

Software Requirements Specification

for

Smart Security

Version 3.0

Prepared By:

Tiffany Wong, Helen Hua, Josie Spencer, Aruna Srinivasiah, Connie Diu, Dericka Logan

Advisor:

Yuan An

Stakeholders:

Tiffany Wong, Helen Hua, Josie Spencer, Aruna Srinivasiah, Connie Diu, Dericka Logan

Table of Contents

Table of Contents	2
1. Document History	4
2. Introduction	5
2.1 Purpose	5
2.2 Overview	5
2.3 Scope	5
2.4 Definitions	5
3. Overall Description	6
3.1 Product Perspective	6
3.2 Product Functions	6-7
3.3 User Characteristics	7-8
3.4 Constraints	8
3.5 Assumptions and Dependencies	8-9
3.6 Requirements Apportioning	9
4. Specific Requirements	9
4.1 External Interfaces	9
4.1.1 User Interfaces	9-16
4.1.2 Hardware Interfaces	16
4.1.3 Software Interfaces	16
4.1.4 Communication Interfaces	17-19
4.2 Functions	19
4.2.1 Creating User Accounts	19-20
4.2.2 Logging into Smart Security	20-21
4.2.3 Creating Arm/Disarm Alarm Phrases	21
4.2.4 User Arming Alarm	21

4.2.5 User Disarming Alarm	21
4.2.6 Tripped Alarms	21-22
4.2.7 Creating Contacts/Entrants List	22
4.2.8 Report of Tripped Alarms	22
4.3 Logical Database Requirements	22
4.3.1 Use and Retention	22
4.3.2 Types of Data	22
4.3.2.1 Account	22-23
4.3.2.2 Alarm	23
4.4 Design Constraints	23
4.4.1 Standards Compliance	23
4.4.1.1 Data Naming	23
4.4.1.2 Logging	23-24
4.5 Software System Attributes	24
4.5.1 Reliability	24
4.5.1.1 System and Software Design	24
4.5.1.2 Error Handling	24
4.5.2 Availability	24
4.5.3 Security	25
4.5.3.1 User Information	25
4.5.3.2 System	25
4.5.4 Maintainability	25
4.5.5 Portability	25
4.5.5.1 Browsers	25-26
4.5.5.2 Operating Systems	26
4.5.6 Performance	26
4.5.7 Scalability	26

1. Document History

Name	Date	Reason	Version
Tiffany Wong, Helen Hua, Josie Spencer, Aruna Srinivasiah, Connie Diu, Dericka Logan	November 4, 2017	Initial Draft	1.0
Tiffany Wong, Helen Hua, Josie Spencer, Aruna Srinivasiah, Connie Diu, Dericka Logan	January 30, 2018	Added previous state machine diagram , added revised Google Home integration state machine diagram, and included current screenshots of web application	2.0
Tiffany Wong, Helen Hua, Josie Spencer, Aruna Srinivasiah, Connie Diu, Dericka Logan	April 3, 2018	<ul style="list-style-type: none">• Corrected typos• Change certain terms• Updated constraints• Updated external interfaces with updated screenshots• Updated state machine diagram• Updated specific requirements	3.0

2. Introduction

2.1 Purpose

This document provides the requirement specifications for the Smart Security system. It specifies user interface attributes, functional and nonfunctional requirements, and long-term ideas for the evolution of the system.

2.2 Overview

The Smart Security system allows users to secure their homes by alerting them when an unauthorized person enters. Users can add approved entrants through the web application. Users can set the alarm with a voice command to the Google Home, and end it the same way.

2.3 Scope

This document describes the software requirements for the Smart Security system. This system is designed to be used in small, relatively quiet homes. The developers will use this document to guide development and ensure that our system works as expected. The system will be tested against these requirements, and this document can be used to describe the system to interested parties or potential stakeholders.

2.4 Definitions

Alarm - the mode of the system during which the user is alerted to unauthorized entrants.

Authorized - an authorized entrant can disarm the alarm for a certain amount of time.

Entrant - a person who enters the home. An entrant may be authorized or unauthorized. The user is a permanently authorized entrant.

Home - the Google Home or Google Home Mini device incorporated in the system.

Residence - the apartment or other living space that the user is protecting using the system.

Unauthorized - an unauthorized entrant will cause the alarm to sound.

User - the primary owner of the device.

3. Overall Description

3.1 Product Perspective

To put the product into perspective, Google Home will be our main source of use for this project, to transform it into a smart security alarm system that will perform various functions when paired with a smartphone. Google Home is a smart home device that allows the user to ask questions and receive answers, play songs, have Google Assistant take care of your day, stream entertainment, all while using your voice to control your smart home. We will be using Google Home to use it for sole purposes of turning it into a security alarm system that is catered towards college students and new graduates in a cheap, multifunctional, and transportable way. Users will use their voice and/or the web application to interact with the alarm system. The specific utilization of our product will be highly attractive to these user groups because of a strong need for an immediate solution to having a secure base in their residence.

3.1.1 System Interfaces

The system interfaces include User Interfaces, Hardware Interfaces, Software Interfaces, and Communication Interfaces. Each interface has a purpose and functionality in the system. Details are mentioned in the External Interface section on pages 9-14 of the document.

3.2 Product Functions

1) Logging into Smart Security

- This function allows for the user to log into the Smart Security web application using their Google account credentials. It is essential to log in with their Google account because their Google account is connected to their Google Home device. The specifics of each component of logging into Smart Security is mentioned in detail 4.2 Functions section, subsection 4.2.2 Logging into Smart Security.

2) Creating Arm/Disarm Alarm Code

- This function allows for the user to create a code for activating and deactivating the alarm through Google Home. The specifics of each component of creating arm/disarm alarm code is mentioned in detail 4.2 Functions section, subsection 4.2.3 Creating Arm/Disarm Alarm Phrases.

3) User Arming Alarm

- This function allows for the user to activate the alarm through Google Home with the arming alarm code as well as the alarm should be activated until the alarm is disarmed. The specifics of each component of the user arming alarm is mentioned in detail 4.2 Functions section, subsection 4.2.4 User Arming Alarm.

4) User Disarming Alarm

- This function allows for the user to deactivate the alarm through Google Home with the disarming alarm code. The alarm should be disarmed until the alarm is armed. The specifics of each component of the user disarming alarm is mentioned in detail 4.2 Functions section, subsection 4.2.5 User Disarming Alarm.

5) Tripped Alarms

- This function allows for the alarm to be tripped when Google Home picks up invasive sounds as well as a text message shall be sent to the user when the alarm is tripped. The specifics of each component of tripped alarms is mentioned in detail 4.2 Functions section, subsection 4.2.6 Tripped Alarms.

6) Creating Contacts/Entrants List

- This function allows for the user to create a contact list of approved entrants via the web application. The specifics of each component of creating contacts/entrants list is mentioned in detail 4.2 Functions section, subsection 4.2.7 Creating Contacts/Entrants List.

7) Report of Tripped Alarms

- This function allows for the user to be able to review a list of tripped alarms on the web application. On the report, the list of tripped alarms will include the time and date of the occurrence. The specifics of each component of the report of tripped alarms is mentioned in detail 4.2 Functions section, subsection 4.2.8 Report of Tripped Alarms.

3.3 User Characteristics

The Smart Security project is meant to offer a solution that is more convenient than manually having to type in a password to alarm and disarm your security system every time you want to give someone access to enter the home. This project will offer a solution to allow users to monitor the security of this homes while they are out. Also, the application will require no (or very little) learning before use.

There are two types of users that interact with the system: users of the web application/Google Home and the entrants. Both types of users have a different use of the system so each of them is going to have their own set of requirements.

The web application/Google Home users can set the alarm with a voice command to the Google Home. The web application/Google Home user must be able to log into the Smart Security system, create

arm/disarm alarm codes, arm/disarm the alarm, view tripped alarms report, and create contact/entrants list. This means that the user is a permanently authorized entrant.

The entrants are the people that enter the home. This type of user does not interact with the web application but only with the Google Home. The entrants can say a code phrase, and based on the code phrase the user will be notified if an authorized entrant or unauthorized entrant is trying to enter their home.

It is also important that the application allow users to add as many approved entrants as possible. Otherwise, it will not be an alternative to handling multiple users. Lastly, the application must be reliable. The application must be able to accurately notify the user via text message when an approved entrant enters their home. Also, the application must be able to accurately notify the user via text message when an unapproved entrant enters their home. There is no room for error when dealing with home security.

3.4 Constraints

The primary design constraint is voice recognition. The Smart Security project involves the Google Home being able to recognize the voices' of the different entrants that enter the home, and being able to analyze the voice to determine the status of the entrant (approved or unapproved). The Google Assistant being able to recognize the difference between different speakers will be a major design consideration.

The Internet connection is another constraint for the web application. The application will retrieve the users' data from the database over the Internet, so it is important that they have an Internet connection for the web application to function.

Google Assistant is essentially the biggest constraint to our project. The Google Home uses Google Assistant to interact with the user through voice. The problem with Google Assistant is that it is only able to recognize language. This will be a problem because it means that Google Home will not be able to pick up and recognize sounds that are not languages.

3.5 Assumptions and Dependencies

The basis of the Smart Security project is detecting unauthorized entrants by reacting to invasive sounds. If the Google Home is not able to do that, another method will be required in order to trigger the alarm. This will depend on testing the device itself which will be acquired at a later stage of development.

The Smart Security project involves using voice recognition to distinguish whether the speaker is in the approved contacts list. This will depend on whether the Google Assistant API has documentation on how to do this. If voice recognition is not possible, alternative ideas include using a code phrase that would arm and disarm the Google Home. Another option is to have authorized entrants state their name upon

entering the residence and cross checking that with the contact list to make sure that they are approved.

Another assumption is that the Google Home can be activated for extended periods of time which will depend on the mechanics of the device. If the Google Home can not be activated indefinitely, an end time will need to be provided by the user.

3.6 Requirements Apportioning

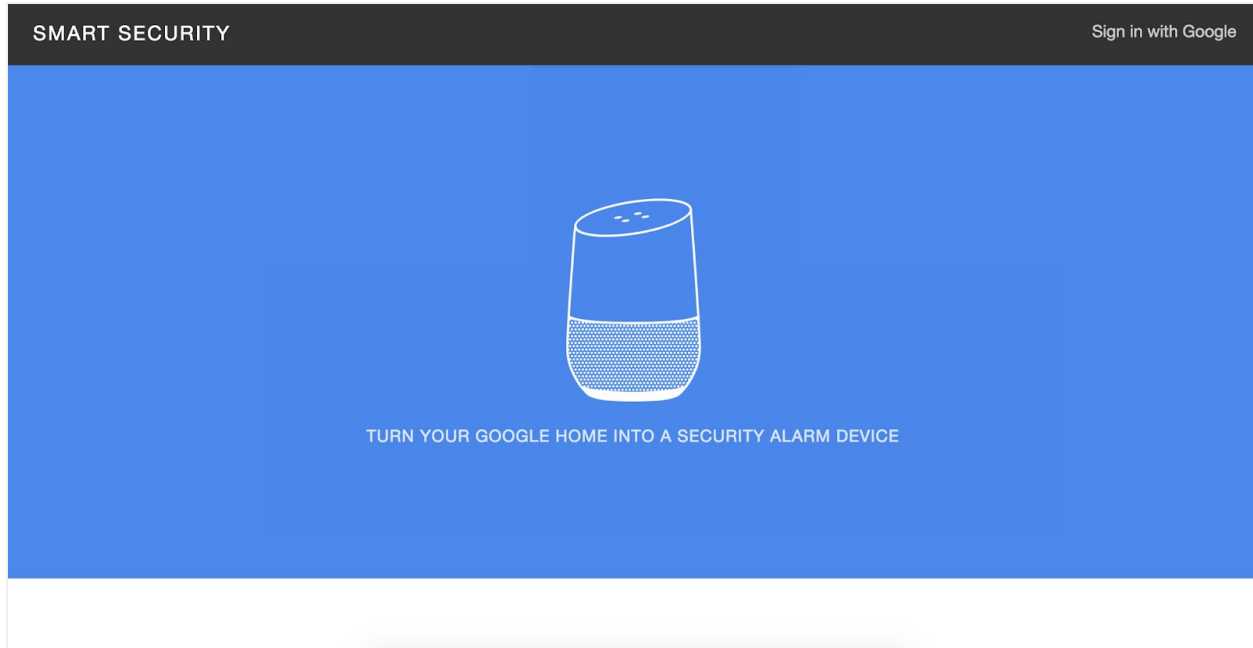
Priority Level	Description
1	Priority 1 items are crucial to the core functionality. The items in this category will be completed and tested prior to the release of the application.
2	Priority 2 items are features that will enhance the user experience but are not crucial to the development of the application. The items in this category may not be finished prior to the release of the application but they will not impact the core functionality.
3	Priority 3 items are features that are not crucial to the core functionality and are not within the scope of the system.

4. Specific Requirements

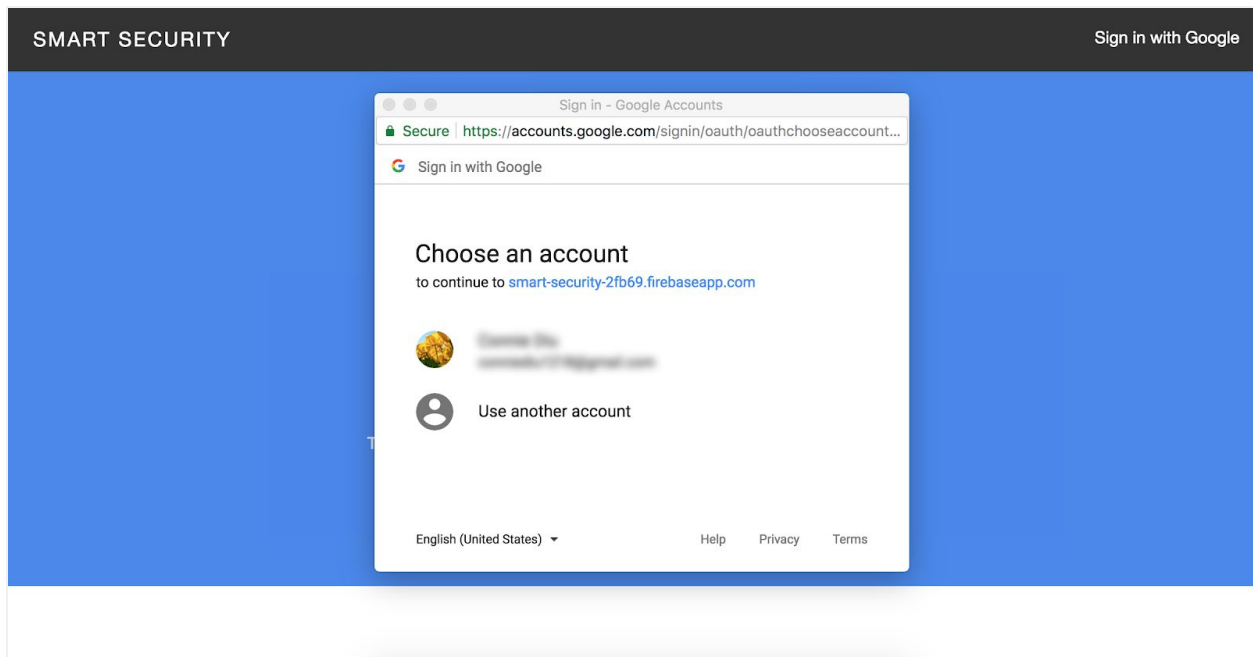
4.1 External Interfaces

4.1.1 User Interfaces

4.1.1.1 Front page - user shall be first brought to the front page of the web application. **Priority 1**




4.1.1.2 Login pop-up screen - user shall be able to use their Google account to log into the web application. **Priority 1**



4.1.1.3 Alarms page - user will be directed to the alarms page once they are logged in. The data will include the date and time of when the alarm was tripped. **Priority 1**

SMART SECURITY

AlarmsEntrants



ALARMS

03/14/2018

Time: 4:51 PM

authorized

03/14/2018

Time: 4:51 PM

authorized


03/14/2018

Time: 7:40 PM



4.1.1.4 Entrants page - user will be able to user will be able to view, add, or delete approved entrants. **Priority 1**

SMART SECURITY

AlarmsEntrants



APPROVED ENTRANTS

Name	Phone	Email	Options
Tiffany Wong	xxx-xxx-xxxx	abc123@drexel.edu	
Helen Hua	xxx-xxx-xxxx	abc123@drexel.edu	

Add Entrant

© 2017 Drexel University. All Rights Reserved.

4.1.2 Hardware Interfaces

4.1.2.1 Our web application will have to interact with Google Home. Furthermore, our web system will be supported through the web and hence users will need to have to use a computer or any devices that allows the user to access the Smart Security web application. Users will need mobile devices to receive text messages.

4.1.3 Software Interfaces

4.1.3.1 For our system, we used Google's DialogFlow and Firebase for the backend and Google integration. For the web application, html, javascript, php, and css was used. **Priority 1**

4.1.4 Communication Interfaces

4.1.4.1 Smart Security web application will communicate with Google Home in order to get information about tripped alarms and armed/disarmed alarms. Additionally, our system will need a database system that will store user information, system transaction and therefore needing a pull protocol to retrieve information to the web application. Furthermore, we will need to use a push protocol to send text message notifications to the user's mobile devices. **Priority 1**

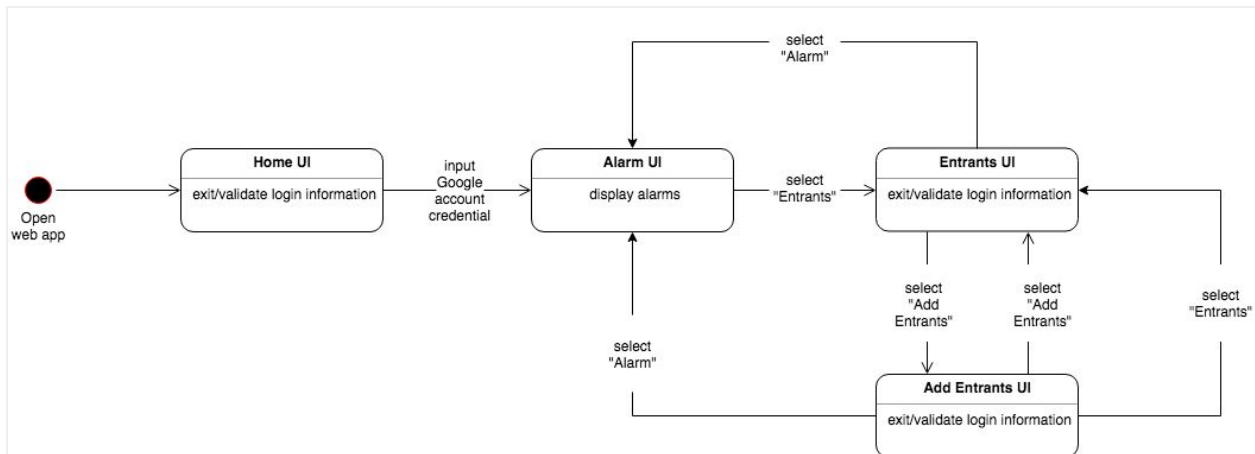


Figure 1. A state machine diagram for illustrating the different states of using the web application at each interface.

Home UI

In this state, the user sees the Home Screen UI, which tells users about the application. There is a link on the right corner for users to log in with their Google account. Once the user is logged on with a valid account, the state changes to the “Alarm UI” state.

Alarm UI

In this state, the user sees a log of past alarms, including the time they were set, the time they were tripped, and whether the entrant was authorized or unauthorized.

The user will be able to see the following buttons on the page:

- “Entrants” button - on selecting the “Entrants” button, the state transitions to the “Entrants UI” state

Entrants UI

In this state, the user sees a list of their approved entrants.

The user will be able to see the following buttons on the page:

- “Alarm” button - on selecting the “Alarm” button, the state transitions to the “Alarm UI” state
- “Add Entrants” button - on selecting the “Add Entrants” button, the state transitions to the “Add Entrants UI” state

Add Entrants UI

In this state, the user sees fields in which they can enter a new entrant’s information.

The user will be able to see the following buttons on the page:

- “Alarm” button - on selecting the “Alarm” button, the state transitions to the “Alarm UI” state
- “Entrants” button - on selecting the “Entrants” button, the state transitions to the “Entrants UI” state
- “Add Entrants” button - on selecting the “Add Entrants” button, the state transitions to the “Add Entrants UI” state

4.2 Functions

4.2.1 Logging into Smart Security

4.2.1.1 Users shall log into Smart Security with their Google account credential. **Priority 1**

4.2.2 Creating Arm/Disarm Alarm Code

4.2.2.1 Users shall create an activate alarm code through Google Home. **Priority 1**

4.2.2.2 Users shall create a deactivate alarm code through Google Home. **Priority 1**

4.2.3 User Arming Alarm

4.2.3.1 Users shall activate the alarm through Google Home with the arming alarm phrase.

Priority 1

4.2.3.1.1 The owner shall be able to arm the alarm with disarming alarm code.

4.2.3.1.2 Authorized contacts shall be able to arm the alarm with disarming alarm code.

4.2.3.2 Alarm shall be activated until alarm is disarmed. **Priority 1**

4.2.4 User Disarming Alarm

4.2.4.1 Users shall arm the alarm through Google Home with the disarming alarm code. **Priority 1**

4.2.4.1.1 The owner shall be able to disarm alarm with disarming alarm code. **Priority 1**

4.2.4.1.2 Authorized contacts shall be able to disarm alarm with disarming alarm code.
Priority 1

4.2.4.2 Alarm shall be disarmed until alarm is armed. **Priority 1**

4.2.5 Tripped Alarms

4.2.5.1 Alarm shall be tripped when Google Home picks up invasive sounds. **Priority 1**

4.2.5.2 A text message shall be sent to user when the alarm is tripped. **Priority 1**

4.2.5.2.1 Text message shall ask user to approve or deny entrance. **Priority 1**

4.2.5.2.1.1 User sends approve.

4.2.5.2.1.1.1 Alarm shall disarm.

4.2.5.2.1.2 User sends deny.

4.2.5.2.1.2.1 Alarm shall immediately play a loud alarm sound for a minute.

4.2.6 Creating Entrants List

4.2.6.1 Users shall be able to create a contact list of approved entrants through the web application. **Priority 1**

4.2.6.1.1 The contact's name, phone number, and email address is required. **Priority 1**

4.2.6.2 Users shall be able to delete an entrant from the list of approved entrants through the web application. **Priority 1**

4.2.7 Report of Tripped Alarms

4.2.7.1 User shall be able to review a list of tripped alarms on the web application. **Priority 1**

4.2.7.2 The list of tripped alarms will include the time, the date, and response. **Priority 1**

4.2.7.2.1 Time format: (hh:mm)

4.2.7.2.2 Date format: (mm/dd/yyyy)

4.2.7.2.3 Response: authorized or unauthorized

4.3 Logical Database Requirements

4.3.1 Use and Retention

4.3.1.1 Database should be able to store 1,000,000 transactions sent from Smart Security application. **Priority 3**

4.3.1.2 Database system must be available 24 hours a day except during scheduled downtime and/or emergency fixes. **Priority 1**

4.3.1.3 Database will be used to load data to the website. **Priority 3**

4.3.1.4 Database will be used to store data from the website. **Priority 3**

4.3.2 Types of Data

4.3.2.1 Account

4.3.2.1.1 Main account must have a Google account associated. **Priority 1**

4.3.2.1.2 Friendly contacts must have a name, phone number, and email. **Priority 1**

4.3.2.1.2.1 Contact name must be a string.

4.3.2.1.2.2 Contact phone number must be Integers.

4.3.2.1.2.3 Email address shall be in the format: "address@domain.extension".

4.3.2.2 Alarm

4.3.2.2.1 Alarm must have an activate and deactivate code. **Priority 1**

4.3.2.2.1.1 Code must be integers.

4.3.2.3 Google Home

4.3.2.3.1 Google Home must integrate with web application **Priority 1**

4.4 Design Constraints

4.4.1 Standards Compliance

4.4.1.1 Data Naming

4.4.1.1.1 Data input from Smart Security application should adhere to format agreed upon in the database system before inserting or editing. **Priority 1**

4.4.1.1.2 Naming convention of variables in the application code must match what the data is for. Username should be username and password should be password. **Priority 1**

4.4.1.2 Logging

4.4.1.2.1 Changes to database such as inserting, editing and deleting must be recorded in a log file. This does not include logging data, just logging the changes taken place. **Priority 1**

4.4.1.2.2 Alarms set up in Smart Security must be recorded in an alarm log file before and after the changes. **Priority 1**

4.4.1.2.3 Alarms disarmed in Smart Security must be recorded in an alarm log file before and after the changes. **Priority 1**

4.4.1.2.4 All maintenance time and downtime must be recorded. **Priority 1**

4.5 Software System Attributes

4.5.1 Reliability

4.5.1.1 System and Software Design

4.5.1.1.1 Systems must be built modularly to ensure ease of replacement when upgrading software and/or hardware. **Priority 1**

4.5.1.1.2 Before maintenance takes place, thorough testing of at least 1 hour must be done on the new updates and/or bug fixes. **Priority 1**

4.5.1.1.3 There must be a backup server for Smart Security application and the database in case of unexpected downtime of the original server. **Priority 1**

4.5.1.1.4 Data must be backed up every 30 minutes to the backup server. **Priority 2**

4.5.1.2 Error Handling

4.5.1.2.1 Invalid login attempts to the system is limited to 5 tries before system locks user for certain amount of time period. **Priority 2**

4.5.1.2.2 Input that does not follow standard structure will display an error message to user and will not be put into the database. **Priority 1**

4.5.2 Availability

4.5.2.1.1 The Smart Security application and database system must be operating and running 99% of the calendar year. **Priority 1**

4.5.2.1.2 Both systems should be available for normal use 24 hours Monday to Sunday except for scheduled downtime on Sundays. **Priority 3**

4.5.3 Security

4.5.3.1 User Information

4.5.3.2 Users who entered their information to the system will be notified that the Smart Security will use their phone number to send text notifications regarding their alarm system. **Priority 1**

4.5.3.3 All user sensitive information must be encrypted. **Priority 2**

4.5.3.2 System

4.5.3.2.1 The application and database server should be monitored every 5 seconds while it is operating to ensure that there are no hacking attempts. **Priority 3**

4.5.3.2.2 Access to Smart Security application is limited to those who have an account that is approved by the administrator(s) of the system. **Priority 1**

4.5.3.2.3 Systems should be updated every 2 weeks to keep security measures up-to-date. **Priority 3**

4.5.3.2.4 Normal users must not be able to access administrator side of the both systems. **Priority 1**

4.5.4 Maintainability

4.5.4.1 The system will have maintenance done only on Sunday nights from around 3:00 AM to 3:30 AM and user are notified 3 days ahead about the scheduled downtime. **Priority 3**

4.5.4.2 Each maintenance should not exceed 20 minutes of downtime of both Smart Security application and the database server. **Priority 3**

4.5.4.3 The systems may undergo emergency downtime only if a exceptional bug is found and must be addressed as soon as possible. **Priority 3**

4.5.5 Portability

4.5.5.1 Browsers

4.5.5.1.1 Smart Security should operate normally on the following browsers:

4.5.5.1.1.1 Google Chrome version 55.0 or higher. **Priority 1**

4.5.5.1.1.2 Firefox version 51.0 or higher. **Priority 1**

4.5.5.1.1.3 Safari version 9.0 or higher. **Priority 1**

4.5.5.1.1.4 Internet Explorer 11 or higher. **Priority 3**

4.5.5.2 Operating Systems

4.5.5.2.1 Smart Security should operate on the following operating systems:

4.5.5.2.1.1 Android version 5.0 Lollipop or higher.

4.5.5.2.1.2 iOS version 9 or higher.

4.5.5.2.1.3 Windows version 7 or higher.

4.5.5.2.1.4 Mac OS X version 10.10 Yosemite or higher.

4.5.6 Performance

4.5.6.1 Application should open and load within 1.5 seconds from the time user enters the web address to access Smart Security (after accounting for their Internet speed). **Priority 2**

4.5.6.2 The system response time when loading and saving data should be less than 1 seconds. **Priority 2**

4.5.6.3 After maintenance, system should be up and running within 1 minute. **Priority 3**

4.5.7 Scalability

4.5.7.1 Database must be able to handle a minimum of 1 transaction per second. **Priority 1**

4.5.7.2 Smart Security application must be able to handle over 5 different connections from different networks. **Priority 1**

4.5.7.3 System should be able to add more administrators whenever it is needed. **Priority 2**

4.5.7.4 More resources should be added if there is more users using the system than the system can handle. **Priority 3**