

iUCF
Concepts of Operations
Object Oriented Processes for Software Development, Fall 2010

Modification history:

Version	Date	Who	Comment
v0.0	08/15/00	G. H. Walton	Template
v1.0	09/07.2010	Jason Bender	Completed Concepts of Operations.
V1.3	09/20/2010	Isidro Arribas	Reformatted document to match the rest of the required documents.

Team Name: iUCF Development Team

Team Members:

- Isidro Arribas, iarribas@ist.ucf.edu
- Jason Bender, jmbender@knights.ucf.edu
- Clifton Crane, fightingpenguinz@knights.ucf.edu
- Derek Crane, derekcrane@knights.ucf.edu
- Renan Pereira, renanpereira@gmail.com

Contents of this Document

The Current System

The Proposed System

- Needs
- Users and Modes of Operation
- Operational Scenarios
- Operational Features
- Expected Impacts
- Analysis

The Current System

The current build of our application, iUCF, contains the basic user interface that we will be utilizing. By creating a tab view application, we simplify navigation for the user and maximize space efficiency and overall usability. The initial screen displayed to the user (the home tab) consists of various icons that link to the most commonly used features.

The only feature developed thus far is the campus map, which is still largely under construction. Users have the option to view the map, find a building, or find their current location on campus.

The Proposed System: Needs

- A. Need to establish a database with various tables to connect to from within the application.
 - B. Need to complete the development of the various features to be included inside the application (see section: operational features)
 - C. Need to complete the UI including the development of graphics, icons, map overlays, etc...
 - D. Need to establish app ID and various provisioning profiles for distribution so that the application can be submitted to Apple and distributed to via iTunes.
 - E. Need to set up support email account and web address for app (required for submittal to Apple)
 - F. Need to complete app and package binary with proper profiles in order to submit for review
-

The Proposed System: Users and Modes of Operation

The application we are developing has only one mode of operation. There are no special admin privileges or restricted views that only certain parties can access. Our target audience is the entire UCF population whether they are students, faculty, or just visiting.

Since the functionality of the app is general to all parties interested in information regarding the UCF campus, separate views for students, faculty, and visitors is not necessary. Therefore all users will have access to the complete functionality of the application.

The Proposed System: Operational Scenarios

The application's main feature will be the interactive map that will allow users to pinpoint their exact location on campus or simply search for a specific building. To operate the map, a user will open the app and click the map link on the home screen. A menu will be presented from which the user can choose which function of the map they wish to utilize. If the user decides to search for a building, an http request is sent to the web browser with specific map values embedded inside. The request generates a map pinpointing the buildings location and then displays the map to the user. The buildings coordinates are stored in a database and are retrieved when need. A problem that is foreseen develops when the user searches by a specific building name. If they search for "Engineering 3" or "Engineering Three" and the entry is stored in the database as "Engineering III", there could be no result returned. This proposes a serious problem that would easily frustrate users. A system needs to be devised to prevent this from occurring.

Other problems may arise in the use of user-driven content. If a user sets a time to receive an alert that their parking permit is going to expire, that time variable needs to be stored to the phones memory. It is needed to be made sure that the memory in which the variable is stored is in permanent memory and not temporary. Furthermore, when the user resets their phone or restarts it, those variable values are preserved without any problems.

The Proposed System: Operational Features

The proposed system contains many features that we have yet to implement. Features that still need to be developed further are: interactive map section, sports section (schedules, team news, other events), campus organization (information, meetings, etc...), parking section (student lots, lot locations, parking services, etc...), food section (locations of all campus restaurants), upcoming event feed, and housing information.

The above listed features are what we plan to implement in the current build of the application. If time permits, more functionality will be added. Other proposed ideas include the use of user-driven content. For instance, a user may be able to establish a profile and set expiration dates for various events and subsequently receive reminders for those events at their own discretion. For example, a student could set the date for the expiration of their parking permit and then denote that they wish to receive an alert 5 days prior to the scheduled event.

The map feature is also web driven. Therefore, if you do not have access to the web (3G or wifi) you will not be able to utilize the feature. In case of this scenario, the user should be notified that they do not have a connection and the feature is not operational because of that.

The Proposed System: Expected Impacts

We plan to release the application into iTunes for worldwide distribution. Since our app targets a specific audience (UCF), we expect that the impact of the app will be restricted to that audience and location. We expect that it will be largely utilized by the student body rather than by faculty/visitors. Student should find it helpful not only to find their way around but to stay plugged in and up to date on the latest events. The app will be especially helpful to new students. New students will be able to locate restaurants, parking, and find out about campus organizations. Furthermore, by targeting new students, your target audience changes virtually every semester. Thus the salability of the application remains at a high state, meaning there will always be a population of new students to market to.

The Proposed System: Analysis

- A. Expected improvements:
 - a. Added functionality of map features (get directions from where you are)
 - b. Build updates post-release will keep content fresh and increase re-usability
 - c. Addition of more features beneficial to target audience
- B. Disadvantages:
 - a. Targeting a very small audience compared to the rest of the world
 - b. Information is readily available on the web if students take the time to look for it
 - c. Must have internet connection to use most of the features
- C. Limitations:
 - a. Time constraints
 - b. Want to attempt to keep the app under 20MB so it can be downloaded via 3G

D. Risks:

- a. Possible copyright issues. (Using school name and logo, etc...)
- b. Apple app store may not accept application (strict approval process)

E. Alternatives and Tradeoffs:

- a. Could not use school logos to avoid any possible copyright issue but risk exposure restriction as students may not recognize application as representing UCF
- b. Could develop application to encompass multiple schools rather than just UCF therefore expanding the target audience but this comes at a great time cost.

iUCF
Project Management Plan
Object Oriented Processes for Software Development, Fall 2010

Modification history:

Version	Date	Who	Comment
v0.0	08/15/00	G. H. Walton	Template
v1.0	09/07/2010	Clifton Crane	Added static project information such as group members, project title, and course information.
v1.1	09/18/2010	Renan Pereira	Initiated Project Overview, Reference Documents, Applicable Standards, Project Team Organization, Tools and Computing Environment, Software Life Cycle Process, Configuration Management
V1.2	09/20/2010	Isidro Arribas	<ul style="list-style-type: none">• Revised Project Overview.• Reworded a couple of sections to be easier to understand.• Fixed formatting for better viewing.• Changed name from iCampus to iUCF.

V1.3	09/21/2010	Derek Crane	<ul style="list-style-type: none"> Changed CVS repository to SVN repository.
------	------------	-------------	---

Team Name: iUCF Development Team

Team Members:

- Isidro Arribas, iarribas@ist.ucf.edu
- Jason Bender, jmbender@knights.ucf.edu
- Clifton Crane, fightingpenguinz@knights.ucf.edu
- Derek Crane, derekcrane@knights.ucf.edu
- Renan Pereira, renanpereira@gmail.com

Contents of this Document

Project Overview
 Reference Documents
 Applicable Standards
 Project Team Organization
 Deliverables
 Software Life Cycle Process
 Tools and Computing Environment
 Configuration Management
 Quality Assurance
 Risk Management
 Table of Work Packages, Time Estimates, and Assignments
 PERT Chart
 Technical Progress Metrics
 Plan for tracking, control, and reporting of progress

Project Overview:

The iUCF application is an iPhone application for the UCF campus. This application will include many features geared towards students, such as an interactive map, sport and event schedules, date deadline alerts, and information on local services and housing. The application will be portable, in its code and structure, which would allow other universities to easily adapt it to their needs.

Reference Documents

- Concept of Operations <include link here>
- Google Maps API Reference
<<http://code.google.com/apis/maps/documentation/javascript/reference.html>>
- Apple iOS Developer Library <<http://developer.apple.com/library/ios/navigation/>>

Applicable Standards

- Coding Standard: A proper indentation and cleanly commented coding standard will be held.
 - Document Standard: All documentation will follow the general format presented in the class website's templates. All modifications will be logged in the document's Documentation History. All project documents will be stored in the group's SVN repository and are subject to versioning. Font for all documents will be Arial size 11pt. All table text will be Center Left aligned.
 - Artifact Size Metric Standard: This section is not applicable to our project.
-

Project Team Organization

Our team is comprised of five UCF Computer Science students: Jason Bender, Renan Pereira, Derek Crane, Isidro Arribas, and Clifton Crane. Management of the group will be done as a community effort, where all project decisions are discussed and implemented by all members. However, Jason has been designated Lead Developer due to his extensive prior experience with iPhone and Objective-C programming. All project documentation will be a team effort; each member will be initially assigned a specific document, or portion of a document, to complete, but all members are free to add and modify all documents as needed. The group will communicate through email, postings on the group's Google Groups website, and face-to-face meetings after class.

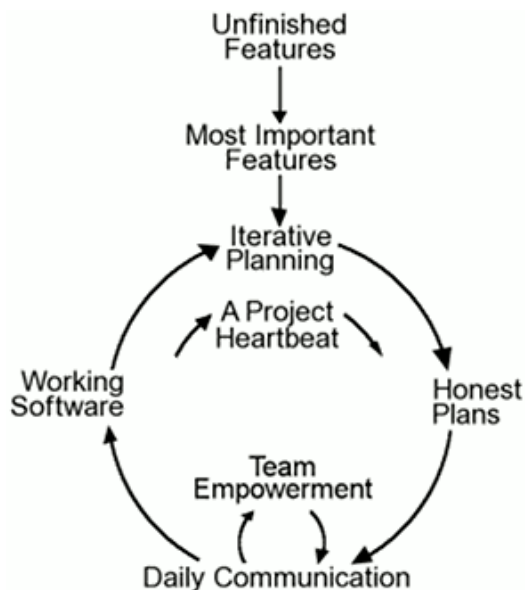
Deliverables:

Artifact	Due Dates <some will have multiple deliveries>
Meeting Minutes	1 hour after each face-to-face meeting
ConOps	09/23/2010
Project Plan	09/23/2010
SRS	09/23/2010
High-Level Design	10/26/2010
Detailed Design	10/26/2010
Test Plan	09/23/2010

User's Manual	11/30/2010
Final Test Results	11/30/2010
Source, Executable, Build Instructions	11/30/2010
Project Legacy	11/30/2010

Software Life Cycle Process

Due to the project's highly dynamic nature, an Agile Development (XP) approach will be implemented.



Tools and Computing Environment

Operating System: Mac OSX, iOS (iPhone's operating system)

Development Environment: Xcode

Programming Languages: Objective-C, Javascript, HTML, SQL, PHP

Database: MySQL

Additional Notes: Some components of the application will be stored in a web server and accessed using the phone's web browser.

Configuration Management

Each group member is responsible for uploading all documentation and source files to the project's SVN repository in order to maintain proper versioning. Files will be considered "Final" versions only after approved by all members of the team.

Quality Assurance

Quality Assurance testing will have two components: iPhone application testing and web application testing. Testing of web application will be done by all team members. Since iPhone application testing requires special hardware this portion of testing will be completed by any member(s) of the team with access to the required an iPhone or iPhone emulator. Test Cases to be used in QA process will be described in detail in the Test Case Plan.

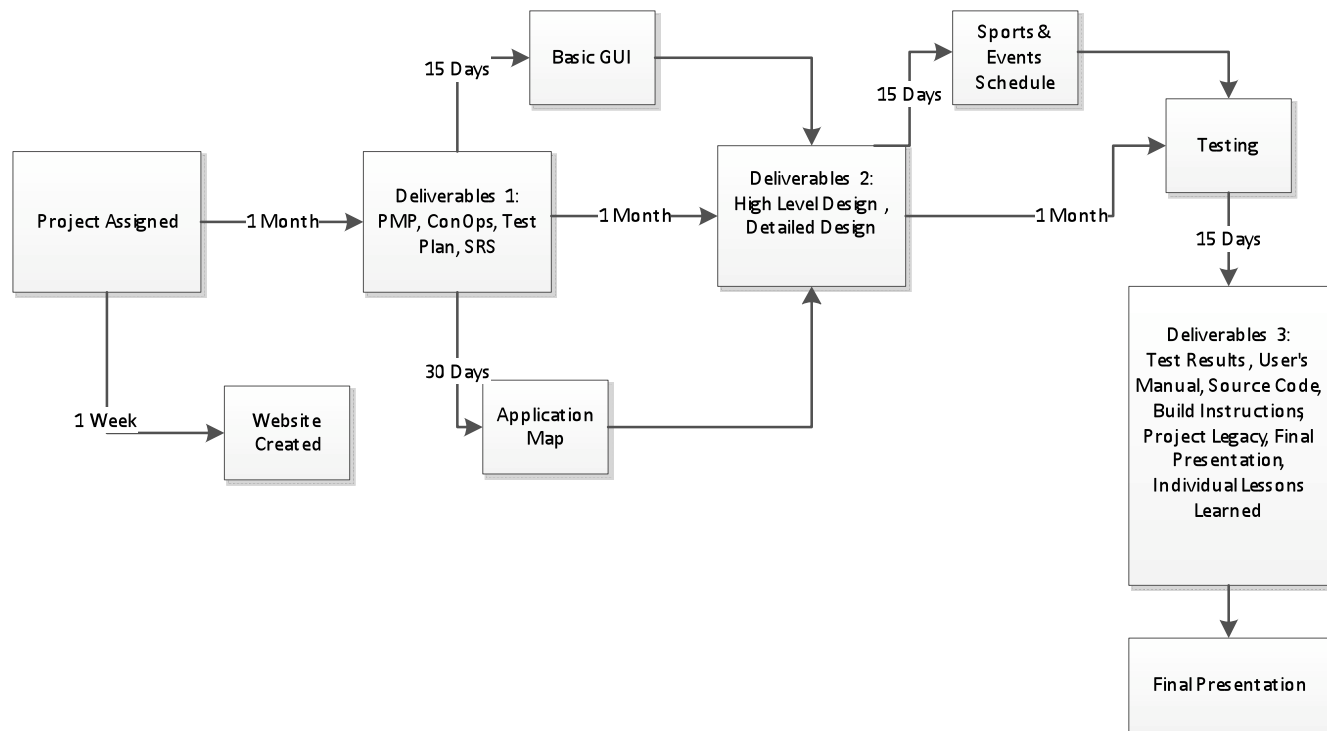
Risk Management

Risk	Risk Prevention
Losing team member due to class drop	Maintain constant communication with all members of the team to ensure everyone is staying current with class material. Also make sure documentation and source code is always current and clear to other team members.
Application becoming outdated	Use database and web server based storage for application data, allowing updates to be performed on the server side, instead of patches.

Table of Work Packages, Time Estimates, and Assignments

Work Package	Expected Duration	Assigned To
Documentation	30 days	All Team Members
Basic GUI	15 days	Jason Bender
Application Map	30 days	All Team Members
Sports and Events Schedule	15 Days	All Team Members
Testing	15 Days	TBD
Final Revision	3 days	All Team Members

PERT Chart



Technical Progress Metrics

Work Package	Expected Duration	Actual Duration	Assigned To
Documentation	30 days	7 days	All Team Members
Basic GUI	15 days	3 days	Jason Bender
Application Map	30 days	7 days	All Team Members
Sports and Events Schedule	15 Days		All Team Members
Testing	15 Days		TBD
Final Revision	3 days		All Team Members

Plan for tracking, control, and reporting of progress

Each member will update their assigned documents weekly, assuming changes have taken place. All changes to documentation and source code will be tracked using the document's Modification History and version control.

On each team member's update, all group members will read and verify the document(s) and give feedback and/or suggestions.

iUCF
Software Requirements Specification
Object Oriented Processes for Software Development, Fall 2010

Modification history:

Version	Date	Who	Comment
v0.1	09/07/10	Clifton Crane	Added static project information such as group members, project title, and course information.
V0.2	09/15/10	Derek Crane	Converted to .odt and started filling out some details.
V0.3	09/16/10	Jason Bender	Added Use Case diagram and filled out other sections.
V0.4	09/21/10	Development team	Added several new requirements.
V0.5	09/21/10	Derek Crane	Added the final requirements.

Team Name: iUCF Development Team

Team Members:

- Isidro Arribas, iarribas@ist.ucf.edu
- Jason Bender, jmbender@knights.ucf.edu
- Clifton Crane, fightingpenguinz@knights.ucf.edu
- Derek Crane, derekcrane@knights.ucf.edu
- Renan Pereira, renanpereira@gmail.com

Contents of this Document

Introduction

- Software to be Produced
- Reference Documents
- Applicable Standards

Product Overview

- Assumptions
- Stakeholders
- Event Table
- Use Case Diagram

- Use Case Descriptions

Specific Requirements

- Functional Requirements
- Interface Requirements
- Physical Environment Requirements
- Users and Human Factors Requirements
- Resource Requirements
- Quality Assurance Requirements

Supporting Material

SECTION 1: Introduction

Software to be Produced:

- The software we are producing is an iPhone application based on various elements of UCF. Refer to the Concepts of Operations document for more information pertaining to specific features.

Reference Documents:

- Concept of Operations
- Project Plan

Applicable Standards:

- The software must provide users the ability to view a campus map with or without an internet connection.
-

SECTION 2: Product Overview

Assumptions:

- The software will run on iPhone 3 and iPhone 4.
- The software will be small in size and memory usage. Less than 20 mega-bytes in size and approximately 30 mega-bytes of memory usage.
- The software will be accepted in the app-store and be available as COTS to the public.

Stakeholders:

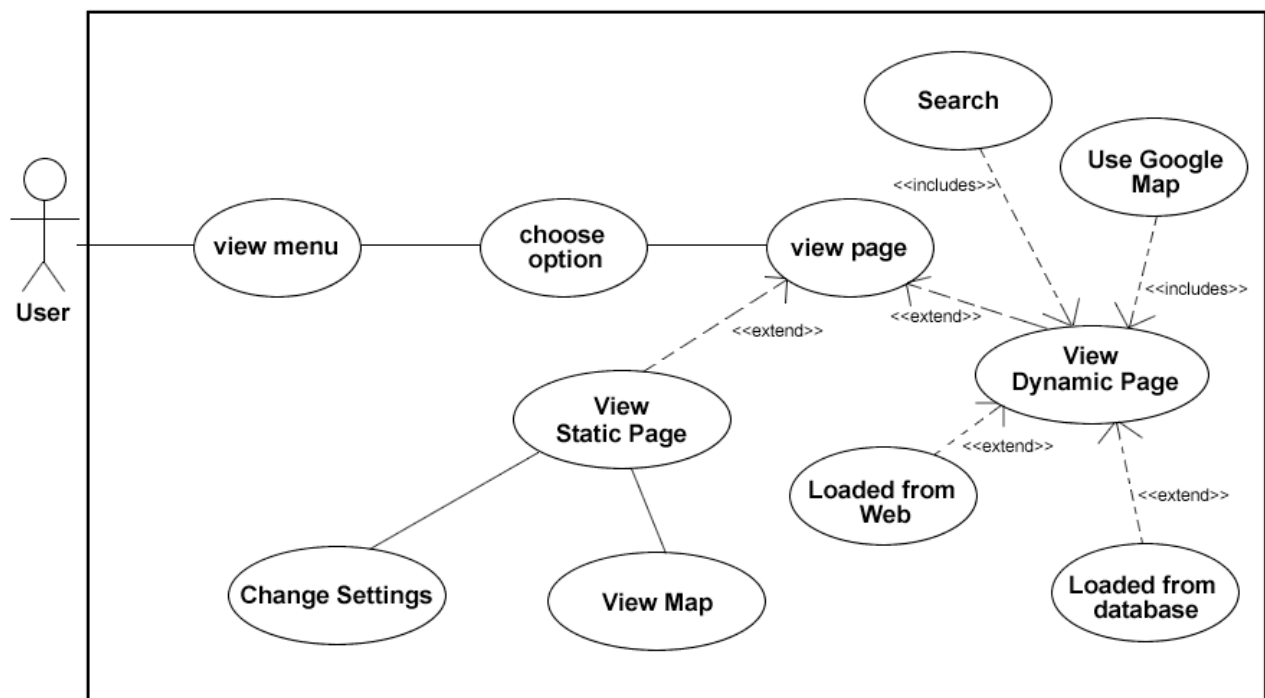
- Our team: Our development team has a stake in the project as it directly affects our course grade.
- Dr. Turgut: Our teacher has a stake in the project since she is teaching us the course material.
- UCF student body: These are our potential customers so they obviously are the largest stakeholders.

Event Table:

Event Name	External Stimuli	External Responses	Internal data and state
Search	A user of the software uses the search feature.	The software will display all associated fields from the data-base depending upon the search	The software will enter the display search

		string.	results state.
View Map	A user of the software is actively viewing and manipulating the map.	The software will handle events such as scrolling and pinpointing the user's location.	The software will enter the view-map state.
Click Map Pinpoint	A user of the software that is actively viewing the map and click on a building	The software will fetch building info from a database and populate a webpage with that information and then display it to the user.	Software will enter a web-view state

Use Case Diagram:



Use Case Descriptions:

- User opens application and views menu
- User chooses an option from the menu
 - Request is processed and appropriate view displayed
 - Page may be dynamic
 - Content fetched from database or from web and page is populated accordingly
 - Page may be static
 - Content is loaded internally from data embedded in application
- User chooses change settings option

- Static page is loaded with multiple setting options
 - Altering a setting changes variable values in memory
- User chooses view map option
 - Static map is loaded and shown to user
- User chooses option that uses google maps
 - Proper map is fetched from the web and displayed
- User chooses search feature
 - Application communicates with a database to return search results
- User chooses another option other than listed above
 - Page either loaded internally or fetched from the web and displayed

SECTION 3: Specific Requirements

Functional Requirements:

No: 1
Statement: The search function shall distinguish between character, numerical, and acronym representations of data.
Source: Development team.
Dependency: None
Conflicts: None
Supporting Materials: Example: {3, three, iii} or {Harris Engineering Center, HEC}
Evaluation Method: Unit testing.
Revision History: Derek Crane, Clifton Crane, Jason Bender – 9/21/2010.

No: 2
Statement: The developers shall use the iPhone SDK and operating system.

Source: Development team.

Dependency: None

Conflicts: None

Supporting Materials: None

Evaluation Method: Successful code compilation.

Revision History: Derek Crane, Clifton Crane, Jason Bender - 9/21/2010.

No: 3

Statement: The system shall require a remote server driven database.

Source: Development team.

Dependency: 1

Conflicts: None

Supporting Materials: None

Evaluation Method: Table entries.

Revision History: Derek Crane, Clifton Crane, Jason Bender - 9/21/2010.

No: 4

Statement: The interactive map shall support pinpoint location defined by the user.

Source: The development team.

Dependency: 1

Conflicts: None.

Supporting Materials: Google Maps

Evaluation Method: Interactive Map Test.

Revision History: The development team, 09/21/2010.

No: 5

Statement: The system shall have the ability to change user settings.

Source: The development team.

Dependency: 7.

Conflicts: None.

Supporting Materials: Use Case Diagram

Evaluation Method: Testing done by development team and testers.

Revision History: The development team, 09/21/2010.

No: 6

Statement: The system shall allow users to input custom alerts.

Source: Development team.
Dependency: 7.
Conflicts: None
Supporting Materials: Concept of Operations
Evaluation Method: Users can input custom alerts.
Revision History: Derek Crane - 9/21/2010

Interface Requirements:

No: 7
Statement: The interface shall give the user viewing access to sub functions while providing the ability to return to the previous or main menu.
Source: The development team.
Dependency: None.
Conflicts: None.
Supporting Materials: None.
Evaluation Method: Layout and Design Test will be the evaluation method.
Revision History: Clifton Crane, 09/21/2010

Physical Environment Requirements:

No: 8
Statement: The physical environment for the project shall meet the physical environment requirements of the

phone being used.

Source: Apple Inc

Dependency: None.

Conflicts: None.

Supporting Materials:

<http://www.apple.com/iphone/specs.html> iPhone 4 Tech Specs

iPhone 3GS Tech Specs

Evaluation Method: It is up to the user to maintain proper physical environmental requirements pertaining to their phone.

Revision History: Clifton Crane, 09/21/2010

Users and Human Factors Requirements:

No: 9

Statement: The application shall support users with moderate iPhone application and internet skill levels.

Source: The development team.

Dependency: None.

Conflicts: None.

Supporting Materials: None.

Evaluation Method: Testers with no knowledge of the functions of our application used to evaluate.

Revision History: Clifton Crane, 09/21/2010

Resource Requirements:

No: 10
Statement: The development team and users shall have the physical device required by the application (iPhone).
Source: The development team.
Dependency: All requirements are dependent on this requirement unless the application is loaded into the simulator by the development team.
Conflicts: None.
Supporting Materials: Apple's iPhone Website
Evaluation Method: User owns the physical device.
Revision History: Clifton Crane, Derek Crane, 09/21/2010

No: 11
Statement: The development team shall register a developer account with Apple for distribution.
Source: The development team, Apple Inc.
Dependency: None.
Conflicts: None.
Supporting Materials: Apple Developer
Evaluation Method: The development team will confirm the requirement has been met.
Revision History: Clifton Crane, Derek Crane, 09/21/2010

Quality Assurance Requirements:

No: 12
Statement: The database shall be maintained every semester
Source: Users
Dependency: 3
Conflicts: None
Supporting Materials: None
Evaluation Method: If all buildings, departments, and sports schedules are up to date.
Revision History: Derek Crane - 09/21/2010

iUCF
Test Plan
Object Oriented Processes for Software Development, Fall 2010

Modification history:

Version	Date	Who	Comment
v0.0	08/15/00	G. H. Walton	Template
v1.0	09/07/2010	Clifton Crane	Added static project information such as group members, project title, and course information.
v1.1	09/17/2010	Clifton Crane	Changed name from iCampus to iUCF. Completed sections one and two, began adding tests to part 4.
V1.2	09/18/2010	Clifton Crane	Added the Scheduler test to part 4.

			Completed part 3 as well.
V1.3	09/20/2010	Isidro Arribas	<ul style="list-style-type: none"> • Reformatted document to match Project Management Plan. • Reworded multiple sections for better understanding. • Added extra information to better explain the project.
V1.4	09/21/2010	Clifton Crane	<ul style="list-style-type: none"> • Added Mac operating system requirements for testing done on the simulator.

Team Name: iUCF Development Team

Team Members:

- Isidro Arribas, iarribas@ist.ucf.edu
- Jason Bender, jmbender@knights.ucf.edu
- Clifton Crane, fightingpenguinz@knights.ucf.edu
- Derek Crane, derekcrane@knights.ucf.edu
- Renan Pereira, renanpereira@gmail.com

Contents of this Document

Overall Objective for Software Test Activity

Reference Documents

Description of Test Environment

Overall Stopping Criteria

Description of Individual Test Cases

Overall Objective for Software Test Activity

The overall objective of the software testing period is to remove any bugs, incorrect information or typos, and ensure the system is stable and functions properly. The display layout will also be included in the test to ensure images are clear, correctly positioned and of the correct dimension. The testing will be done on both the iPhone 3G and the iPhone 4, both which are manufactured by Apple, Inc. The development team will be sent issues that testers encounter, allowing the team to correct these problems. This process will result in a more efficient and effective application.

Reference Documents

- Concept of Operations <include link here>
- Project Plan <include link here>
- SRS <include link here>

Description of Test Environment

We will be testing our application on the iPhone 3G and the iPhone 4. We will also run the tests on the simulator included with the iPhone SDK. Testers wanting to perform tests using the simulator will require a computer with a Mac operating system installed. The testing will be done by the development team, along with persons selected by the development team. By using testers with no knowledge of the application it will allow them to find unknown problems or issues, resulting in an increase of overall success of the project. The testing is going to be done on the phone will be using the same environment which will be distributed to the client. Testing done on the simulator will be done in a different environment than that of the final product; however the testing done on the simulator will also help the development team ensure that the product is as reliable as possible.

Overall Stopping Criteria

If errors are found during the testing period, the testing will continue unless the error encountered is fatal to the application. If a fatal error is found, the tester will try to recreate this error for a second time to help the development team determine the cause of the fatal error. Testing can be done for a set duration of time, where after the testers and development team will meet and discuss the results of their tests. If errors are reported, the development team will try to fix those errors before the same test is ran again. The tester will be asked to recreate the initial problem, and ensure that the changes made by the development team were effective. If testers find no errors, the testers will be asked to run the tests at least two more times using different data to determine if in fact the application is running properly.

The testers will not only check for obvious errors, as MySQL errors or a JavaScript error, but also check for logical errors. This means that the testers will have to be aware of what the application is going to provide, allowing them to make reasonable feedback to the tests. If the testers have any questions about how the application should function they can always contact the development team. After that, the development team will determine if the application is good enough to be delivered.

When the team determines if the product is good enough to be delivered, they take into the account that the application and its sub-functions are working properly. Being that our application is being developed for a smart phone, functionality is not the only concern for error. Cosmetic errors are also unacceptable and must be fixed before delivery. The application requires that no known errors exist when released. Errors that were not detected during the testing period will be fixed as they are found.

Description of Individual Test Cases

Describe EACH individual test to be run: (so if you plan to run 20 test cases, you would answer the following questions for each of them. Either a tabular format or a bulleted format is acceptable.)

- Test Objective: <exactly what are does this specific test demonstrate?>

- **Test Description:** <exactly what will you test? What test data will be used (specifically -- what data values, what data files? This data must be determined in advance. So if you plan to use a test file, include a link to the file here. If you plan to key in specific data, include the data here. Make sure the data you choose will allow you to achieve your stated objective for this test.>
- **Test Conditions:** <Under what conditions will you run this test? This is relevant for software for which there are multiple "modes". For some systems, the test conditions are totally described in the above test environment section -- in this case, the response to this would be "See Test Environment".
- **Expected Results:** <If the test executes correctly, what will be the result -- i.e., exactly what will the output look like; what will be the resulting data in the database, ...>

1. Layout and Design Test	
Test Objective:	<ul style="list-style-type: none"> • Show that the design layout is complete, correctly aligned and well displayed. • Demonstrate that the menu options are working properly. • Check that links direct to the correct location.
Test Description:	<p>The tester will navigate through the application, ensuring that all menus are properly linked to their appropriate sub functions. The tester will also make sure that everything is well aligned and images are clear and easily viewable. They will also check for any graphical glitches that might occur. As well as checking that any links provided in the application point to the correct location.</p> <p>No data is needed for this test.</p>
Test Conditions:	See Test Environment
Expected Results:	If the test results are successful, all menus and images will be displayed properly (clarity, alignment) on all versions of phones being included in the test. The navigation menus and links will work properly and point to the correct location.
2. Building Search Test	
Test Objective:	<ul style="list-style-type: none"> • Demonstrate that the search-by-building feature is not only functioning, but also providing the correct information. • Show that the application is successfully accessing the online database to pull the building information.
Test Description:	The tester will be given a set of buildings that are located on the university's campus and stored in the database. The tester will then use the search function of the application to locate desired buildings from the provided list. The tester will look for any inaccuracy in the information, or way things are displayed.
Test Conditions:	Conditions for the test will require that the user is connected to the internet, via 3G connection or wireless connection.

Expected Results:	If the test results are successful, the queried building will be found in the online database and the corresponding building information will be sent and displayed to the user.
3. Interactive Map Test	
Test Objective:	<ul style="list-style-type: none"> • Ensure that the interactive map is functioning properly • Easy navigation of the interactive map.
Test Description:	<p>The tester will be given time to learn how to use the interactive map. The tester will then test each function the map has to offer. The tester will look for any inaccuracies that might happen, as well as any graphical glitches displayed. The tester will also describe to use how easy the navigation of the interactive map was. Allowing the tester to give suggestions on how the interactive map would be easier to use.</p> <p>No data is needed for this test.</p>
Test Conditions:	Conditions for this test will require that the user is connected to the internet, via 3G connection or wireless connection.
Expected Results:	If the test results are successful, that means the tester was able to navigate the map properly, and locate the building they were looking for on the map. The tester was also able to click on the pin at the location of the building, and display corresponding information. Also, the tester had an easy time navigation and finding the information that was being searched for.
4. Static Campus Map Test	
Test Objective:	<ul style="list-style-type: none"> • Show that users are successfully able to access the static campus map provided with the application.
Test Description:	<p>The tester will ensure that the static campus map is properly loaded and displayed on the phone. The tester will also include the clarity and scale of the image in their results, as well as any inaccuracies that may occur.</p> <p>No data is needed for this test.</p>
Test Conditions:	Internet connection is not required for this test. See Test Environment.
Expected Results:	If the test results are successful, the map will be loaded properly and displayed clearly.
5. Scheduler Test	

Test Objective:	<ul style="list-style-type: none">• Show that the user scheduler sub function is properly storing events.• Scheduler retrieves events properly.• Scheduler reminds the user of upcoming events.
Test Description:	The tester will input a series of events into the calendar and determine if the events are being stored correctly. This will be done by after storing the information, checking to see if the information stored matches the information displayed. The tester will also check if the application lets him or her know about upcoming events.
Test Conditions:	Internet connection is not required for this test. See Test Environment.
Expected Results:	If the test results are successful, the application will properly store all user defined events, access those event properly, and display the event to the user when it is in the near future.
