting minutes, and the project plan.	6	5
0.5	WBS 7.1.1	As a student I need to attend customer meetings.	16	16
0.6	WBS 8.1	As a student I need to write the meeting minutes.	6	4
0.7	WBS 7.1.1	As a student I need to set up the video conferencing for further customer meetings.	4	2
0.8	WBS 8.2	As a student I need to make structure and start writing the project plan.	30	35
0.9	WBS 2.1.1	As a student I need to come up with the functional requirements.	40	64
0.10	WBS 3.2	As a student I need to help assign roles to the team members.	2	2
0.11	WBS 1.1.2	As a student I need to decide which collaboration tools to use.	6	8
0.12	WBS 8.2	As a student I need to come up with a project and team name.	4	2
Σ			128	156

6.3 Testing

Only the third party products were tested during the sprint 0. Team focused on testing the online team collaboration tools including Trello, described in Section 2.4.1, Gravity, described in Section 2.4.2 and TargetProcess, described in Section 2.4.3. The outcomes of our research are described in Section 2.4.

6.4 Occurring risks

One of the major risks team came across were the limitations of the selected collaboration tools used for controlling the Scrum process. As team members were learning all the perks of using the Scrum methodology it was continuously discovered that neither Trello nor Gravity software

tools suited team's needs and more time searching for another tool and setting it up again had to be spent. A few difficulties regarding the version control system Git prevents merging individual versions of the project plan document. This is considered as an risk so team spent significant amount of time solving these issues.

6.5 Customer feedback

The outcomes of preliminary studies have been presented to the customer. The customer was also walked through the collaboration tools and provided with the access to all of the team's resources so that the team and the customer would be able to collaborate more tightly. The customer was satisfied with the amount of the work done and approved the TargetProcess to become the team's main collaboration tool. Since the team managed to finish all of the selected user stories the customer agreed with the accelerated end of Sprint 0.

6.6 Retrospective

Since each complex software project requires that the team would go through the certain initiative tasks such as the assignment of the team roles, agreeing on the meetings, setting up the communication and collaboration tools etc., the Sprint 0 cannot be really thought of as a fully-fledged sprint. Team mainly focused on pre-studying the scrum methodology, communicating with the customer and the supervisor, clarifying the requirements and researching both the collaboration and development tools.

The working was started immediately after the first meeting with the customer was held. Besides the time spent on setting the tools we mainly focused on writing the project plan as this is a project where the amount of documentation is expected to be extensive.

In order to reflect on the past sprint and to learn from the mistakes done it is necessary to answer two essential questions: "What went well" and "What could be improved".

6.6.1 What went well

Three online collaboration tools (Trello, Gravity and TargetProcess) were evaluated so that the suitable one could be find. The initiation tasks were finished earlier than expected and the Sprint 1 could be started earlier. All of the members' computers were set so that all of the tools would work. The team also got familiar with the concept of user stories, assigned a few of them to Sprint 0 and managed to finish all of them. All of the members were committed to spend the same amount of working hours.

6.6.2 What could be improved

During the first days the team was not operating efficiently due to the fact that team started to write the project plan without any clear specification of the document structure. Once the

structure was agreed on the team members were able to work on the single tasks efficiently again without any need for unnecessary communication overhead and negotiations. Lack of experience in using the Scrum process resulted in imprecise usage of the user stories concept. The team should carry out additional research regarding the Scrum process in order to follow the specified guidelines properly. The team also did not measure the time spent on the different tasks. In order to reflect on the preliminary time estimation the time spent on working on the single tasks must be tracked. Late use of the proper collaboration tool resulted in inability to generate the burndown chart for Sprint 0.

7 | Sprint 1

In this chapter planning and work-flow regarding Sprint 1 will be described. All from setting goals to implementation and testing. On the end team will evaluate whole sprint and try to answer the following questions: What went well? What could be improved?

7.1 Sprint planning

After assembling all the tools in Sprint 0, team decided to start with the implementation of the core modules. As team's understanding of task improved, they were able to come up with user stories from the perspective of user, customer, developer and a student. All user-stories were given to the customer so they can be prioritized. All but user-stories concerning team school obligations, like writing project plan, minutes, meetings with supervisor and attending lectures. Team decided to have school tasks as a user stories so time can be better tracked. School user-stories were mandatory and already added as a user-stories of sprint 1. On Monday 02.09.2013. team had meeting with the customer where time needed for every user story were estimated. Result of that meeting was list of rest of the user-stories for sprint 1.

For the documentation part, team made a document "project plan". Supervisor wanted these sections to be integrated in the project report instead. Therefore team decided to make the content for the report, and to make the general structure for it. Also integrate the parts previously written in the project plan into the project report. Team also planed to add the additional sections that were missing.

All implementation related user stories for sprint 1 are presented in Table 7.1. All the documentation related stories for sprint 1 are presented in Table 7.2. All the project management related stories for sprint 1 are presented in Table 7.3.

Table 7.1: Implementation user stories selected for sprint 1

ID	Ref.	Description	Hours	
			Est.	Sp.
I1.1	C1	As a server I need to be able to listen for clients.	30	22
		Research server sockets.	6	6
		Implement server listener.	8	6
		Create the moc client.	8	6
		Connect with mock client.	8	4
I1.2	M6	As a server I need to be able to send one command to the client.	4	4
I1.3	M6	As a client I need to be able to receive one command from the server.	2	2
I1.4	M6	As a client I need to be able to receive commands from the server.	4	4
I1.5	M6	As a client I need to be able to "play" one command.	4.5	4.5
I1.6	M3	As a manager I want to see the number of connected devices to the server.	0.5	0.5
Σ			45	37

Table 7.2: Documentation stories selected for sprint 1

ID	Ref.	Description	Hours	
			Est.	Sp.
D1.1	WBS 8.2	As a student I need to restructure the project plan, so it is a report instead.	24	32
D1.2	WBS 8.2	As a student I have to finish the introduction chapter.	22	32
D1.3	WBS 8.2	As a student I have to finish the pre-study chapter.	20	10
D1.4	WBS 8.2	As a student I have to finish the planning chapter.	16	8
D1.5	WBS 8.2	As a student I have to finish the sprint 0 chapter.	8	10
D1.6	WBS 8.2	As a student I have to write planning and architecture sections of sprint 1 chapter.	10	20
Σ			100	112

7.1.1 Duration

This Sprint will be 2 weeks long. From 02.09.2013 to 15.09.2013. The team agreed on the date for presentation and showing the running demo – Thursday 12.09.2013. Estimated velocity is 240h since team agreed on 30 working hours per person per week.

7.2 Preliminary studies

In this section all research connected to finishing user-stories, taken before work on actual code, will be described, as well as a decisions team made in order to successfully finish sprint 1.

7.2.1 Bonjour software

The first research of technologies that could help with achieving client-server communication, without writing everything from scratch, was Bonjour software¹. It is a group of technologies that includes service discovery, address assignment, and hostname resolution. Bonjour locates devices such as printers, other computers, and the services that those devices offer on a local network using multicast Domain Name System (mDNS) service records. It is the Apple's implementation of Zero configuration network (Zeroconf). Although good and very useful tool it is not supported on Android, and since Android is the platform chosen for the DigialLighter product further research must have been carried out. Zeroconf software written in Java with similar set of services as Bonjour software seemed to be the best option.

¹<http://www.apple.com/support/bonjour/>

Table 7.3: Project management stories selected for sprint 1

ID	Ref.	Description	Hours	
			Est.	Sp.
P1.1	WBS 7.1.1	As a student I have attend the weekly meetings with the customer	22	20
		Week I - Meeting	6	6
		Week I - Minutes	2	2
		Week II - Prepare demonstration	2	2
		Week II - Demonstration	6	6
		Week II - Minutes	6	4
P1.2	WBS 7.1.2	As a student I have to attend the weekly meetings with the supervisor	12	11
		Week I - Meeting	4	4
		Week I - Minutes	2	1
		Week II - Meeting	4	4
		Week II - Minutes	2	2
P1.3		As a student I need to attend the team building.	7	9
P1.4	WBS 7.3	As a student I need to participate to the lectures about team dynamics.	32	25
		Course of group dynamics.	30	24
		Summarize of course.	2	1
Σ			73	65

7.2.2 Zero configuration (Zeroconf)

Zeroconf² is a methodology and a set of special technologies that automatically create a usable wireless network based on the Internet Protocol Suite (TCP/IP) when computers or network peripherals are interconnected. It does not require manual operator intervention or special configuration servers. It is assembled of technologies that includes service discovery, addressing assignment, and host name resolution.

7.2.3 JmDNS library

Further research brought JmDNS³. Java implementation of multi-cast DNS that can be used for service registration and discovery in Local Area Network. It works on most JDK1.6 compatible Virtual Machines, it comes as a library and it is easy to integrate with Android. JmDNS fulfill all of the expectations, but while learning about it, team found out that Android itself have built in Network Service Discovery(NSD) that do the same thing.

7.2.4 Android Network Service Discovery (NSD)

NSD⁴ is supported from API version 16. It allows users to identify other devices on the local network, register services, broadcast connection information, scan for registered services and connect. Even with API limitation it is a part of Android platform, no third part libraries are needed, it will evolve with Android and therefore always be working. Min API version was not of concern to the customer, so team made a decision of using Android Network Discovery(NSD) for client-server core module.

7.3 Sprint goals

Team's goal for this Sprint is to deliver working demo over core client-server module. This includes registering services, listening for the client and sending simple signals to the client from the server application. Scanning for the services, connecting, receiving signals and play commands on the client. Establishing simple communication protocol and implementing Test-Flight in both applications, the server and the client.

The goal for the documentation is to have a good structure for the report, and integrate the project plan in the project report. It must be evaluated whether to write more or remove sections after integrating. The goal is also to finish the chapters that are a part of this sprint, which means that team need to finish the sprint 0 chapter, and write the sprint 1 chapter.

²<http://www.zeroconf.org/>

³<http://jmdns.sourceforge.net/>

⁴<http://developer.android.com/training/connect-devices-wirelessly/nsd.html>

7.4 Architecture

For the core communication and organization of our product the client-server architecture was selected as shown in Figure 7.1. Choosing this architecture was very intuitive to do as the project should consist of two applications and overall tasks should be partitioned. Client application should be able to light up different sequences of lights depending on server command signals. And server application should be responsible for awaiting incoming clients connections, mapping clients to the grid and providing command play signals to the clients by unicast and broadcast. Communication should be established over wireless network using same router for both applications, the user and the server, and without need for third party server nor Internet connection.

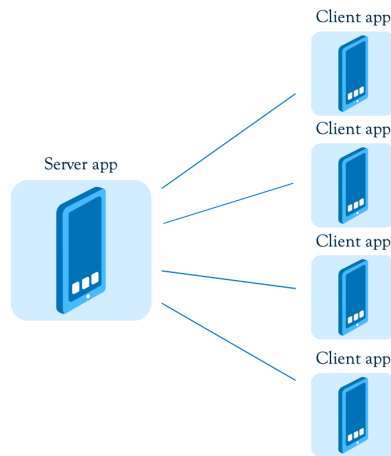


Figure 7.1: Client-server architecture

7.4.1 Logical View

Figure 7.2 gives an overview of the client class structure and collaboration. As planned client have all of the functionality covered by user stories.

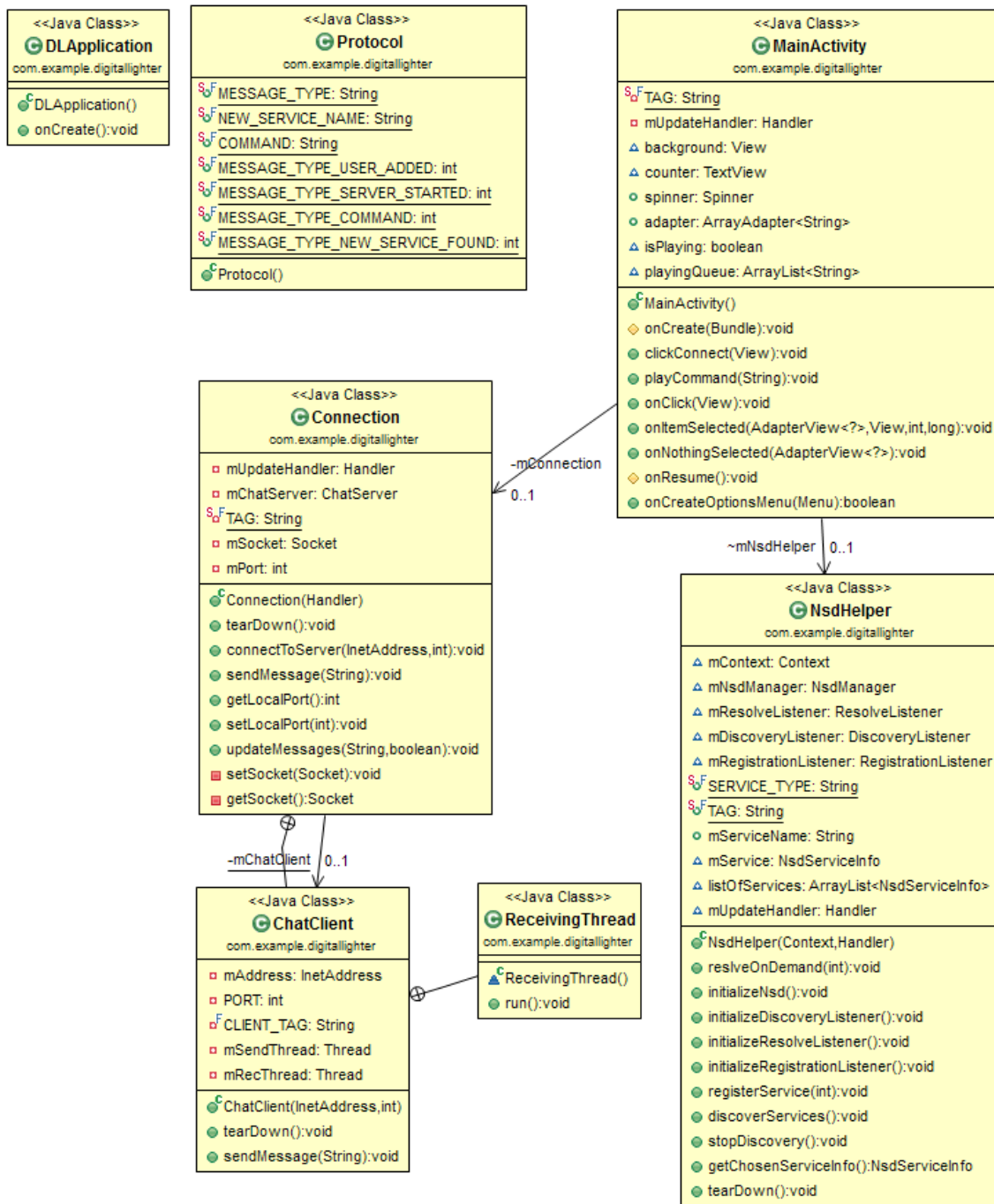


Figure 7.2: Sprint 1 client class diagram

Figure 7.3 gives an overview of the server class structure and collaboration. Separated thread "ServerThread" for incoming clients was made to optimize server. Sending command signals is also moved from UI thread to "SendingThread" in order to make application more responsive.

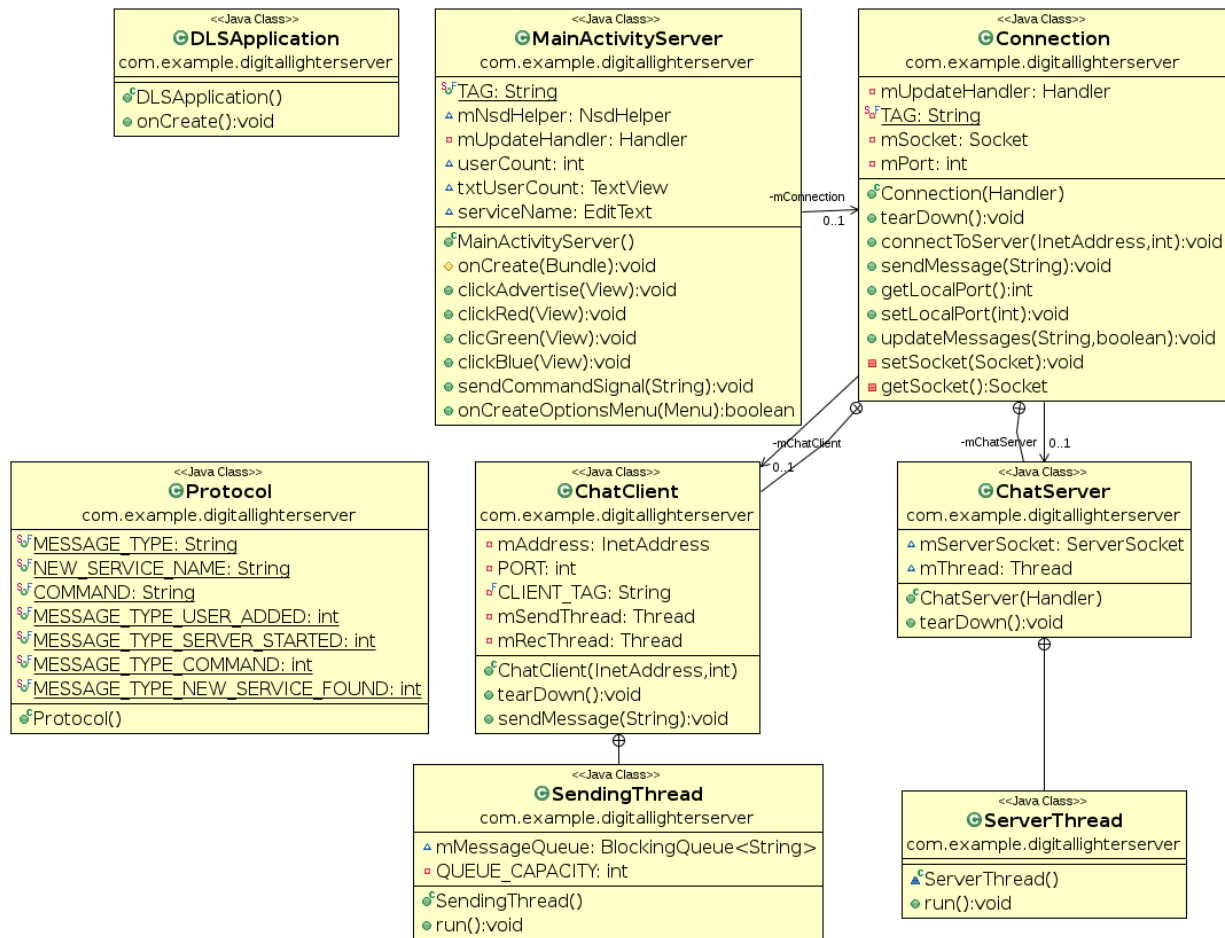


Figure 7.3: Sprint 1 server class diagram

On the sequence diagram in Figure 7.4 it is shown how applications can be used. It gives an idea of how the data flows between different parts of the system when it is up and running.

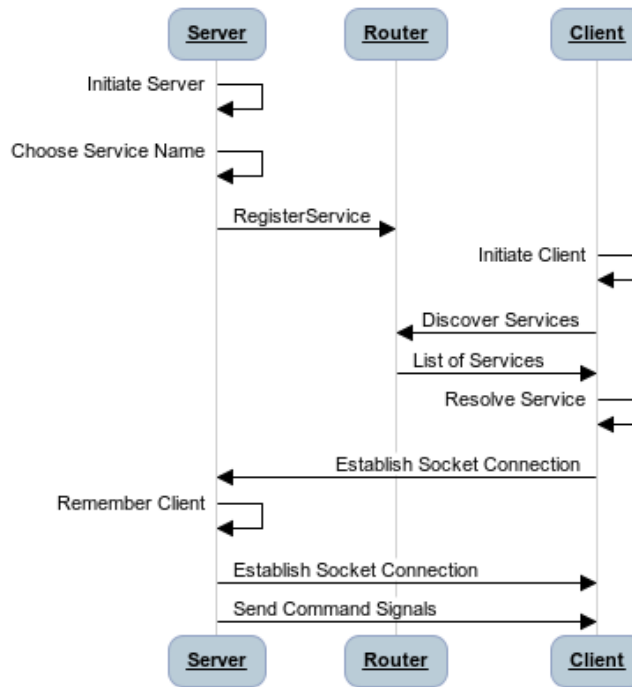


Figure 7.4: Sprint 1 Communication

7.4.2 Physical View

The deployment diagram, displayed in Figure 7.5 describes the run-time architecture of the system. It shows the configuration of the hardware elements (nodes) and shows how software elements and artifacts are mapped onto those nodes.

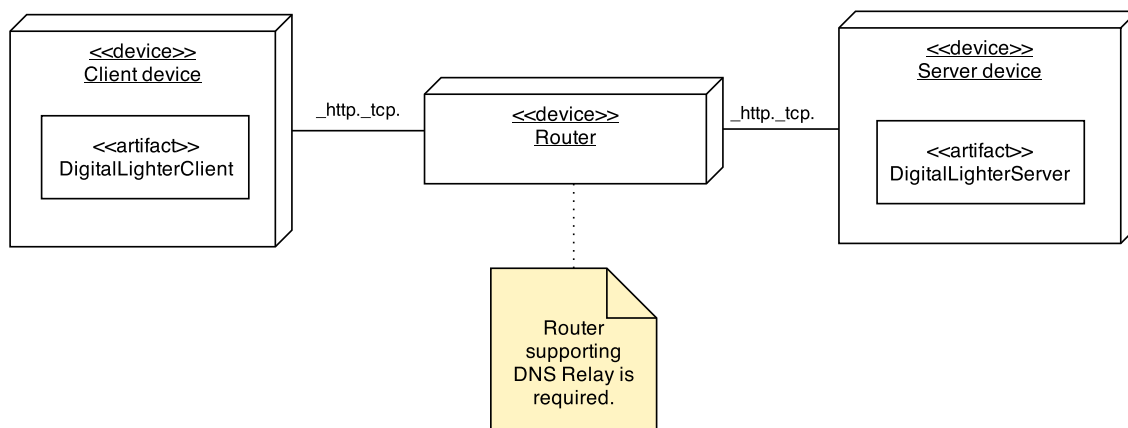


Figure 7.5: Deployment diagram

Client is the application that runs on audience Android device. It is able to discover available services, connects to the one and receives command signals from the server. After receiving commands the client is able to flash screen according to given signal.

Router is inevitable part of client-server communication since project is based on wireless network and zeroconf technologies. In order to support all of the demands it has to include a DNS relay service. The relay embraces storing of service names and allows look-up stored data to the clients.

Server is the application that runs on manager's Android device. It has to register service it provides at router DNS table. Once registered it have to be able to listen for the clients and broadcast command signals to them.

7.5 Occurring risks

The risk table 3.3, taken from the risk management section in the planning chapter, shows the risks that can occur this project. For this sprint most of them did not occur. This is because the hardest technical parts, which is the image processing, is not scheduled until the upcoming sprints.

The user stories for this sprint was very clear, and the customer did not change any of them in the last minute. The team overestimated some of the user stories, so pressure with the time frame was not experienced.

The first risk in the risk table describes team members being sick. This did not occur, but what did occur was that some of team members had to be absent in some of the working hours. The team solved this by having these team members work more independently, and with more flexible hours.

For now only Android 4.1 adds support for multicast DNS-based discovery. One of the risks is that Android will not make it available for lower platforms. In that case a bit over 50% of the devices will not be supported as shown on the Figure 7.6, and future development of our prototype will require some of the third part libraries we discovered in prestudy or similar. Supporting lower platforms in the future, or not, is however never stated in any official document of the Android documentation.

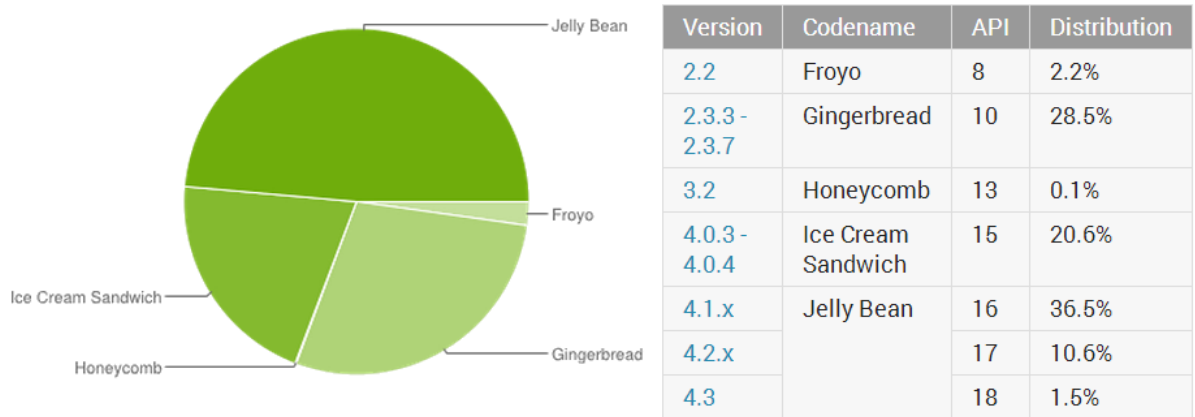


Figure 7.6: Android version distribution⁵. Data collected during a 7-day period ending on October 2, 2013. Any versions with less than 0.1% distribution are not shown

7.6 Customer feedback

The goal of this sprint was to implement the Prototype 1 as described in Section 3.7.3 but it was not specified how the working increment should be presented to the customer. The team decided to record a demonstration video and present it to the customer through the YouTube online service. The customer was delighted by this idea and encouraged the team to continue in recording the videos for the next demonstrations.

The backlog of user stories was presented to the customer so that he could choose the stories for the next sprint. Since he was satisfied with the team performance during last sprint customer suggested that the team should pick the stories itself as long as the choice suites the intention to finish Prototype 2 until the end of the next sprint.

Customer however disagreed with the time estimated for the single stories contained in the backlog and expressed his concern that the team might be underestimating the complexity of the given tasks. Therefore he changed the time estimated for almost each story by extending it.

The video presented to the customer can be found on YouTube under the name Prototype 1⁶.

7.7 Retrospective

This section reflects on the past sprint. In order to learn from the mistakes done and thus to improve the workflow it is necessary to answer two essential questions: "What went well" and "What could be improved".

Actual burn down chart can be seen on Figure 7.7.

⁵Source: <http://developer.android.com/about/dashboards/index.html>

⁶<http://www.youtube.com/watch?v=ofIa5QC6AKA>

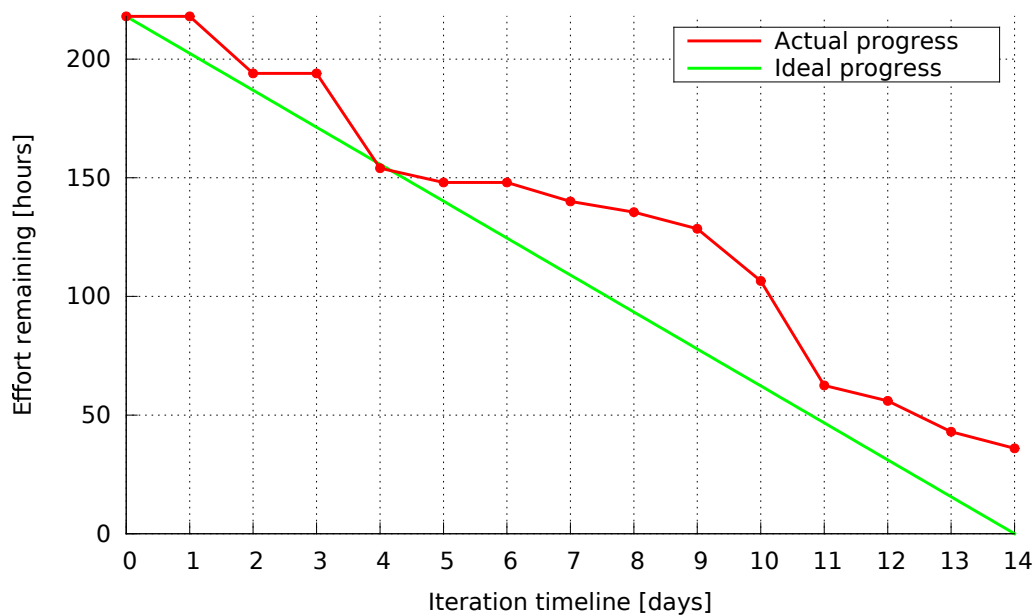


Figure 7.7: Burn down chart for sprint 1

7.7.1 What went well

For the implementation part the sprint 1 goals were reached. The working demonstration video over core client-server module was delivered. The server side registers services, listens for the client, and sends simple signals. The client is scanning for the services, connecting, receiving signals and play commands on the client. The customer was very satisfied with the video, and suggested recording our future prototypes as well. Besides this we reached the goal to incorporate TestFlight in both applications. Also milestone, Obedient client – Prototype 1, was reached.

For the documentation the work done in the report is good. The report is better structured now, and it actually looks like a report, which is good. The supervisor liked the structure as well. A lot of sections were added when the team integrated the project plan in the report, so that the introduction, the preliminary studies and the planning chapter would be as complete as possible. In for example the preliminary studies chapter the team has not written all of the studies about the image processing part yet, because the team will research this more in later sprints.

7.7.2 What could be improved

The sprint planning in this sprint could have been better. Some of the implementation user stories were over estimated. On the other hand, documentation user story **D1.3** was not completed. This was caused by lack of time due to refining older parts of report, which were marked as finished. To solve this better till next time the team should spend more time doing the planning poker. This makes sure that everyone understands the task better.

The structure of the report is good, but the number of chapters would probably have to be re-

duced. The supervisor suggested to move the test plan chapter in the planning chapter, since it was such a small chapter. The supervisor also wanted the team to work more with the requirements chapter. There should be a better connection between the user stories and the requirements.

Another con this sprint was that when the product was tested on the mobiles there were problems with the network. Since the mobiles were on different subnets, the client could not find the server. We had to borrow a router and bring with us to be able to test.

8 | Sprint 2

In this chapter the planning and work-flow regarding Sprint 2 will be described. Everything from setting the goals to implementation, and testing. At the end there will be evaluation of the whole sprint, and answers to following questions: What went well? What could be improved?

8.1 Sprint planning

The customer was very satisfied with the demonstration video for sprint 1, and suggested recording future prototypes as well. After the successful sprint 1 review, the planning of next sprint was started. Since the customer was very happy with the sprint 1 delivery, he gave the team an opportunity to decide for themselves what they want to work on in sprint 2, and then ask for his approval. It has been decided by the team to focus on multiple client support, and since sprint 1 just focused on connecting one client to the server, then it would also be a good opportunity to refine the code which was implemented so far. The customer thought this was a really good idea, because working on this now could help in future. Catching problems in early phases is much better than discovering them too close to the deadline. Then it might not be possible to have enough time to fix those problems. Therefore sprint 2 will focus on adding support for multiple clients connection and sending different signals to different clients. Existing code should be revised and improved so that in next sprint the main effort can be focused on image processing module.

In the supervisor meeting the day after the sprint review, the project report draft was shown to supervisor. He was satisfied with the new structure, but he wanted the team to finish more sections. He also insists on creating a work break down structure for the project. There were also some chapters that could be merged, which means we should refine the structure of the report. These suggestions were included into the sprint plan and according to time estimation some stories were adjusted.

All implementation related user stories for sprint 2 are presented in Table 8.1. All documentation related stories for sprint 2 are presented in Table 8.2. All project management related stories for sprint 2 are presented in Table 8.3.

Table 8.1: Implementation user stories selected for sprint 2

ID	Ref.	Description	Hours	
			Est.	Sp.
I2.1	M2	As a server I need to be able to have multiple clients connected.	35	30
I2.2	M6	As a server I need to be able to send multiple commands to all the connected clients.	6	4
I2.3	C3	As a client I need to be able to play the commands.	8	8
I2.4	M6	As a server I need to be able to send the same signal to multiple clients	8	6
Σ			57	48

Table 8.2: Documentation stories selected for sprint 2

ID	Ref.	Description	Hours	
			Est.	Sp.
D2.1	WBS 8.2	As a student I need to finish the pre-study chapter.	22	26
D2.2	WBS 8.2	As a student I need to finish the planning chapter.	16	20
D2.3	WBS 8.2	As a student I need to finish requirements chapter.	30	26
D2.4	WBS 8.2	As a student I need to finish the architecture chapter.	24	12
D2.5	WBS 8.2	As a student I need to finish sprint 1 chapter.	12	16
D2.6	WBS 8.2	As a student I need to write planning and architecture sections of sprint 2 chapter.	16	18
Σ			120	118

8.1.1 Duration

This Sprint will be 2 weeks long. From 16.09.2013 to 29.09.2013. The team agreed on the date of presentation and showing the running demo – Thursday 26.09.2013. Estimated velocity is 240h since team agreed on 30 working hours per person per week.

8.2 Sprint goals

The goal for Sprint 2 is to deliver a working demo with a more refined core client-server module. This still includes registering services, listening for the client and sending simple signals to the client from the server application, but with a more refined code. Of course the client should still scan for the services, connect, receive signals, and play the commands. The goal is still to use the

Table 8.3: Project management stories selected for sprint 2

ID	Ref.	Description	Hours	
			Est.	Sp.
P2.1	WBS 7.1.1	As a student I have attend the weekly meetings with the customer	22	18
		Week I - Meeting	6	6
		Week I - Minutes	2	3
		Week II - Prepare demonstration	2	2
		Week II - Demonstration	6	4
		Week II - Minutes	6	3
P2.2	WBS 7.1.2	As a student I have to attend the weekly meetings with the supervisor	12	9
		Week I - Meeting	4	3
		Week I - Minutes	2	3
		Week II - Meeting	4	1
		Week II - Minutes	2	2
		Σ	34	27

established simple communication protocol. All of the above is extended so it works for several clients connected to the server.

Goals about the report are to refine the report and write the sections needed to be up to speed with the report. This means that sprint 1 chapter has to be finished same as sprint 2 chapter.

8.3 Architecture

Biggest changes in architecture, comparing to previous sprint, is possibility of connecting multiple clients to the server and sending same commands with multiple playing instructions to the clients.

Adding support for multiple clients was straight forward. Changing one socket instance for array of sockets. Sending same signal to multiple clients is done by iterating trough all the sockets and sending same signal that have been previously sent to single client.

Change of the protocol and example of playing command instruction message is given in Implementation Section 8.4

8.4 Implementation

For sending/receiving commands with multiple playing instructions communication protocol had to be changed. New command message that server sends to client is shown on Figure 8.1.

#ff0000:5000|#00ff00:3000|#0000ff:1000

Figure 8.1: Command message with multiple instructions

Coding:

- color to play(hexadecimal),
- color-time separator,
- duration(milliseconds),
- one instruction separator.

Particular command on Figure 8.1 is composed of 3 instructions. First one is "Play red for 5 seconds", than "play green for 3 seconds" and on the end "play blue for 1 second".

Playing this messages on client side is matter of parsing the instruction with proper separator, and setting up time counter that will change background color in right moment.

8.5 Testing

During regular testing of application functionality, it was noticed that client get restarted from time to time. It was impossible to replicate the use case that lead to this error. Crashing and restarting of clients were random and rare. After catching some of the log error messages, it was possible to see that all of them have TestFlight tag. As a solution the TestFlight library has been removed from both applications, the client and the server. At dozens of additional tests error didn't occur and this issue is considered solved.

8.6 Occurring risks

The risk table, presented in Table 3.3, shows many different risks that can occur. For this sprint most of them did not occur. This is because the most challenging technical parts are not developed yet. This is scheduled for sprint 3.

The user stories for this sprint were also very clear, and really not that different from the last sprint. The only difference was that instead of one client, there are now several clients connected to the server. As mentioned earlier the customer was very happy with the demonstration in

sprint 1, and he let the team choose the stories for this sprint. He approved the chosen stories, and because of this no changes in requirements were not met in the last minute.

In this sprint the team also did a better job with the time estimation. All of this avoids a lot of issues, which is advantage.

8.7 Customer feedback

Similarly as in the previous sprint the demonstration video showing the resulting implementation of the Prototype 2 was first presented to the customer. According to the customer the video sufficed to prove the Prototype 2 was achieved and that all of the chosen user stories regarding the implementation were finished.

Contrarily to the previous sprint the customer decided to choose the user stories from the backlog for the next sprint himself. He again suggested the time estimation for specific stories that he considered to be of the greater difficulty. the team also suggested customer that the "bonus tasks" could be utilized. These tasks would be optionally added to the sprint backlog and if there is a time left the team would continue working on these tasks. Customer agreed with this suggestion.

As the team spend some time on carrying out the research related to the image processing it presented its discoveries to the customer. Specific techniques and algorithms which could possibly lead to the working implementation were suggested however the customer strongly disagreed with implementing the required image processing module "from scratch". The customer's idea was to exploit already existing solutions that might possibly exist and be available to free use.

Since the supervisor strongly suggested to list the compulsory requirements and agree with the customer on using them the team suggested both functional and nonfunctional requirements and presented them to the customer. Customer agreed which gave rise to the list of requirements specified in Section 4.2.

The customer was asked to give the advice on certain chapters of the report the team was working on and his main suggestion was to strictly follow the rules of UML specification when it comes to describing the architecture using UML diagrams. He also suggested to simplify the use-case diagram.

The video presented to the customer can be found on YouTube under the name Prototype 2¹.

8.8 Retrospective

This section reflects on the past sprint. In order to learn from the mistakes done and thus to improve the workflow it is necessary to answer two essential questions: "What went well" and "What could be improved".

Actual burn down chart can be seen on Figure 8.2.

¹<http://www.youtube.com/watch?v=98Q3nYTAx1w>

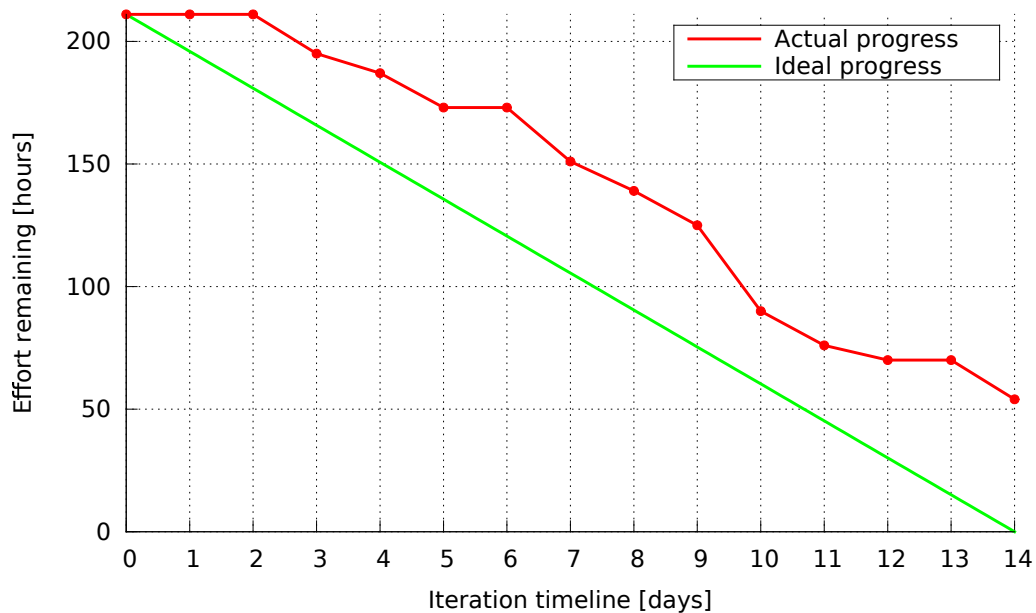


Figure 8.2: Burn down chart for sprint 2

8.8.1 What went well

The team reached the sprint 2 goals and completed all implementation stories. The team delivered the demo with more refined code, and as planned this code supported multiple clients. The clients were able to play the commands the server sent. Another milestone was reached – Obedient crowd – Prototype 2.

The work done in the report is very good – the report is better structured now. Some extra hours were spent on going through the whole report. This was to see if we need to add more in the specific sections, or if we should remove something.

Even though some members had other responsibilities, and had to absent in some of the working hours, the team still got everything from implementation part done in time for the sprint review. Everyone worked really well independently, as well as in team.

8.8.2 What could be improved

Documentation user stories **D2.3** and **D2.4** were not finished in this sprint as planned. The reason is again lack of time because a need for reviewing previously written parts of report and correcting them. Therefore writing documentation after the matter is settled and used should be performed, not documenting immediately.

When there is work that needs to be done in the report, the person who writes this section should also read terminology, or what other team members have written earlier. This will help to get a better flow for the reader. This will also save a couple of hours when refining is needed afterwards.

9 | Sprint 3

In this chapter the planning and work-flow regarding Sprint 3 will be described. Everything from the setting of our goals to implementation and testing. At the end team will evaluate the whole sprint, and try to answer on following questions: What went well? What could be improved?

9.1 Sprint planning

In the planning part of sprint 2, there have been introduced an idea or goal for refining and extending the existing code, customer agreed, with condition, that sprint 3 will be focused on image processing part.

As the image processing module was one of the major risks, there was done some preliminary research even in the end of sprint 2 about possible approaches. One specific way to deal with the problem (using OpenCV and Hough transformation [4]) was introduced to customer in planning part of meeting. Customer was not satisfied with such a low-level approach and wanted us to use some existing tools. Therefore there was need for additional preliminary studies concerning existing projects or libraries that could be used. You can read more about additional preliminary studies below.

Also there was made a proposal of additional "pre-demo" video showing the progress of implementation on Thursday 4th of October 2013 during the regular meeting and customer gladly accepted this offer.

All implementation related stories for sprint 3 are presented in Table 9.1.

All the documentation related stories for sprint 3 are presented in Table 9.2. All the project management related stories for sprint 3 are presented in Table 9.3.

9.1.1 Duration

This sprint is 2 weeks long. From 30th of September 2013 to 13th of October 2013. We agreed on the date of presentation and showing the running demo – on Thursday 11 of October 2013. Estimated velocity is 240 hours since we agreed on 30 working hours per person per week.

Table 9.1: Implementation user stories selected for sprint 3

ID	Ref.	Description	Hours	
			Est.	Sp.
I3.1	M4	As a server I need to be able to process images from the real world.	32	28
I3.2	WBS 6.2	As a server I need to be able to process images from the virtual world.	20	23
I3.3	M4	As a server I need to be able to detect one client from its light.	40	46
I3.4	M4	As a server I need to detect multiple clients from light.	5	8
I3.5	M4	As a server I need to map all available devices to grid.	10	10
Σ			107	115

Table 9.2: Documentation stories selected for sprint 3

ID	Ref.	Description	Hours	
			Est.	Sp.
D3.1	WBS 8.2	As a student I need to finish the requirements chapter.	18	16
D3.2	WBS 8.2	As a student I need to finish the architecture chapter.	26	20
D3.3	WBS 8.2	As a student I need to finish the sprint 2 chapter	16	20
D3.4	WBS 8.2	As a student I need to write planning and architecture sections of sprint 3 chapter	34	30
Σ			94	86

9.2 Preliminary studies

As explained in Section 9.1 the customer was presented the possible technical approaches which, if followed, might lead to the successful implementation of the image processing (device detection) module. The customer is well aware of the fact the image processing part itself should be considered rather complex and that it might be tricky to achieve good results while attempting to detect the lit devices' screens. Therefore he stressed out it was his strong requirement that the team would find, research and utilize already existing solutions.

One of the customers advices on this topic was to contact the project supervisor and ask him to either recommend the relevant department in NTNU or even better the specific persons who might be able to provide needed know-how or at least suggest which approach to choose in order to eventually obtain the most precise results. If the team were to fail getting the relevant information the reasonable amount of time should be spent on carrying out additional research

Table 9.3: Project management stories selected for sprint 3

ID	Ref.	Description	Hours	
			Est.	Sp.
P3.1	WBS 7.1.1	As a student I have attend the weekly meetings with the customer	20	13
		Week I - Meeting	6	3
		Week I - Minutes	2	1
		Week II - Prepare demonstration	2	5
		Week II - Demonstration	6	3
		Week II - Minutes	4	1
P3.2	WBS 7.1.2	As a student I have to attend the weekly meetings with the supervisor	12	7
		Week I - Meeting	4	2
		Week I - Minutes	2	2
		Week II - Meeting	4	2
		Week II - Minutes	2	1
		Σ	34	20

regarding existing solutions, customer stated.

The team's supervisor suggested two persons working for IDI¹ department at NTNU who might be of any help. Nevertheless, nor either of them or the other persons who they suggested were able to recommend any existing solution. Thus the team spend more time on researching the existing solutions using online accessible sources. The outcomes were as follows:

9.2.1 Shape detection

Since the objective is to detect the device's display which is lit and surrounded by the relatively dark surroundings (and thus sufficiently contrasting) the problem of detecting the device can be simplified to basically detecting the known shape - rectangle. Even though the detection of the specific shape is well described in many online sources including the OpenCV documentation². The algorithm is based on transforming the image into grayscale, blurring it, thresholding and detecting the edges. Though the team were not able to find any existing solution which could be used right away as the customer required.

¹<http://www.idi.ntnu.no/>

²http://docs.opencv.org/doc/tutorials/imgproc/shapedescriptors/find_contours/find_contours.html

9.2.2 Color blob detection

Another widely used approach comprises of detecting the areas of the specified color, so called *color blobs*. This technique seemed to be even easier to implement as compared to the shape detection. What is more the libraries and working examples exist and those can be utilized conveniently.

CVblob-for-android

The open-source library CVblob-for-Android³ was specifically designed for Android platform and it builds on the OpenCV library *cvblob*⁴. It allows for the detection of the blobs of the specified color and it even features the center of mass estimation. Nevertheless after inspecting the code it was decided to choose the alternative code that provides the OpenCV among its functioning examples (see Section 9.2.2) and which is easier to adjust.

OpenCV color-blob-detection

OpenCV4Android SDK includes a few examples demonstrating the capabilities of the library. One of them, *color-blob-detection*⁵ turned out to suite the needs of DigitalLighter project the best. The example enables the user to select arbitrary color that would be used to detect color blobs. It was rather straightforward to adjust the source code so that the specifically chosen colors would be used to detect the devices' screens. Even though the obtained code cannot be considered as an already working technology as the customer required, given the outcomes of the research and the current knowledge of the team's members this was the most reasonable solution.

9.3 Sprint goals

The goal of this sprint is having a working application on a mobile phone that can work in two modes. Input type and output are for both modes common – input is an image or video and output is location of detected mobile phones in a given matrix (e.g. 4x4 matrix) with color that mobile's screen is lighting. In the first mode, real video from mobile's camera will be treated as an input and on the other hand in the second mode mock data (image or video) are treated as an input.

You can see example of input with matrix 2x2 in Figure 9.1. Appropriate output of device detection module is (left top tile is [0,0] and right top tile is [1, 0]): {blue,[0,1]}, {red,[0,0]}, {green,[1,1]}.

³<https://code.google.com/p/cvblob-for-android/>

⁴<https://code.google.com/p/cvblob/>

⁵<https://github.com/Itseez/opencv/tree/master/samples/android/color-blob-detection>

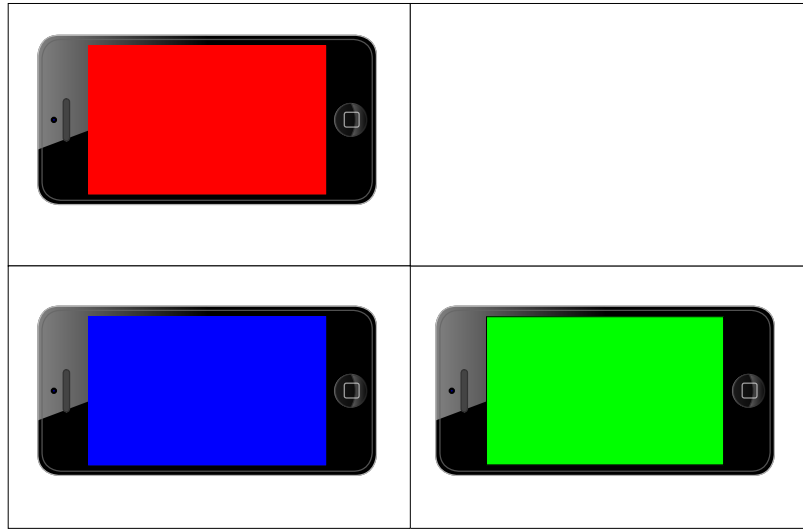


Figure 9.1: Example of ideal data input for device detection module with 2x2 matrix

9.4 Architecture

In this section it will be described device detection component module using 4+1 architectural view model.

9.4.1 Logical view

You can see a class diagram of new classes created in sprint 3 in Figure 9.2. We can divide these new classes into three categories.

Into the first category we count classes **LightDetector**, **TileMapper** and **PointCollector**. These classes are a core of device detection module. The only visible method for working with this module is **PointCollector**'s method **collect**. This function accepts two parameters: first is the image where mobile phone's screen detection should be done and second is a list of colors of screens which should be detected. Method **collect** is run asynchronously (as a new thread due to possible high time demanding operations) and therefore it was designed as a design pattern **Observer**. As class **PointCollector** implements interface **Observable** (also known as a *Subject* [9, p. 326]), its **Observers** must implement method **update**. To this function is passed as a argument hash map with keys as a colors and values as a lists of tile positions with appropriate color. Class **LightDetector** is responsible for detecting location of color blobs in given image. Results as a list of pixel's position of blobs are passed as a return value of function **getBlobCoords**. These values are fetched by instance of **PointCollector** and passed to **TileMapper**, which is responsible for mapping points into appropriate tiles in grid.

In second category there is only class **CameraActivity**, which is responsible for handling outputs from camera or mock device and also gives appropriate feedback on screen of mobile (draws grid and marks detected blobs).

In last category there is class **ColorManager**, which is standalone class and in application there is only need for one instance. Therefore it was designed as a **Singleton** pattern [9, p. 144]. Its main purpose is to handle transformations of different color formats such as special library, network and internal formats.

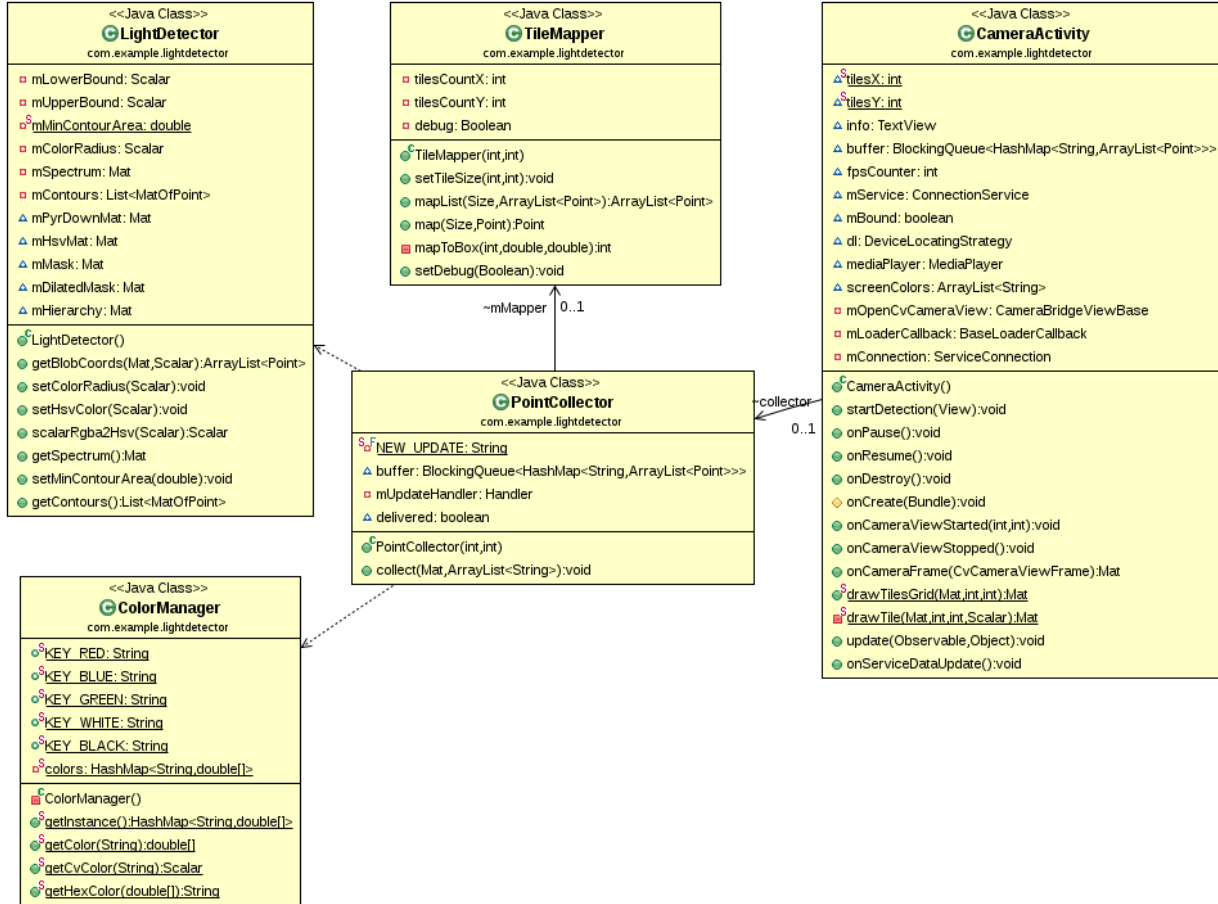


Figure 9.2: Sprint 3 device detection module class diagram

9.4.2 Physical view

You can see the physical view of whole product represented in Figure 9.3. Even though in this particular case device *Camera* part of *Server* device, the architecture and code was designed generally so *Camera* can be stand alone device.

9.4.3 Process view

You can see the process view represented in Figures 9.4 and 9.5. The activity diagram, displayed in Figure 9.4, can be perceived as a subactivity diagram of action *detect clients location* from

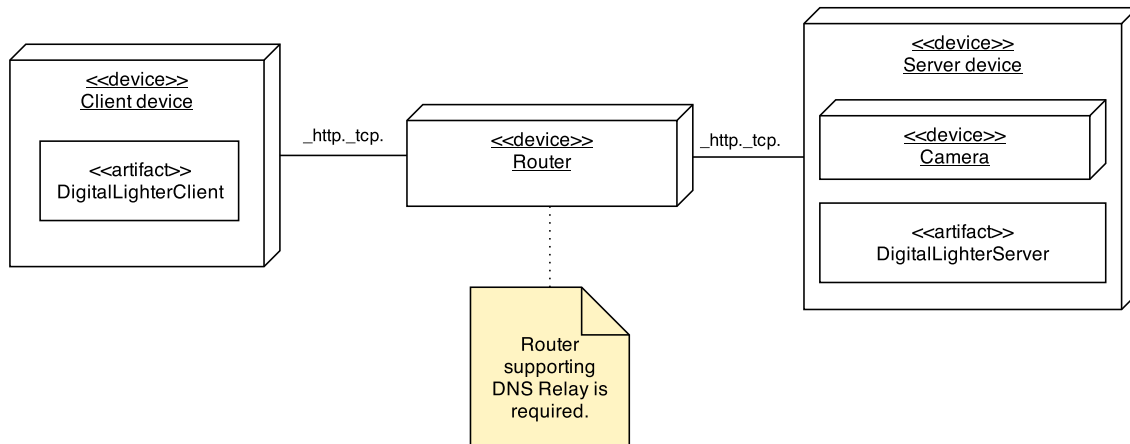


Figure 9.3: Deployment diagram

Figure 5.5. It should be mentioned, that each time new image was taken by camera, new thread is created and therefore several processing in the same time can be performed.

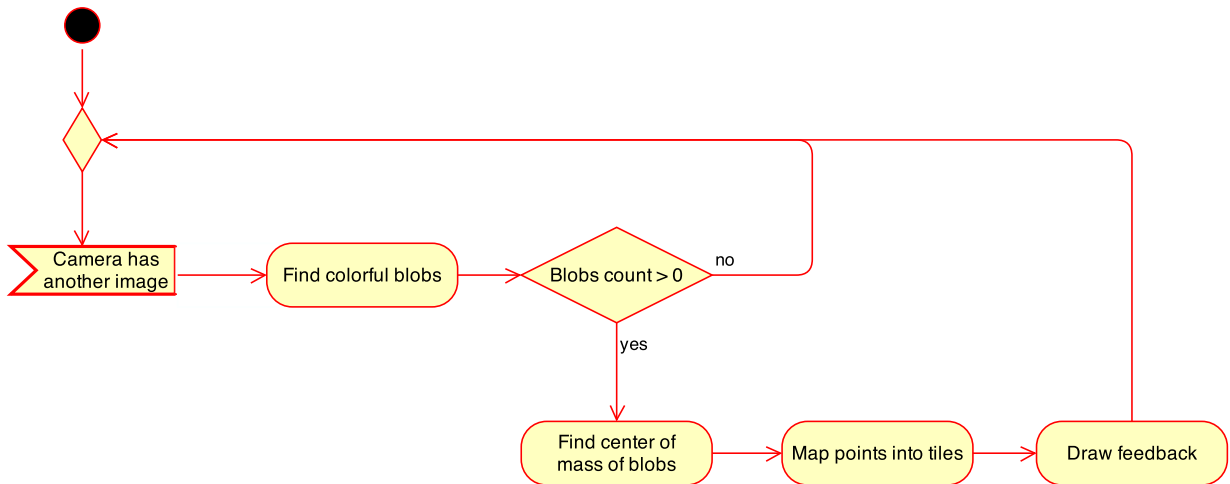


Figure 9.4: Sprint 3 activity diagram

9.5 Implementation

Few problems during implementation had occurred. Since the OpenCV library for Java is in early stage, there is some functionality missing. One of these is a method `open(String)` of class `VideoCapture`⁶. In C++ version of OpenCV there exist such a method and it allows programmers to use a video for input for image processing. After short research, a fix of this bug was found⁷, but this feature will be added in release 2.4.7. Due to this discovery, there has been abandoned

⁶<http://docs.opencv.org/java/org/opencv/highgui/VideoCapture.html>

⁷<http://code.opencv.org/issues/3207>

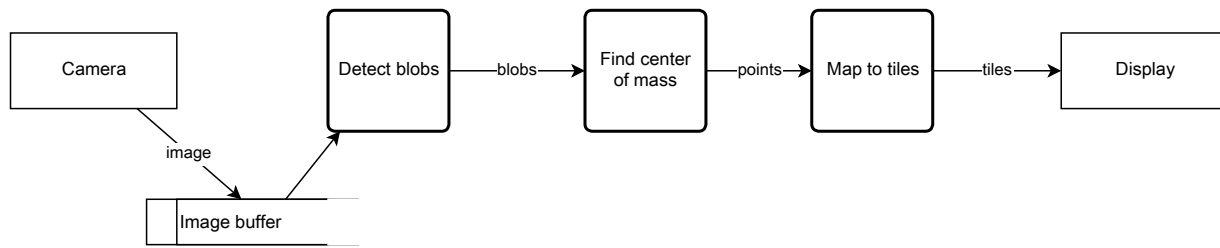


Figure 9.5: Sprint 3 data flow diagram

(at least until OpenCV 2.4.7 is released) a plan for mocking a video and only single pictures were used for testing.

9.6 Testing

A capability of Java to run on multiple platforms were utilized during testing and initial testing was performed without any Android device. There have been created a testing set⁸ of simple images used for black box testing.

After this test, it has been decided to merge module with the rest of the application and perform some integration tests. To outline a characteristics of further testing you can see videos^{9 10} demonstrating current progress of implementation.

9.7 Occurring risks

This sprint, the *dead end with technology* item from Table 3.3 occurred. Since the required functionality (loading mocked videos) was proposed by customer mainly from testing reasons, the fact that the team met dead end was announced to customer with proposition, that the team can spend more time on researching other solutions or simply the final product will lack this feature. The customer have chosen second choice.

9.8 Customer feedback

Since the implementation went better then expected the team was able to present the demonstration video already after one week. This video shows the requirements that Prototype 3 specifies (see Section 3.7.3) so the customer was pleased by the fact that the team is ahead of plan. It was therefore agreed that the team would attempt to fulfill at least some of the requirements that Prototype 4 specifies. After next week additional demonstration video was presented to the

⁸<https://github.com/dohnto/DigitalLighter/tree/master/source/others/LightDetectorTester/res/drawable>

⁹<http://www.youtube.com/watch?v=TcuMlvvAwSQ>

¹⁰<http://www.youtube.com/watch?v=fhWFAJY7Q0g>

customer. It showed the ability of multiple clients to be recognized by the server simultaneously. Customer was satisfied with this solution.

During the implementation the team realized it might be tricky to achieve good results as far as the image processing is concerned. Thus it suggested that the acceptance tests could be presented to the customer in advanced so that he would be able to decide whether the project is being developed in the way he requires. Acceptance tests might be created in the form of the images or the conceptual videos that the system must be able to process and/or display correctly. The customer stated this is very mature suggestion and agreed with that.

The customer also added the user story regarding collecting the mobile devices for the final presentation as this might not be easy to achieve and it must be plan well ahead of time.

The first video presented to the customer can be found on YouTube under the name Prototype 2.1¹¹. The second video is published under the name Prototype 3¹².

9.9 Retrospective

This section reflects on the past sprint. In order to learn from the mistakes done and thus to improve the workflow it is necessary to answer two essential questions: "What went well" and "What could be improved". You can see the burn down chart in Figure 9.6.

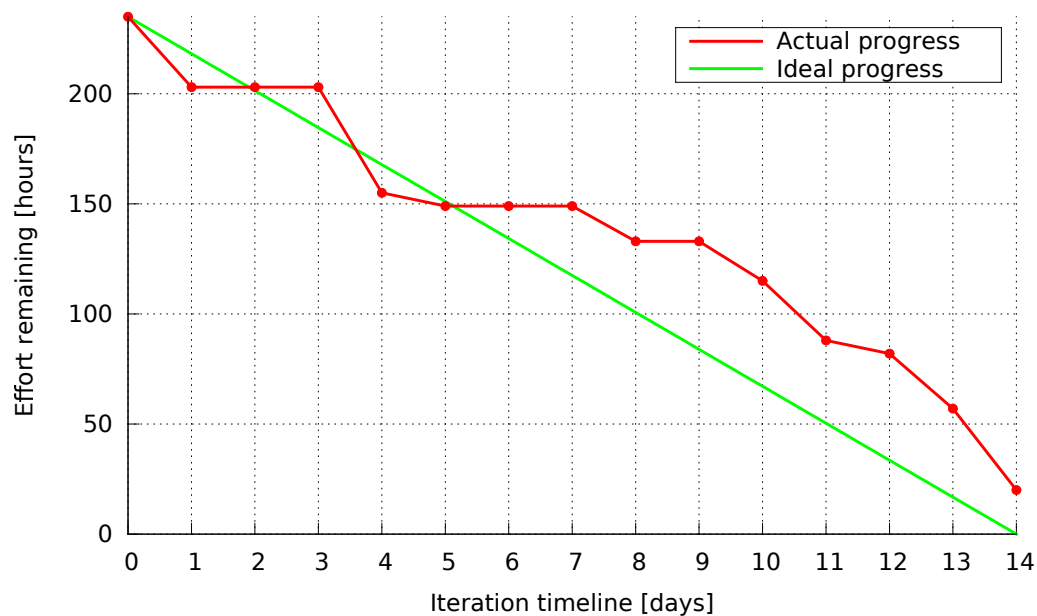


Figure 9.6: Burn down chart for sprint 3

¹¹<http://www.youtube.com/watch?v=TcuM1vvAwSQ>

¹²<http://www.youtube.com/watch?v=fhWFAJY7Q0g>

9.9.1 What went well

Since the preliminary studies concerning image processing started very soon and instead of writing device detection module from scratch an existing code was used, the core of implementation was finished ahead of schedule. Also all documentation stories were reached according to plan.

9.9.2 What could be improved

Due to misunderstanding about customer's demands regarding using existing solution, extra time was required for additional preliminary studies resulting to the same output as the first preliminary study concerning image processing.

In the end, the customer approved using OpenCV as a best option. Therefore the communication with customer should be more precise and accurate and if any ambiguity occurs, it should be consulted with customer as soon as possible.

From the beginning it was obvious, that the device detection module's performance is affected by level of light. During the testing the team was uncertain when the performance is sufficient, because the question of lighting was not discussed enough. After this experience, an proposal to customer was made: if similar situation occurs, acceptance tests should be prepared and approved by the customer.

Last but not least, implementation story **I3.2** was not finished, due to *dead end* described in Section 9.5.

Also, daily stand ups were often omitted due to late attendance. Even though the workload is 30 hours per week per person, the efficiently spend time is much lower, therefore the workload was decreased to 25 hours per week per person (it better reflects the reality).

10 | Sprint 4

In this chapter the planning and work-flow regarding Sprint 4 will be described. Everything from setting our goals to implementation and testing. At the end the team will evaluate the whole sprint and try to answer the following questions: What went well? What could be improved?

10.1 Sprint planning

The team with the customer planned how to take the project further on a regular planning meeting. At this point it was possible to send control signals to multiple clients, and then detect the screens that lit up. It was decided that the plan for this sprint was to combine those two modules, and try to display a short animation. This meant that detection of possible clients' locations must be performed and then connecting its position in a grid with an appropriate socket linked with client.

After this locating, commands to all clients must be continuously send with information, what colors should they lit. To be more efficient and do not use too much traffic, which could result into delays on network, it was proposed by customer that video should be preprocessed and large chunks of data with timing should be sent to clients.

Next, team discussed with the customer long term planning of the product. Since this product cannot be finished and deploy for everyday and commercial use due to time restriction, the features included and not included in final prototype were discussed. The customer proposed two different goals. One involved mobile device tracking (on rock concert after detection phase, it is probable, that audience wont be static but they will be moving, and therefore if animation should be displayed in proper way, these movements should be detected). After a short discussion, it was decided that this is a long term goal, which is out of scope of this project. The focus was put on finishing all core modules and therefore a different goal was adopted. This goal will be described in Section 10.2.

As the team wanted the customer feedback more often, it was again settled, than a next pre-delivery demonstration will be made for meeting on 17th of October.

All implementation related stories for sprint 4 are presented in Table 10.1.

In sprint 4 the team also planned to work more on the report. The plan was to finish sprint 3, since that sprint itself finished the week before. It was also wanted to start working on sprint 4 in report. Since more modules were implemented and their integration must be done, discussion

Table 10.1: Implementation user stories selected for sprint 4

ID	Ref.	Description	Hours	
			Est.	Sp.
I4.1	M4	As a server I need to link the devices' location with their ids.	52	48
I4.3	M6	As a server I need to play the whole media to the grid.	47	44
Σ			99	92

about architecture started to be intensive. The main goal was to keep architecture as general as possible and open for all future (out of scope) features. Therefore the idea of architecture become more clear and work on writing architecture chapter could start. The team where also getting a better understanding of what the end product would look like, therefore it was planned to start working a little bit on the evaluation as well. The supervisor also came up with some suggestions for small improvements on the report, which the team planned to follow up. Also it was decided to make to the user stories more consistent. Last but not least, it was also planned to separate the implementation stories from the documentation, and the project management stories.

All the documentation related stories for sprint 4 are presented in Table 10.2.

Table 10.2: Documentation stories selected for sprint 4

ID	Ref.	Description	Hours	
			Est.	Sp.
D4.2	WBS 8.2	As a student I need to finish the architecture chapter.	26	21
D4.3	WBS 8.2	As a student I need to finish the sprint 3 chapter	10	8
D4.4	WBS 8.2	As a student I need to write planning and architecture sections of sprint 4 chapter	15	15
D4.4	WBS 8.2	As a student I need to create structure and write introduction for the evaluation chapter	20	15
Σ			71	59

All the project management related stories for sprint 4 are presented in Table 10.3.

10.1.1 Duration

This sprint is 2 weeks long. From 14th of October 2013 to 27th of October 2013. We agreed on the date of presentation and showing the running demo – on Thursday 25th of October 2013. Estimated velocity is 200 hours since we agreed on 25 working hours per person per week.

Table 10.3: Project management stories selected for sprint 4

ID	Ref.	Description	Hours	
			Est.	Sp.
P4.1	WBS 7.1.1	As a student I have attend the weekly meetings with the customer	20	14
		Week I - Meeting	6	3
		Week I - Minutes	2	2
		Week II - Prepare demonstration	2	2
		Week II - Demonstration	6	4
		Week II - Minutes	4	3
P4.2	WBS 7.1.2	As a student I have to attend the weekly meetings with the supervisor	6	5
		Week I - Meeting	0	0
		Week I - Minutes	0	0
		Week II - Meeting	4	3
		Week II - Minutes	2	2
		Σ	26	19

10.2 Sprint goals

The goal of this sprint is to have application which integrates already implemented modules. This application should be able to display using clients a short animation of a traffic light as can be seen in Figure 10.1. This is in fact prototype 4 – *Traffic light control* as can be seen in Section 3.7.3.

10.3 Architecture

In this section it will be described architecture for sprint 4 using 4+1 view model. The implementation work can be divided into two logical parts: linking clients' sockets with their position in grid and processing media (animation) into chain of commands for clients. Therefore the architecture description will be in each view divided into two parts.

10.3.1 Logical view

In Figure 10.2 can be seen class diagram of device locating component. On the top of the hierarchy there is an interface `DeviceLocatingStrategy`, which defines common behavior for two classes: `DeviceMapper` and `DeviceTracker`. The first mentioned is an abstract class and defines shared behavior for all (future) types of implementations of linking sockets with positions. One of these



Figure 10.1: Traffic light

classes is `DeviceMapperSimple` which uses simple dummy algorithm. This algorithm simply let lit clients with white color one by one and detects this color. Detected tile is position of a mobile device.

Class `DeviceTracker` is class that is used as a getter of list of devices with their positions. Core of this class is missing, it is only present so it can be extend as a future feature. Future purpose of this class is to track already detected devices and update their positions during playing media.

In Figure 10.3 you can see class diagram of second component – media player. Class `ImageMapper` is responsible for loading single pictures. These pictures are used by class `CommandCreator`, which can create commands generically during run and as a parameter can be passed number of frames that should be parsed in advance and send to client in single command. For sending to clients in appropriate moments is responsible instance of class `MediaPlayer`. As you can see, the `MediaPlayer` class keeps an instance of class `DeviceLocatingStrategy` which servers as a getter for sockets and positions.

10.3.2 Physical view

In this sprint, no changes were made in physical view. Therefore you can see the physical view in deployment diagram in Figure 9.3 from sprint 3.

10.3.3 Process view

Process view for sprint 4 can be seen in activity diagram in Figure 10.4. First, to all clients is sent control signal to lit with black color. After some delay (because of possible network delay, there is

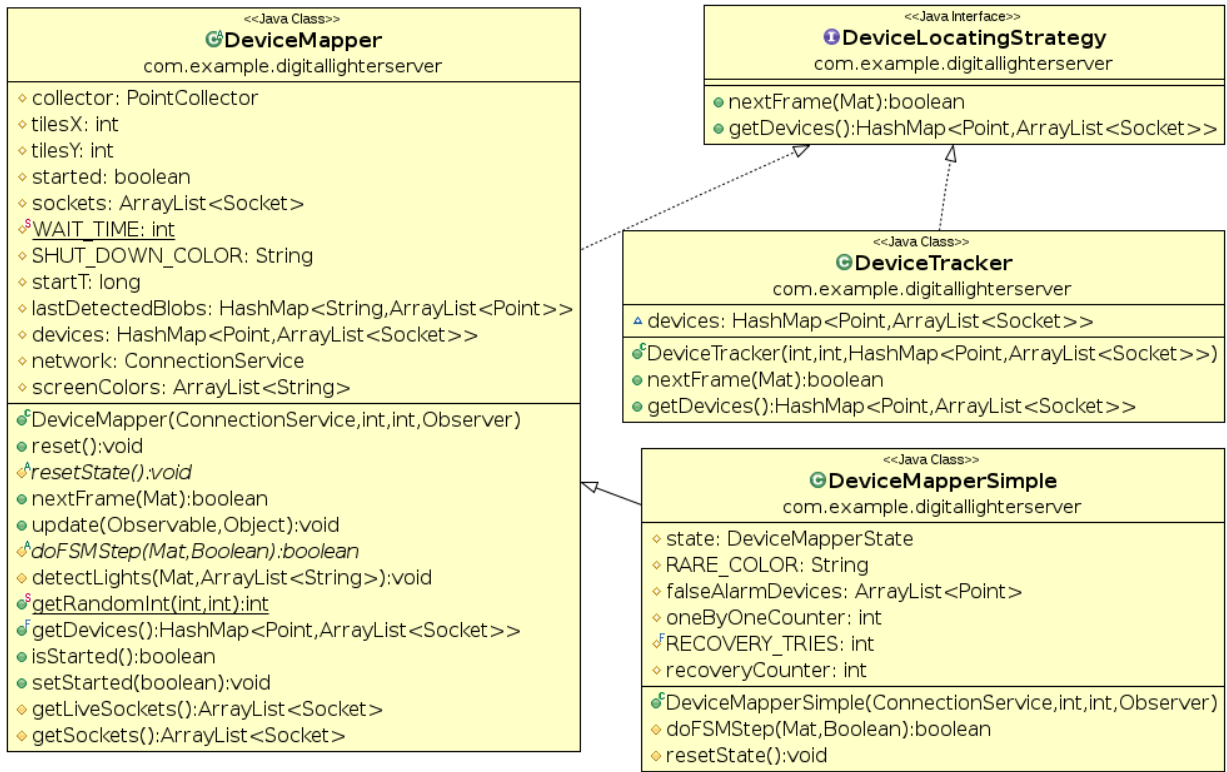


Figure 10.2: Sprint 4 device locating module

need for waiting), all white blobs are detected, this represents *false alarm* blobs. Next, iterating through all clients starts. In each iteration to only one clients is sent to lit with white color. Again, after some delay a new detection is performed and difference between currently detected white blobs and false alarm blobs is calculated. The difference is a position of client.

It should be mentioned, that both activities *Detect white color* are in fact slightly modified subactivities, which can be seen in Figure 9.4. Therefore this diagram in Figure 10.4 is subactivity of activity *Detect clients location* of Figure 5.5.

The media player component is from point of process view rather straightforward, and therefore there will be no figure presented.

10.4 Testing

There was no unit testing for device locating component, because mocking of network module, which is needed for this testing was estimated with too high value. Therefore testing was performed with all modules integrated. The team focused on finding the lowest value for waiting which can be seen in Figure 10.4. This value was set on 200 milliseconds due to empirical research. It should be mentioned, that this value suppose very low delay between sending lit control signal and its performing by client. Three mobile devices were used for testing purposes. You can see

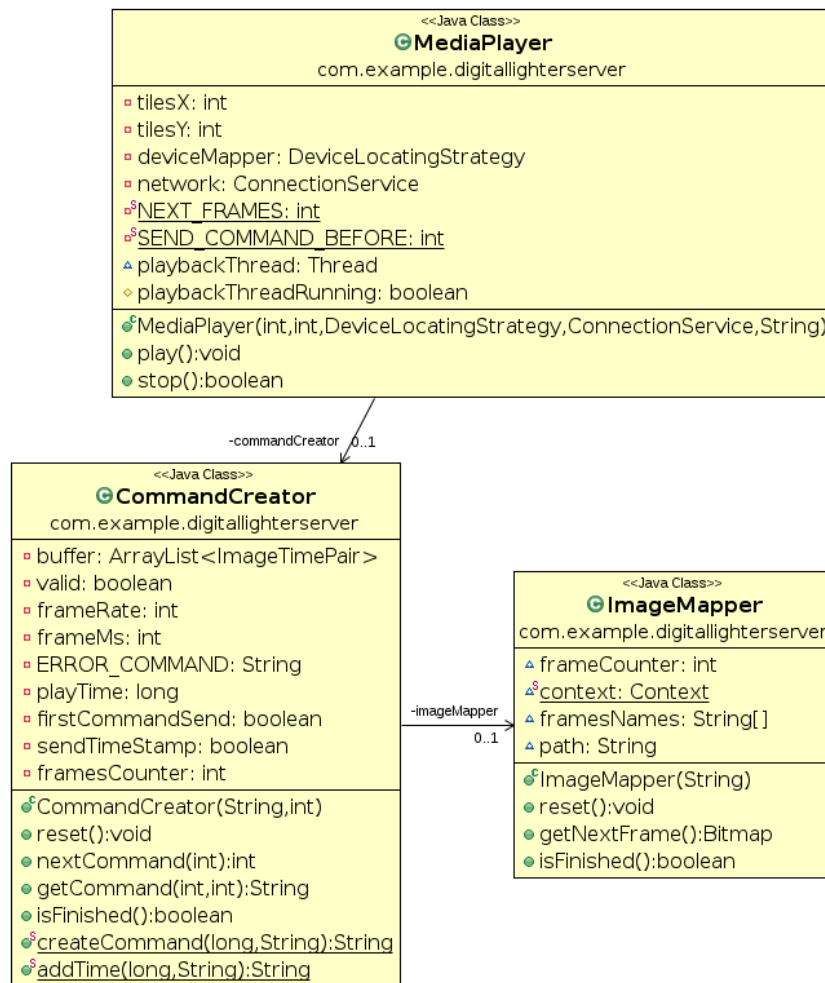


Figure 10.3: Sprint 4 media player module

an example of the form of testing in prototype 4 demonstration video¹.

Media player component did not require connected clients and therefore unit testing could be performed. For this testing, traffic light media² was chosen. After verification, component was integrated to the system and integration testing in real situation was performed.

10.5 Occurring risks

In this sprint, working prototype was very close to final product. Almost all code modules were implemented and integrated and therefore testing of whole prototype could be done quite exhaustively. Unfortunately there was lack of mobile devices and also the team could not borrowed

¹<http://www.youtube.com/watch?v=6mV5ajoZomI>

²<https://github.com/dohnto/DigitalLighter/tree/master/source/DigitalLighterServer/assets/3x3/traffic-light>

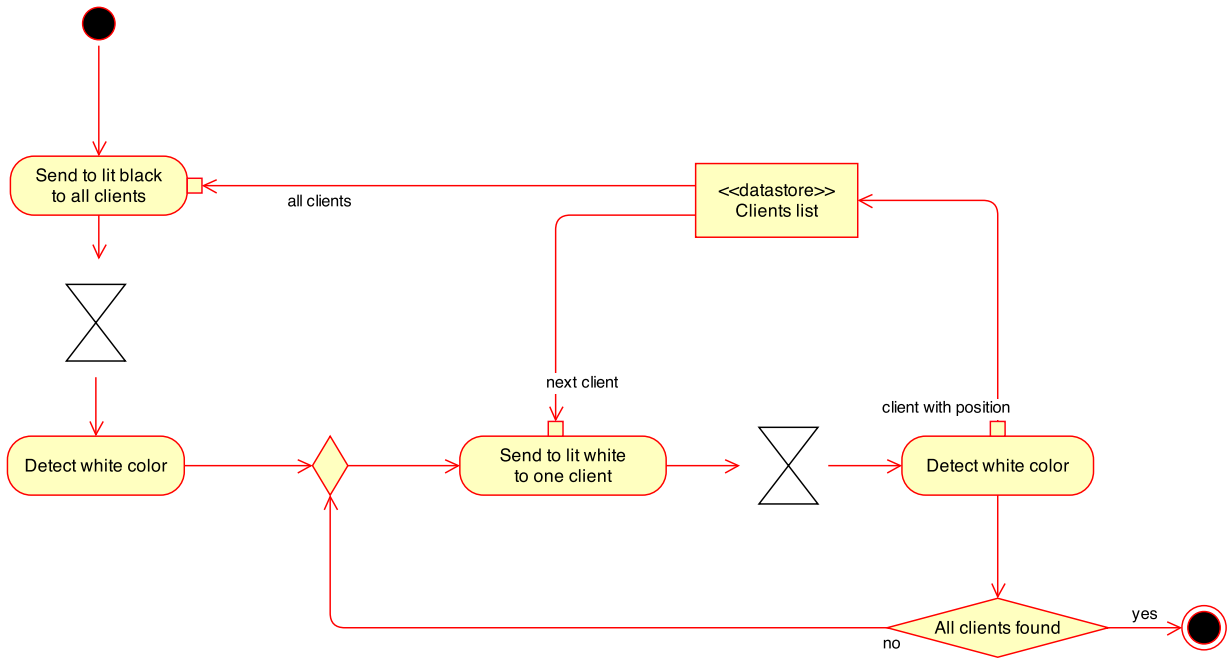


Figure 10.4: Sprint 4 activity diagram

any router supporting DNS relay. Both customer and supervisor were asked for advice where to gain required hardware, but only negative or very general hints were provided. Therefore the testing was performed just with few devices, which cannot be counted as complete. There is possibility, that server application is going to have problems with large number of clients.

One team member announced very early in the process phase his absence for second week of this sprint. Since the absence was planned for in advance, the team was able to deal with this problem. Also the missing member was active beyond working hours.

10.6 Customer feedback

As the team succeeded to implement the most parts of Prototype 4 (see Section 4.2) the only part left was to create the demonstration of the traffic light, the three (or more) devices ordered vertically displaying red, yellow and green color, the setting reminiscent of the ordinary traffic light. The team managed to finish the implementation in the first half of the sprint so the demonstration video was presented to the customer one week in advance.

The customer expressed his contentment with the goals achieved. He however stated the team should focus more on the speed of detection of the devices. Using current "one-by-one" approach is too time consuming. What is more from now on the team should focus mostly on coming up with the ideas how to present the product in the most immersive way.

On the second meeting during this sprint the team suggested the ideas about what to utilize the created product for. The customer decided to choose the equalizer video effect that would be

displayed on the phones while accompanying the played music.

The video presented to the customer can be found on YouTube under the name Prototype 4³.

10.7 Retrospective

This section reflects on the past sprint. In order to learn from the mistakes done and thus to improve the workflow it is necessary to answer two essential questions: "What went well" and "What could be improved". You can see the burn down chart for sprint 4 on Figure 10.5.

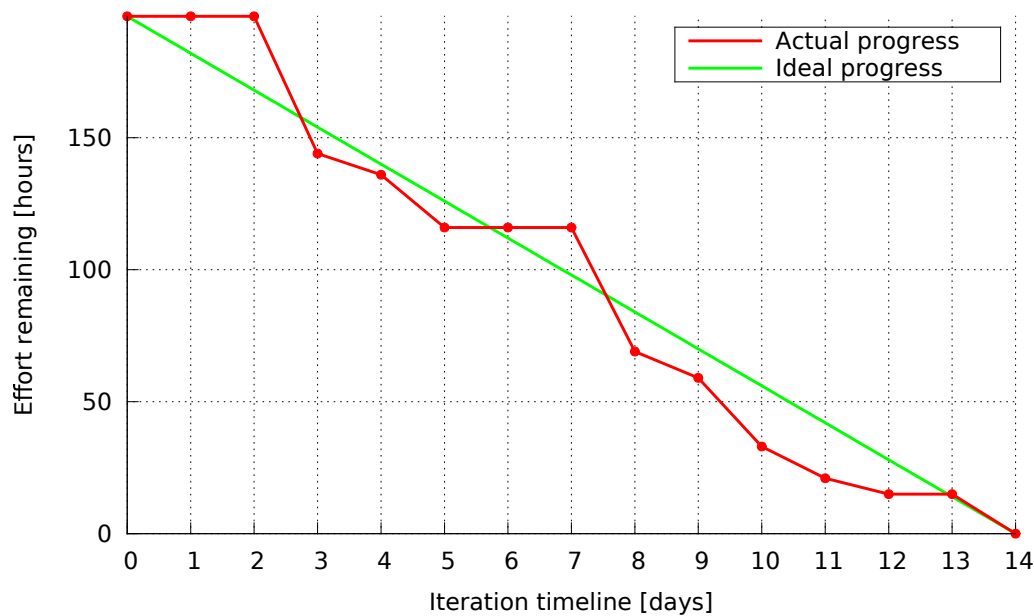


Figure 10.5: Burn down chart for sprint 4

10.7.1 What went well

All implementation related user stories were finished and as a consequence the goal for sprint 4 and the milestone *Traffic light control* were reached. The work in implementation part of this project start to escalate and team started to be enthusiastic.

The team was able to prepare and deal with situation, that one member was absent for a whole week.

³<http://www.youtube.com/watch?v=6mV5aJoZomI>

10.7.2 What could be improved

Stand-up meetings were not that often anymore. Due to other obligations of all members of team and late attendance during work hours, it was difficult find a common time for these meetings. Attendance should be more accurate and if absence is known ahead, it should be mentioned to other members.

11 | Sprint 5

As in previous sprint chapters, work-flow through sprint 5 will be described. What makes this sprint different from the others, is that the team had to manage occurred risk on communication module. A good part of this chapter will be devoted to resolution of risk, but it will not lack the rest of the processes such as planning, establishing goals and retrospective.

11.1 Sprint planning

The planning started with customer meeting at Thursday 24th of October 2013. For the customer, the main event and goal of this sprint, is to finalize the prototype and record the demo video. Therefore planning includes tasks as: collecting phones from friends, researching the companies that do mobile development and borrowing the phones from them, renting the room and recording equipment from university. Team agreed to borrow phones from friends, but if that did not work backup plan was to borrow phones from companies.

The recording final demo is planned for Thursday 7th of November, and informing potential phone owners that will be able to share their phones with us is planned at least one week before the event. This way table of potential phones can be created, and risk of not collecting the sufficient number of devices can be reduced and solved. Recording equipment is planned to be rented or reserved at least 3 days before the event.

As far as the implementation is concerned the team focused mainly on device detection part as the customer required to speed up this process. In order to achieve this task the *tree algorithm* for detection was developed (see Section 11.4). Plan is that the final demo would show an equalizer since the theme for the whole project was a rock concert. Therefore the team also needed to implement a predefined gallery. In sprint 5 team plan to work further on the report as well to finish sprint 4, start writing sprint 5 and to work further on the evaluation and the conclusion chapters.

All implementation related stories for sprint 5 are presented in Table 11.1. All the documentation related stories for sprint 5 are presented in Table 11.2. All the project management related stories for sprint 5 are presented in Table 11.3.

Table 11.1: Implementation user stories selected for sprint 5

ID	Ref.	Description	Hours	
			Est.	Sp.
I5.1	M6	As a client I need to synchronize my internal clock with server.	40	45
I5.2	M4	As a server I need to speed up the detection process.	36	38
I5.3	M5	As a manager I want to be able to create a playlist.	8	6
I5.4	M5	As a manager I want to have predefined gallery to be played.	12	14
Σ			96	103

Table 11.2: Documentation stories selected for sprint 5

ID	Ref.	Description	Hours	
			Est.	Sp.
D5.1	WBS 8.2	As a student I need to finish the sprint 4 chapter	10	15
D5.2	WBS 8.2	As a student I need to write planning and architecture sections of sprint 5 chapter	30	25
D5.3	WBS 8.2	As a student I need to finish the evaluation chapter	10	8
D5.4	WBS 8.2	As a student I need to create structure of the conclusion chapter	5	6
Σ			55	54

11.1.1 Duration

This sprint is 2 weeks long. From 28th of October 2013 to 10th of November 2013. Team agreed on the date of presentation and showing the running demo – on Thursday 7th of November 2013. Estimated velocity is 200 hours since every team member will devote 25 working hours per week.

11.2 Sprint goals

On the end of this sprint team expect to have working, the prove of concept demo. As core modules are finished, team will invest time in tuning existing and implementing some of the supporting functionalities such as: speeding up the detection process and preparing predefined gallery to be played. Goal for this sprint is also a time sync between server and clients. Reason is bypassing possible network latency, and enabling clients to play the content more precise.

Table 11.3: Project management stories selected for sprint 5

ID	Ref.	Description	Hours	
			Est.	Sp.
P5.1	WBS 7.1.1	As a student I have attend the weekly meetings with the customer	28	19
		Week I - Meeting	6	4
		Week I - Minutes	2	2
		Week II - Prepare demonstration	32	27
		Week II - Demonstration	6	5
		Week II - Minutes	4	3
P5.2	WBS 7.1.2	As a student I have to attend the weekly meetings with the supervisor	8	6
		Week I - Meeting	6	4
		Week I - Minutes	2	2
		Week II - Meeting	0	0
		Week II - Minutes	0	0
P5.3	WBS 7.3	As a student I need to participate to the lectures about sales.	20	11
Σ			78	58

11.3 Architecture

At first team was thinking about implementing the clock synchronization from scratch. Using one of the popular algorithms as Berkeley algorithm¹, and adding additional responsibility to server or client application.

The Berkeley algorithm requires feedback communication from clients by implementing receiving message system on server. Calculating the time offset on server and adjusting local times on clients. The protocol will have to expended in order to support clock sync communication. Although this approach will work, lot of changes on existing architecture have to be done.

As focus of this project is on proving the concept, and time for big implementation parts or structure changes is not sufficient, team agreed on researching alternative solutions before start implementing. This is also a part of risk management strategy where team have to be careful not to implement things they are not suppose to, according to Table 3.3. Further research has brought the team to Network Time Protocol (NTP)².

Good thing about NTP is that it is used for variable-latency data networks. It was only needed to find application that provide NTP service, and change client code to call this service and

¹<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=29484>

²<https://tools.ietf.org/html/rfc5905#section-14>

determine what is a local time offset before connecting to Digital Lighter server application. In that case, change of existing architecture will be minimal, and it will play only on client application. No server, or protocol will have to be changed.

As solution Time Server³ application have been chosen. It will work on same device as Digital Lighter server application, and it will provide same time as Digital Lighter application would.

11.3.1 Logical view

Logical view of added module is shown in Figure 11.1. Class **SNTPClient** makes call to time server in separate thread, and in **doInBackground** method retrieve **NtpMessage** object in order extract exact time from server. **NtpMessage** represents a NTP message, as specified in RFC 2030⁴. The message format is compatible with all versions of NTP and SNTP. This class does not support the optional authentication protocol, and ignores the key ID and message digest fields. For convenience, this class exposes message values as native Java types, not the NTP-specified data formats. For example, timestamps are stored as doubles (as opposed to the NTP unsigned 64-bit fixed point format). However, the constructor **NtpMessage(byte[])** and the method **toByteArray()** allow the import and export of the raw NTP message format.

Change have influenced message interchange between server and clients as well. The difference comparing to old sequence diagram is given at Figure 11.2. New calls are colored red. It is important that time server get started before registering service to the network.

11.3.2 Physical view

Deployment digram is also changed, and now includes NTP server as artifact. Additional artifact can be seen on Figure 11.3

³<https://play.google.com/store/apps/details?id=com.icecoldapps.timeserver>

⁴<http://www.rfc-base.org/txt/rfc-2030.txt>

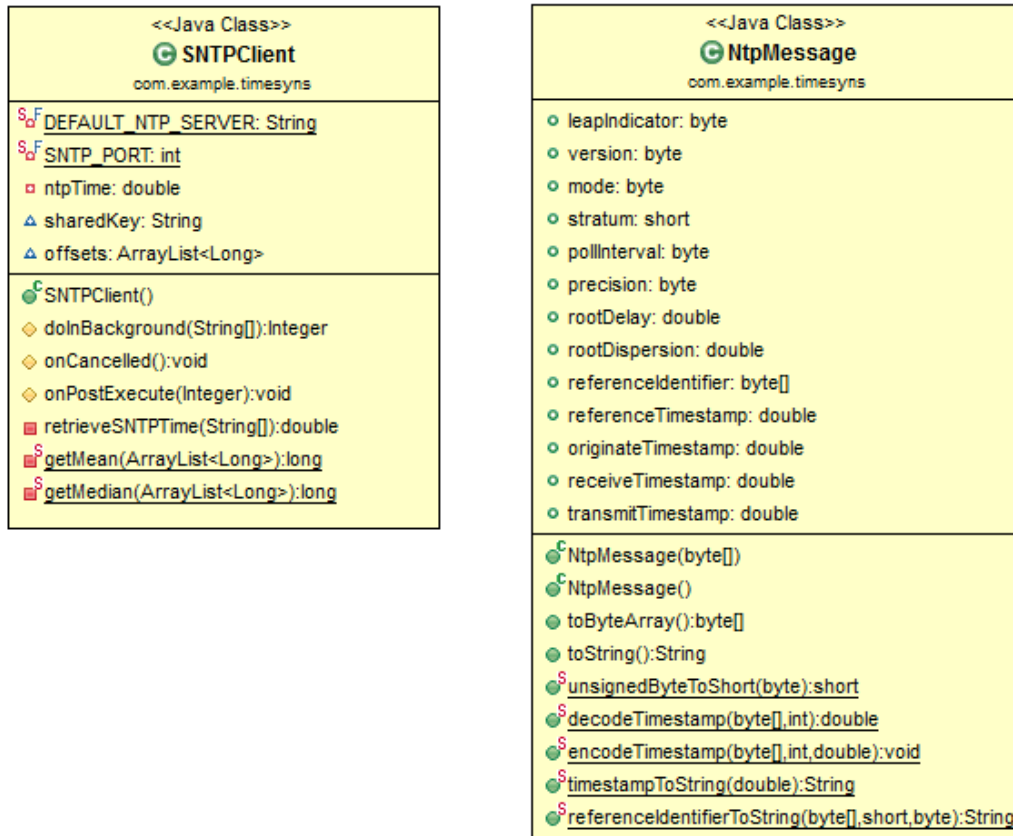


Figure 11.1: Time sync class diagram

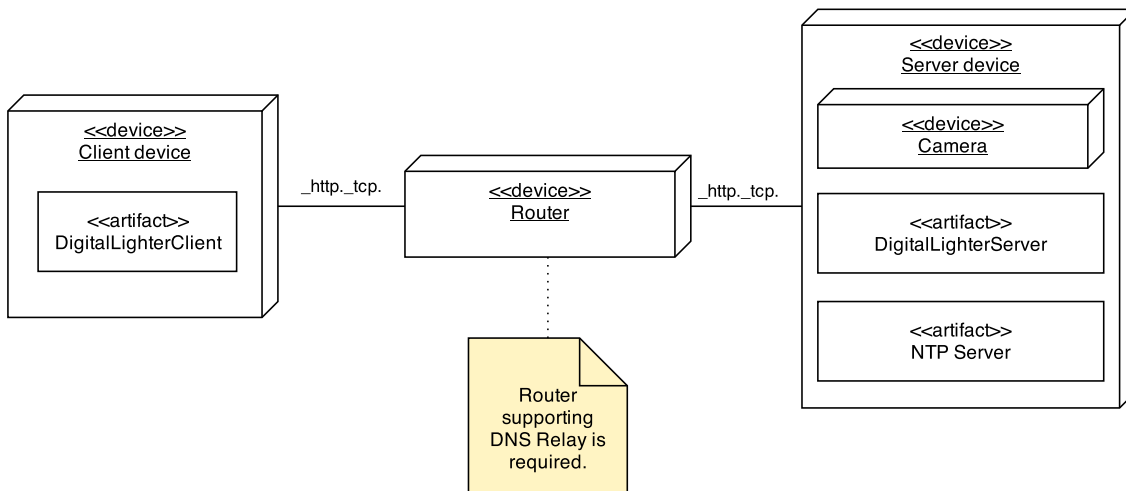


Figure 11.3: Deployment diagram

Observing deployment and sequence diagrams, it is easy to see that this solution didn't need

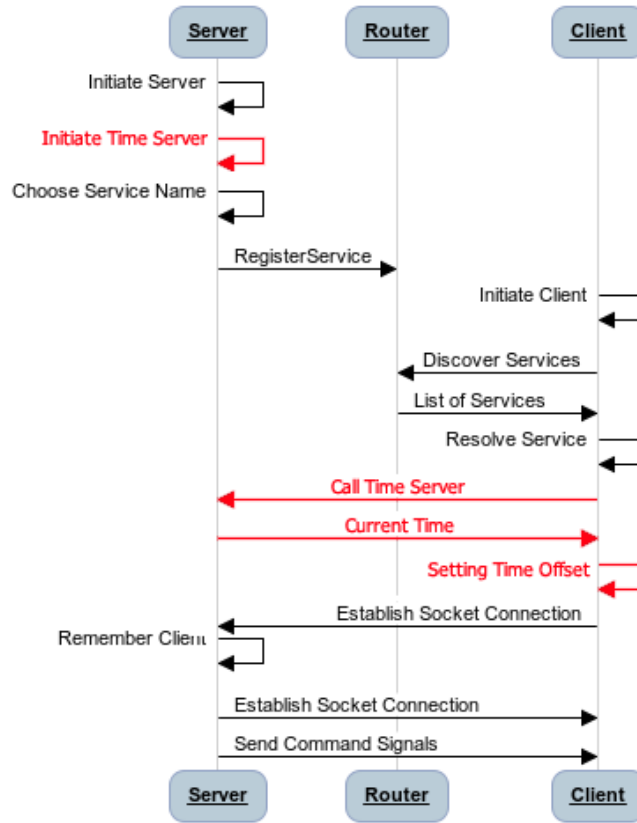


Figure 11.2: Sequence diagram

existing system to change. But in same time new functionality have been added and big amount of time have been saved.

11.4 Implementation

This section is devoted to describing the improved algorithm used for faster detection of the devices. So far the simple detection using "one-by-one" approach as explained in Section 10.3.3 was used. The drawback of this algorithm is that it performs with time complexity $T(n) = O(n)$ and thus it might get inadmissibly long. Thus more advanced algorithm based on divide and conquer design paradigm was developed. It will be described in greater detail in next sections.

11.4.1 Algorithm

The algorithm is based on the assumption that the image processing module originally designed and implemented during Sprint 3 (see Section 9.4) is capable of recognizing multiple different colors. Then it possible to recursively divide all the devices into the groups of the same color until there is only one device left in each subgroup. This algorithm is referred to as the *tree*

algorithm since the recursive division gradually creates the tree where each node excluding the leaf nodes has n children where $n = \text{number_of_colors_used_during_detection}$.

Figure 11.4 depicts the example of the detection process of the 16 devices while recognizing among 4 different colors: red, green, blue, yellow. In this example the ideal conditions where each device precisely fits into one single matrix tile are set.

One step of the algorithm can be represented as generating the new level of the tree (STEP 0, STEP 1 and STEP 2 in Figure 11.4). Given the specific tree level each node represents the group of devices that are to be lit with the same color. Devices are assigned to the groups randomly and it is required that each group would consist of the the same number of devices if possible (otherwise some groups might differ by 1 device).

Once it was decided which device will be lit by which color the server sends relevant signals to all of the devices and then detects the color blobs in the image. It saves the position of detected colors and proceeds to the next step.

The algorithm ends when there is no more than one device in each color group. Then the saved history of the colors sent to each device can be compared to the actual detected colors and their positions which results in detecting the real position of each device.

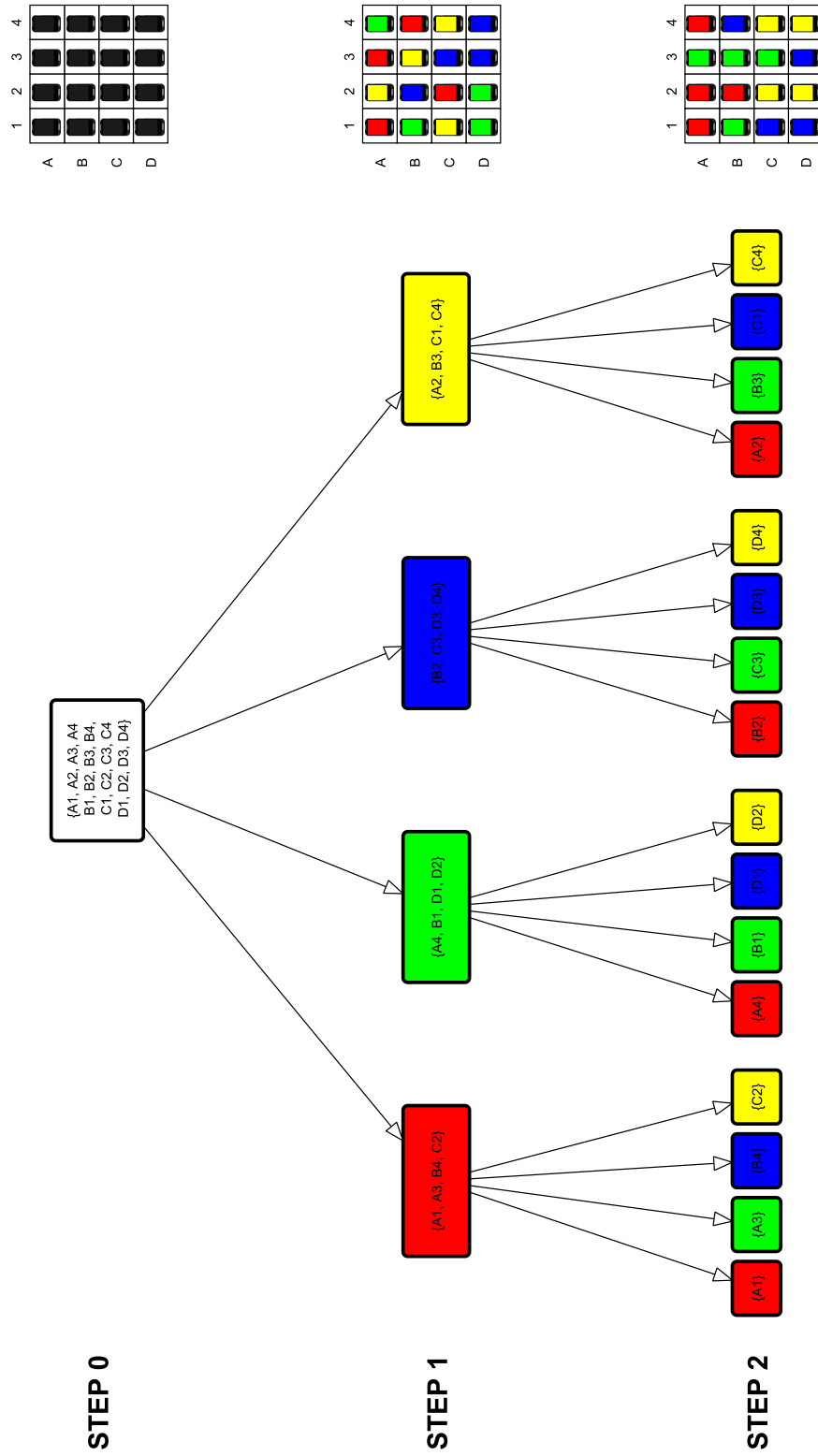


Figure 11.4: Tree algorithm detection example. There are sixteen devices and four detected colors which means it is possible to recognize the position of all devices within two steps.

11.4.2 Limitations

The main drawback of this algorithm is the position resolution. It is only capable of distinguishing between all devices if there are no more than one device in each tile. Otherwise the ambiguities occur and the algorithm finishes in the state where some devices are assigned more then one possible positions.

Nevertheless it has been decided to disregard this issue as this project only serves the purpose of proof of concept and if needed these minor limitations might be improved on performing additional techniques that would solve the problem.

11.4.3 Modification

It might happen that certain devices are not recognized using this algorithm. Therefore this algorithm is accompanied by the former "one-by-one" approach. This fallback option is only invoked if there happen to be devices that failed to being recognized.

11.4.4 Performance

Given the divide and conquer base of this algorithm the time complexity is logarithmic, $T(n) = O(\log n)$, as compared to linear time complexity of the former algorithm. This is though the best-case scenario where all the devices are correctly recognized. In the worst case scenario where no device is recognized during performing tree algorithm the time complexity would be $T(n) = O(\log n + n)$.

11.4.5 Communication protocol

After testing synchronization module it was found that old communication protocol, described in Section 8.4, cannot be left unchanged. In order to sync at start or at least during the playing sequence, clients need to receive starting timestamps for every command. This change can be seen on Figure 11.5, with following coding:

- server time in milliseconds at which client should play specific instruction. Time offset calculated from NTP call is added to this time so all clients can have same value in sum,
- time stamp-color separator,
- color to play(hexadecimal),
- one instruction separator.

32154851:#ff0000|32198458:#00ff00

Figure 11.5: New command message with multiple instructions.

11.5 Testing

After adding NTP module some testing considering time synchronization was executed. During this phase there was only 4 devices at team disposal. Details about this devices and their roles can be seen in Table 11.4

Table 11.4: Test devices in sprint 5

Device	Android os	Role
Samsung Galaxy S2	4.1.1	Router
Samsung Galaxy S3	4.1.2	Server
HTC Desire	2.3	Client
Samsung Google Nexus S	4.1.1	Client

Test case:

1. Start router. Since no real device is available emulate router functionality with one of the phone's tethering option. Name the hot-spot "Sprint5".
2. Connect server device to the network "Sprint5".
3. Start NTP time server on server device.
4. Start Digital Lighter server application.
5. Name service "testSprint5" and register it to the network.
6. Connect client device to the network "Sprint5"
7. Refresh list of available services on network and select service "testSprint5".
8. Connect to selected service.
9. Repeat steps 6-8 for all client devices.
10. Choose syncMedia⁵ and set frame rate to 2.
11. Detect all present clients and play media.
12. Observe client applications behavior.

Expected server behavior:

1. When started, Time Server application, stays as notification icon on notification bar.
2. When service is registered, Toast⁶ message of confirmation is displayed - "testSprint5 is registered".

⁵syncMedia - media, when played all client devices blink synchronously at given frame rate.

⁶<http://developer.android.com/guide/topics/ui/notifiers/toasts.html>

3. When clients connect, text label representing number of connected devices will increment.
4. When client device is detected grid square in which device is detected will light up.

Expected client behavior:

1. When list of available services is refreshed, all services should be visible in list view.
2. When "testSprint5" service is selected, the Toast message of confirmation is displayed - "testSprint5 service selected".
3. When "Connect" button is pressed, Toast message of confirmation is displayed - "Connected".
4. Few seconds after being connected, Toast message of background synchronization process should be displayed - "Time diff: SOME_VALUE". Where SOME_VALUE is client time offset comparing to Time Server.
5. When instructed to play, all clients should blink 2 times per second synchronously.

Test 1-7 observed behaviors:

Server: 1 ✓ 2 ✓ 3 ✓ 4 ✓

Client: 1 ✓ 2 ✓ 3 ✓ 4 ✓ 5 ✗

Inability of clients to synchronize blinking commands, led to new round of debugging. After the changes described in Section 11.4.5 new set of tests have been executed.

Test 8-12 observed behaviors with changed communication protocol:

Server: 1 ✓ 2 ✓ 3 ✓ 4 ✓

Client: 1 ✓ 2 ✓ 3 ✓ 4 ✓ 5 ✓

After additional tests, the changes made in this sprint, based on functionality and their reflection of user stories, are considered valid and verified.

11.6 Occurring risks

First time during development process, team have encountered the risk that might influence successful finish of the project. Restarting of client devices during runtime is noted again. This issue didn't happen for a while and source of it have been previously attached to TestFlight library as described in Section 8.5. As number of testing devices become bigger, multiple phones started to behave the same way and it was obvious that TestFlight library was not the real problem. According to risk management strategy recorded under "Dead end with technologies" in risk table 3.3, team decided to do thoroughly research for alternative solutions.

Further research lead to the reason of client restarting - race condition inside `NsdManager` class. The issue is registered on the Android issues list⁷ and has been fixed on Oct 18. 2013. Release is planned for Android 4.4 so only changing Android NSD for some of the previous technologies could solve the problem.

At that time, working on next core module (image processing), with knowledge that issue will be gone with next release of Android system, was preferable by the customer. For that reason team exchanged few e-mails with the customer explaining that previous decision to stick to Android NSD have multiple bad effects on the recording of final prototype:

- Team will be in possession of big number of Android devices for a limited period of time. If clients restart in middle of the presentation that will take away notably amounts of time.
- NSD works just on devices powered by Jelly Bean(4.1.x) or KitKat(4.4). Most of the phones team have plan to borrow are not brand new and do not support Android NSD service.

Finally the customer agreed on adding another user story with high priority - Replace NSD communication module with one of the technologies previously researched.

First fix Best NSD replacement technology according to previous research, according to sprint 1 preliminary studies in Section 7.2.3, was JmDNS library. Client and server communication module has been changed with JmDNS. Library is written for Java and is well documented. Migration to new technology was therefore fast, but it had some issues when working on Android platform. Clients are not able to resolve registered service. Detection of existing service is possible, but when client tries to resolve additional information shown on Figure 11.6 like Address, JmDNS will return null.

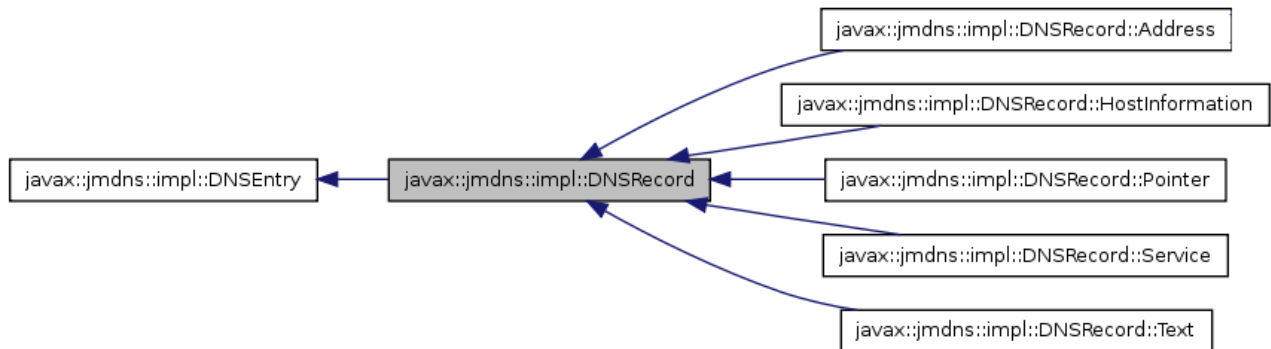


Figure 11.6: DNS record inherit graph.

It is important to say that this was not the case with all of the devices. Some of them, like Samsung Galaxy S2, was able to resolve detected services. It was clear that partially working solution is not good enough. After trying to fix the issue, and searching for answer on-line team decided to research additional alternative solutions.

⁷<https://code.google.com/p/android/issues/detail?id=35585>

Final fix Further research led to David Simmons multicast support application⁸ on Android devices. His solution works on all Android devices beginning with system version 2.2. Big surprise was trying his application for detecting NSD created service, and JmDNS created services. It was working almost instantly with NSD, and all services ever created on test router were easily recognized and resolved. Since race condition were happening only on client devices, and empirically best match for registering/detecting and resolving service was NSD/netlib, team decided to keep different libraries that communication modules for server and client are based on. Final communication modules and their respective libraries are shown on Figure 11.7

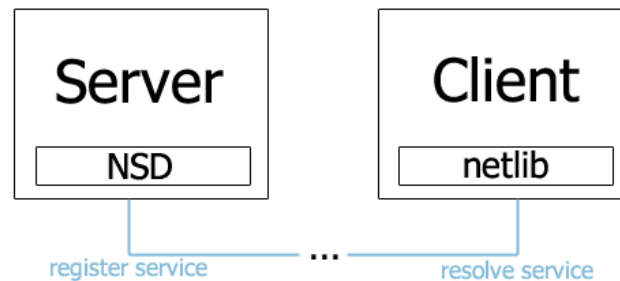


Figure 11.7: Illustration of communication modules specific libraries.

11.7 Customer feedback

The main goal of this sprint was to collect as many devices as possible and use them to display various types of imagery including the equalizer effect which was chosen by the customer as the possibly most spectacular for the viewer. As the team encountered a few serious performance issues regarding the network communication and the time synchronization it was not able to deliver the demonstration video of the flawless animation. On the contrary due to technical issues the team was only able to display obviously not properly working animations on the borrowed phones. Nevertheless two demonstration videos were recorded and presented to the customer.

The customer was familiarized with the problems that lead to the faulty performance. He encouraged the team to prepare new demonstration videos that would show the imagery specifically adjusted in a way that it would not be obvious the team faces synchronization issues: lower fps, static images or abstract effects to list a few suggestions.

The first video presented to the customer can be found on YouTube under the name Demonstration - Equalizer [out of sync]⁹. The second video is published under the name Demonstration - Czech flag¹⁰.

⁸<https://github.com/simmons/netlib>

⁹<http://www.youtube.com/watch?v=cDBcgPt1Sq8>

¹⁰<http://www.youtube.com/watch?v=MmTdFDsHTbk>

11.8 Retrospective

This section reflects on the past sprint. In order to learn from the mistakes done and thus to improve the workflow it is necessary to answer two essential questions: "What went well" and "What could be improved". Actual burn down chart is shown on Figure 11.8

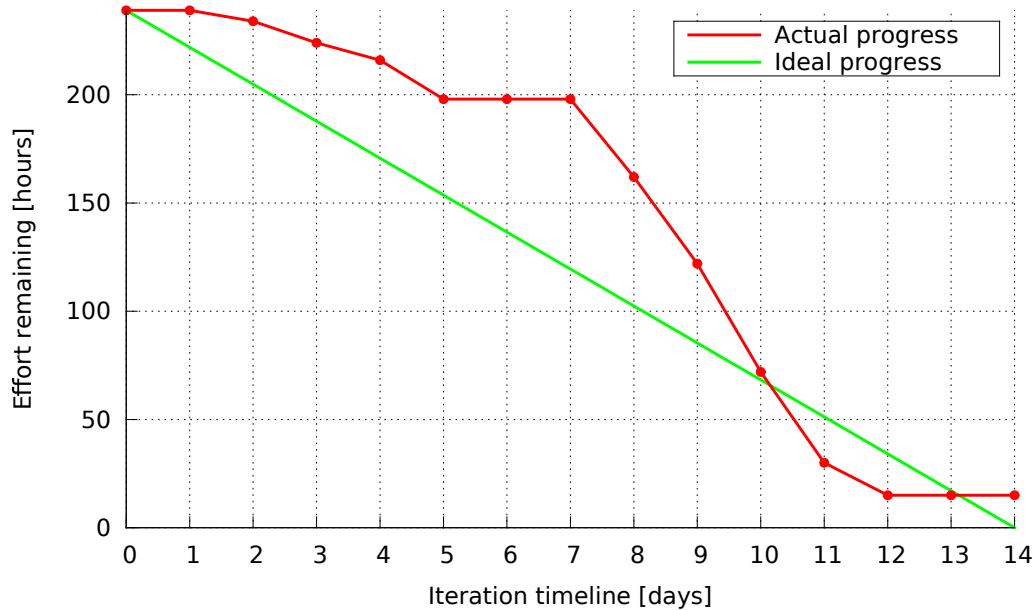


Figure 11.8: Burn down chart.

11.8.1 What went well

Team collected more than enough devices for demonstration purposes. All devices have been collected from friends and colleagues since they were notified about recording date a whole week in advance, so there was no need to involve big companies in process. Recording equipment and room at the university were rented on time.

All user stories considering implementation were finished on time, and occurred risk on communication module have been solved. Solving the issue was one of the main reasons for sufficient number of devices since new client module supports legacy devices. Detection algorithm speedup allowed better use of time devoted to recording. As a result of this improvements two demonstration videos have been recorded for the customer.

11.8.2 What could be improved

Connecting high number of devices (approximately 30) causes network low performance. For some messages it take more than 3 sec to arrive to client. This makes problems with device detection and synchronization. Team was aware of possible issue but didn't mark it as a high

risk, because test was never launched on more than 3 devices as clients. As a result of this oversight equalizer video was hard to reproduce on clients and, after few attempts, not recorded at all. This could be avoided by thinking of networking issue as necessary to solve.

Team estimated that number of devices for recording the video will be higher than 30. With this on mind most media that could be played was made for large-scale grid, including 25 pixels and more. After need to eliminate some of the devices not much of media left to be played. Team should prepare as diverse grids as possible in order to get better use of devices during next demonstration recording.

Also documentation user stories **D5.3** and **D5.4** were not completed because of underestimation of time needed. Team is reaching the last sprint, and it would be good if more time is spent on estimation because there is no more sprints to which user stories can be delegated.

12 | Sprint 6

In this chapter the planning and work-flow regarding Sprint 6 will be described. This is the last sprint and it is also of shorter length. Since the implementation of the product is already done this chapter will omit certain sections such as Architecture, Implementation and Testing which are present in the previous fully-fledged sprints chapters. At the end the team will evaluate the whole sprint, and try to answer on following questions: What went well? What could be improved?

12.1 Sprint planning

It was formerly intended to devote the last sprint to the tasks related with finishing and refining the report as well as preparing the final presentation and rehearsing. Nevertheless as the application performance issues occurring during the last sprint prevented the team from accomplishing to record the final demonstration video part of the total sum of person-hours available in this sprint must be spent on creating the video again.

The planning of this sprint started already during the last customer meeting held on 7th November where he suggested to use different types of imagery to be displayed on borrowed phones (see Section 11.7). One of the tasks therefore should be to create suitable images and/or videos in multiple scales as it was uncertain how many phones would be available during recording. The team members should also try to collect as many mobile devices as possible and also borrow the video recording device.

Next significant part of the sprint should be devoted to creating the demonstration video which would fulfill the requirement to amaze and entertain the viewers. As already planned this sprint will also mainly focus on finalizing and refining the report, preparing it for the submission, creating the final presentation and rehearsing.

It should not be forgotten to deploy both server and client side application on Google Play as this is one of the non-functional requirements (see Section 4.2).

Due to the fact the final version of the project must be submitted on the fixed date this sprint is shorter by four days and the workload must be adjusted accordingly. The team has eight work days to its disposal which makes it for 160 person-hours for the whole team.

All implementation related stories for sprint 6 are presented in Table 12.1. All the documentation related stories for sprint 6 are presented in Table 12.2. All the project management related stories

Table 12.1: Implementation user stories selected for sprint 6

ID	Ref.	Description	Hours	
			Est.	Sp.
I6.1	M5	As a server I need to provide the manager both with multiple static images and videos suitable for low fps.	8	10
I6.1	N3	As a student I have to deploy the application on app store	2	2
Σ			10	12

Table 12.2: Documentation stories selected for sprint 6

ID	Ref.	Description	Hours	
			Est.	Sp.
D6.1	WBS 8.2	As a student I need to finish the sprint 5 chapter	18	20
D6.2	WBS 8.2	As a student I need to finish the sprint 6 chapter	4	6
D6.3	WBS 8.2	As a student I need to finish the evaluation chapter	12	12
D6.4	WBS 8.2	As a student I need to finish the conclusion chapter	10	10
D6.5	WBS 8.2	As a student I need to finish the Appendix	10	12
D6.6	WBS 8.2	As a student I need to refine the whole report	10	15
D6.7	WBS 8.2	As a student I need to write the reflection paper	6	7
Σ			70	92

for sprint 6 are presented in Table 12.3.

12.1.1 Duration

This sprint is 10 days (8 work days) long. From 11th of November 2013 to 20th of November 2013. The final presentation of the project will be held on Thursday 21st of November 2013. Estimated velocity is 160 hours since we agreed on 25 working hours per person per week.

12.2 Sprint goals

The goals set to this sprint comprise of recording the final demonstration video using the newly created imagery, editing the video, finalizing and refining the report, creating the presentation and rehearsing. It was also agreed with the supervisor that the team would present demonstrate the presentation in draft one day before 21st November so that the supervisor would be able to give the last advices.

Table 12.3: Project management stories selected for sprint 6

ID	Ref.	Description	Hours	
			Est.	Sp.
P6.1	WBS 7.1.1	As a student I have attend the meeting with the customer	8	3
P6.2	WBS 7.1.2	As a student I have to attend the meeting with the supervisor	8	5
P6.3		As a student I have to prepare for the final presentation	44	50
P6.4		As a student I have to prepare for submission of the report	10	8
P6.5		As a student I have to create final demo-video	10	12
Σ			80	80

12.3 Implementation

As the application is already finished there were no need for the implementation during Sprint 6. However this section will summarize the types of imagery designed for the recording of the final demonstration video and explain the choices.

12.3.1 Static images

As the main implementation problems the team was facing during last sprint stem from the inaccurate time synchronization it has been decided to use static images as one of the sources of displayed imagery. Static images are perfectly suitable for showing the detection algorithm together with sending the commands work as expected while avoiding the synchronization issues.

As the team was able to collect around 18 devices the 4x4 pixels resolution pictures were used. The resolution can be considered too low for displaying any meaningful content, there are some possibilities (requiring a bit of imagination) though. The example can be seen in Figure 12.1.



Figure 12.1: Depiction of the famous cartoon character Marge Simpson displayed on 4 x 4 pixels.

12.3.2 Animations

Despite the synchronization issues it is still possible to play the animations of as low value of FPS¹ as 5 or less. The examples of the animations can be seen in the video published under the name Prototype 6 on YouTube: <http://www.youtube.com/watch?v=bBIvLHwQe7I>.

12.4 Occurring risks

The lack of time due to the fact the recording of the final demonstration video must have been redone might be considered a major risk. This is specified as a *Unrealistic time estimate* in Table 3.3 summarizing the measures that should be taken in order to reduce impact of occurring risks. The team decided to perform the suggested measure "*Work overtime*" and actually spent more hours on finalizing the product than formerly estimated.

12.5 Customer feedback

In the middle of the sprint the special meeting with customer was held in order to present him the draft of the final presentation video. Customer stated that the video proves the team reached all the goals and fulfilled all the requirements specified. He also suggested to extend the video and add more content. This will be done while creating the final demonstration video.

The video presented to the customer can be found on YouTube under the name Prototype 6².

¹Frames Per Second

²<http://www.youtube.com/watch?v=bBIvLHwQe7I>

12.6 Retrospective

This section reflects on the past sprint. In order to learn from the mistakes done and thus to improve the workflow it is necessary to answer two essential questions: "What went well" and "What could be improved".

You can see the burn down chart in Figure 12.2.

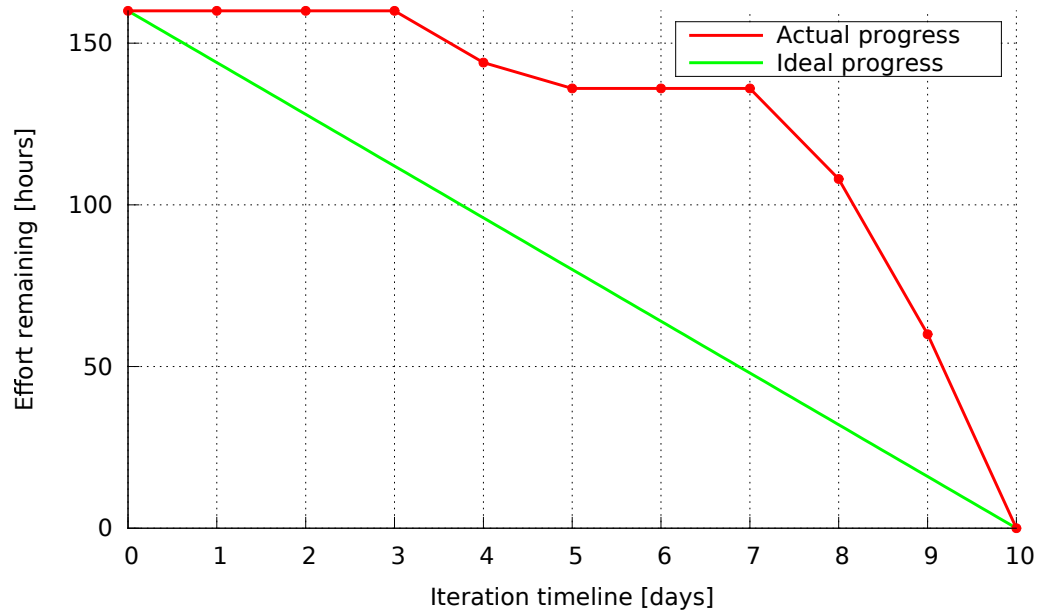


Figure 12.2: Burn down chart for sprint 6

12.6.1 What went well

It must be noted that the team managed to prepare better for the final demonstration video recording as compared to the last sprint where it only encountered the performance issues for the first time. The devices operated mostly as expected and it was possible to record sufficiently enough video material which was later used for creating the presentation.

12.6.2 What could be improved

The team failed to finish the work within estimated time frame which lead to taking the necessary measures such as working overtime and prioritizing the most important tasks while omitting some less significant ones.

13 | Evaluation

In this chapter the team will evaluate different aspects of the project. Team collaboration, the project assignment, the cooperation with both customer and supervisor together with risks and role assignments will be discussed. It will be also mentioned how the team tackled challenges during the project, and how the team handled the project management. Briefly explained the team will discuss why and how the outcome ended up being what it is today. This chapter is split into three major parts, which is group evaluation, project evaluation and technology evaluation.

13.1 Group evaluation

In this section the team will elaborate on how the group functioned as a whole, and how the team collaborated. The team will also evaluate both the customer and the supervisor, before moving on to the evaluation of the goals the team set at the beginning of this project. After this the team will evaluate the challenges met during this project and how this affected the work. At last the risk and role assignment will be evaluated.

13.1.1 Team dynamics

This section is about how the group functioned as a whole, how the team communicated, how the work was distributed, and at last it discusses the team motivation.

In general The group was randomly put together, and the number of people in this group was four. Each person can be seen as an individual with different personality and interests. A perfect atmosphere was not expected. Fortunately the team had no major conflicts, only constructive discussions. For the most part the team functioned well together, and offered each other help when needed. Overall the team feels like the team work, and progression of the project was good.

Communication Good communication within a team is important. For a team to perform it needs a safe and stable working environment. The importance of this was addressed early on. It

was decided that every decision made in the group should be a result of a democratic process, which meant making decision by the majority vote.

The communication was mostly done collocated. We agreed on sitting in the same work area as much as possible. This turned out to be a good rule most of the time. When every one was sitting together it was never difficult to get help from the rest of the group. This enabled the team, to quickly discuss points in doubt, and continuing with the work. This also gave the team a smoother work flow, a quicker progression, and it gave the team more time to revise and improve the work.

Because of completely different schedules, it was not always possible to sit collocated. Often there was a team member missing at working hours. There were a few situations where the team would have benefited from having the entire group present, but there was nothing the team could have done. Luckily it did not affect the project too much. The team would solve the problem by using Facebook. If someone had other obligations, and they were not able to be present, then the team kept in contact through Facebook. This was a suitable solution since the team members were available online often. This made coordination an easier task, since all of the team members could respond as soon as possible.

There were a few problems with the group communication. The team was too much dependent on the team member responsible for the communication. The team did not always get copies of emails written to the supervisor or the customer. This caused a little confusions regarding the meeting hours from time to time. It was also the cause of important information coming a bit late. The team should have had more strict rules for this.

In hindsight the team is able to point out a couple of things that could have made the communication better. The team could have done more stand ups. The lack of stand ups was another problem caused by the different schedules. The team tried to solve this problem by having longer meetings when needed, or suggested by a team member. Still, the team should have had more clear rules about stand ups, and this could have made the communication easier. More communication with other groups taking this course was another idea that could have improved the team communication. Although the tasks were different, the exchanging of ideas and sharing experiences on both group dynamics and processes could have affected the team in a positive way.

The team also had some social gatherings to get to know each other better, and throughout the project the team got well known with each other. This made the communication easier for us as a team. By learning from each other, and learning to put aside differences, one can say that everyone has grown as individuals. In spite of a couple of bumps in the road, the team feel that the issue of communication was handled pretty well.

Work distribution The distribution of work was challenging at times. The team had little experience with image processing, and therefore it was hard to be sure about the exact work load. Naturally this also made it difficult to maintain a balanced workload throughout the project. In the beginning the team agreed on what was needed to be done, and then group members had to take individual responsibilities to perform work on the tasks they were comfortable with. Towards the ending of the project it was a bit stressful, because of troubles with the final demonstration

video. Suddenly the team needed to spend more hours than expected. This is something the team should have anticipated, and the team should have distributed the work a better.

Motivation All team members were very motivated. Everyone wanted to receive a good grade in this course, since the course was very relevant for us. In this course the team had to use knowledge from several different courses in previous years. Another motivation was the project assignment. The fact that the assignment was interesting increased our motivation and efforts. It was exciting and the team had the opportunity to learn something new. Our customer was also a great motivator, and by forcing us to deliver something at the end of each sprint, the team always felt like the milestones was reached. This way we always felt one step closer at the end of each sprint, and it became very motivating to continue the hard work.

13.1.2 Customer and project task

Our customer, Peder Kongelf, was pleased with the results. During the last customer meeting he informed the team that the project goal was reached, which was to make the audience become a screen at a rock concert. This was a proof-of-concept project, and if someone wanted to do further work on our application the only remaining parts would be performance and scalability. The team wanted to have a bigger "wow-effect" on the last video demonstration, but we did not have enough resources to conduct this. With the demonstration video it is only possible to see the potential the project has. All in all the customer was satisfied with this result and explicitly stated that the team accomplished all required tasks.

The team is very impressed with the customer. The project given to us for this course was extremely exciting and the team got to work with challenging and new technology, which was really great. The fact that the task was interesting increased our motivation and efforts. The customer had plenty of experience with working as an consultant, and he was eager to teach us as much as possible. The team also wanted to learn as much as possible from him. He was giving us good advice through out the whole project. One of the best advices he provided regarded working together in one place instead of working as the individuals. Looking back the team realizes that otherwise communication challenges could have been much tougher. Our customer also helped us with the scrum methodology. He taught the team a lot about everything from epics, user stories, task, estimation, to reviews. All of this was extremely useful for the team.

Early on the customer suggested weakly meetings, which was good. He always gave the team great feedback, as well as giving an impression of having confidence in the team. Sometimes, because he had a great amount of work load him selves, the meetings where shortened. Still, he was involved and enthusiastic about the project. Of course the team expected some communication problems from time to time. These challenges regarded confusion over meeting hours, but the team addressed these issues very smoothly.

13.1.3 Supervisor

Almost every week the group met up with the supervisor. In these meetings the progress made by the group was presented, and the supervisors came with feedback on the report and other

work. It was very useful to know whether the team needed to increase the work speed or not. The supervisor also gave feedback on the content of our report, which was beneficial for the team.

Before the meetings the team sent out a meeting invitation to the supervisor. If there were something in particular the team had questions about, then the relevant documentation was emailed to the supervisor before the meeting, so the he could get a chance to look at it before the meeting.

In the beginning the meetings were useful in order to assure that the team were on the right track. After a while there was less need for a weekly meeting, because the team had a better understanding of what the report should look like. Maybe a shorter meeting would have been more sufficient for the middle part of the project. All in all the supervisor gave us valuable feedback, especially with the report and the final presentation.

13.1.4 Goals

In this section the team will evaluate the project goals, the group goals, and our personal goals, to see if the team achieved them.

Project goals

Finish the project within the scheduled timetable The team finished the project within scheduled timetable. It was a little stressful towards the ending, but this goal was reached.

Finish the project with the specified requirements The project was finished with the specified requirements, and the team made a demonstration video for the final presentation to show off the system. This goal the team achieved.

Group goals

Receive a good grade or a good recommendation At this stage it is difficult to determine whether this goal was achieved or not. Looking back at setting these goals the team realizes that setting more measurable goal would have been better. The team has learned from this, and will take this experience with us fore future projects.

Personal goals

Agnethe One of my personal goals was to learn more about group dynamics and collaboration. This I have definitely done. Working as a team can be challenging at times, and I have learned a lot from this experience. Another goal of mine was to learn more about writing reports. Writing this huge report has thought me a lot about technical writing, and I feel more prepared for writing reports in other courses later on. A goal that is hard to determine at this point is whether I am

better at holding presentations since this report is supposed to be finished before holding the final presentation. The last goal I would like to mention is that when I wrote the goals in the beginning of this course I wrote that there is a great chance that I would work as an IT-consultant. Looking back at this I realize that I really enjoyed this experience. Working as an IT-consultant is something I would like for the future.

Jan When looking back on the whole project timespan now I see that one of the main purpose of this course was to teach students to cooperate and communicate both together in development team and also with the other project related authorities such as the supervisor and the customer. I gained the idea of what does it take to work under agile methodology Scrum and I appreciate this experience as I might come across it in my future carrier. I also improved my skills as far as the Java/Android programming is concerned but not as much as I expected mostly due to the fast tempo and the need of doing other tasks. Even though sometimes boring I appreciate that writing technical documentation was a part of this project as I definitely improved my English writing skills. As for the course itself I must admit I was a bit disappointed by the organization and the lack of lectures and/or other sources of information related to the project. I enjoyed working with my team colleagues though and to conclude I feel this course was beneficial for me.

Milos After 13 weeks of team work my primary goals and expectations from this course have been fulfilled. I learned how to be part of the team and how to collaborate with other people in achieving common goals. Further more I was enjoying working with new teammates. Overall atmosphere was great, team spirit was on high level and we even spent time together beside this project. For this reason, we were able to finish all of the set requirements and manage occurred risks and minor conflicts without big effort. Writing documentation helped me with technical English, just as expected, and practicing SCRUM gave me a picture of how real companies manage their development cycle. For me this project was absolute success and I will be proud to add it to my CV.

Tomas During this course, I have enormously improved my written English. It was also useful to learn about technical writing in English language, since some of the rules are different than in Czech. Working in group and collaboration was sometimes difficult, due to different views of team members, and there is often need of compromises. I have increased my knowledge in topic of architecture and software design and I have learned a lot of new diagrams. Unfortunately due to the work distribution, my Android platform development skills have not improved as I expected, because other more knowledgeable team members were responsible for these parts.

13.1.5 Challenges

The team will in this section reflect on the challenges experienced, why they occurred, and what we could have done differently.

Time and time estimation Throughout the project, time restrictions have been a challenge because most of the members in the group have had other projects during the semester. Agnethe had two different projects in addition to the customer driven project. Milos also had another project that was quite time consuming in the beginning. Jan and Tomas also had another projects besides the customer driven project, and Jan also took Norwegian classes. All of this made it hard to find suitable working hours, and time.

The actual amount of logged work fell a little short of the expected amount of hours. Still, the team were able to successfully execute the tasks planned. The team also feels like the shortage of hours did not have any significant detrimental effect to the project.

Regarding the time estimation, the team thinks the estimates were within reasonable bounds, but still it could have been better estimations for some of the user stories. To improve these time estimation the team could have done a better job using the planning poker. While reviewing this the team realizes that realistic time estimation is valuable, and that this project has taught us a lot about this.

Language and cultural barriers Within the team there were three different nationalities. There was one Norwegian, one Serbian and two Czechs. Since there were different nationalities the communication had to be done in English. English was spoken internally in the group. The communication with both the supervisor and the customer was also done in English. It was a challenge to have all communication in English. Solutions and technical issues is probably easier to discuss in a persons native language. This caused challenges during the decision makings. It was spend more hours then expected discussing some issues, because it was hard to express everything in English. It was also hard to get a clear understanding of the different arguments. Sometimes Jan and Tomas switched to Czech, to be able to help each other. Sometimes it was easy to lose focus during discussions when the language was switched, but most of the time it was very useful and it progressed the discussion.

Since everyone was used to study at different universities, with different cultures, and with different ways of doing things. The team sometimes had different methods for solving problems. If the team could not reach a decision, it was decided to ask the supervisor or customer to get additional information, and then try to make a new decision. Very often this helped to shed new light on the situation. If the team still could not make a decision the majority vote was used. For the most of the time language and cultural differences was not a problem. Everyone was excited to learn, and teach each other.

13.1.6 Role assignment

As mentioned earlier the role assignment were adapted a bit, so they could fit the project better. Also the roles assigned to each group member was more of a guideline, rather than a binding responsibility. This was mainly because for some of the team members, this was their first scrum project. Some of the roles required more experience and knowledge then the group member had. The team solved this problem by working together, and helping each other.

Looking back at the role assignment it is not possible to deny that there are things that could

have been done differently. Since everyone was working with mostly unfamiliar technologies in the beginning, it might have been a good idea to make each of the role's responsibility more clear, and enforcing them in the early stages of the project. Also, the team could have had spent more time researching the roles actually needed. Even though it was decided to embrace this as an equal development team, the roles might have been useful for us if these improvements had been done.

These are the roles assigned, and this is how they were evaluated.

Project Leader The project leader was supposed to be responsible for the project progression, and delegate tasks to the other team members. This did not work very well, because the tasks was prioritized by the customer. If one task was done, the team member had picked the task with the highest priority to do next. All of the team members took responsibility for the progression of the project, and tried to make sure that it went according to plan. The description of this role also claimed that the Project Leader had final call in arguments. This was not the case, since it was decided from the beginning that this as an equal development team.

System Architect The system architect was responsible for checking and analyzing all the layers in the product, but this was also something not only one person took responsibility for.

Scrum Master The Scrum Master role was assigned to one person originally. Nevertheless, all team members took responsibility for following up on this. It became natural to suggest to have stand ups during working hours, since it was not always easy too keep track of what the other team members were working on. Having the whole team responsible for scrum methodology worked for this project, because of the team members having different schedules.

Communication Responsible This is the only role that was used throughout the whole project. The reason for this is because it was easier for the team to keep track, and it was also easier for both the customer and supervisor to have only one person to communicate with.

QA Responsible The QA was responsible for the quality of all documents and the end product. Still, the whole team felt responsible for delivering a product with good quality. The QA was also supposed to help determining if stories and acceptance criteria was well defined. For this project it was difficult because it was hard to do with little experience. A QA is also responsible for organizing testing, and this includes that unit tests are well written, provide developers with high level test cases for the stories, and performing exploratory testing on early builds. For this project the testing part was not a priority. All of the above resulted in that this role was not used.

Documentation Responsible This is a role that all team members took responsibility for. It was smart having one person responsible for setting up a good structure in the beginning. After a while everyone participated where they could.

13.1.7 Risk evaluation

In this section the risks from the risk Table 3.3 in the planning section, will be evaluated. Among these risks the team only experienced a few of them.

Sickness/Absence Some of the team members were exchange students, and they wanted to travel and experience Norway during the semester. Naturally it led to some absence at times. When this happened the team members worked extra hard for this not to affect the project. This was also the solution if members got sick, and for this project this was a good solution.

Coding problems To some extent the team experienced some coding problems, but the team figured it out. At the end it was possible for us to reach the project goals. This is a risk that probably occurs in most projects. Because of reflecting over the risks the team was prepared that might happen.

Testing problems This risk did not occur, since the customer told us not to focus on the testing part.

Changing requirements The team did not experience problems with requirements changing too much. The communication with the customer was really good, and possibly because of the reflection over this risk, the problem was avoided.

Dead end with technologies While doing research for existing solutions the team experienced difficulties. The team could not find a good existing solution. Luckily the team found OpenCV, and by using this it was possible to finish the project.

Unrealistic time estimation It was not easy to estimate all of the user stories. Sometimes the team underestimated and other times the team overestimated. Towards the ending of the project it was a bit stressful, but mostly this did not affect this project.

Customer too ambitious The team did not experience that the customer was too ambitious. The reason for this was because the communication with the customer through out the project was very good. The team and the customer had good sprint reviews, and planned the next sprint together.

Hardware problems The team had some hardware problems. For this project the team were dependent on borrowing phones to be able to test the system. It was not easy to borrow phones from a larger amount of friends at the same time. The hardware problem was that the team did not have enough resources. Luckily the team was able to borrow phones two times. The first time the team experienced some troubles with the school network. Phones connecting to different subnets at NTNU made it difficult test. To solve this problem we needed our own router. The problem

was solved by shooting the final demo-video at one of the team members apartment. If this risk had been more concrete from the beginning, the team could might have foreseen this.

13.2 Project evaluation

This section will start with an evaluation of the preliminary studies, and the planning phase of this project. Then the team will evaluate the utilization of the scrum methodology, before moving on to a summary of the meetings minutes.

13.2.1 Preliminary studies

In this section the team will evaluate the preliminary studies done for this project.

Similar projects The team is satisfied with the fact that similar projects were researched, because it gave inspiration for which effects the Digital Lighter should be able to play.

Software development methodology The research of the methodologies was time well spent, because it helped the team to learn more about Scrum, and how to carry it out in practice.

Development technologies This research was the most important one. It was also what the team learned the most from. Looking back the team realizes that spending even more time researching could have been helpful, but with the given time frame this would be a too big of a risk.

Project management tools The team became better acquainted with the Scrum while spending time finding a suitable project management tool. To understand what kind of features this kind of tool needed, the team gained a better understanding of Scrum, which was positive. What was negative was that the free tools were definitely not perfect, and the team spent more time then necessary trying to choose between several incomplete tools.

Documentation tools Researching documentation tools gave the team an opportunity to learn more about LaTeX, which was planned to use for writing the report. This was very helpful since not all team members had experience using it.

13.2.2 Planning

It was useful to plan the project, because the team became more aware of the strengths and weaknesses within the team. Therefore the time spent planning the project was not wasted. It also helped the team working more goal-oriented. In the beginning of the project the sprints were planned together with the customer. This was valuable for us, because the team avoided confusion

regarding the requirements. The team also learned a lot about estimation of user stories. Doing the planning by the team itself, this knowledge became very useful. Reviewing the time spent on planning, there is no point denying that the team could have planned the sprints better. This might have helped with avoiding stress towards the end.

13.2.3 Scrum

The Scrum development methodology was chosen early on even though some of the team members were not too familiar with it. Still there were many good reasons why the team chose to utilize Scrum. One of the reasons was the risk of sudden changes in the requirements. Another was that it would be easier to monitor the progress. Starting the implementation early was also a good argument. Also the fact that the team would most likely be using agile methods in future projects. This was also the methodology the customer proposed to use.

Using Scrum was difficult at first as only some of us had any previous experience. In addition, the project scope was not properly set at the beginning, which made building the product backlog difficult. The team did not follow the Scrum convention for user stories as an example. All of the stories had to be changed later on to make them more consistent. The backlog also contained stories for writing the report and attending meetings. The team decided to add this to report anyways, because it showed very clearly how the had carried out the project from the start till the end.

Even though the team tried to adapt Scrum to the project, one of the negative experiences was the stand ups. As a student there are several courses to take into account, and this made daily stand ups difficult to follow through. Therefore Scrum could have worked better if the team worked more as a unit with more strict working hours from the beginning.

The positive experience were the concepts of sprints. Everything from planning to the retrospective was well conducted, and the team learned a lot. The team also held a demo presentation for both the customer and the advisor at the end of every sprint, and this helped with keeping them updated, and opened for some feedback. Based on this feedback it was possible make the necessary changes in time.

Even though the team have had some difficulties with Scrum, the team is glad for using this methodology. Scrum made it simple to implement changes in the project, and continuously produce results that could be discussed with the customer. It would have been challenging to complete the project with a waterfall methodology, because it was important to start the implementation early.

In the end the team was able to deliver a product that both the team and the customer was very happy with, which in hindsight can make say that the team was right to choose Scrum.

13.2.4 Meetings summary

Looking back the team feels like the meetings were useful. It was essential to have a good communication with the customer, because it was important not to get sudden changes in the requirements. It was also important to be sure that the customer was satisfied with the product

throughout the project. While writing this report the team realized the advantage of having good meeting minutes. It was useful to be able to go back and check thoughts on decisions and thoughts on the project in general, while writing.

In the Appendix it is possible to see an example of the meeting minutes. The customer meeting minutes can be found in Appendix C, and the supervisor meeting minutes is Appendix D.

13.2.5 Course feedback

In this section there will be an evaluation of the Customer driven development course. The team will evaluate the lectures given in this course, and evaluate the course in general.

The lectures

In this section the team will evaluate the different lectures that were held in this course.

Course in group dynamics This course focused on making the participants aware of that everyone is different, and handles things differently. Another important focus was the awareness of collaboration. The course made the team more aware of that working together will get us further. It also made the team aware that the other group members can help with reflection during a decision making. The course also forced the team to discuss the group different goals, and the different expectations.

This course was one of the more useful lectures, since it improved the awareness of people's differences and that their way of behaving is different, and that forced the team to discuss how to make decisions and solve conflicts if they occurred. To give some criticism, it can be mentioned that this course should have been earlier in the semester.

Presentation in agile software development and estimation techniques This lecture focused mainly on Scrum methodology, but it also compared agile software development to other methodologies. This lecture was helpful for the team because i was decided to work in Scrum. Since only one team member had used this methodology before, it was helpful for the team to learn more about the basics.

Seminar in technical writing This lecture was useful since it gave the team a understanding of how a technical report should be written. The only critique is that some of the feedback the team got was not consistent with the feedback given by the supervisor.

Course in presentation technique This lecture was some what useful. The team got some tips on sales, and how to hold a good presentation. These tips will be taken into account when preparing for the final demonstration.

Final comments

This has proven to be a very valuable course. It has given the team a proper insight into the how's and why's of a software development process. Everything from the very start to the end of a project. The team have learned more about how to plan, estimate and research during a project. Also the team learned more about implementation and new technologies. Of course the team also gained some experiences with utilizing the scrum methodology, and working together as a team.

It has been exciting to work with a real customer, and learning more about working as a consultant. This was huge motivation for the team, because the team really wanted to deliver something the customer would be pleased with. All in all this has been a successful project, which the team has learned a lot from.

13.3 Technologies and tools evaluation

In this section you can read about main used technologies and tools during developing. Emphasis is placed on problems encountered during using these tools and technologies.

13.3.1 Android

A brief description of Android operating system have been given in Section 2.3.2. Building system under this platform had some advantages and disadvantages. Main points of both groups will be presented but bottom line goes in favor of positive points - Android OS was a good choice.

Disadvantages:

- Platform fragmentation - Right now, there are so many different devices out there, it is extremely difficult to put together an application that even runs on all of them.
- Late support for multicast DNS-based discovery - For now only Android 4.1 adds support this feature, and that leave more than 30% of the devices behind. Even working devices have issue described in Section 11.6.
- OpenCV library for Android is still relatively new and missing some of the original library features as described in Section 13.3.2.

Advantages:

- Good documentation - Android is very well documented and covered with training tutorials and lessons ¹. This helped the team members who have not previously met with programming mobile applications.
- Large community - This means a lot of answered questions, and lot of developers ready to help you with problems. In some situations it was enough to check few forums, and developing problems will be solved.

¹<http://developer.android.com/training/index.html>

- Biggest part of the market share - Considering the survey carried on by the well established analyst, the company IDC, as of Q2 2013 Google's Android OS held almost 80 % of the market share². This means that even with cut of 30% because of platform fragmentation, team can target the biggest number of the devices. With changed networking module as described in Section 11.6 Digital Lighter Client application can be deployed on more than 95% of the Android devices. This also means that team was able to easier collect sufficient number of devices for final demo video.
- Big number of third party libraries - Popularity of platform made everyone porting all kind of libraries to Android. Team was using some of them during the development process like OpenCV, JmDNS and netlib³.

13.3.2 OpenCV

OpenCV is an open source computer vision library briefly described in Section 2.3.2. It is a very useful library, which saved the team a lot of time instead of implementing image processing parts themselves – whole light detection module was just enhanced demonstration example of OpenCV functionality.

On the other hand, some problems with this library occurred. OpenCV Java API is a relatively new and therefore some implementation is missing. One example of this was described in Section 9.5. This raised a lot of problems, which resulted into discarding of user story.

OpenCV Java API can be therefore counted as a immature product but with a high potential in future, provided OpenCV developers will focus more on this API. At this moment, new release 2.4.7 if OpenCV is 44 days late according to the plan⁴ with still 32 opened bugs or new features.

13.3.3 Git

The team adopted Git as its version control system since early phase as described in Section 2.5.2. Most of the members had experience with some other VCS such as Subversion and therefore the idea of versioning was not new. On the other hand, three of four members of the team had no experience with Git itself.

Even though Git is a powerful tool with a lot of features, the main reason of using was to share code. Since development was rather linear, branching was used rarely. On the other hand, support of releases was used quite often ⁵. This was very comfortable for creating a final report, when the team could easily see exactly what features were included in each prototype and also it was possible to see the GUI of product in particular phase.

Of course, troubles during merging occurred. Since merging conflicts can be sometimes demanding and usually requires a deep knowledge about merging code, sometimes it was impossible for one

²<http://www.idc.com/getdoc.jsp?containerId=prUS24257413>

³<https://github.com/simmons/netlib>

⁴<http://code.opencv.org/projects/opencv/roadmap>

⁵<https://github.com/dohnato/DigitalLighter/releases>

person to solve it. These conflicts were in most cases solved by discussion with all interested members.

After all the experience with Git is rather positive. It is always good to be able to work with tools, that a lot of companies use in real development.

13.3.4 Target Process

Target Process is a collaborative project management tool described in Section 2.4.3. This tool was chosen after two another tools were tried. It is very complex tool for large teams and it provides many different views on project (called boards) and also offers to personalize these views.

This tool was used mainly for a collaboration with customer. Before sprint planning, user stories for next sprint were prepared and inserted into Target Process. Customer could easily prioritize these stories by dragging and dropping. He could also see spend hours on particular story and also the progress bar depicting progress of this story.

Sometimes, due to lack of a discipline, the spend hours and finished stories and tasks were not filled in time and therefore the retrospective cannot be that precise. Anyway, Target Process was very mature and useful tool for customer communication with much more features that was for this project necessary.

13.3.5 TestFlight

TestFlight service, described in Section 2.3.3, was adopted because of customer proposal. The service was used in early phases of development for customer comfort when testing the product.

Since the beginning of implementation, some non-reproducible errors occurred in client application. After some testing, the team suspected TestFlight of those problems. TestFlight for Android is relatively new service ⁶ and there it was decided to discard it from the implementation. After some time, the team has discovered, that the issues were caused by Android system itself, as described in Section 11.6, but the TestFlight was never used again, because the demonstration for customer were performed in form of videos.

Therefore it is difficult to evaluate tool, that has been used only for a short period of time.

13.3.6 Technical issues

During testing light detection module, problem situation⁷ with mobile devices camera was encountered. In dark, if a tracked object (mobile phone lighting with single color) or camera itself is moving, everything is working as expected. But if both object and camera are static, after some time, the camera starts to adjust the colors. You can see that situation in Figure 13.1. This

⁶<http://blog.testflightapp.com/post/49971420302/android>

⁷Tested with Samsung Galaxy S3 and Samsung Galaxy S2.

is probably caused by exposure settings, and further research concerning that topic would have to be done. By empirical research there were established colors (e. g. blue and white) which are

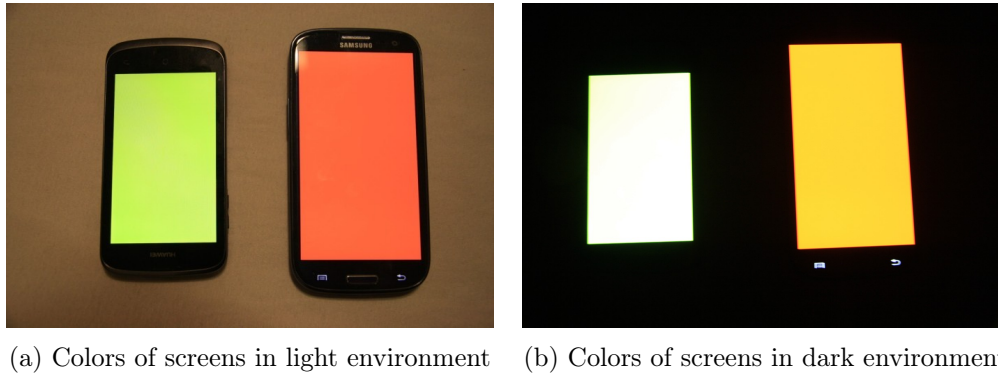


Figure 13.1: Colors of screens in different light environments

least affected by this behavior.

14 | Conclusion

In this chapter the summary of the project results will be summarized. Both applications which the final product consists of will be described and their main features together with the main limitations will be listed. The functional and non-functional requirements will be reflected on so that it would be cleared that all of the project objectives were fulfilled.

14.1 Final product

It was agreed with the customer that the final product would be split into two separate applications, the server-side and client-side part. Despite the fact the main responsibilities of the server (including processing the images and performing the detection, location and time synchronizations) represent computationally intensive tasks the customer insisted on implementing the server as a mobile application. The main reason is to enable the prospective power users or even developers to utilize the framework also for the other domains besides displaying imagery on the music concert crowd. The client-side on the other hand might be considered relatively simple application that only registers to the server and continuously changes the screen color according to the received textual commands.

14.1.1 Server-side application

Logic The server-side application is implemented in Java programming language using standard Google Android API and it supports the devices running on Android OS version 2.2 or higher. For the image processing module the Java version of OpenCV library was used and the networking is handled using the freely accessible library netlib. The time synchronization module uses the NTP time protocol and as an NTP server it uses the Android application Time server¹ which must be downloaded and run prior to the server-side application. The application Digital Lighter Server can be downloaded through the online store Google Play²

For the detection the server uses the *tree algorithm* as explained in Section 11.4 with the fallback option of linear *one-by-one* algorithm. It can be chosen by the user how many colors should be used for the detection but the recommended value is three, specifically blue, white and red. The

¹<https://play.google.com/store/apps/details?id=com.icecoldapps.timeserver>

²<https://play.google.com/store/apps/details?id=com.silentducks.digitallighterserver>

server is packed with a few images and animations of the various sizes and aspect ratios so that they could be played on the client devices right away.

GUI The graphical user interface should be as simple as possible as it only requires the user to set the server name, choose the number of colors for the detection and select the media to be played. GUI of the server can be seen in he Figure 14.1.

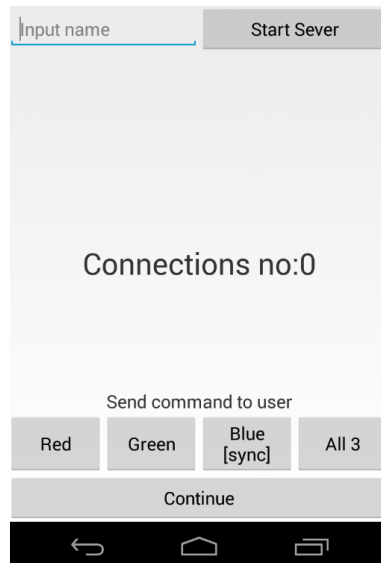


Figure 14.1: Server UI

14.1.2 Client-side application

Logic As described in the introduction of this chapter the client application should be as simple as possible, so that it would run seamlessly on the wide range of Android phones and so that no user would come across any inconveniences preventing them from using the application. The client application only enables the user to choose one of the services that are offered on the local network and connect to them. From this point the client only waits for the signals to be continuously received from the server, parsing the textual protocol and responding to the single commands by changing the color of its screen.

The application Digital Lighter Client can be downloaded through the online store Google Play³.

GUI Similarly to the server-side application also the client GUI should be as simple as possible. The only task the user should perform is to manually select the offered service and connect. Then there is an option to hide the user interface buttons which can improve on the success rate of the detection. The GUI is depicted in the Figure 14.2

³<https://play.google.com/store/apps/details?id=com.silentducks.digitallighter>

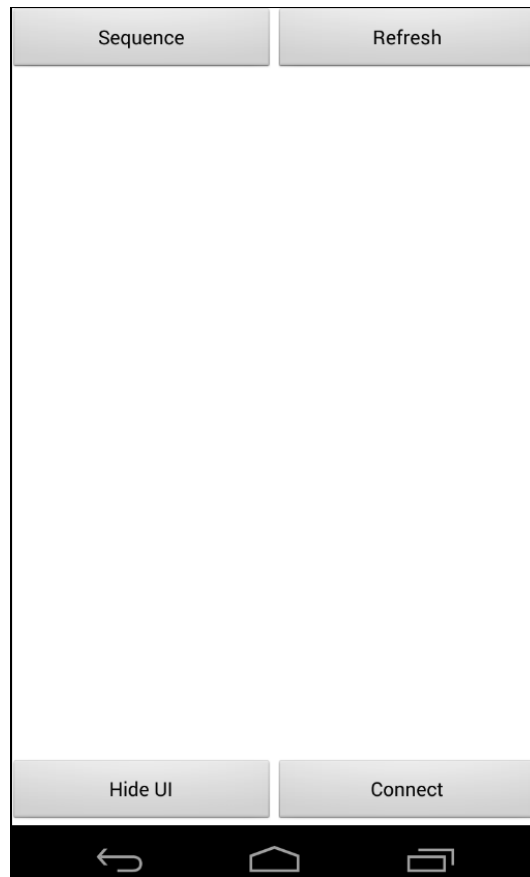


Figure 14.2: Client UI

14.2 Limitations

Though fully functioning the final product still suffers certain limitations and only works properly under certain constrained conditions. This is due to the fact the project represents proof of the concept product which mainly focuses on presenting the required features then scaling them for the real world conditions. The major limitations are as follows:

Limited FPS The server is only able to display animations and videos with FPS⁴ value not exceeding 5 FPS.

Limited detection color palette The performance of the devices detection and location is influenced by what color is used for the detection. The experiments showed the combination of white, blue and red color gained best success ratio as stated in 13.3.6.

⁴Frames Per Second

Dark environment It is required that the client devices to be detected are placed in relatively dark environment so that the colors of their screens would contrast with the low ambient light.

Fixed position After the detection is done the devices should not move more than in the range of the tile they were originally placed. This is due to the fact that device tracking was not implemented during this project as it was not considered that important task.

14.3 Further work

One of the main challenges for future is to improve the performance. Higher FPS value would allow the user to display more interesting animations or even videos. In sense of the number of client devices connected the scalability should be also improved.

Apart from the these major improvements both the development team and the customer were able to come up with useful fixes and attractive ideas about where there project might lead in the future. The main suggestions are as follows:

Device tracking One of the first steps in further development is device tracking. Once the mobile device is detected, its position is saved but not updated during time. It is probable, that on rock concerts, the audience wont be static, but it will be in motion. The current product would not be able to deal with that situation and the content would not be displayed properly. One of the solutions might be to perform the initial detection and during playing the media it is known what color should each mobile device display and therefore the playing could be considered as a another continuous detection of devices.

Grid adjustment Further work can also be focused on grid adjusting. In non-artificial situations, it is possible, that the audience will be located only in the edge of the camera's snapshot. The current prototype would not adjust to that situation, although it would be desired. It would be adequate to keep the grid's dimension but to adjust the its resolution and position it to the edge, where audience is located. One of the benefits of this extension is that the person who is controlling device with camera would not have to position and zoom it precisely.

Improved image processing Even though device detection worked fine in dark environment, there are fields in which it could be improved. The tree algorithm, described in Section 11.4.1 will not work properly, when more devices are in the same tile. Although there is a fallback algorithm, which will succeed in this case, its time complexity is too high for large audience. Hence, a new algorithm with similar performance as the tree algorithm is desired. Last but not least, if the detection worked even in worse light conditions it would significantly increased the potential of the product.

Extension of domain One of the interesting ideas for extension is not to display single color on a client device, which therefore works as a one pixel, but to display several pixels or even part of an image. It would be possible to display complex images or videos even on few devices as can be seen in Figure E.1. This concept would required not only detection of position of devices but also finding its orientation.

14.4 Summary

During this project the mobile software product titled Digital Lighter was designed. It's main purpose is to entertain the audience of the music concerts and other social events who use their smartphones or tablets to participate in the huge human-screen.

The objective of the development team was to design and implement this application as well as manage the project using agile methodology Scrum, adapt to the requirements of the customer and deliver the extensive report summarizing the whole development process.

The team committed 1363 person-hours for all the project obligations and eventually it succeeded to deliver the working product which was approved of both customer and supervisor and submit the detailed report describing the development process.

Bibliography

- [1] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. Manifesto for agile software development. <http://agilemanifesto.org/>, 2001.
- [2] M. Cohn. *Agile Estimating and Planning*. Pearson Education, 2005.
- [3] John Dooley. *Software Development and Professional Practice*. Apress Series. Apress, 2011.
- [4] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, January 1972.
- [5] Marlon Dumas and Arthur H. M. ter Hofstede. Uml activity diagrams as a workflow specification language. In *Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools, «UML» '01*, pages 76–90, London, UK, UK, 2001. Springer-Verlag.
- [6] Petter L. H. Eide. Quantification and traceability of requirements. Technical report, NTNU, 2005.
- [7] R.E. Fairley. *Managing and Leading Software Projects*. Wiley, 2009.
- [8] Center for Medicare and CMS Medicaid. Selecting a development approach. 2005.
- [9] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
- [10] Philippe Kruchten. The 4+1 view model of architecture. *IEEE Softw.*, 12(6):42–50, November 1995.
- [11] Ken Schwaber and Jeff Sutherland. The scrum guide. <https://www.scrum.org/Scrum-Guides/>, 2013.
- [12] The Institute of Electrical and Eletronics Engineers. Ieee standard glossary of software engineering terminology. IEEE Standard, September 1990.
- [13] S. Viscardi. *The Professional ScrumMaster’s Handbook*. Packt Publishing, Limited, 2013.

A | Manager manual

In this part, installation and basic usage guide will be presented.

A.1 Installation

Download application *DigitalLighterServer* from Android Play store¹ and install this application on your Android device.

A.2 Usage

After starting the server application, you will see screen similar to Figure A.1.

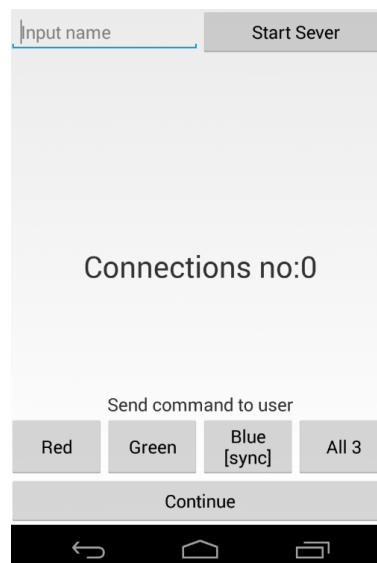


Figure A.1: Create server screen

¹<https://play.google.com/store/apps/details?id=com.silentducks.digitallighterserver>

First step you as a manager you want to do is create a new service. This can be done by typing a name (e. g. concert stage name or artist's name) into the input box and pressing button *Start Server* as shown in Figure A.2.



Figure A.2: Choose name of a server

After this action, client users can connect to this server. For every connected client, *Connections* count will increase and therefore you can see how many people want to be involved in your light show.

By pressing buttons *Red*, *Green*, *Blue* and *All 3* you can see what is the distribution of connected phones in area. After pressing a button, the clients will lit with appropriate color. You can also see the delay of network by using colors *Red* and *Green*, which are played immediately when client receive that command. On the other hand, *Blue* command is synchronized. Whenever you are ready, you can continue by clicking on button *Continue*. This will lead you to new screen, which can be seen in Figure A.3.

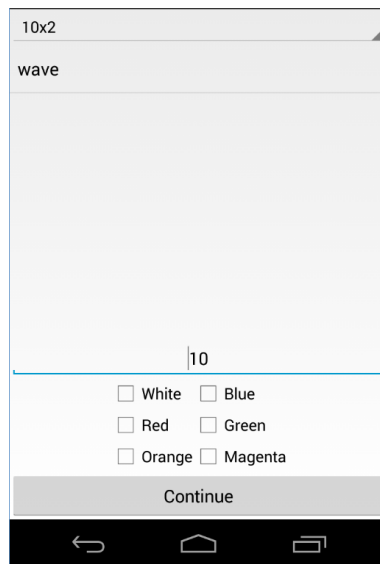


Figure A.3: Choosing media screen

In this screen you can choose media to be played by first choosing appropriate grid size from first list. After that a second list with concrete media is updated according to grid size and you have to choose specific item. After that, in next input field you can change FPS² value and last you can choose which colors should be used for detection of position of devices.

When you are ready, you can continue by clicking on button *Continue*. This will take you into last screen, similar to Figure A.4.

In this screen, there you can see realtime output from camera on your mobile phone. You can also see the grid and two buttons *Detect* and *Stop*. When you are ready, just position the camera

²Frames Per Second

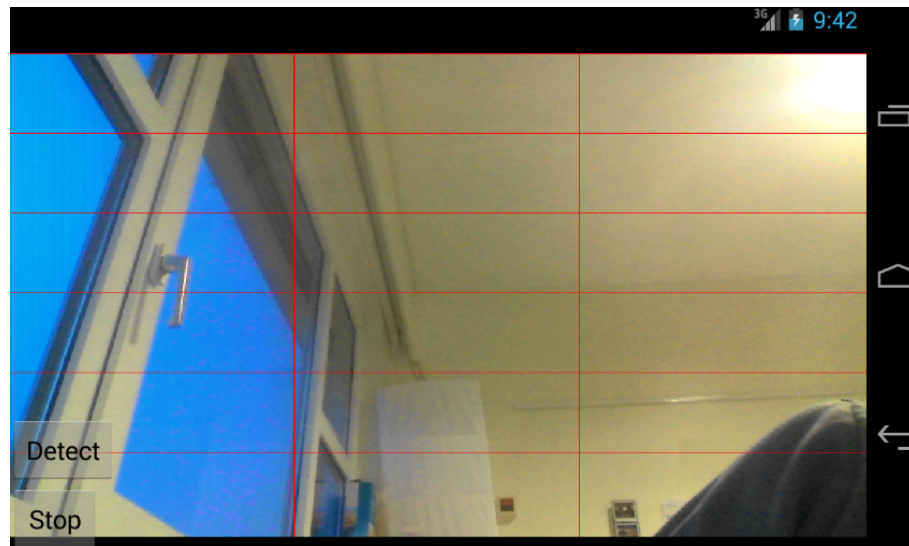


Figure A.4: Camera screen

in a way, that the audience is in screen, and click on button *Detect*. Now, you have hold still the so the clients' mobile devices do not change particular tile in grid. Detection of position can take a while, the length is dependent on number of connected clients. After the detection is ready, the media you have chosen will immediately starts to play on clients. When you have enough of fun, you can pres *Stop* button, and then use classic Android button for returning to previous screen and choose different parameters of media playing and repeat the process.

B | Client user manual

In this part, installation and usage guide will be presented.

B.1 Installation

Download application *DigitalLighter* from Android Play store¹ and install this application on your Android device.

B.2 Usage

After starting the application you can see screen similar to one shown in Figure B.1

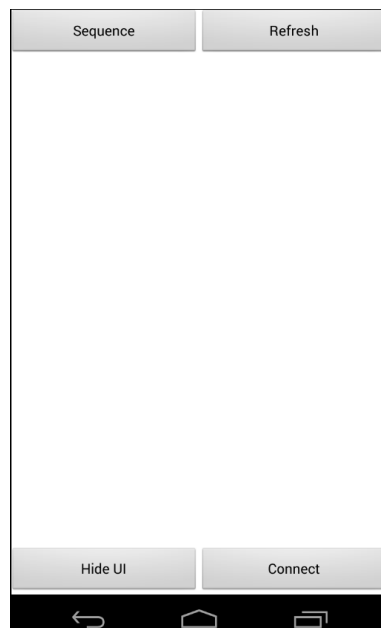


Figure B.1: Clients screen

¹<https://play.google.com/store/apps/details?id=com.silentducks.digitallighter>

When you are ready to connect to service, simply click on button *Refresh*, which will cause refreshing the list of available servers. Since that moment, the list is updated automatically. Choose your server and click on button *Connect*. From this moment, the colors of your screen are controlled by remote server. You can improve better impression from light show by clicking on button *High UI*, which will cause hiding of all control buttons and server list. When you do not want to participate in light show, you can disconnect by clicking on button *Disconnect*. Note, that this button will appear only, when you are already connected to some server.

C | Customer meetings

The meeting template consist of several parts: place, date and time, when the meeting was organized. Next, there are lists of presented and absent stakeholders. After that list of decisions with supporting reason decided on the meeting follows. Next table summarize action items – tasks, which must be done and their deadlines. After that there are listed action items which were finished since last meeting. Last table summarize pending action items throughout all meetings. This is followed by summary, which is text description of the meeting and important decisions. Template is concluded by information about next meeting.

You can see example of single customer meeting in next pages.

Minutes of meeting with the Customer (#3)

Place: Gloschaugen, Accenture lab and Oslo

Date: 29 August 2013

Time: 16:00

Members present: Peder Kongelf, Agnethe Soraa, Tomas Dohnalek, Jan Bednarik, Milos Jovac

Members not present: (none)

Decisions:

#	Decision	Reason
1	More information about minutes.	Provide more formal information about decisions. Reduce chance of misunderstanding.
2	More technical issues in appendix.	Provide more formal information about decisions. Reduce chance of misunderstanding.
3	Send customer link with notes.	Customer demand.
4	Research and use Testflight.	Customer proposal.
5	Scaling decision.	We are probably not able to make application for rock concert in 13 weeks.
6	Not focusing much on testing.	Customer proposal.
7	Milestones are not connected to sprints. Additional milestones can be added. We can use hard-coding for milestones.	Customer proposal.
8	We have to create a better agenda.	To be more efficient during meetings.
9	We will use Gantt chart and burn-down chart.	To check if we are working according to schedule.
10	Documentation should show the changes not to redefining it.	To track our work progress.
11	We have decided to end our Sprint0 prematurely and start Sprint1 on Monday 2 September.	We want to start working on core of the project asap.

12	Be more active and independent. Try to make decisions. Even a part of team can make decisions, not whole team is necessary.	Not everyone has to decide about everything.
13	Instead of story points we will use hours. We will fill worked hours on story into Gravity.	Gravity supports only hours, it is useful to compare estimate hours and real hours.

Action items:

#	Action item	Who	Due date
1	Correct previous minutes	Jan	30 Aug
2	New labels in gravity: admin, report, tools/tech	Tom	30 Aug
3	Research HockeyApp	Jan	12 Sep
4	Put paragraph in report about scaling down the project and main tasks is domain.	Milos	30 Aug
5	When we start working replace QA paragraph in Project plan with explanation how we want this app to work.	all	cont.
6	Add decision why we choose Gravity (or other tool) to report	Jan	30 Aug
7	Change our milestones. Use milestones from google docs. Turn them to stories.	Milos	30 Aug
8	Change the limitation paragraph in the research.	Agnete	30 Aug
9	Find tool to make a Gantt chart and burndown chart.	Jan	30 Aug
10	Colorcode skill table. Do risk analysis.	Tom	30 Aug
11	Put into report that we all are equal, we do not need roles.	Jan	30 Aug
12	Plan the meeting for Monday.	all	30 Aug
13	Stories out of scope should be mentioned in final report.	all	cont.
14	Set up collaboration tool (Gravity did not work)	all	30 Aug
15	Finish the first draft of the project plan	all	2 Sep

Finished action items:

#	Action item
1	Research booking of videoconference room
2	Set up LaTeX template for meeting minutes and report
3	Set up AgileZen (we decided for other tool - Gravity)

Pending action items:

#	Action item	Who	Due date
---	-------------	-----	----------

1	Research and set up the issue tracking system.	25 August
2	Add sprint goals for sprint 0.	22 August
3	Write a project plan.	30 August

Summary: We started with discussing the meeting minutes from the last meeting. Customer suggested we should put more information in meeting minutes, even the technical ones (possibly in appendix). We presented to the customer the collaboration tool Gravity. Customer suggested to use certain tags such as *admin*, *research*, *report*, *tools/technologies*, etc. Customer also suggested to use the HockeyApp service to distribute the application.

We then presented our milestones but customer did not agree and advised us to take more agile approach. Customer suggested a few specific milestones and tasks, we will utilize them. The utilization of Gantt chart and burndown chart was suggested in order to keep track on whether we are behind plan. We moved on to discuss the report. According to the customer the report should display changes so we should not rewrite the content. We might also use more diagrams and charts.

We then discussed the user stories. We should present the new roles e.g. *concert manager*. During the effort estimation we will use hours instead of points. customer then decided how important each story is.

Next meeting: 2 September 2013, exact time TBA

Meeting adjourned at 18:00.

D | Supervisor meetings

The meeting template consist of several parts: place, date and time, when the meeting was organized. Next, there are lists of presented and absent stakeholders. After that list of decisions with supporting reason decided on the meeting follows. Next table summarize action items – tasks, which must be done and their deadlines. After that there are listed action items which were finished since last meeting. Last table summarize pending action items throughout all meetings. This is followed by summary, which is text description of the meeting and important decisions. Template is concluded by information about next meeting.

You can see example of single supervisor meeting in next pages.

Minutes of meeting with the Supervisor (#3)

Place: Gloschaugen, Room iArbeid

Date: 13 September 2013

Time: 10:00

Members present: Anh Nguyen Duc, Agnethe Soraa, Tomas Dohnalek, Jan Bednarik, Milos Jovac

Members not present: (none)

Decisions:

#	Decision	Reason
---	----------	--------

Action items:

#	Action item	Who	Due date
1	Stress testing.	Milos	23 September
2	Description of architecture whole project, server-client architecture	all	cont
3	Prepare Project report draft for next meeting	all	19 September
4	Utilize time estimation spreadsheet from compendium	Jan	20 September

Finished action items:

#	Action item
1	Discuss with customer compulsory requirements and write them into final report.
2	Write subsection Requirements in Project scope section.
3	Add chapter Pre-study
4	Add chapter Evaluation, Terminology.
5	Methodology move to different section, not in General Terms.

Pending action items:

#	Action item	Who	Due date
1	Break-down structure section.	Agnethe	23 September
2	Write Time-estimation in report.	all	27 September
3	In General terms put subsection Customer Constrains, move the stuff we have there.	Agnethe	27 September
4	Do the time estimation and time tracking.	all	start asap

Summary: We started the meeting with revising what we have done last week, we tried to demonstrate the working prototype, but we came across technical issues so after all we demonstrated the prototype using youtube video. We discussed the contents of project report and supervisor suggested to describe the architecture and design patterns we used. Supervisor suggested that we should use time estimation and time tracking.

Next meeting: 20 September 2013, 10:15

Meeting adjourned at 11:00.

E | Figures



Figure E.1: Concept of application based on Digital Lighter