Department of Computer Science and Engineering The University of Texas at Arlington

GLADOS PRESENCE Home Managment System

Christian Burkhart Shawn Chandwani Jose Flores George Oroni John Podolanko

Last Updated: July 21, 2014 1:08 AM

Table of Contents

Table o	f Contents	2
Docum	ent Revision History	5
List of I	Figures	6
List of '	Tables	7
1. Prod	luct Concept	8
1.1	Purpose and Use	8
1.2	Intended Audience	8
2. Prod	luct Description and Functional Overview	9
2.1	Features and Functions	9
2.2	External Inputs and Outputs	10
2.3	Product Interfaces	11
3. Cust	tomer Requirements	
3.1	Centralized Server	
3.2	Compatible Light/Fan Switches	
3.3	Electronic Door Lock	19
3.4	Digital Thermostat	19
3.5	RFID Reader/Writer	19
3.6	Programmable RFID Bands / Tags	20
3.7	Android [™] Application	20
3.8	Database Management System (DBMS)	20
3.9	User Account Administration	21
3.10	User Settings Profiles	21
3.11	User Login	22
3.12	Application Menu Bar	22
3.13	Switch Lights On/Off	23
3.14	Set Fan Speeds	23
3.15	Lock/Unlock Door	24
3.16 SRS Ve	Adjust Thermostat	24 GLaDOS

3.17	User Notifications	25
4. Pac	kaging Requirements	
4.1	Included Components	
4.2	Module Weight	
4.3	Module Sizes	27
4.4	Android TM Application Publication	27
4.5	Power Supplies	27
4.6	User Manual	
5. Perf	formance Requirements	
5.1	RFID Recognition Distance	
5.2	Settings Activation Latency	
5.3	Database Query Response Time	
5.4	Server Response Time Over WLAN	
5.5	Server Response Time Over 3G/4G Networks	
5.6	Android [™] Application Startup Delay	
6. Safe	ety Requirements	
6.1	Electrical Circuit Grounding	
6.2	Packaging Safety	
6.3	Operating Temperature	
7. Maiı	ntenance and Support Requirements	
7.1	Maintenance Scope	
7.2	Android [™] Version Support	
7.3	Source Code Commenting	
7.4	Source Code / Schematics / Documentation Availability	
7.5	User Manual	
7.6	Installation Support	
7.7	Language Support	
7.8	Multiple User Support	
8. Oth	er Requirements	
8.1	Connectivity Tolerance	
8.2	Application Security and Privacy	
8.3	Intrusion Prevention	
SRS V	ersion 1.0 3	GLaDOS

	8.4	Manual Override	39
9.	Accep	otance Criteria	40
	9.1	Verify User Account Creation	40
	9.2	Verify Light Control	41
	9.3	Verify Fan Control	41
	9.4	Verify Door Lock Control	42
	9.5	Verify Thermostat Control	42
1(). Use	Cases	43
	10.1	Switch Lights On/Off – Sensor-Based	43
	10.2	Switch Lights On/Off – Application-Based	43
	10.3	Lock/Unlock Door – Sensor-Based	44
	10.4	Lock/Unlock Door – Application-Based	44
	10.5	Change Temperature – Sensor-Based	44
	10.6	Change Temperature – Application-Based	45
	10.7	Change Fan Speed – Sensor-Based	45
	10.8	Change Fan Speed – Application-Based	45
	10.9	Create User Account – Application-Based	46
	10.10	User Sign In – Application-Based	46
	10.11	Create Settings Profile – Application-Based	47
	10.12	View/Change Settings Profiles – Application-Based	47
11	. Feas	ibility Assessment	51
	11.1	Scope Analysis	51
	11.2	Research	51
	11.3	Technical Analysis	52
	11.4	Cost Analysis	52
	11.5	Resource Analysis	53
	11.6	Schedule Analysis	54
12	2. Futu	re Items	57
	12.1	Customer Requirement 3.18: Home Stereo Interface	57
	12.2	Customer Requirement 3.19: TV Interface	57
	12.3	Customer Requirement 3.20: Security Camera Interface	57

Document Revision History

Revision Number	Revision Date	Description	Rationale
0.1	3-Jul-14	SRS First Draft	Initial SRS distribution
0.2	16-Jul-14	Initial Revisions / Cleanup	Made customer's & director's revisions
0.3	18-Jul-14	Post-Peer Review & Feasibility Finalization	Made revisions based on peer feedback
1.0	21-Jul-14	SRS Gate Review	Use cases rewritten to match requirements

Figure #	Title	Page #
2-1	Login screen	11
2-2	Create user screen	12
2-3	Home screen	13
2-4	Light control screen	14
2-5	Thermostat control screen	15
2-6	Fan control screen	16
2-7	Door lock control screen	17
10-1	Light control use case diagram	48
10-2	Door lock control use case diagram	48
10-3	Temperature control use case diagram	49
10-4	Fan control use case diagram	49
10-5	Door lock control screen	50

List of Figures

List of Tables

Table #	Title	Page #
2-1	External Inputs and Outputs	10
11-1	Cost Analysis of Hardware Required	53
11-2	Function-Point Analysis	54
11-3	Adjustment Factors	54
11-4	Unadjusted & Adjusted Function-Point Totals	55
11-5	Jones' First Order Estimation	55
11-6	Semi-Detached CoCoMo Estimation	56

1. Product Concept

This section describes the purpose, use, and intended user audience for Presence, the RFID/sensor-based home management system. The purpose of this product is to make the life of the user easier by providing automation of everyday common tasks through sensors. This product will also have an accompanying Android[™] application that both communicates with and provides control over certain household functionalities.

1.1 Purpose and Use

Presence has several functionalities and requirements that it is expected to perform for its user. It automates several common tasks at home such as turning lights off/on, regulating the temperature, locking/unlocking doors, and other tasks. Presence can allow tasks to be automated or manually configured. The purpose behind these features is to provide the user with a system that automates the most common and menial of daily tasks and to give the user a single platform to manually configure these tasks.

Since the main purpose of this product is aimed at helping the user automate certain aspects of their daily life, Presence will autonomously make decisions based on information received from sensors or manual input. Presence will sense when a user is present and adjust settings in the home. In addition, Presence is designed to simplify the user's daily life.

One scenario in which Presence will be useful is when a user returns home with their hands full carrying grocery bags or a briefcase. This situation is where Presence really works on behalf of the user. By utilizing sensors and having a centralized server to process this data, the user does not have to set something down to find their keys to unlock the door and then turn on a light.

Overall, Presence's goal is to make the user's life much simpler.

1.2 Intended Audience

The target audience for Presence includes energy conservationists, tech-savvy individuals, and smartdevice enthusiasts of all ages. The energy conservationist will find this device useful because it will turn off lights and adjust the thermostat when the user's presence is not sensed. The tech-savvy and smartdevice enthusiasts will enjoy the numerous automated functions Presence has to offer. There is an assumption that the audience will own an AndroidTM mobile device and have a wireless network in order to take full advantage of Presence's feature set.

2. Product Description and Functional Overview

This section provides the reader with an overview of Presence. The primary operational aspects of the product, from the perspective of the end users, maintainers, and administrators, are defined here. The key features and functions found in Presence as well as critical user interactions and user interfaces are described in detail.

2.1 Features and Functions

Presence will consist of a server, RFID tags and sensors, and infrared transmitters and receivers. They will all work in conjunction with an Android[™] application. The server will run a lightweight Linux operating system that enables control of sensors and various switches and devices in the home from lights to ceiling fans to door locks to security cameras. Users will carry an RFID tag in their possession, which will trigger events to happen in the home when in close proximity to the RFID sensors.

User settings can be automated. For instance, when a user comes home, the RFID sensor detects the user and sends a signal to the server which unlocks the door and turn on the lights. The system also adjusts the thermostat to the user's preferred settings. When the user leaves and the sensor no longer detects a targeted presence, the door is locked, the lights are turned off, and the thermostat is shut off or adjusted to conserve energy.

Presence's associated mobile application allows the user to create, modify, and delete various home settings on a per-user basis. Presence also allows manual control over various household settings. For instance, if a user is at home and decides to watch a movie, the user can select a setting mode from the mobile application that turns off the lights in the room while the user is present. When the movie is finished, the user can select a different setting mode that turns the lights back on. If the user wishes to go to sleep, the user can select a setting mode that turns off the lights, adjusts the thermostat, and turns on a fan. These are just a few of several configurable possibilities.

Presence will accommodate several users, each of which can have their own settings configurable on the server from the mobile application. Some settings such as thermostat control will have a hierarchal list of users whose preferences override those of others.

Lastly, the administrator side of the mobile application allows a user to monitor conditions and override settings in the home when no one is present. The user may view the feed from security cameras and may also manually adjust settings. For instance, if a user is out of town for a few days, the user may wish to turn on the lights and the TV in the evenings to indicate the presence of a person in the home to would-be burglars.

2.2 External Inputs and Outputs

Name	Туре	Description	Use
Application - Create Account	Input	The user inputs their account information to create an account	The user will create the account which they can use to interface with the system
Application - Login	Input	The user inputs their username and password into the application and is then able to interface with the system	The user will be able to log into their account
Application - Set Lights	Input / Output	The User inputs an action to the android application	The user will be able to toggle the switch for the lights
Application - Set fans	Input / Output	The User inputs an action to the android application	The user will be able to toggle the switch for the fan settings
Application - Lock/Unlock Doors	Input / Output	The User inputs an action to the android application	The user will be able to toggle the door locks
Application - Set Temperature	Input / Output	The User inputs an action to the android application	The user will be able to set the temperature of the thermostat
Application - Create Profile	Input	The user will be able to create a profile with preferred setting.	The user can switch between these anytime and set their preferred one as the default for their RFID tag
Server	Input / Output	The server communicates with all devices and the application and stores all information	Used to store/retrieve information and send control signals to all devices
Lights	Output	The lights will communicate with the server to make changes	The lights will turn on/off
Fans	Output	The fans will communicate with the server to make changes	The fans will switch between off/low/medium/high
Thermostat	Output	The thermostat will communicate with the server to make changes	The temperature will change
Door Locks	Output	The door locks will communicate with the server to make changes	The door lock will toggle between locking/unlocking

Table 2-1 External inputs and outputs

2.3 Product Interfaces



Figure 2-1 Login screen



Figure 2-2 Create user screen

ان. ج	
Log Out	
Lights Temprature	
Door Locks Fans	
Create New Profile	
My Profiles	

Figure 2-3 Home screen



Figure 2-4 Light control screen



Figure 2-5 Thermostat control screen



Figure 2-6 Fan control screen



Figure 2-7 Door lock control screen

3. Customer Requirements

This section specifies the hardware and software requirements set forth by the customer for Presence. It will cover each software feature and give a description to establish a look-and-feel as well as give an idea why it was something that the customer wanted in the product. The requirements will also be given a priority to give a general idea of how important each feature is to Presence.

3.1 Centralized Server

3.1.1 Description: There will be a centralized server that controls all of the required household peripherals, stores user accounts and settings, and maintains the database. It will also interface with the RFID reader/writer to associate RFID tags to specific users.

3.1.2 Source: GLaDOS

3.1.3 Constraints: Must have sufficient processing power and memory to control multiple household peripherals.

3.1.4 Standards: None

3.1.5 Priority: 1 – Critical

3.2 Compatible Light/Fan Switches

3.2.1 Description: Light and fan switches will need to be integrated into the system either by wire or wirelessly to receive commands from the server.

3.2.2 Source: GLaDOS

3.2.3 Constraints: Must be compatible with server, and sufficient open-source API software may be difficult to find.

3.2.4 Standards: None

3.2.5 Priority: 1 – Critical

3.3 Electronic Door Lock

3.3.1 Description: An electronic door lock will need to be integrated into the system either by wire or wirelessly to receive commands from the server.

3.3.2 Source: GLaDOS

3.3.3 Constraints: Must be compatible with server, and sufficient open-source API software may be difficult to find.

3.3.4 Standards: None

3.3.5 Priority: 1 – Critical

3.4 Digital Thermostat

3.4.1 Description: A digital thermostat will need to be integrated into the system either by wire or wirelessly to receive commands from the server.

3.4.2 Source: GLaDOS

3.4.3 Constraints: Must be compatible with server, and sufficient open-source API software may be difficult to find.

3.4.4 Standards: None

3.4.5 Priority: 1 – Critical

3.5 **RFID Reader/Writer**

3.5.1 Description: An RFID reader/writer will be used to interface RFID bands with the server.

3.5.2 Source: GLaDOS

3.5.3 Constraints: Must be able to secure RFID tags to prevent unauthorized use.

3.5.4 Standards: None

3.5.5 Priority: 1 – Critical

SRS Version 1.0

GLaDOS

3.6 Programmable RFID Bands / Tags

3.6.1 Description: RFID bands or tags will be associated to individual users for identification and must be programmable.

3.6.2 Source: GLaDOS

3.6.3 Constraints: Must have security features to prevent unauthorized reading/spoofing.

3.6.4 Standards: None

3.6.5 Priority: 1 – Critical

3.7 AndroidTM Application

3.7.1 Description: The AndroidTM application will be used to interface with Presence. The application will give the user the ability to create an account that will be associated with their RFID tag and manually set the settings for subsystems associated with Presence such as lights, fans, door locks, and thermostats.

3.7.2 Source: GLaDOS

3.7.3 Constraints: None

3.7.4 Standards: Android[™] APK

3.7.5 Priority: 1 – Critical

3.8 Database Management System (DBMS)

3.8.1 Description: There will be a backend database management system set up on the server to store the information and preferences of the users. The server will create queries based on the RFID tag information when an RFID tag has been scanned.

3.8.2 Source: GLaDOS

3.8.3 Constraints: None

3.8.4 Standards: SQL

3.8.5 Priority: 1 – Critical

3.9 User Account Administration

3.9.1 Description: The AndroidTM application will allow a user to create, modify, and delete user accounts and associate them with RFID tags. This information will be stored in the DBMS.

3.9.2 Source: GLaDOS

3.9.3 Constraints: None

3.9.4 Standards: Android APK, SQL

3.9.5 Priority: 1 – Critical

3.10 User Settings Profiles

3.10.1 Description: The Android[™] application will allow a user to create, modify, and delete multiple profiles to adjust different settings in the home. These profiles will be stored in the DBMS.

3.10.2 Source: GLaDOS

3.10.3 Constraints: None

3.10.4 Standards: Android APK, SQL

3.10.5 Priority: 2 – High

3.11 User Login

3.11.1 Description: The first screen shown by the AndroidTM application is the 'Login' screen, where the user can enter their username and password. The user can also choose whether or not to have the device remember the login information to skip the 'Login' screen in the future. The user's credentials will be authenticated by the DBMS on the server.

3.11.2 Source: GLaDOS

3.11.3 Constraints: None

3.11.4 Standards: Android[™] APK

3.11.5 Priority: 1 – Critical

3.12 Application Menu Bar

3.12.1 Description: The Android[™] application will have a single menu bar at the top of every page after the user is logged in. The menu bar will contain a 'Sign Out' button for the user as well as a 'Refresh' button, which will get an updated status of all devices controlled by the server in the home.

3.12.2 Source: GLaDOS

3.12.3 Constraints: None

3.12.4 Standards: Android[™] APK

3.12.5 Priority: 3 – Moderate

3.13 Switch Lights On/Off

3.13.1 Description: The server can switch lights on/off automatically or from manual configuration via the Android[™] application. The application will display the current switch position of each light switch under Presence control.

3.13.2 Source: GLaDOS

3.13.3 Constraints: Requires connection of light switch via controller mechanism accessible to Presence. Refer to Packaging Requirement 4.1.

3.13.4 Standards: None

3.13.5 Priority: 1 – Critical

3.14 Set Fan Speeds

3.14.1 Description: The server can adjust fan speeds to off/low/medium/high. It can be done automatically or from manual configuration via the Android[™] application. The application will display the current switch position of each fan under PRESENCE control.

3.14.2 Source: GLaDOS

3.14.3 Constraints: Requires connection of fan switch via controller mechanism accessible to Presence. Refer to Packaging Requirement 4.1.

3.14.4 Standards: None

3.14.5 Priority: 1 – Critical

3.15 Lock/Unlock Door

3.15.1 Description: The server can lock/unlock the door automatically or from manual configuration via the Android TM application. The application will display the current lock position of the door under PRESENCE control.

3.15.2 Source: GLaDOS

3.15.3 Constraints: Requires connection of electronic door lock via controller mechanism accessible to Presence. Refer to Packaging Requirement 4.1

3.15.4 Standards: None

3.15.5 Priority: 1 – Critical

3.16 Adjust Thermostat

3.16.1 Description: The server can adjust the temperature on the digital thermostat automatically or from manual configuration via the AndroidTM application. The application will display the current temperature setting and let you increase/decrease the temperature number for the thermostat.

3.16.2 Source: GLaDOS

3.16.3 Constraints: Requires connection of thermostat via controller mechanism accessible to Presence. Refer to Packaging Requirement 4.1.

3.16.4 Standards: None

3.16.5 Priority: 1 – Critical

3.17 User Notifications

3.17.1 Description: When the user isn't home and no authorized RFID tags are sensed in the vicinity, Presence immediately notifies the master user via application alert on the master user's mobile device if any devices under server control are manually changed.

3.17.2 Source: Customer

3.17.3 Constraints: None

3.17.4 Standards: None

3.17.5 Priority: 2 – High

4. Packaging Requirements

This section specifies all of the packaging requirements that must be met for Presence. An included AC adapter will provide power for the server. A user manual will be provided to assist the user with the setup process as well as common tasks, best practices, and troubleshooting procedures. Presence will be easy to configure using the associated AndroidTM application.

4.1 Included Components

4.1.1 Description: The packaging will include the following: (1) server with a pre-loaded operating system, (1) electronic door lock, (2) Wi-Fi-enabled light switches, (1) digital thermostat, (2) Wi-Fi-enabled fan switches, (1) RFID reader/writer, (1) master RFID tag and (1) additional RFID tag, (1) user manual specified in Packaging Requirement 4.6, and appropriate power supply accessories specified in Packaging Requirement 4.5.

- 4.1.2 Source: GLaDOS / Customer
- 4.1.3 Constraints: None
- 4.1.4 Standards: None
- 4.1.5 Priority: 1 Critical

4.2 Module Weight

- **4.2.1 Description:** The weight of any one module will not exceed 10 pounds.
- 4.2.2 Source: GLaDOS
- 4.2.3 Constraints: None
- 4.2.4 Standards: None
- 4.2.5 Priority: 3 Moderate

4.3 Module Sizes

- **4.3.1 Description:** The size of any one module will not exceed 8" x 6" x 2".
- 4.3.2 Source: GLaDOS
- 4.3.3 Constraints: None
- 4.3.4 Standards: None
- 4.3.5 Priority: 3 Moderate

4.4 AndroidTM Application Publication

4.4.1 Description: The AndroidTM application will be offered for free download from the Google PlayTM Store.

4.4.2 Source: GLaDOS

4.4.3 Constraints: Android[™] mobile device must allow third-party applications to be installed

4.4.4 Standards: Android[™] APK file format and Google Play[™] publishing standards

4.4.5 Priority: 1 – Critical

4.5 **Power Supplies**

4.5.1 Description: There will be an AC adapter for the server and appropriate batteries for any non-wired devices such as the door lock and the thermostat.

4.5.2 Source: GLaDOS

4.5.3 Constraints: None

4.5.4 Standards: US National Electrical Manufacturers Association (NEMA) 5-15R

4.5.5 Priority: 1 – Critical

4.6 User Manual

4.6.1 Description: A user manual will be provided for the initial setup of Presence. The user manual will also guide user through use case scenarios to familiarize the user with the system. It will include best practices for installation as well as troubleshooting procedures.

4.6.2 Source: GLaDOS

4.6.3 Constraints: None

4.6.4 Standards: English (US) language only

4.6.5 Priority: 3 – Moderate

5. Performance Requirements

This section specifies the requirements that should be met in order to assure that Presence functions properly. The Android[™] application is a key component and should interface with the server with as little latency as possible. The server and sensors must also gather and send data through the Wi-Fi connection quickly.

5.1 **RFID Recognition Distance**

5.1.1 Description: A user's RFID tag should be detectable within 4 inches

5.1.2 Source: GLaDOS

5.1.3 Constraints: The quality, frequency, and passivity of the RFID tag will determine how far away it can be to be recognized by an RFID reader.

5.1.4 Standards: None

5.1.5 Priority: 3 - Moderate

5.2 Settings Activation Latency

5.2.1 Description: Upon receiving a settings change command, the system should take no more than 2 seconds to activate the user's profile and/or settings.

5.2.2 Source: GLaDOS

5.2.3 Constraints: None

5.2.4 Standards: None

5.2.5 Priority: 2 – High

5.3 Database Query Response Time

5.3.1 Description: Since the database management system will be on the server itself, database lookups should take less than 1 second.

5.3.2 Source: GLaDOS

- 5.3.3 Constraints: Onboard processing power and memory on server
- 5.3.4 Standards: None
- 5.3.5 Priority: 3 Moderate

5.4 Server Response Time Over WLAN

5.4.1 Description: The Android[™] application will be used to control Presence. The server should take no longer than 5 seconds to respond to a command given by the application over the Wireless Local Area Network.

5.4.2 Source: GLaDOS

5.4.3 Constraints: The Wi-Fi signal strength and specific standard used

5.4.4 Standards: IEEE 802.11a/b/g/n/ac

5.4.5 Priority: 2 - High

5.5 Server Response Time Over 3G/4G Networks

5.5.1 Description: The AndroidTM application will be used to control Presence. The server should take no longer than 10 seconds to respond to a command given by the application over 3G/4G cellular networks.

5.5.2 Source: GLaDOS

5.5.3 Constraints: Bandwidth of Internet connection and network congestion

5.5.4 Standards: 3G/4G Wireless Standards

5.5.5 Priority: 3 – Moderate

SRS Version 1.0

5.6 AndroidTM Application Startup Delay

5.6.1 Description: The time it takes to launch the Android[™] application should be less than 10 seconds once it is opened.

5.6.2 Source: GLaDOS

5.6.3 Constraints: Android[™] mobile device hardware and resource utilization

5.6.4 Standards: None

5.6.5 Priority: 3 – Moderate

6. Safety Requirements

This section specifies the various safety considerations associated with Presence. All requirements deemed necessary to prevent the product from causing serious physical harm to users will be covered.

6.1 Electrical Circuit Grounding

6.1.1 Description: All wires, integrated circuit chips, sensors, and other electrical components must be properly grounded to avoid the possibility of electrical shock to the user from a leaking current. The hardware will be fully enclosed to avoid accidental electrocution.

6.1.2 Source: GLaDOS

6.1.3 Constraints: None

- 6.1.4 Standards: IEEE Standard 142: Foundation for Grounding
- 6.1.5 Priority: 1 Critical

6.2 Packaging Safety

6.2.1 Description: The server shall not be composed of packaging that includes unusually sharp edges or materials that can induce harm to an individual.

- 6.2.2 Source: GLaDOS
- 6.2.3 Constraints: None

6.2.4 Standards: None

6.2.5 Priority: 2 – High

6.3 Operating Temperature

6.3.1 Description: The server shall not expel an excessive amount of heat during its operation. The internal hardware should dissipate enough heat and allow sufficient airflow so that the server operates at a reasonable temperature that will not burn the user or the surrounding environment.

6.3.2 Source: GLaDOS

6.3.3 Constraints: None

6.3.4 Standards: None

6.3.5 Priority: 2 – High

7. Maintenance and Support Requirements

This section specifies the requirements involved with both maintaining and providing continuous support for Presence after delivery. It will specify the scope of maintenance and the methods by which such maintenance will be provided for Presence. It will include documentation that will be used as well as how the development team can be contacted if any problems or questions arise from the use of Presence.

7.1 Maintenance Scope

7.1.1 Description: Maintenance of Presence will include, but is not limited to, support for both hardware and software failure of Presence and update support for the AndroidTM application.

7.1.2 Source: GLaDOS

7.1.3 Constraints: None

7.1.4 Standards: None

7.1.5 Priority: 4 – Low

7.2 AndroidTM Version Support

7.2.1 Description: The supplemental AndroidTM application will support a minimum API Level of 16, which corresponds to AndroidTM Operating System version 4.1.2 (Jelly BeanTM).

7.2.2 Source: GLaDOS

- 7.2.3 Constraints: None
- 7.2.4 Standards: None
- 7.2.5 Priority: 2 High

7.3 Source Code Commenting

7.3.1 Description: The source code for the AndroidTM application affiliated with Presence will be extensively commented to allow for better readability.

7.3.2 Source: GLaDOS

7.3.3 Constraints: None

7.3.4 Standards: None

7.3.5 Priority: 3 – Moderate

7.4 Source Code / Schematics / Documentation Availability

7.4.1 Description: All source code developed by GLaDOS will be open source and will be made available in the project archive to future development teams. All server and circuit schematics associated with the design of Presence as well as all documentation – including System Requirements Specification, Architectural Design Specification, Detailed Design Specification, and System Testing Plan – will also be made available for public use.

7.4.2 Source: GLaDOS

7.4.3 Constraints: None

7.4.4 Standards: None

7.4.5 Priority: 3 – Moderate

7.5 User Manual

7.5.1 Description: The user manual will contain specifics on the various functionalities and features and will include screenshots. The user manual will also have a troubleshooting section dedicated to any issues most commonly encountered in the testing phase.

7.5.2 Source: GLaDOS

7.5.3 Constraints: None

7.5.4 Standards: English (US) language only

7.5.5 Priority: 2 – High

7.6 Installation Support

7.6.1 Description: Professional installation of Presence will be recommended and provided by GLaDOS and/or approved third-party associates.

7.6.2 Source: GLaDOS

7.6.3 Constraints: None

7.6.4 Standards: None

7.6.5 Priority: 2 - High

7.7 Language Support

7.7.1 Description: English will be the primary language for all user interfaces and documentation. Additional languages will not be supported at this time.

- 7.7.2 Source: GLaDOS
- 7.7.3 Constraints: None
- 7.7.4 Standards: None
- 7.7.5 Priority: 3 Moderate

7.8 Multiple User Support

7.8.1 Description: When multiple users are in the home, the master user's settings will be given priority. In the event that multiple users are present and the master user is not, a default household profile will be activated.

7.8.2 Source: GLaDOS

7.8.3 Constraints: None

7.8.4 Standards: None

7.8.5 Priority: 3 - Moderate

8. Other Requirements

This section specifies any requirements necessary for Presence that did not directly fall under any of the previous sections of this document.

8.1 Connectivity Tolerance

8.1.1 Description: Presence should be tolerant of internet connectivity and be able to perform localized in-home functions if no internet connection is detected.

8.1.2 Source: GLaDOS

8.1.3 Constraints: None

8.1.4 Standards: None

8.1.5 Priority: 2 – High

8.2 Application Security and Privacy

8.2.1 Description: When communicating via the Internet, Presence and/or the AndroidTM application should ensure the security and privacy of internally stored or transmitted user information.

8.2.2 Source: GLaDOS / Customer

8.2.3 Constraints: None

8.2.4 Standards: OWASP Web Application Security Standards

8.2.5 Priority: 2 – High

38

8.3 Intrusion Prevention

8.3.1 Description: The server will be guarded by a firewall and will require authentication to prevent unauthorized tampering. RFID tags will be uniquely keyed and password-protected to prevent unauthorized reading by foreign sensors allowing a third-party to spoof the RFID and gain unauthorized access into the home.

8.3.2 Source: GLaDOS / Customer

8.3.3 Constraints: None

8.3.4 Standards: None

8.3.5 Priority: 2 – High

8.4 Manual Override

8.4.1 Description: In the event of loss of power to the system or other system failure, the user should not be locked out of the home. A physical key will still unlock the door and other utilities may be manually turned on and off by the user in any case.

8.4.2 Source: GLaDOS / Customer

8.4.3 Constraints: None

8.4.4 Standards: None

8.4.5 Priority: 1 – Critical

9. Acceptance Criteria

This section specifies the requirements and procedures used to demonstrate the product and its individual functions to the customer. All of the criteria in this section must be met in order for Presence to be accepted. The instructions are very high level due to the product concept still being in its infancy. As the project moves forward, these criteria will be revised and updated as necessary.

9.1 Verify User Account Creation

Requirement Number	Name
3.8	Database Management System
3.9	User Account Administration
3.10	User Settings Profiles
3.11	User Login

9.1.1 Requirement(s) addressed:

9.1.2 Verification Procedure: The home page Android[™] application will have 'Login' and 'Create User' buttons that will take any user to the appropriate locations. We will test this requirement by tapping the 'Create User' button and making multiple accounts with linked RFID tags. We will then test the tags on the HMS to verify that our custom settings were applied. Each account will have a different setting and all of them together will cover every module's unique feature. This will allow us to quickly and accurately verify whether or not one of the accounts created did not get saved in our database.

9.2 Verify Light Control

9.2.1 Requirement(s) addressed:

Requirement Number	Name
3.2	Compatible Light Switches
3.13	Switch Lights On/Off

9.2.2 Verification Procedure: The AndroidTM application will be opened to the light control screen once a valid user account is signed in. We will click the button associated with the different lights and look to see if the light turns off and on. We will also verify that the current state of the light(s) is being updated on the application.

9.3 Verify Fan Control

9.3.1 Requirement(s) addressed:

Requirement Number	Name
3.2	Compatible Fan Switches
3.14	Set Fan Speeds

9.3.2 Verification Procedure: The AndroidTM application will be opened to the fan control screen once a valid user account is signed in. We will tap the button designated to adjust the speed of the fan and verify that the fan is spinning accordingly. We will test all four speeds: off, low, medium, and high.

9.4 Verify Door Lock Control

9.4.1 Requirement(s) addressed:

Requirement Number	Name
3.3	Electronic Door Lock
3.15	Lock/Unlock Door

9.4.2 Verification Procedure: The Android[™] application will be opened to the door control screen once a valid user account is signed in. We will try unlocking and locking the door through the application and verify that it did indeed lock or unlock. We will also manually lock and unlock the door lock to verify that the screen shows some indication of the lock's current state.

9.5 Verify Thermostat Control

9.5.1 Requirement(s) addressed:

Requirement Number	Name
3.4	Digital Thermostat
3.16	Adjust Thermostat

9.5.2 Verification Procedure: The AndroidTM application will be opened to the thermostat control screen once a valid user account is signed in. We will increase and decrease the temperature settings and verify that the thermostat shows the new setting. We will also manually adjust the temperature from the thermostat and verify that it gets updated in the application.

10. Use Cases

This section specifies how the intended end-user will interact with Presence at home and on their AndroidTM mobile device. Each use case listed below will identify the scenario, the actor, when the use case begins (TUCBW – The Use Case Begins With), and when the use case ends (TUCEW – The Use Case Ends With).

10.1 Switch Lights On/Off – Sensor-Based

10.1.1 Scenario: The user scans their RFID tag with the RFID reader, and that information is transmitted to the server which switches on the lights in that room. As the user scans their RFID tag upon leaving the home, that information is transmitted to the server which switches off the lights.

10.1.2 Actor(s): User

10.1.3 TUCBW: The user scans their RFID tag with RFID reader.

10.1.4 TUCEW: Presence automates switching the lights off/on.

10.2 Switch Lights On/Off – Application-Based

10.2.1 Scenario: The user opens Presence AndroidTM application on their mobile device and taps the 'Lights' button. The application displays the toggle buttons associated with each light controlled by Presence and their current switch position. At this point, the user can tap any of the toggle buttons to switch the lights off/on.

10.2.2 Actor(s): User

10.2.3 TUCBW: The user opens the Android[™] application.

10.2.4 TUCEW: The user can see the switch position of the lights and adjust them manually.

10.3 Lock/Unlock Door – Sensor-Based

10.3.1 Scenario: The user scans their RFID tag with the RFID reader, and that information is transmitted to the server which unlocks the door. As the user scans their RFID tag upon leaving the home, that information is transmitted to the server which locks the door.

10.3.2 Actor(s): User

10.3.3 TUCBW: The user scans their RFID tag with RFID reader.

10.3.4 TUCEW: Presence automates locking/unlocking the door.

10.4 Lock/Unlock Door – Application-Based

10.4.1 Scenario: The user opens Presence Android[™] application on their mobile device and taps the 'Door Locks' button. The application displays the toggle button associated with each door lock controlled by Presence and their current lock state. At this point, the user can tap any of the toggle buttons to lock/unlock doors.

10.4.2 Actor(s): User

10.4.3 TUCBW: The user opens the Android[™] application.

10.4.4 TUCEW: The user can see the lock state of the doors and adjust them manually.

10.5 Change Temperature – Sensor-Based

10.5.1 Scenario: The user scans their RFID tag with the RFID reader, and that information is transmitted to the server which adjusts the thermostat. As the user scans their RFID tag upon leaving the home, that information is transmitted to the server which adjusts the thermostat to conserve energy.

10.5.2 Actor(s): User

10.5.3 TUCBW: The user scans their RFID tag with RFID reader.

10.5.4 TUCEW: Presence automates thermostat adjustment.

10.6 Change Temperature – Application-Based

10.6.1 Scenario: The user opens Presence AndroidTM application on their mobile device and taps the 'Temperature' button. The application displays the current temperature setting of the thermostat as well as '+' and '-' buttons to adjust the temperature. At this point, the user can set the desired temperature and then tap 'Submit' to confirm the settings.

10.6.2 Actor(s): User

10.6.3 TUCBW: The user opens the Android[™] application.

10.6.4 TUCEW: The user can see the thermostat change to accommodate the new settings.

10.7 Change Fan Speed – Sensor-Based

10.7.1 Scenario: The user scans their RFID tag with the RFID reader, and that information is transmitted to the server which adjusts the fan speed. As the user scans their RFID tag upon leaving the home, that information is transmitted to the server which readjusts the fan speed.

10.7.2 Actor(s): User

10.7.3 TUCBW: The user scans their RFID tag with RFID reader.

10.7.4 TUCEW: Presence automates adjusting fan speed.

10.8 Change Fan Speed – Application-Based

10.8.1 Scenario: The user opens Presence Android[™] application on their mobile device and clicks the 'Fans' button. The application displays the toggle buttons associated with each fan controlled by Presence and their current switch position. At this point, the user can tap any of the toggle buttons to set the fan speed to off/low/medium/high.

10.8.2 Actor(s): User

10.8.3 TUCBW: The user opens the Android[™] application.

10.8.4 TUCEW: The user can see the switch position of the fans and adjust them manually.

10.9 Create User Account – Application-Based

10.9.1 Scenario: The user opens Presence Android[™] application on their mobile device and taps the 'Create Account' button. The user will be taken to a screen where they will fill out basic personal information and associate an RFID tag, and tap 'Submit' once completed. The next screen will require the administrator's RFID tag to be read in order to confirm the new user. After validation, the user will be returned to the 'Home' screen and the new user will be able to log in and adjust settings.

10.9.2 Actor(s): User

10.9.3 TUCBW: The user opens the Android[™] application.

10.9.4 TUCEW: The new user account is created and the user can log in with their credentials and adjust settings.

10.10 User Sign In – Application-Based

10.10.1 Scenario: The user opens Presence Android[™] application to the login screen and enters their username and password. The user may also select, if they wish, to have their login information remembered on that device allowing them to bypass the login screen in the future.

10.10.2 Actor(s): User

10.10.3 TUCBW: The user opens the Android[™] application.

10.10.4 TUCEW: The user logs in with their credentials and is taken to the 'Home' screen.

10.11 Create Settings Profile – Application-Based

10.11.1 Scenario: The user opens Presence AndroidTM application on their mobile device and taps the 'Profiles' button. The user will be taken to a screen with two buttons: 'View Profiles' and 'Create Profile'. The user taps 'Create Profile' which takes them to a screen where they will name the profile, choose settings for each device in the home, and tap 'Submit' once completed. A pop-up will then ask the user if they wish to make this new profile their current profile. After tapping 'Yes' or 'No' on the pop-up, the user will be returned to the 'Home' screen.

10.11.2 Actor(s): User

10.11.3 TUCBW: The user opens the Android[™] application.

10.11.4 TUCEW: The user can see the new profile in the 'View Profiles' screen. The current profile will be indicated by bold text.

10.12 View/Change Settings Profiles – Application-Based

10.12.1 Scenario: The user opens Presence AndroidTM application on their mobile devices and taps the 'Profiles' button. The user will be taken to a screen with two buttons: 'View Profiles' and 'Create Profile'. The user taps 'View Profiles' where a named list of profiles that the user created appears. The current profile will be indicated by bold text. At this point the user can tap any profile and a pop-up will appear with the following selectable options: 'Set to Current Profile', 'Edit Settings', and 'Delete Profile'.

10.12.2 Actor(s): User

10.12.3 TUCBW: The user opens the Android[™] application.

10.12.4 TUCEW: The user can see their list of settings profiles and the current settings will match the current profile.



Figure 10-1 Light control use case diagram



Figure 10-2 Door control use case diagram



Figure 10-3 Temperature control use case diagram



Figure 10-4 Fan control use case diagram



Figure 10-5 User control use case diagram

11. Feasibility Assessment

This section will analyze the feasibility of Presence and associated Android[™] application. It will analyze the scope, research, technical details, cost, available resources, and schedule estimation for the project. These analyses are based on the team's knowledge and previous experiences with similar projects.

11.1 Scope Analysis

The scope of work required for the critical requirements in this project is reasonable given the project deadline. The most complex critical element will be interfacing a server with multiple different household peripherals. The customizable settings for users include lights on/off, fan speeds, thermostat adjustments, and doors locked/unlocked. Interfacing each one of those may require a separate API, and open-source code may be scarce. Given those constraints, the team has done extensive research and has a thorough understanding of the technologies and the risks involved. In conclusion, producing a working prototype of all the critical requirements looks feasible. This estimation is based on each of the team members' understanding of the complexity of this project. Requirements will be prioritized so that critical- and high-priority requirements are completed first.

11.2 Research

The team has completed extensive hardware research in order to refine the requirements to determine feasibility. Devices researched include the Raspberry Pi microcontroller, 5V relay switches, Wi-Fi-enabled light and fan switches, digital thermostats, and electronic door locks, infrared transmitters and receivers, RFID readers/writers, RFID tags/bands, and AndroidTM platforms.

The interface between the server and the peripheral controllers for each of the other subsystems has also been researched. The team has found that there is not a wide selection of open-source API's to connect the server to the controllers such as a light switch controller. However, open-source controller API's are not non-existent either.

The customer is also very knowledgeable with years of teaching experience and has been a valuable asset to the team regarding information related to the project. The customer will continue work with the team to solidify key requirements and help devise methods of implementation.

11.3 Technical Analysis

The team is comfortable with a specific server to be used as the central control for Presence but would like to do more research before making any final commitments. However, this choice is subject to change in the event that refined requirements no longer make it feasible. Presence will incorporate the server with other subsystems that will be controlled by microcontrollers. These subsystems include light control, temperature control, door control, and fan control.

The light controls will be responsible for turning on/off the lights. Preliminary design implies that this subsystem will comprise of a microcontroller and possibly relay switches. The relays would be used to open and close the light's circuit hence turning it on and off. There are also Wi-Fi-enabled light switches which can communicate with the server via a Wireless Local Area Network.

The temperature control will be responsible for controlling the room temperature based on the user's desired settings. A digital thermostat will be a key component. Wi-Fi will be the most likely form of communication between the thermostat and the server.

The door control will lock and unlock the door. A key component of this subsystem will be an electronic door lock that will be connected to a microcontroller and controlled remotely. There are Wi-Fi-enabled door locks, which may be used to interface with the server.

Lastly, the fan control will be responsible for switching between fan speeds (off/low/medium/high). It will comprise of a relay switch and a microcontroller.

11.4 Cost Analysis

The team's research and estimation of how much Presence will cost to build indicates that it will be within the \$800 allotted budget. All required items needed to develop the Android[™] application such as the Integrated Development Environments, Google Drive repository, and bug finders are free except for the publication of the application on Google Play[™] which comes to \$25. All of the hardware devices researched to meet the requirements amount approximately \$520, and no individual hardware component exceeds \$100, so there is room to expand or upgrade if necessary. Table 11-1 gives a more specific breakdown of the cost analysis.

Part	Cost
Server & Components	\$100.00
Wireless Thermostat	\$100.00
Wireless Door Lock	\$85.00
Wireless Fans (2)	\$90.00
Wireless Light Switch (2)	\$70.00
RFID reader/writer	\$50.00
RFID bands (2)	\$25.00
Google Play [™] subscription	\$25.00
Total	\$545.00

 Table 11-1 Cost analysis of hardware required

11.5 Resource Analysis

The team is made up of a Computer Engineer, a Software Engineer, and three Computer Scientists. Four out of five team members have previously worked or are currently working in different fields the computer industry. As a result, the team has a broad range of knowledge and experience valuable to making Presence a feasible product.

Permanent roles have been assigned to team members based on their strengths and experience. The team's Computer Engineer will be responsible for all hardware decisions and requirements. This will include circuit design, microcontroller specification, and Assembly programming. One of the team's three Computer Scientists has significant professional software development experience for AndroidTM and will be the lead software architect. The team's Software Engineer will be in charge of software and hardware testing as well as risk management. Another Computer Scientist will be the project manager by taking charge of all documentation and customer relations. That Computer Scientist will also be the security analyst. The final team members will be in charge of finances and budgeting. All team members have excellent programming skills, and two of the five members have extensive experience dealing with hardware.

The team's only weaknesses will be encountered during performance tuning, mechanical design, and any potential multiprocessing or distributed systems requirements.

11.6 Schedule Analysis

The team used three methods to estimate the length of the project. The initial method the team decided to use was Jones' First Order Estimation. This method generates function points based on each individual module of the project.

Function-Point Analysis				
Function Type	Low Complexity	Medium Complexity	High Complexity	Totals
Number of Inputs	5x3	2x4	1x6	29
Number of Outputs	4x4	1x5	1x7	28
Inquiries	4x3	0x4	0x6	12
Logical Internal Files	5x7	1x10	0x15	45
External Interface Files	4x5	0x7	0x10	20
Unadjusted Function-Point Total 134				

 Table 11-2 Function-point analysis

Table 11-2 shows the various function types and the analysis of our project with respect to each type. The unadjusted function-point total came out to 134. The logical internal files make up a majority of the function-points. The number of inputs and number of outputs also make up a very significant portion of the function-points due to application overhead and control of multiple peripherals.

Influence Multipliers	Effort Value (0-5)
Data Communications	4
Distributed Data Processing	0
Performance	4
Heavily-Used Configuration	5
Transaction Rate	5
Online Data Entry	1
End-User Efficiency	5
Online Update	1
Complex Processing	0
Reusability	2
Installation Ease	2
Operational Ease	1
Multiple Sites	0
Facilitate Change	3
Total	33
Adjustment Factor	0.98

Table 11-3 Adjustment factors

To find the adjustment factor, the team assigned values to each of the 14 influence multipliers. Table 11-3 shows each multiplier and the value the team assigned it. The total of all multipliers was used in the following formula to compute the adjustment factor:

Adjustment Factor (AF) = 0.65(Total * 0.01)AF = 0.65(33 * 0.01) = 0.98

As shown above the adjustment factor came out to 0.98, which means the team's adjusted function-point total is slightly less than the unadjusted function-point total.

Adjusted Function-Point Total = 0.98 * 134 = 131.32

134
0.98
131.32

Table 11-4 Unadjusted and adjusted function-point totals

Given the adjusted function-point total, the team performed Jones' First Order Estimation. The project falls under the "Systems" type, therefore, the estimation values are associated with that type of software.

Best Case	Average Case	Worst Case
	0	worst Case
$131.32^{0.43}$	$131.32^{0.45}$	$131.32^{0.48}$
8.14	8.98	10.39
	8.14	8.14 8.98

Table 11-5 Jones' First Order Estimation

Table 11-5 shows that the best-case scenario for reaching project completion is 8.14 months. The worst-case scenario could extend the product to 10.39 months.

At this point, the team decided to use the semi-detached Constructive Cost Model (CoCoMo) as a second method of estimation. It measures effort (E) in person-months and duration (D) in months. The team used an aggressive estimate of 4,000 SLOC and a conservative estimate of 8,000 SLOC.

CoCoMo Equations: $E = a(KSLOC^b)$ $D = c(E^d)$

In the semi-detached CoCoMo, a = 3.0, b = 1.12, c = 2.5, and d = 0.35. The aggressive and conservative estimated calculations are as follows:

Aggressive Estimate (CoCoMo Calculation):

 $E = 3.0(4^{1.12}) = 14.17$ person months $D = 2.5(E^{0.35}) = 6.32$ months

Conservative Estimate (CoCoMo Calculation):
$E = 3.0(8^{1.12}) = 30.8 \ person\ months$
$D = 2.5(E^{0.35}) = 8.3 months$

_. _ _

Aggiessive	Conservative
14.17	30.8
6.32	8.3
2.24	3.71
	Aggressive 14.17 6.32 2.24

 Table 11-6 Semi-detached CoCoMo estimation

Table 11-6 summarizes the findings of the CoCoMo. Based on this model, the best-case and worst-case scenarios for project completion are 6.32 and 8.30 months respectively. By pairing those numbers with the numbers from Jones' First Order Estimation, the team has concluded that the project will take between 6.32 and 10.39 months with a likely completion time of 8.3 months. Given the project time constraint of six months and the fact that the team added some buffer time in the tables above in the event of unforeseen risks and worst-case situations.

To remove any lingering doubt, the team decided to run the Sanity Test (Weiss & Wysocki) which is as follows:

$$E = \frac{O + 4M + P}{6}$$

With an optimistic duration of 6.32 months, a nominal duration of 8.3 months, and a pessimistic duration of 10.39 months, the team calculates the following estimate:

$$E = \frac{6.32 + 4(8.3) + 10.39}{6} = 8.32 \text{ months}$$

The Sanity Test estimation indicated that the project would take 8.32 months to complete which falls within the other two estimation models. Based on the overall feasibility assessment, the team concludes that completing this project within the given timeframe is highly probable.

12. Future Items

This section specifies additional feature requirements for Presence that may be implemented as time and schedule allow, but will likely be pushed into future development in the interests of meeting the project deadline and releasing a stable, viable product.

12.1 Customer Requirement 3.18: Home Stereo Interface

12.1.1 Requirement Description: The system will incorporate customizable musical playlists for users, which will play automatically when a user's RFID tag is detected. It can also be configured to stop/start playing manually from the AndroidTM application.

12.1.2 Constraint: Due to having a feature-rich requirement set early on in the planning phase, the team realizes the importance of delivering a timely, high-quality product. As a result the team has decided to put this feature off until future development.

12.2 Customer Requirement 3.19: TV Interface

12.2.1 Requirement Description: The system will incorporate integration with a television via infrared transmitters and receivers. It can be configured to be turned on automatically when a user's RFID tag is detected, and it can be configured manually from the AndroidTM application.

12.2.2 Constraint: Due to having a feature-rich requirement set early on in the planning phase, the team realizes the importance of delivering a timely, high-quality product. As a result the team has decided to put this feature off until future development.

12.3 Customer Requirement 3.20: Security Camera Interface

12.3.1 Requirement Description: The system will incorporate integration with home security cameras. The user may view live security camera feeds in the home remotely and may switch them on/off at will.

12.3.2 Constraint: Due to having a feature-rich requirement set early on in the planning phase, the team realizes the importance of delivering a timely, high-quality product. As a result the team has decided to put this feature off until future development.