
Software Requirements Document for Safe Home Framework

Authors: Yu Cao, Josh Dehlinger, Ruchi Dhingra and Jinchun Xia

Version	Date	Author	Change
0.1	Nov 30, 2004	Yu	Creation of parts of Section 3.
0.2	Dec 2, 2004	Josh	Creation of Section 1.
0.3	Dec 3, 2004	Josh	Creation of parts of Sections 2 and 3.
0.4	Dec 3, 2004	Ruchi	Preliminary review of Sections 1, 2 and 3.
0.5	Dec 6, 2004	Josh	Completion of Section 3.
0.6	Dec 7, 2004	Ruchi	Creation of Section 2.2 and compilation of intermediate document
0.7	Dec 9, 2004	Jinchun	Creation of Section 2.1.2
0.8	Dec 12, 2004	All	Review.
1.0	Dec 14, 2004	Ruchi	Incorporation of review comments and submission of first release.

Table of Contents

1.	Introduction	4
1.1.	Purpose	4
1.2.	Scope	5
1.3.	Definitions, Acronyms and Abbreviations	5
1.4.	References	6
1.5.	Overview	8
2.	Overall Description.....	10
2.1.	Product Perspective	12
2.1.1.	Concept of Operations	13
2.1.2.	Major UIs	13
2.1.3.	Hardware Interfaces.....	23
2.1.4.	Software Interfaces	23
2.1.5.	Communication Interfaces.....	24
2.1.6.	Memory Constraints	25
2.1.7.	Operations	25
2.1.8.	Site Adaptation Requirements	25
2.2.	Product Functions.....	26
2.2.1.	Use Case Set: Device Integrator/Upgrader UI.....	26
2.2.2.	Use Case Set: Research UI	29
2.2.3.	Use Case Set: Application Development UI.....	30
2.2.4.	Use Case Set: Administration UI.....	33
2.2.5.	Use Case Set: Device Operation UI.....	37
2.3.	User Characteristics.....	40
2.4.	Constraints.....	42
2.5.	Assumptions and Dependencies	43
2.5.1.	Stakeholder Assumptions and Dependencies.....	43
2.5.2.	Software Assumptions and Dependencies	45
2.5.3.	Hardware Assumptions and Dependencies.....	45
3.	Specific Requirements	47
3.1.	External Interface Requirements	47
3.1.1.	User Interfaces.....	47
3.1.2.	Hardware Interfaces.....	48
3.1.3.	Software Interfaces	49
3.1.4.	Communications Interfaces	49
3.2.	Features	50

3.2.1.	Feature - Device Integrator/Upgrader UI.....	50
3.2.2.	Feature - Research UI.....	52
3.2.3.	Feature - Application Development UI.....	53
3.2.4.	Feature - Administration UI.....	55
3.2.5.	Feature - Device Operation UI.....	57
3.3.	Performance Requirements	57
3.4.	Design Constraints	59
3.5.	Software System Attributes	59
3.5.1.	Reliability	59
3.5.2.	Availability	60
3.5.3.	Security.....	61
3.5.4.	Maintainability	63
3.5.5.	Portability	66
3.6.	Other Requirements.....	66

1. Introduction

The Safe Home project was initiated prior to the onset of the Fall 2004 semester by the Department of Computer Science at Iowa State University as means to enhance the reputation of the department at the national level, to provide a hands on learning experience for students, to provide research opportunities for graduate students and faculty and to be a means to obtain research grants. The Safe Home project is an initial development of a technologically enabled home environment to assist elderly and physically challenged persons to live a more comforting lifestyle.

The Safe Home project's software design was given to the COM S 409/509 class as a semester project to develop and detail potential features and requirements that will eventually be integrated into the Safe Home's environment. It is the intention of this project that the COM S 409/509 class develop a Vision Document, previously submitted, as well as a Software Requirements Specification document to be delivered to Dr. Simanta Mitra and the Department of Computer Science by the conclusion of the Fall semester. The contact persons for the Safe Home project are Dr. Simanta Mitra, Dr. Carl K. Chang and Dr. Johnny S. Wong.

The whole Safe Home project is being approached in an iterative manner over the next year or so. The first iteration comprises of a quick development of a basic prototype in order to garner more interest for the project and potentially more grant money. The project can be viewed as having two parts: features of the Safe Home and the Safe Home lab. The Safe Home is the eventual physical home where the Safe Home technology is integrated into an actual living environment. The Safe Home lab will reside in the Department of Computer Science (Atanasoff 115) at Iowa State University. The Safe Home lab will be the setting to develop, implement and test smart devices, perform experiments and configure smart device control logic. Within the actual Safe Home, features will be integrated and added gradually. This portion of the project, in particular, focuses on the UI needs of the researchers and developers associated with this project.

1.1 PURPOSE

This Software Requirements Specification (SRS) provides a description of all the functions, specifications, external behaviors, design constraints, requirements (both functional and non-functional) and other factors

necessary to provide a complete and comprehensive description of the user-interfaces within the proposed Safe Home Framework.

This document was compiled and written for Dr. Simanta Mitra as the primary audience. The secondary audience for this document is the remaining stakeholders of this project within the Computer Science department at ISU; namely Dr. Carl K. Chang, Dr. Johnny Wong and the graduate students that will be working on the design and development of this project in the future.

1.2 SCOPE

The Safe Home Framework project detailed in this SRS is designed to facilitate the needs of the researchers, as well as the smart devices, with the Safe Home environment (Safe Home lab and Safe Home). The Safe Home Framework consists of the software system that will integrate the functionality, communications and operations of all the smart devices with the home and the occupants, researchers and other stakeholders associated with the Safe Home. This document specifically details the requirements for the user-interfaces needed by researchers within the Safe Home Framework.

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Term	Description
API	Application Programming Interface
BLOB	Binary Large Object data file format
COM S	Computer Science
CORBA	Common Object Request Broker Architecture
COTS	Consumer Off-The-Self products
CSV	Comma Separated Value file format
HCI	Human Computer Interaction
HTTP	Hyper Text Transfer Protocol
ISU	Iowa State University
JDBC	Java DataBase Connectivity
JSP	Java Servlet Pages

JVM	Java Virtual Machine
Kbps	Kilobits per second
MB	MegeByte(s)
MS	MicroSoft
RFID	Radio Frequency IDentification
SH	Safe Home/ Smart Home
Safe Home Environment	The physical Safe Home including the smart devices
Safe Home Framework	The design and architecture requirements of the control software
SRS	Software Requirements Specification document
SQL	Structure Query Language
TCP/IP	Transport Control Protocol / Internet Protocol
UI	User Interface
USB	Universal Serial Bus
XML	eXtensible Markup Language

1.4 REFERENCES

The proposed requirements and features detailed in this document have had some influence from other Safe Home projects. The following lists the resources used in the creation of the features and attributes for this project:

- The Adaptive House, University of Colorado at Boulder
 - <http://www.cs.colorado.edu/~mozer/house/>
- AIRE, Massachusetts Institute of Technology
 - <http://www.ai.mit.edu/projects/aire/>
- The Aware Home, Georgia Tech University
 - <http://www.cc.gatech.edu/fce/ahri/>
- CyberManor, Internet Home Alliance
 - <http://www.cybermanor.com/>
- Easy Living, Microsoft
 - <http://research.microsoft.com/easyliving/>

- Gator Tech Smart House, University of Florida
 - <http://www.erc.ufl.edu/>
- House, Massachusetts Institute of Technology
 - http://architecture.mit.edu/house_n/
- IIB, Trinity College Dublin
 - http://www.cs.tcd.ie/research_groups/IIB/
- i-Land, Amiente
 - <http://www.darmstadt.gmd.de/ambiente/i-land.html>
- MavHome, University of Texas at Arlington
 - <http://mavhome.uta.edu/>
- Prima, Inria
 - <http://www-prima.imag.fr/>
- Smart Spaces Lab, National Institute of Standards and Technology
 - <http://www.nist.gov/smartspace/>

In addition to resources used for gathering Safe Home features from existing Safe Home projects, the following reference was used to gather knowledge about the Niagara Framework:

- Niagara Framework, Tridium
 - <http://www.tridium.com/products/niagara.asp>

Thirdly, control system resources were utilized to gather background knowledge of basic control system concepts and operations. These resources included:

- Software and Controls Requirements, Advanced Technology Solar Telescope
 - <http://atst.nso.edu/library/docs/SPEC-0005.pdf>
- Control System Design, University of Newcastle
 - <http://csd.newcastle.edu.au/control/>
- Control Systems Centre, University of Manchester
 - <http://www.csc.umist.ac.uk/>

Lastly, this document is based upon the features described by the Vision Document previously prepared as well as the composite Safe Home Framework features list compiled by Dr. Simanta Mitra.

1.5 OVERVIEW

As the baby boom generation gracefully ages, the United States will be consumed with the necessity to care for an abundance of able but elderly people in the coming two decades. This necessitates home environments that can adequately care for able-bodied, elderly people. At the same time, pervasive technology is becoming more and more prevalent and accepted in society to a point where future elderly generations will be aided by technology in order to maintain independence. In parallel, the Department of Computer Science at Iowa State University is looking for innovative research opportunities to distinguish its program, provide unique learning and research problems to its faculty and students and become eligible for further research grants. These two factors lead to the initiation of the Safe Home project.

The Safe Home project was established to investigate the primary steps into establishing a Safe Home environment to provide able-bodied elderly people a home that allows them to remain independent from the health care systems (such as nursing homes) for as long as possible. The Safe Home is to employ technological solutions to care for and make life easier for these people.

Any Safe Home project will rely on a set of smart devices throughout the home to aid a person with everyday living. These smart devices will need to be interconnected in various ways to allow for control and communication among them as well as interaction with the user(s). The Safe Home Framework, the primary focus of this document, will manage the core of the underlying software necessary to operate all of these devices for the various types of users (occupants, researchers, etc). At a high level, the Safe Home Framework allows:

- Researchers to develop and run experiments for the smart devices.
- Occupants to control and interact with the smart devices through various interfaces to make their lives more comfortable.
- Smart devices to communicate with outside systems (such as the 9-1-1 emergency system) under well defined circumstances or conditions.
- Smart devices to communicate with each other under defined circumstances or conditions.

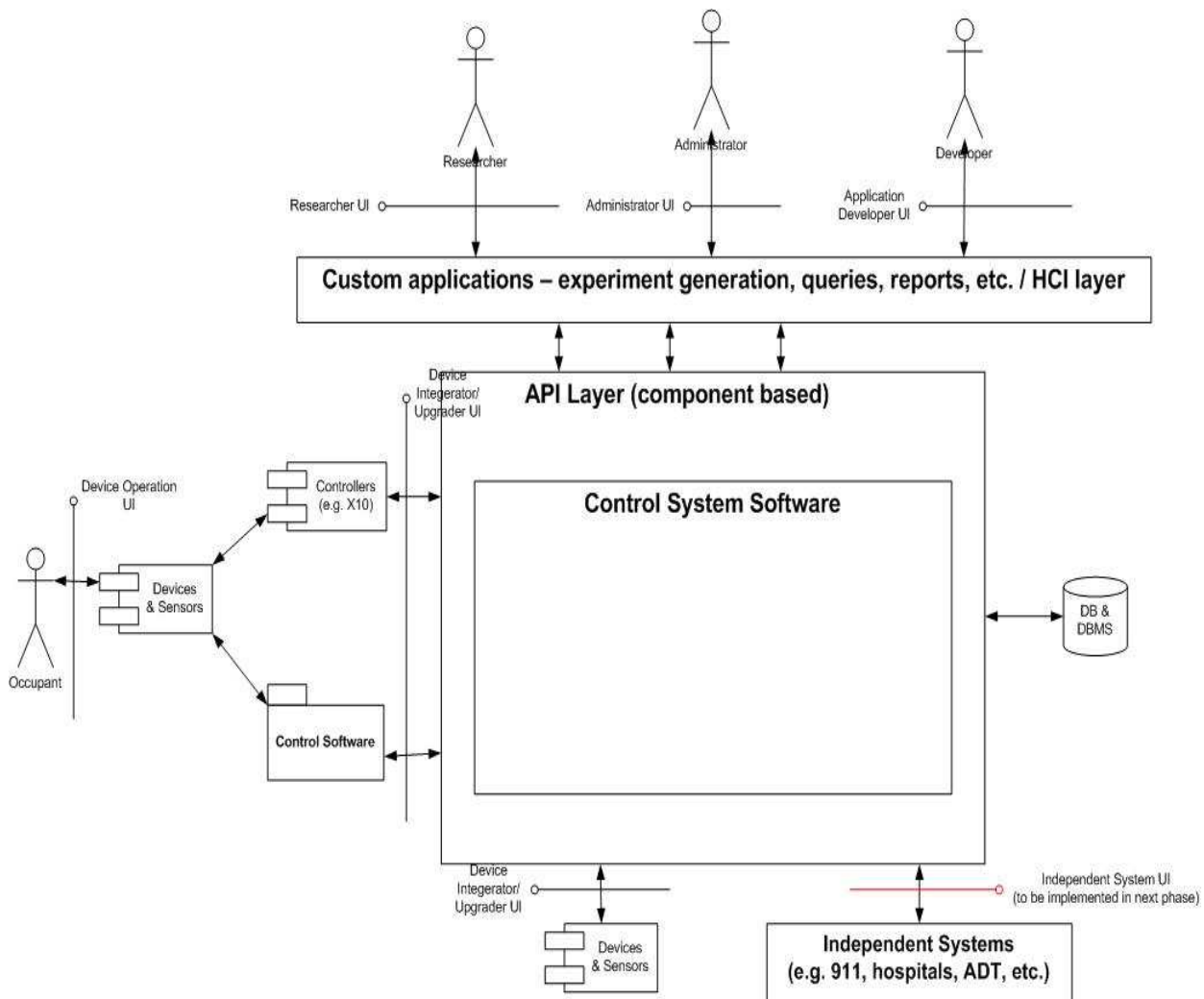
These are just some of the general features that the Safe Home Framework will provide to the various users involved in the Safe Home project. In addition to these features, the Safe Home Framework will be:

- Scalable to allow future device additions into the system.
- Versatile to allow researchers to perform a wide variety of simulations and experiments.
- Adaptable to allow for "plug-and-play" like applications and control software to monitor and run the smart devices.

2 Overall Description

A Safe Home aims to provide occupants maximum comfort, safety and security. This project is specifically designed for aged people and people with minor disabilities. The Safe Home consists of two major components: the actual home with the smart devices and the supporting software. As previously mentioned, this document specifically concentrates on the features and requirements of the Safe Home Framework's UI as it pertains to the researchers.

The following figure provides a potential software architecture for the Safe Home Framework:



The concentration of this document is at specifying the requirements for five different interface points illustrated in the above architecture. The only interface not detailed within this document that is pictured in the potential software architecture, shown previously, is the external system UI. We reserve the specifications for this UI for future releases. The interface points described in this document include: the Device Integrator/Upgrader UI, the Research UI, the Application Development UI, the Administration UI and the Device Operation UI. Each of these UIs provide distinct functions to different users and have different features and requirements. Below, we provide a brief description of each UI.

This document considers the researchers and developers the target (primary) users. A user can have access to multiple UIs, through a tab-based navigation thus allowing him to assume multiple roles.

Device Integrator/Upgrader UI

This UI allows a researcher, developer or administrator to perform several necessary functions pertaining to the management of the smart devices. The main functions that this UI provides to the user is to allow the user to add, configure, trouble-shoot and/or remove smart devices from operating within the Safe Home environment. This UI will be accessible by researchers, developers and administrators.

Research UI

This UI allows a researcher to perform several necessary functions pertaining to research experiments being performed using the Safe Home Framework and the Safe Home environment. This UI provides two main features: management of control routines and management of experiment relay and an experiment simulation mechanism. The management of control routines provides the user with authority over the control routines driving the smart devices within the Safe Home environment. The replay and simulation mechanism aids a researcher in either replaying an experiment previously performed or by simulating an experiment. A simulated experiment executes within the Safe Home Framework without using the actual, physical devices within the Safe Home environment. This UI will be accessible by researchers only.

Application Development UI

The application development interface allows a researcher or developer to perform several necessary functions pertaining to developing applications (either control routines, devices applications or experiment applications)

for integration into the Safe Home environment using the Safe Home Framework. This UI mainly provides management of the addition of application executable code into the Safe Home Framework and associating it with a device(s). This UI will be accessible by researchers and developers.

Administration UI

This UI allows an administrator to perform several necessary functions pertaining to access control, user settings and logging actions. Access control marshals access rights and read-write-execute permissions to the system to appropriate users and allows an administrator to define or modify access controls. User settings functions accessible in this UI include the ability for adding, configuring, managing and deleting user settings. Lastly, the logging mechanism to adjust or delete the actions that are logged by the system for security purposes. This UI will be accessible by administrators.

Device Operation UI

This UI allows an occupant to perform several necessary functions pertaining to adjusting the settings of the smart devices within the Safe Home environment. The main function of this interface is to allow occupants to override and/or change the setting for executing control routines on the smart devices. .

2.1 PRODUCT PERSPECTIVE

Safe Home projects have popped up at research universities and at major industry companies in the last few years. Some of these projects have been centered upon "smart homes" where the concentration is more geared towards luxury living rather than assisted living. This project, however, concentrates on the latter focus: to provide an assisted living environment for its occupants.

Similar projects can be found at Florida University, MIT, Colorado State University, University of Texas at Arlington, University of Colorado at Boulder, Georgia Tech University, Stanford University, etc. Some of these projects are called Smart Home projects and include luxury services too. References to many of these projects were given in the References portion of this document found in Section 1.3.

Any Safe or Smart Home will require a rather large piece of control software to act as a common base between the smart devices and the control logic. In addition, intermediary software is necessary when desiring to have

additional interfaces for researchers, developers, administrators, etc. To suit this need, the Tridium Corporation has the Niagara Framework commercially available for similar applications. A reference to the Niagara Framework was given in the References portion of this document found in Section 1.3.

2.1.1 Concept of Operations

The set of Safe Home Framework UIs detailed in this document will provide researchers, developers and, to a lesser extent, occupants and administrators with the initial interfaces to provide these users with a core set of functionalities to bootstrap the Safe Home project at Iowa State University. The UIs will be web-based to provide easy and efficient access.

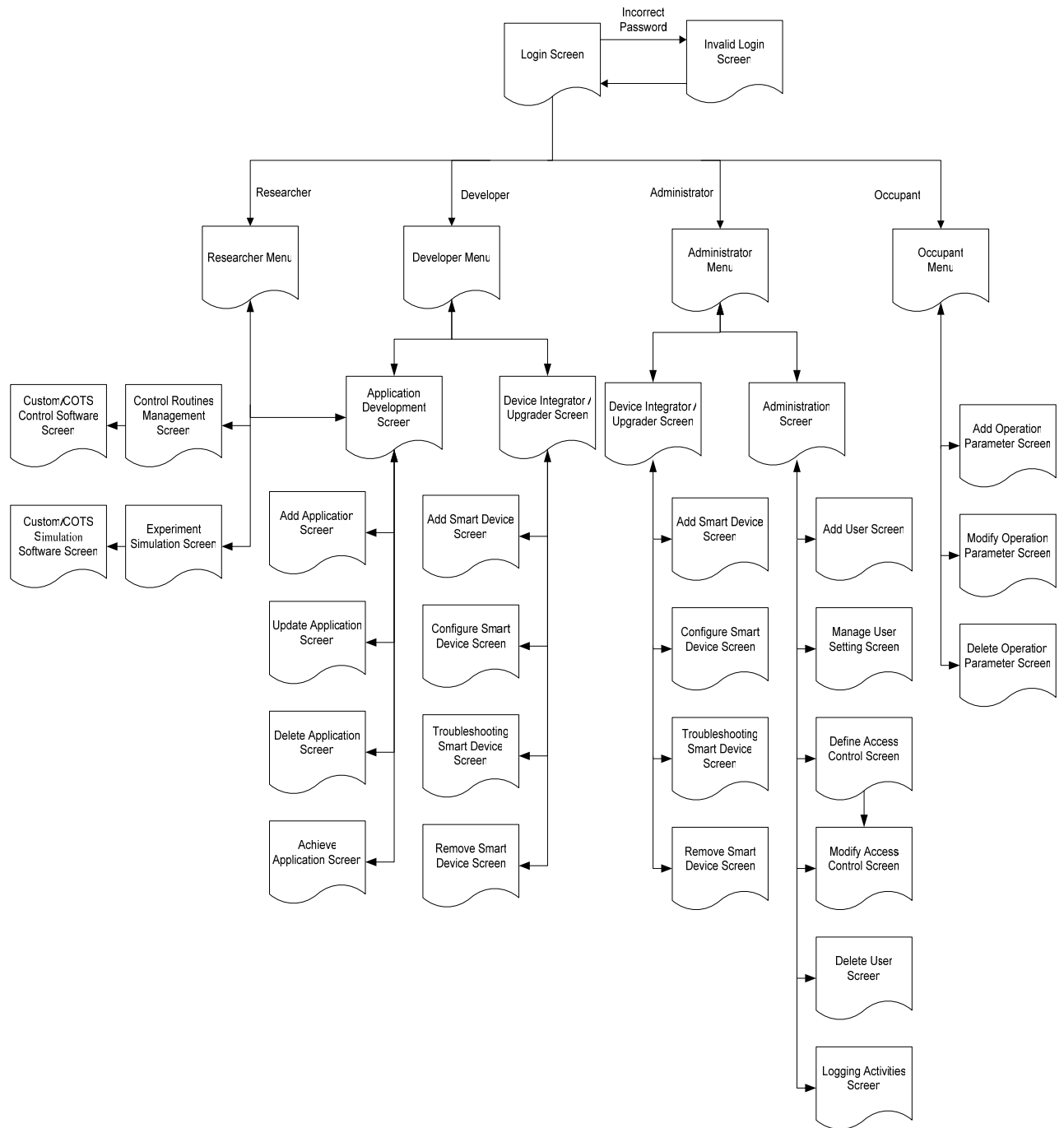
2.1.2 Major UIs

This section details the major UIs within the Safe Home Framework. A UI screen flow diagram is presented as well as a set of representative screen shots of the UIs may look like.

2.1.2.1 Overall Flow Diagram

The following flow diagram depicts the overall structure and behavior of the Safe Home system. Researchers can access Research UI, Device Integrator/Upgrader UI, and Application Development UI. Developers can access Device Integration/Upgrader and Application Development UIs while the Administrators can use Device Integrator/Upgrader and Administration UIs. The Occupants can only use the Device Operation UI.

The overall flow diagram is:

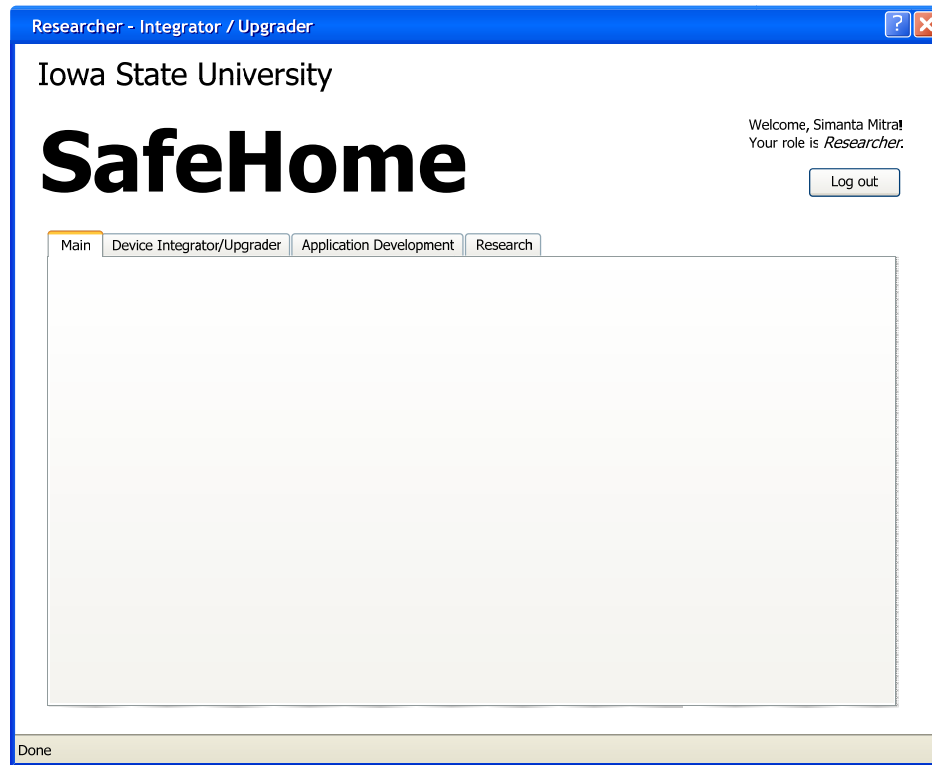


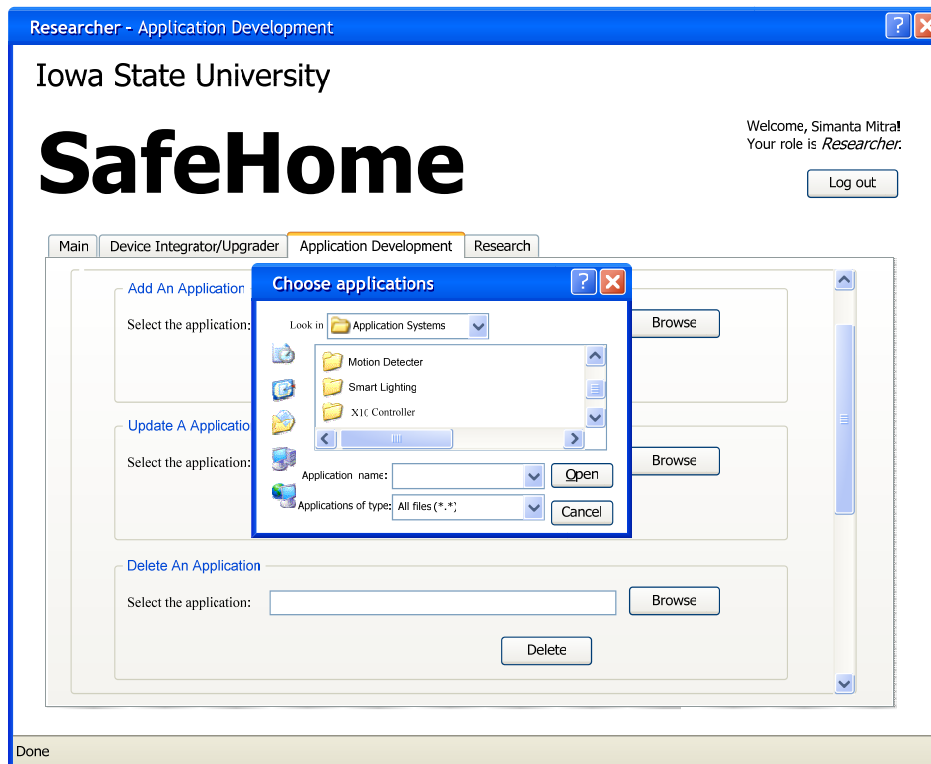
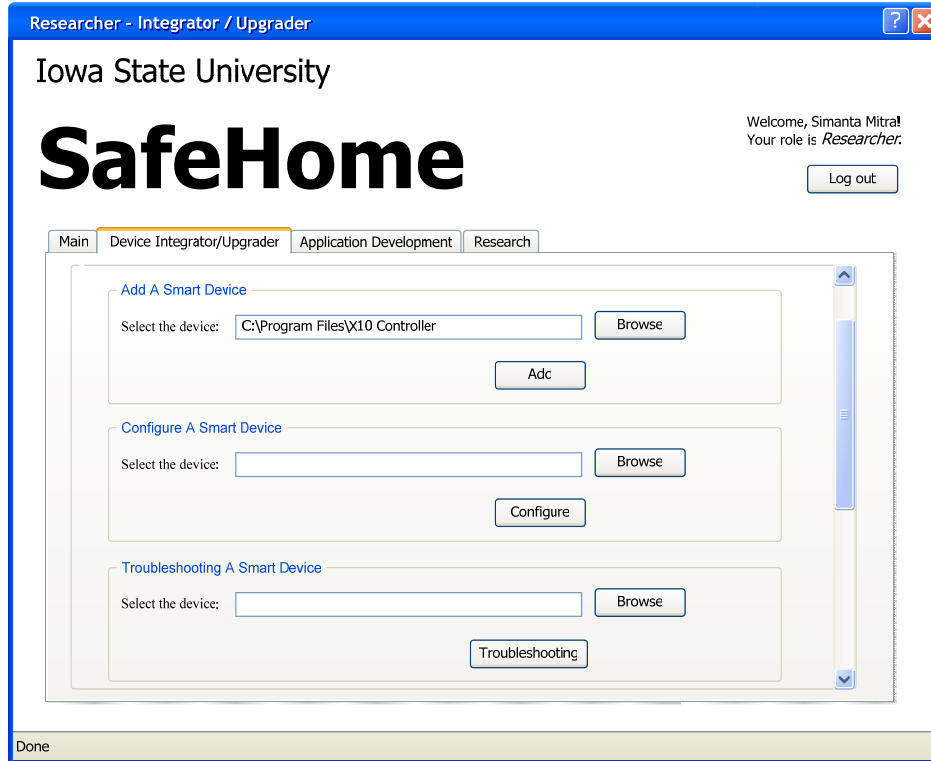
2.1.2.2 2.1.2.2 Example Screenshots

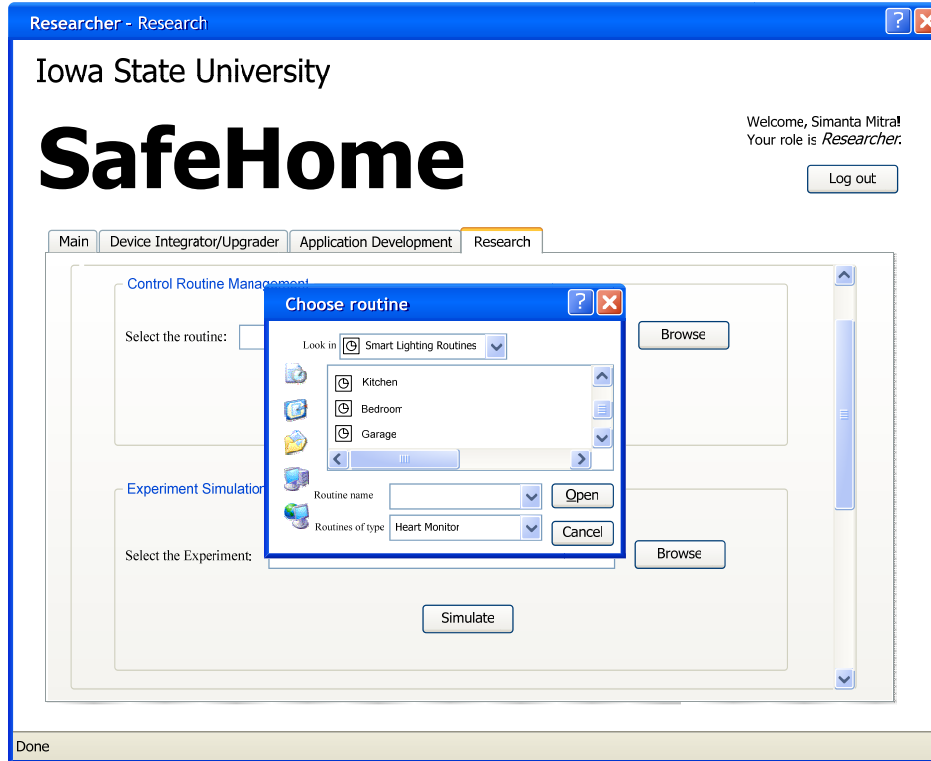
This section presents a representative sample of UIs.

2.1.2.2.1 Researcher Screenshots

The following four screenshots show the user interfaces when they log onto the system as a researcher:

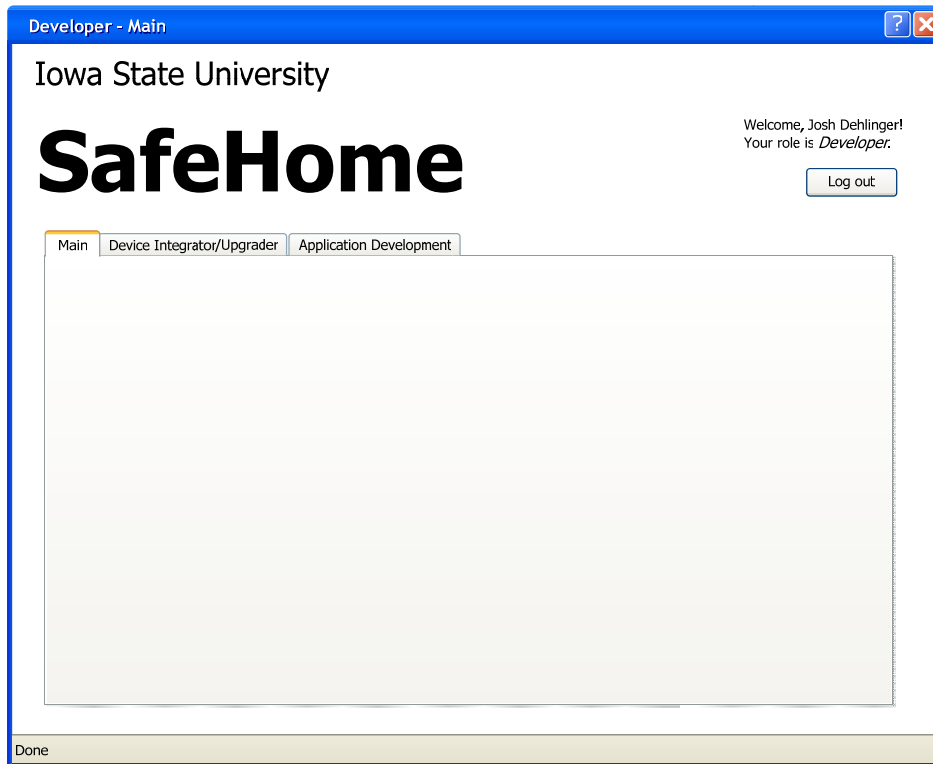


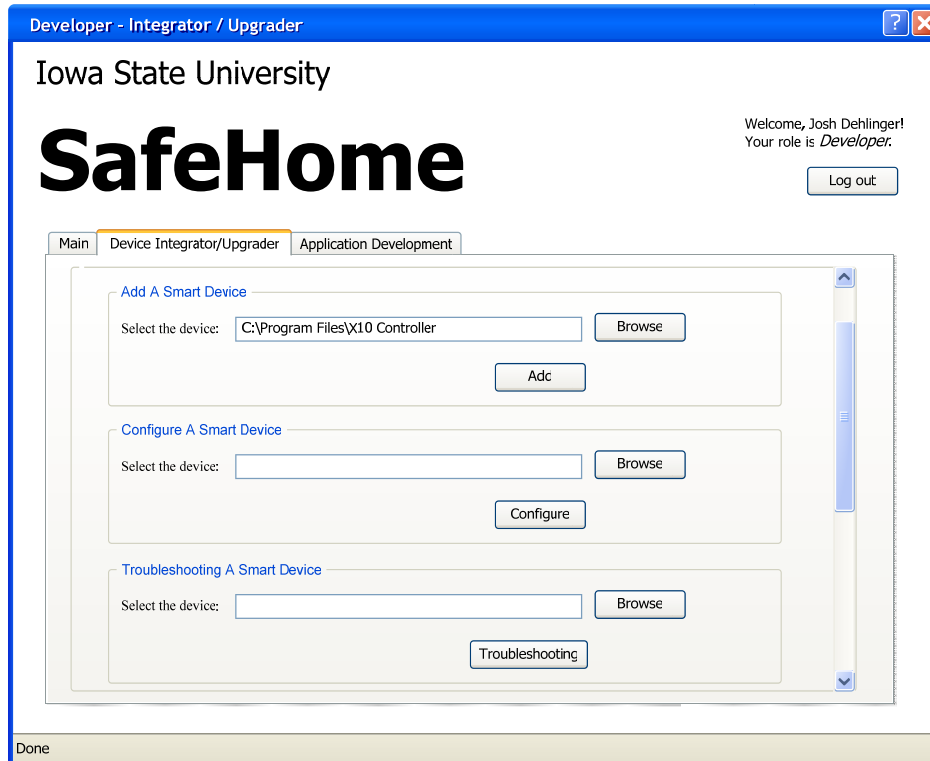




2.1.2.2.2 Developer Screenshots

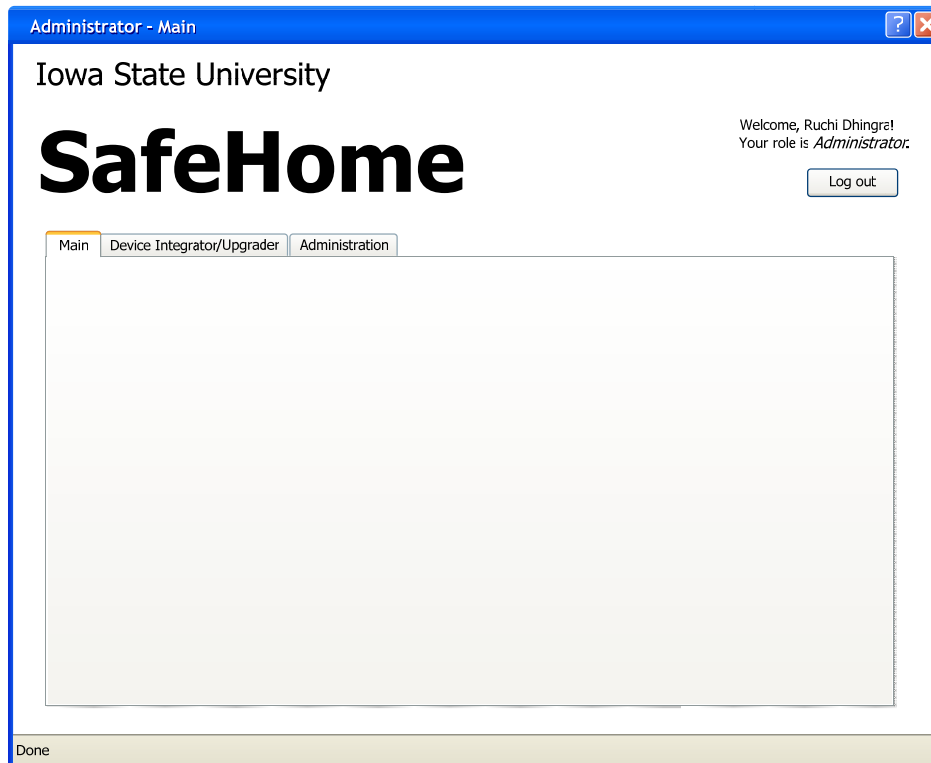
The following three screenshots show the user interfaces when they log onto the system as a developer:

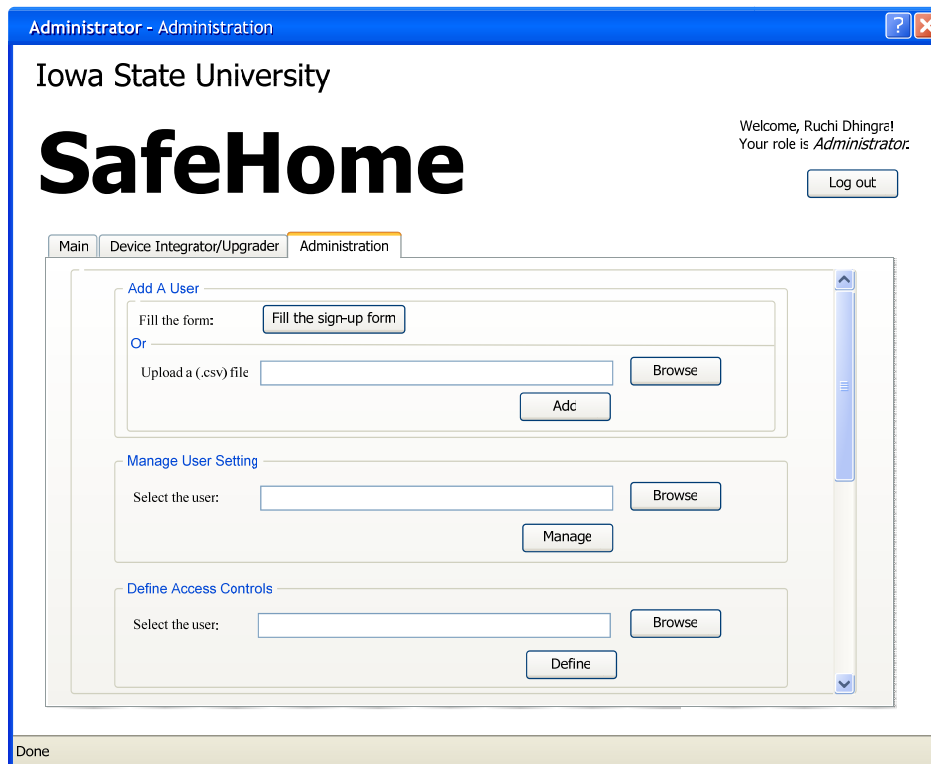
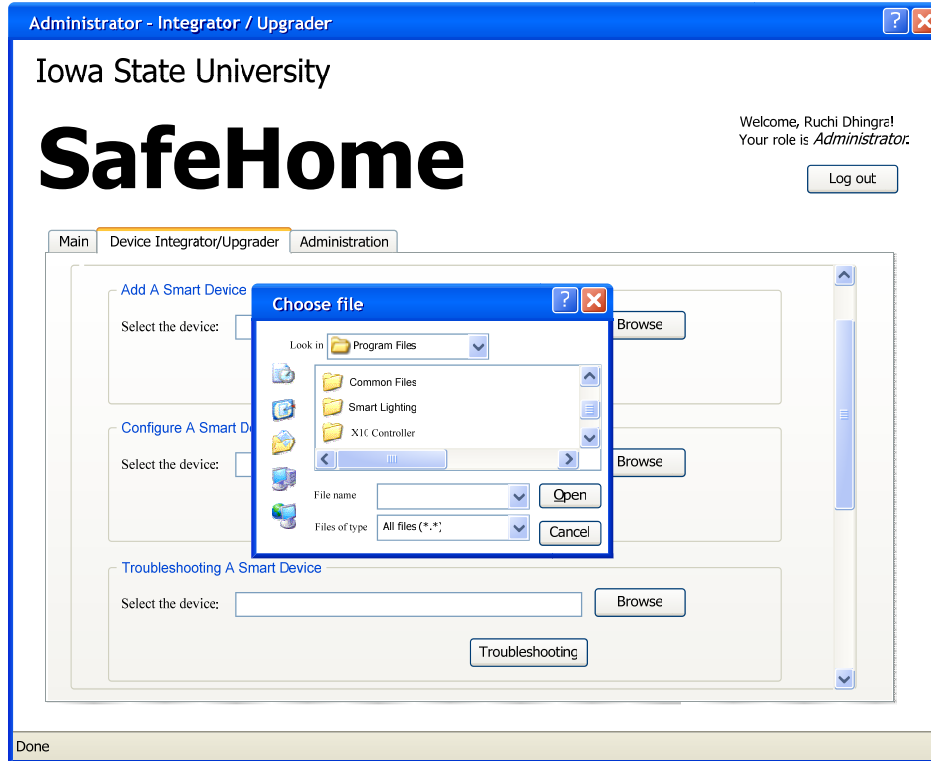




2.1.2.2.3 Administrator Screenshots

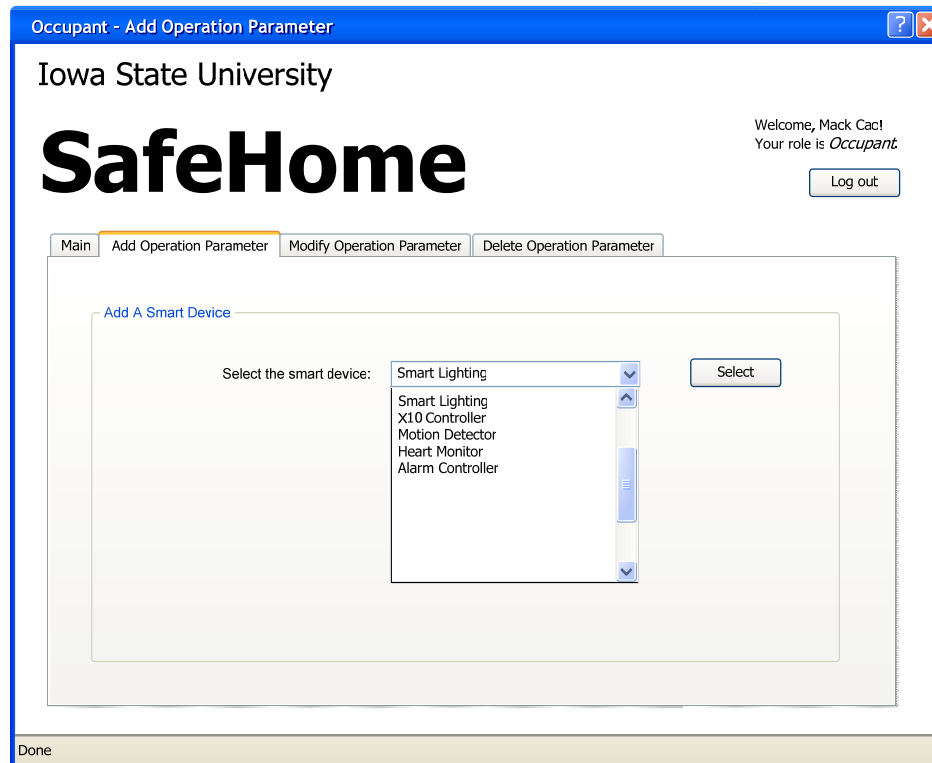
The following three screenshots show the user interfaces when they log onto the system as an administrator:





2.1.2.2.4 Occupant Screenshots

The following screenshot shows the user interface when they log onto the system as an occupant:



2.1.3 Hardware Interfaces

The proposed Safe Home Framework UIs will require a variety of hardware interfaces to operate on:

- 2.1.3.1 *Network Hardware* - Each computing system employing the Safe Home Framework shall have a network interface card.
- 2.1.3.2 *Network Connection* - Each computing system employing the Safe Home Framework physical network connection (telephone line, cable line, wireless link, etc) shall have a minimum bandwidth of 56 Kbps.
- 2.1.3.3 *Web-enabled Systems* - Each computing system employing the Safe Home Framework shall have a display monitor capable of displaying web pages.
- 2.1.3.4 *Color Interface* - Each computing system employing the Safe Home Framework display monitor shall have the ability to display a minimum of 16 bit color.
- 2.1.3.5 *Interface Pixel Rate* - Each computing system employing the Safe Home Framework display monitor shall have a minimum of 800 x 600 pixel rate.
- 2.1.3.6 *Input Device* - Each computing system employing the Safe Home Framework shall have a keyboard input device.
- 2.1.3.7 *Navigation Device* - Each computing system employing the Safe Home Framework shall have a navigation device such as a mouse, track ball, stylist, etc.
- 2.1.3.8 *Smart Device Connection* - Each Safe Home smart device shall be connected to the Safe Home Framework via a controller or device driver.

2.1.4 Software Interfaces

The proposed Safe Home Framework UI will require a variety of software interfaces:

- 2.1.4.1 *JVM Versioning* - Each computing system employing the Safe Home Framework shall have a current version (a release within the last year as of Dec. 2004) of the JVM installed.
- 2.1.4.2 *JSP Rendering Capability* - Each computing system employing the Safe Home Framework shall have software capable of rendering JSP pages.
- 2.1.4.3 *Java Beans Rendering Capability* - Each computing system employing the Safe Home Framework shall have software capable of rendering web pages using Java Beans.
- 2.1.4.4 *Java Script Rendering Capability* - Each computing system employing the Safe Home Framework shall have software capable of rendering web pages using Java Script.
- 2.1.4.5 *Database Necessity* - The Safe Home Framework shall use a database to store data.
- 2.1.4.6 *Database Connectivity* - The Safe Home Framework database shall be accessible via JDBC.
- 2.1.4.7 *Database Query Language* - The Safe Home Framework database shall be query accessible via SQL2 (also known as SQL 1992).
- 2.1.4.8 *Device Communication* - Each Safe Home smart device shall be able to communicate with the Safe Home Framework via an API.

2.1.5 Communication Interfaces

The proposed Safe Home Framework UI will require a variety of communication interfaces:

- 2.1.5.1 *Network Interface Card* - Each computing system employing the Safe Home Framework shall have a network interface card (repeat of 2.1.3.1).
- 2.1.5.2 *Interface Network Bandwidth* - Each computing system employing the Safe Home Framework physical network connection (telephone line, cable line, wireless link, etc) shall have a minimum bandwidth of 56 Kbps (repeat of 2.1.3.2).
- 2.1.5.3 *Database Accessibility* - The Safe Home Framework database shall be accessible via JDBC (repeat of 2.1.4.6).
- 2.1.5.4 *Network Communication Protocol* - The Safe Home Framework UIs shall communicate using TCP/IP protocol suite with the Safe Home Framework.
- 2.1.5.5 *Network Hardware* - All wireless communication used by the Safe Home Framework to computing systems employing Safe Home Framework UIs shall conform to the 802.11 standards.

2.1.6 Memory Constraints

The proposed Safe Home Framework may require a minimal amount of memory constraints:

- 2.1.6.1 *Interface Memory Capacity* - Each computing system employing the Safe Home Framework shall have a minimum of 100 MBs.

2.1.7 Operations

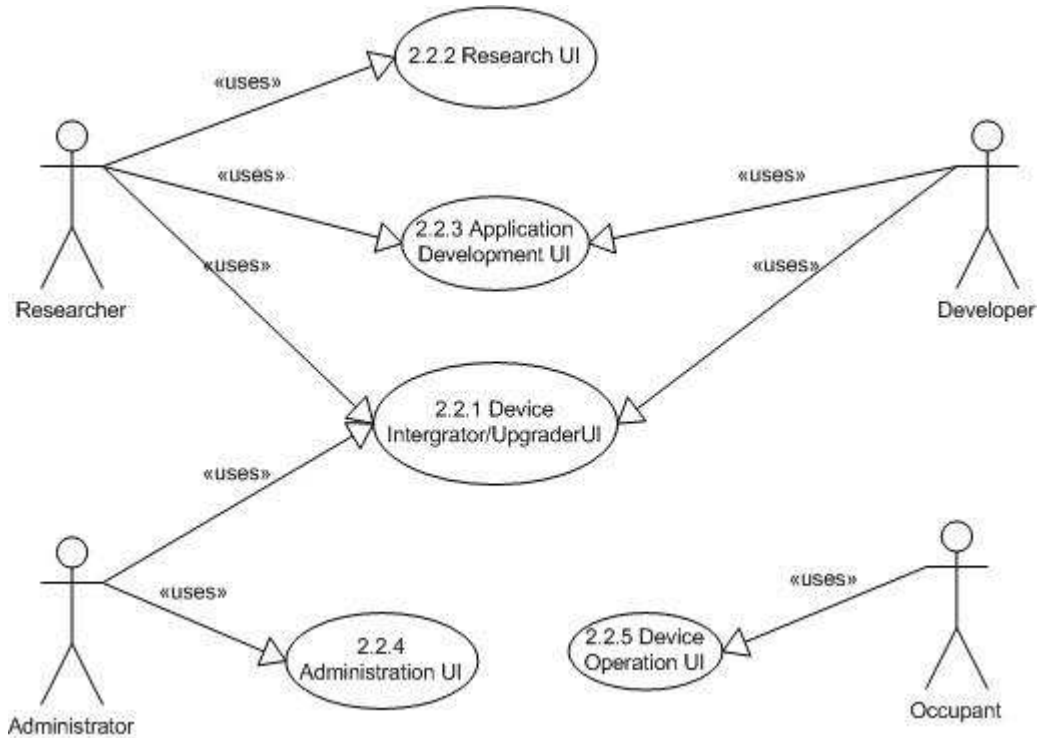
- 2.1.7.1 This section lists the special operations needed during the execution of the Safe Home Framework.
- 2.1.7.2 *Database Backup* - The data store that stores the research data shall be routinely backed up to facilitate an easier recovery process in the event that the data is lost or corrupted.

2.1.8 Site Adaptation Requirements

[None]

2.2 PRODUCT FUNCTIONS

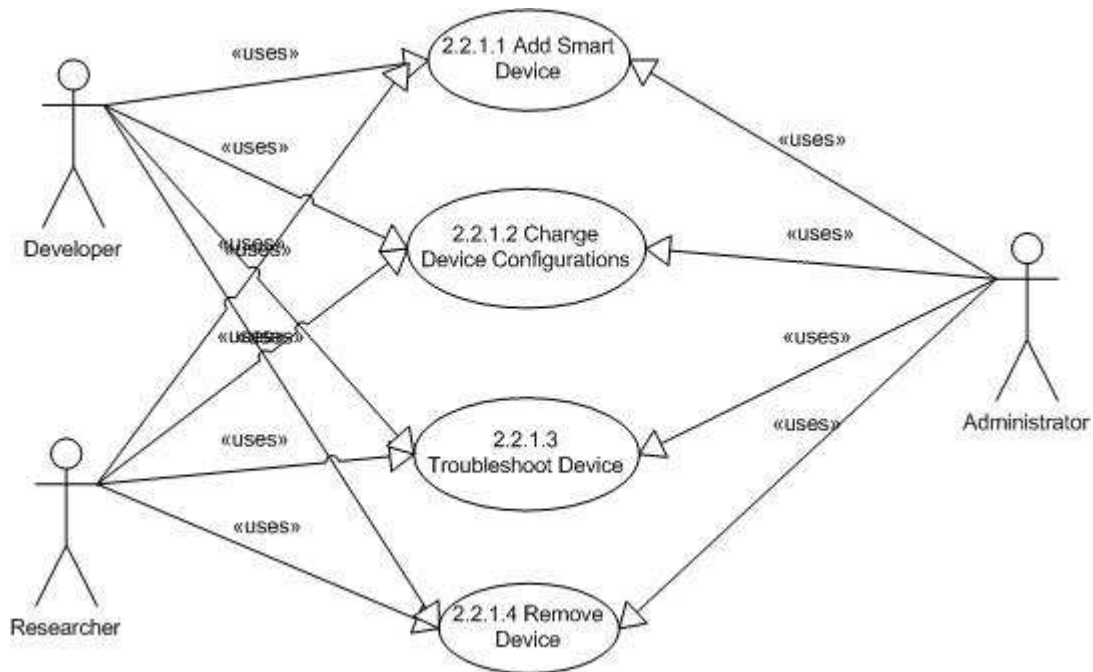
The following gives a high-level use case diagram for the proposed Safe Home Framework UIs described in this document:



There are four actors being considered in this phase as shown in high-level use case diagram: researcher, developer, administrator and occupant. There are five primary UIs within the Safe Home Framework. They are the Device Integrator/Upgrader UI, the Research UI, the Application Development UI, the Administration UI and the Device Operation UI.

2.2.1 Use Case Set: Device Integrator/Upgrader UI

The following depicts the use case for the Device Integrator /Upgrader UI:



Title: Device Integrator/Upgrader UI

Description: A researcher, developer or administrator can add a smart a device, change configurations of an existing smart device, trouble-shoot smart devices or remove an existing smart device.

Actors: Researcher, Developer, Administrator

2.2.1.1 2.2.1.1. Use Case - Add a Smart Device

Basic Flow:

1. The user will select the device (software component) to be added using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The user will select the server and destination folder into which this file/folder is to be copied using a browse-select location mechanism similar to a MS-Windows file management client.
3. The user will install the software.

Alternate Flow:

1. If the file/folder has been previously saved, the user can directly proceed to step 3 and install the software.

Preconditions:

1. The device (software component) should not already have been installed at the desired location.
2. The device (hardware component) should already have been installed as needed.

Postconditions:

1. The device is successfully installed.

2.2.1.2 2.2.1.2 Use Case - Change Device Configurations

Basic Flow:

1. The user will select the device (software component) whose configuration is to be modified using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The user will be directed to custom/COTS configuration management software for the selected device.

Alternate Flow:

1. None.

Preconditions:

1. The device (software component) should already have been installed at the desired location.
2. The device (hardware component) should already have been installed as needed.
3. The custom/COTS configuration management software should have been installed and should be accessible to the user.
4. The device should not currently be in use.

Postcondition:

1. The device configuration is successfully modified.

2.2.1.3 2.2.1.3 Use Case - Troubleshoot Device

Basic Flow:

1. The user will select the device (software component) for trouble-shooting using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The user will be directed to custom/COTS device management software for the selected device.

Alternate Flow:

1. None.

Preconditions:

1. The device (hardware component) should already have been installed as needed.
2. The custom/COTS device management software should have been installed and should be accessible to the user.

Postcondition:

1. The problem is successfully fixed.

2.2.1.4 2.2.1.4 Use Case - Remove a Device

Basic Flow:

1. The user will select the device (software component) to be removed using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The user will delete the device.

Alternate Flow:

1. None.

Preconditions:

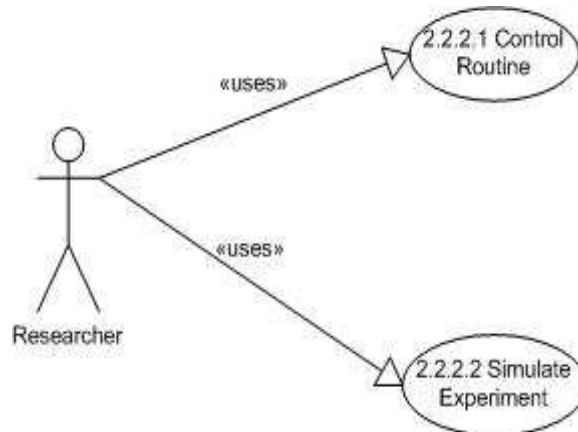
1. The device (software component) should already have been installed at the desired location.
2. The device should not currently be in use.

Postcondition:

1. The device is successfully removed.

2.2.2 Use Case Set: Research UI

The following depicts the use case set for the Research UI:



Title: Research UI

Description: A researcher can control routines and simulate experiments.

Actor: Researcher

2.2.2.1 2.2.2.1 Use Case - Control Routine

Basic Flow:

1. The user will select the routine to be controlled from a list of currently executing routines using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The user will be directed to custom/COTS control software for the selected routine.

Alternate Flow:

1. None.

Precondition:

1. The routine selected should be executing at the time of selection.
2. The custom/COTS control software should have been installed and should be accessible to the user.

Postcondition:

1. None.

2.2.2.2 2.2.2.2 Use Case - Simulate Experiment

Basic Flow:

1. The user will select the experiment to be simulated from a list of available experiments using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The user will be directed to custom/COTS simulation software for the selected experiment.

Alternate Flow:

1. None.

Precondition:

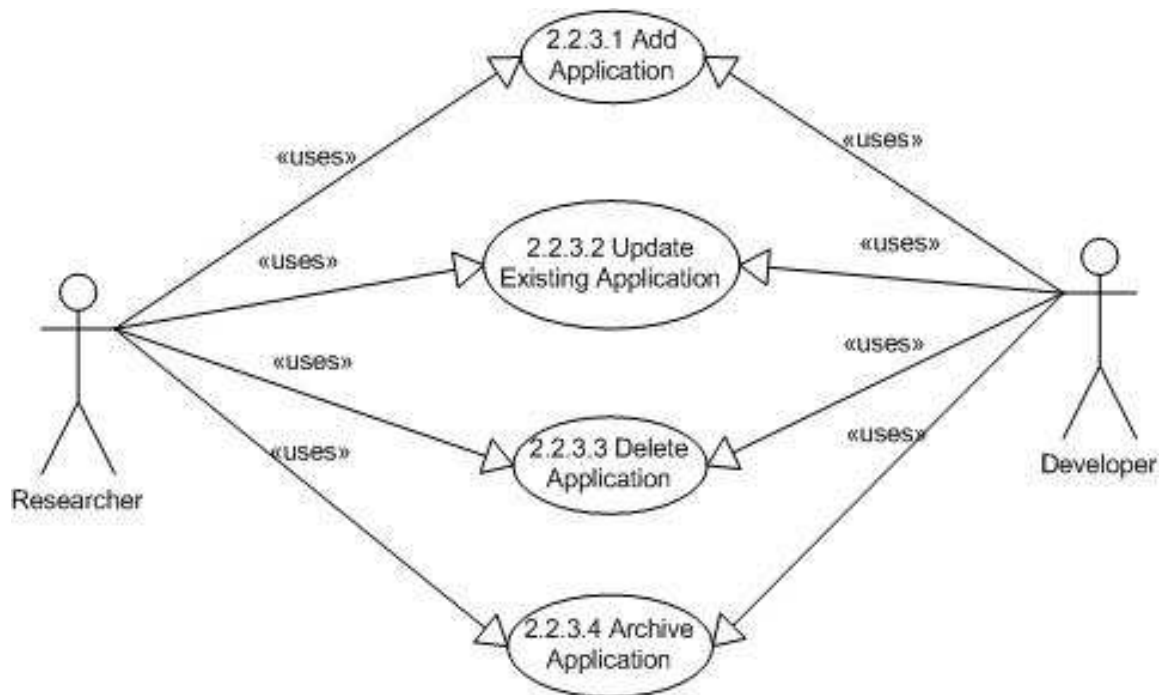
1. The experiment should have been defined and incorporated into the list of available experiments.
2. The experiment should not currently be in execution.
3. The custom/COTS simulation software should have been installed and should be accessible to the user.

Postcondition:

1. None.

2.2.3 Use Case Set: Application Development UI

The following depicts the use case set for the Application Development UI:



Title: Application Development UI

Description: A researcher or developer can add applications, update existing application, delete existing applications or archive existing applications.

Actors: Developer, Researcher

2.2.3.1 2.2.3.1 Use Case - Add Application

Basic Flow:

1. The user will select the application to be added using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The user will select the server and destination folder into which this application is to be copied using a browse-select location mechanism similar to a MS-Windows file management client.
3. The user will install the software.

Alternate Flow:

1. If the application has been previously saved, the user can directly proceed to Step 3 and install the application.

Preconditions:

1. The application should not already have been installed at the desired location.

Postconditions:

1. The application is successfully added.

2.2.3.2 2.2.3.2 Use Case - Update Existing Application**Basic Flow:**

1. The user will select the application to be updated using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The user will be directed to custom/COTS configuration management software for the selected application.

Alternate Flow:

1. None.

Preconditions:

1. The application should already have been installed as needed.
2. The custom/COTS configuration management software should have been installed and should be accessible to the user.
3. The application should not currently be in use.

Postconditions:

1. The application is successfully modified.

2.2.3.3 2.2.3.3 Use Case - Delete an Application**Basic Flow:**

1. The user will select the application to be deleted using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The user will delete the application

Alternate Flow:

1. None.

Preconditions:

1. The application should already have been installed at the desired location.
2. The application should not currently be in use.

Postconditions:

1. The application is successfully removed.

2.2.3.4 2.2.3.4 Use Case - Archive Application

Basic Flow:

1. The user will select the application to be archived using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The user will archive the application to the database.

Alternate Flow:

1. None.

Preconditions:

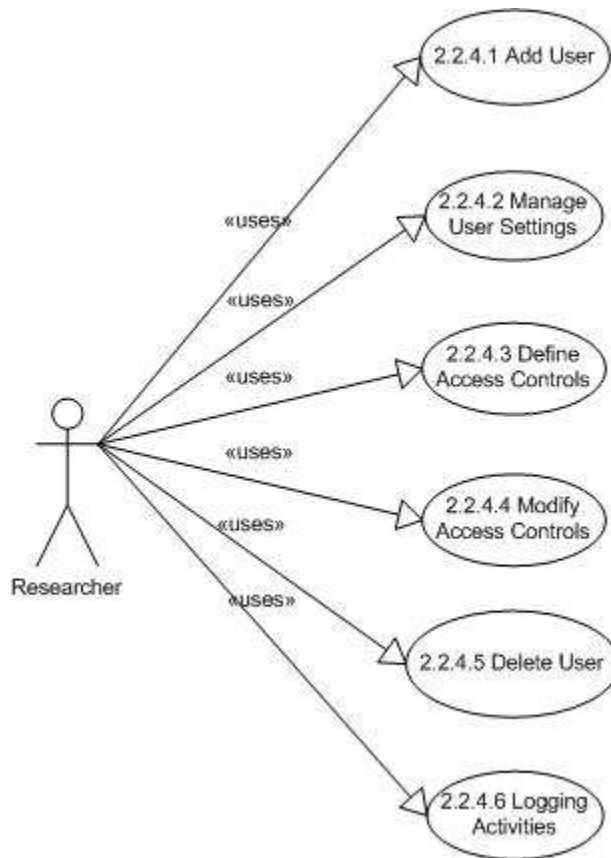
1. The application should not currently be in use.
2. The application should not already be archived in the database.

Postconditions:

1. The application is successfully archived into the database.

2.2.4 Use Case Set: Administration UI

The following depicts the use case set for the Administration UI:



Title: Administration UI

Description: An administrator can add or manage user settings, define or modify access controls, delete user or perform logging activities.

Actor: Administrator

2.2.4.1 2.2.4.1 Use Case - Add User

Basic Flow:

1. The administrator will click on an “Add New User” form.
2. The administrator will enter the following mandatory information on the form:
 - a. User Name
 - b. User’s Full Name
3. The administrator will enter the optional information on the form (information elements to be identified during detailed design). These fields can be left blank if desired.

4. The administrator will submit the form and the user database will be appropriately updated by the system.

Alternate Flow:

1. The administrator can create multiple user accounts by uploading a .csv file that contains all the mandatory information and as many elements of the optional information as desired.

Precondition:

1. The user name should not already exist in the user database.

Postcondition:

1. The user will be added to the user database.

2.2.4.2 2.2.4.2 Use Case - Manage User Settings

Basic Flow:

1. The administrator will select the user whose particulars are to be updated from a list of existing users. This will bring up an “Update User Settings” form.

2. The administrator will update information on the form. All fields except User Name can be modified.

3. The administrator will submit the form and the user database will be appropriately updated by the system.

Alternate Flow:

1. None.

Precondition:

1. The user should already exist in the user database.

Postcondition:

1. The user settings will be updated in the user database.

2.2.4.3 2.2.4.3 Use Case - Define Access Controls

Basic Flow:

1. The administrator will select from a list of existing users the user for whom access controls are to be defined. This will bring up an “Add Access” form.

2. The administrator will enter information on the form (information elements to be identified during detailed design).

3. The administrator will submit the form and the access control database will be appropriately updated by the system.

Alternate Flow:

1. None.

Precondition:

1. The user should already exist in the user database.

Postcondition:

1. The user's access rights will be added to the access control database.

2.2.4.4

2.2.4.5 2.2.4.4 Use Case - Modify Access Controls

Basic Flow:

1. The administrator will select the user whose access control settings are to be updated from a list of existing users for whom these settings have been previously defined. This will bring up a "Modify Access Controls" form.
2. The administrator will update information on the form. All fields except User Name can be modified.
3. The administrator will submit the form and the access control database will be appropriately updated by the system.

Alternate Flow:

1. None.

Precondition:

1. The user should already exist in the user database and should have predefined access control settings.

Postcondition:

1. The user's access control settings will be updated in the access control database.

2.2.4.6

2.2.4.7 2.2.4.5 Use Case - Delete User

Basic Flow:

1. The administrator will select the user to be deleted from a list of existing users..
2. This will bring up confirmation page that displays the selected user's settings.
3. The administrator will confirm the delete and the user database will be appropriately updated by the system.

Alternate Flow:

1. None.

Precondition:

1. The user should already exist in the user database.
2. The user should not be currently performing any activity within the entire system.

Postcondition:

1. The user will be deleted from the user database.

2.2.4.8 2.2.4.6 Use Case - Logging Activities

Basic Flow:

1. The administrator will specify the criteria for the log list – user or application. A “user log list” will show all current logging activities sorted by user name. An “application log list” will show all current logging activities sorted by application name.
2. The system will display the log list selected by the administrator.
3. The administrator will be able to update the logging activities in the list.
4. The modified list will be saved into the database.

Alternate Flow:

1. None.

Preconditions:

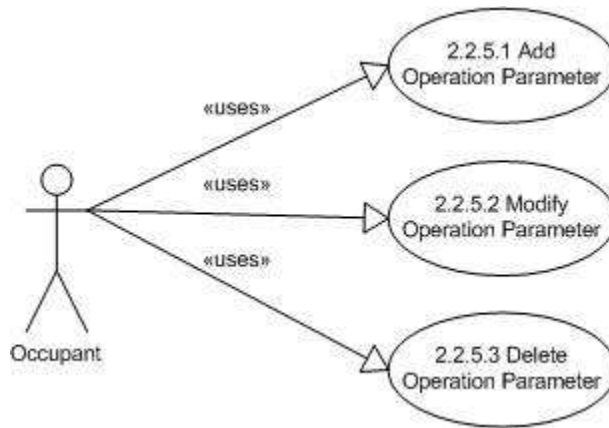
1. None.

Postconditions:

1. The modified log lists are saved into the database.

2.2.5 Use Case Set: Device Operation UI

The following depicts the use case for the Device Operation UI:



Title: Device Operation UI

Description: An occupant can add, modify or delete operation parameter for an existing smart device.

Actor: Occupant

2.2.5.1 2.2.5.1 Use Case - Add Operation Parameter

Basic Flow:

1. The occupant will select the smart device for which an operation parameter is to be added from a list of available devices using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The occupant will add the operation parameter(s) for the device.
3. The new parameters will be updated into the database.

Alternate Flow:

1. None.

Preconditions:

1. The device (software component) should already have been installed and should be included in the available list.
2. The parameter(s) being specified should not be been specified earlier.

Postconditions:

1. The device parameters are successfully defined.

2.2.5.2 2.2.5.2 Use Case - Modify Operation Parameter

Basic Flow:

1. The occupant will select the smart device for which an operation parameter is to be modified from a list of available devices using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The occupant will modify the operation parameter(s) for the device.
3. The new parameters will be updated into the database.

Alternate Flow:

1. None.

Preconditions:

1. The device (software component) should already have been installed and should be included in the available list.
2. The parameter(s) being specified should have been specified earlier.

Postconditions:

1. The device parameters are successfully modified.

2.2.5.3 2.2.5.3 Use Case - Delete Operation Parameter

Basic Flow:

1. The occupant will select the smart device for which an operation parameter is to be deleted from a list of available devices using a browse-select file/folder mechanism similar to a MS-Windows file management client.
2. The occupant will delete the operation parameter(s) for the device.
3. The parameter will be deleted from the database.

Alternate Flow:

1. If the parameter being is critical to the functioning of the device, it will be assigned a default value and instead of being deleted from the database, this default value will be updated into the database.

Preconditions:

1. The device (software component) should already have been installed and should be included in the available list.
2. The parameter(s) being specified should have been specified earlier.

Postconditions:

1. The device parameters are successfully modified.

2.3 USER CHARACTERISTICS

In this section, we briefly discuss the user characteristics of the actors that will interact with the Safe Home Framework including the demographics and key needs of our main users. Since the Safe Home Framework is being planned, designed, developed and deployed in iterative cycles, the focus of this document is limited to the researchers and developers. We believe that these two user groups will initially gain the most benefit from using the Safe Home Framework. Similarly, the remaining stakeholders, namely the Department of Computer Science, will gain the most benefit from incorporating the needs of the researchers and developers (rather than the users) in the Safe Home Framework because they will be the ones to further advance and develop the project. Although the users were not the focus of this document, we have included some minimal user-interface functionality requirements that can be used as a base line to incorporate the future needs of the Safe Home users. In the future, however, the Safe Home Framework, and specifically the UIs to the Safe Home Framework will need to further consider and integrate the needs of the users and support personnel.

We project that the researchers and developers that will be involved with the Safe Home Framework in the near future will consist of professors and students, both graduate and undergraduate students. Typically, these users will be highly educated, technologically perceptive, familiar with an array of different computing environment and UIs and able to acquire new knowledge and skills on a short learning curve. Both researchers and developers have previously, or will in the future, expressed interest in this project and have a generally positive and optimistic outlook on the potential of the Safe Home project.

The developers that will be involved with the development of the Safe Home Framework will have Department of Computer Science graduate students as the majority. These students will typically fall within the 22-28 age group, be predominately males (roughly 70% male, 30% female), with a good working knowledge of the computer systems and programming languages involved with the development of this project. Additionally, a majority of the students will have near 24/7 access to the computers and the lab in which this project will be developed. The main motivation for the developers of the Safe Home Framework will that parts of its development will be integrated into various graduate-level classes or as a part of a paid research assistantship. We believe that it will be somewhat infrequent that professors will have a hand in programming development of the Safe Home Framework. Rather, professors will play a major role in the design phases.

The researchers, on the other hand, will most likely consist of an equal mix of graduate students and professors. Both of these groups will be highly specialized in their area of research and will be motivated by both pay as well as partaking in publishable research.

It is under these demographics of the researchers and developers that the Safe Home Framework UIs, described in the following sections, were developed.

The Safe Home Framework UIs detailed in this document is primarily tailored to suiting the needs of the near-term researchers and developers that will be involved in future advancing the Safe Home project. The primary needs of the researchers and developers include:

1. The ability to add devices and equipment to the Safe Home research lab and integrate these with minimal overhead.
2. The ability to design and conduct research experiments such as simulations, new technology incorporation, performance and usability evaluations, and algorithm development and deployment.
3. The ability to use this lab as a “proof of concept” for research and educational activities.

Again, the needs of the occupants and other stakeholders are being considered in an ancillary project.

The following table gives a summary of the key capabilities offered by the Safe Home Framework for the researcher:

Researcher Benefit	Supporting Features [Use Case Reference]
Full researcher device control UI	Researchers will have a full gamut of control over smart devices. [2.2.1, 2.2.2, 2.2.3]
Experiment replay capability UI	The Safe Home Framework will capture an experiments run data and allow the ability to recreate the experiment at a later time. [2.2.2.2]
Simulation mechanism UI	The Safe Home Framework will allow experiment to be run within the framework without actually using the smart devices. [2.2.2.2]

The following table gives a summary of other key capabilities offered by the Safe Home Framework accessible by researchers and/or other user groups:

Other User Benefits	Supporting Features [Use Case Reference]
Device Settings UI	Occupants will have partial control over the smart devices to allow them to set their preferred uses. [2.2.1.2, 2.2.5]
Administrator Access Management UI	Allows the administrator to set and control access rights to various users to the data and processes within the Safe Home Framework. [2.2.4]
Device Integerator/Upgrader UI	Provides for a flexible, scalable Safe Home Framework by allowing a developer, researcher or administrator to add, configure, trouble-shoot or remove devices through a UI. [2.2.1]

The preceding two tables give a brief conceptual look at the features of the Safe Home Framework as it applies to the different users.

2.4 CONSTRAINTS

In this section, we list the constraints placed upon the Safe Home Framework as given by Dr. Carl Chang, Dr. Johnny Wong and Dr. Simanta Mitra.

- 2.4.1 *Java Development* - The Safe Home Framework shall be developed in the Java programming language.
- 2.4.2 *Component-Object Paradigm* - The Safe Home Framework shall be designed using the component-object design paradigm.
- 2.4.3 *COTS Software* - The Safe Home project shall employ as much COTS software as possible.

2.5 ASSUMPTIONS AND DEPENDENCIES

The main features and requirements of the Safe Home Framework, as described in this document, concentrate on the needs of the researchers and developers involved with the Safe Home project. It was the assumption of this software engineering team that these groups of actors have the most pressing needs that must be captured early on in the iterative development of the Safe Home project. Thus, the features and requirements detailed below heavily concentrate on those actions that a research working within the Safe Home lab may want to take. In the following subsections, we detail the assumptions and dependencies made in this document regarding the stakeholders, software, hardware and miscellaneous associated with the Safe Home Framework UIs.

2.5.1 Stakeholder Assumptions and Dependencies

This section lists the assumptions and dependencies taken upon the authors of this document regarding the two stakeholders (researchers and developers) pertinent to the Safe Home Framework UI described here.

- 2.5.1.1 *Developers Motivation* - It is assumed that developers shall be motivated for developing the Safe Home Framework by either pay, grades (classwork) or research.
- 2.5.1.2 *Windows NT Operating System Experience* - It is assumed that researchers and developers will previous experience or training with the Windows NT operating system.
- 2.5.1.3 *Real-Time Operating System Experience* - It is assumed that researchers and developers will previous experience or training with the WindReveis real-time operating system.
- 2.5.1.4 *TinyOS Experience* - It is assumed that researchers and developers will previous experience or training with the TinyOS operating system.
- 2.5.1.5 *Web-Based Interface Experience* - It is assumed that researchers and developers will previous experience or training with using web-based interfaces.
- 2.5.1.6 *Web-Based Programming* - It is assumed that developers will previous experience or training with the web-based, Java interface programming such as JSP, Java Beans, etc.
- 2.5.1.7 *Hierarchy of Roles* - It is assumed that a hierarchy of roles will be defined external to this document.
- 2.5.1.8 *Requirements Prioritization* - It is assumed that the requirements listed in this document will be prioritized based on inputs from all primary stakeholders and resources available. It is essential that this activity be conducted prior to commencement of the design stage.
- 2.5.1.9 *Stakeholder Priority* - It is assumed that the researchers and developers of the Safe Home project shall currently have the most pressing needs for the Safe Home Framework as seen by Dr. Carl Chang and the Department of Computer Science at Iowa State University.
- 2.5.1.10 *Developers/Users Skills* - It is assumed that the researchers and developers shall have a minimal technological skill set of a graduating computer science undergraduate.

2.5.2 Software Assumptions and Dependencies

This section lists the assumptions and dependencies taken upon the authors of this document regarding the software pertinent to the Safe Home Framework UI described here.

2.5.2.1 *Web Browser* - It is assumed that all computers employing a Safe Home Framework UI shall use Internet Explorer, Netscape Navigator or Mozilla Firefox as their web browser.

2.5.2.2 *Web Browser Versioning* - It is assumed that all computers employing a Safe Home Framework UI shall have current (i.e. released within the last year as of December 2004) web browsers of the browsers listed in 2.5.2.1.

2.5.2.3 *JVM* - It is assumed that all computers employing a Safe Home Framework UI shall have JVM correctly installed and working properly.

2.5.2.4 *JVM Versioning* - It is assumed that all computers employing a Safe Home Framework UI shall have current (i.e. released within the last year as of December 2004) version of JVM.

2.5.2.5 *Real-Time Java Framework* - The Safe Home Framework is dependent on the existence of a Java Framework that shall support the real-time requirements of the UIs of the system.

2.5.2.6 *COTS-JAVA Compatability* - It is a dependency that all COTS software used for the design and development of the Safe Home Framework shall be Java compatible.

2.5.2.7 *COTS-CORBA Compatibility* - It is a dependency that all COTS software used for the design and development of the Safe Home Framework shall be CORBA compatible.

2.5.3 Hardware Assumptions and Dependencies

This section lists the assumptions and dependencies taken upon the authors of this document regarding the hardware pertinent to the Safe Home Framework UI described here.

- 2.5.3.1 *Smart Device Communication* - It is assumed that all smart devices shall be able to communicate to application code via an X10 or similar controller or through a device driver.
- 2.5.3.2 *Smart Device Communication* - It is assumed that all smart devices shall be network-enabled to have the ability to communicate via a network.
- 2.5.3.3 *Web-enabled Communication* - It is assumed that all computers employing a Safe Home Framework UI are web-enabled.

3 Specific Requirements

This section details specific requirements relating to the external interface, features, performance, design constraints, system attributes and other attributes of the Safe Home Framework UIs. We organize a major portion of the requirements in this section by feature (rather than class). Within each feature, we further partition the requirements by subfeatures.

3.1 EXTERNAL INTERFACE REQUIREMENTS

This section discusses the details of the user interfaces for the proposed Safe Home Framework UIs. Please note, Section 2.1.2 of this SRS document displayed a handful of screenshots of the Safe Home Framework UIs. However, these screenshots only provide examples of a potential interface display to give an appropriate product perspective. These prototypes lacked the detailed interface requirements and should only be used as guidelines.

3.1.1 User Interfaces

This section lists the external UI requirements placed upon the external systems interacting with the Safe Home Framework UI.

3.1.1.1 *Interface Access* - All interfaces shall require a password to access them.

3.1.1.2 *Interface Functionality* - The interfaces shall present different functionality to different users depending on the type of user currently logged in.

3.1.1.3 *Interface User Identification* - All interfaces shall indicate what type of user is logged in at all times.

3.1.1.4 *Web Accessibility* - All UIs shall be web accessible.

3.1.1.5 *Communication Accessibility* - All UIs shall be able to receive data from the Safe Home Framework.

3.1.1.6 *Tab-based Navigation* - All interfaces shall utilize a tabbed navigational system (as shown in the screenshots illustrated in Section 2.1.2).

3.1.1.7 *Two-degree Navigation* - All functions within the Safe Home system shall be at most 2 clicks away from any screen within the system.

3.1.1.8 *Interface Uniformity* - The interfaces shall be uniform in presentation, color scheme, presentation style, etc. as determined by the Dr. Carl Chang and Dr Johnny Wong.

3.1.2 Hardware Interfaces

This section lists the external hardware interface requirements placed upon the external systems interacting with the Safe Home Framework UI.

- 3.1.2.1 *Primary Input Device* - The primary input device for the Safe Home Framework UIs shall be a keyboard.
- 3.1.2.2 *Primary Input Device* - The primary input device for the Safe Home Framework UIs shall be a mouse.
- 3.1.2.3 *Primary Output Device* - The primary output device for the Safe Home Framework UIs shall be a computer monitor.
- 3.1.2.4 *Hardware Communication* - All external hardware shall have a means of connecting and communicating with the Safe Home Framework.
- 3.1.2.5 *Hardware Connection* - All external hardware shall have a standard connection such as USB port, serial port, etc.
- 3.1.2.6 *Hardware Interface Support* - All external hardware shall have tech support or a manual to help with the interface.
- 3.1.2.7 *Hardware Data Connection Rate* - All external hardware shall have a data connection rate equal to or as fast as a serial connection.

3.1.3 Software Interfaces

This section lists the external software interfaces requirements placed upon the external systems interacting with the Safe Home Framework UI.

- 3.1.3.1 The Safe Home Framework UIs shall communicate with a database system to send, receive and store data.

3.1.4 Communications Interfaces

This section lists the external communication interface requirements placed upon the external systems interacting with the Safe Home Framework UI.

3.1.4.1 The Safe Home Framework UIs shall communicate via the internet using the HTTP and TCP/IP communication protocols.

3.2 FEATURES

This section details the requirements of the main features in the proposed Safe Home Framework UIs.

3.2.1 Feature - Device Integrator/Upgrader UI

This section details the needed requirements for the Device Integrator/Upgrader UI. The Device Integrator/Upgrader UI consists of three two subfeatures relevant to the UI: an API for communicating in a variety of ways and device management facilities. The requirements for these subfeatures comprising the Device Integrator/Upgrader UI feature are detailed in the following subsections.

3.2.1.1 Device Integrator/Upgrader UI – API Communicating in a Variety of Ways

3.2.1.1.1. *Smart Device Communication View* - The UI shall have a screen to be able to view all smart devices and their communication medium (USB, wireless, wired, etc.) to the Safe Home Framework.

3.2.1.1.2. *Smart Device Communication View Sorting* - The UI shall have the ability to sort the smart devices and their communication medium to the Safe Home Framework by medium type (USB, wireless, wired, etc.) in ascending or descending order.

3.2.1.1.3. *Smart Device Communication View Filtering* - The UI shall have the ability to filter the smart devices and their communication medium to the Safe Home Framework by medium type (USB, wireless, wired, etc.).

3.2.1.1.4. *Smart Device Communication View Sorting* - The UI shall have the ability to sort the smart devices and their communication medium to the Safe Home Framework by smart device name (USB, wireless, wired, etc.) in ascending or descending order.

3.2.1.1.5. *Communication Data Rate View* - The UI shall have the ability to display the average data communication rate for each smart device and their communication medium (USB, wireless, wired, etc.).

3.2.1.1.6. *Communication Data Rate View Sorting* - The UI shall have the ability to sort the smart devices and their communication medium to the Safe Home Framework by their average communication data rate in ascending or descending order.

- 3.2.1.1.7. *Smart Device Communication View Filtering* - The UI shall have the ability to filter the smart devices and their communication medium to the Safe Home Framework by their average communication data rate.
- 3.2.1.2 Device Integrator/Upgrader UI – Device Management Facilities
- 3.2.1.2.1. *Browsing Device Drivers* - The UI shall provide a mechanism to allow a user to browse the current device driver (name, version number and date installed) for each smart device within the Safe Home environment that is currently using a device driver.
- 3.2.1.2.2. *Browsing Device Drivers View* - The screen to browse the all device drivers shall list the smart device in alphabetical order paired with its associated device driver.
- 3.2.1.2.3. *Adding Device Drivers* - The UI shall allow for the addition of device driver for smart devices.
- 3.2.1.2.4. *Removing Device Drivers* - The UI shall allow for the removal a of device driver.
- 3.2.1.2.5. *Removing Device Drivers View* - The UI shall provide a mechanism that will allow a user to select the device (software component) to be removed using a browse-select file/folder mechanism (similar to a MS-Windows file management client).
- 3.2.1.2.6. *Configuring Device Drivers* - The UI shall allow for configuring a of device driver.
- 3.2.1.2.7. *Configuring Device Drivers View* - The UI shall provide a mechanism that will allow a user to select the device (software component) whose configuration is to be modified using a browse-select file/folder mechanism (similar to a MS-Windows file management client).
- 3.2.1.2.8. *Selecting Device Drivers* - The UI shall have a mechanism to select which device driver should be used for each smart device within the Safe Home environment.
- 3.2.1.2.9. *Selecting Device Drivers View* - The screen to select a device driver shall list the possible device drivers to choose from in alphabetical order by device driver name.
- 3.2.1.2.10. *Trouble-Shooting Device Drivers* - The UI shall have an interface to be able to trouble-shoot device drivers.
- 3.2.1.2.11. *Trouble-Shooting Device Drivers View* - The UI shall provide a mechanism that will allow a user to select the device (software component) that needs to be fixed using a browse-select file/folder mechanism (similar to a MS-Windows file management client).
- 3.2.1.2.12. *Trouble-Shooting Device Drivers View* - The screen to display the device driver trouble-shooting shall display a status for each smart device indicating if it is enabled or disabled.

3.2.1.2.13. *Trouble-Shooting Device Drivers View* - The screen to display the device driver trouble-shooting shall display a status for each smart device indicating if it is working properly or not working properly.

3.2.2 Feature - Research UI

This section details the needed requirements for the Research UI. The Research UI consists of three main subfeatures relevant to the UI: control routines, independent execution and experiment replay and simulation. The requirements for these subfeatures comprising the Research UI feature are detailed in the following subsections.

3.2.2.1 Research UI – Control Routines

3.2.2.2.1. *Control Routines* - The platform shall provide a series of control routines for the researchers to establish their research infrastructure and manage the research projects.

3.2.2.2.2. *Device Navigation and Selection* – The user shall be able to browse the existing devices and choose the device he/she wants and create the control routines for the device.

3.2.2.2.3. *Browsing and Controlling of the Currently Executing Single and Group Control Routines* - The platform shall provide an interface to let the user to view and control the currently executing routings or groups.

3.2.2.2.4. *Single Control Routines Creation* – The user shall create a new control routine and define its properties, for example, routing ID, routing input, routing output.

3.2.2.2.5. *Group Control Routines Creation* – The user shall create a new group of control routines and define its properties, for example, routing group ID, routing group components.

3.2.2.2.6. *Update Single and Group Control Routine Properties* - The user shall be able to update the properties for both single control routine and group control routines. For instance, the use shall adjust the single routines for group routine.

3.2.2.2.7. *Management of the Currently Executing Control Routines or Groups* - The platform shall provide an interface to let the user to reschedule the currently executing routings or groups.

3.2.2.2.8. *Independent of the Concurrent Execution* - The platform shall provide separate execution and interface for different groups of control routines

3.2.2.2 Research UI – Independent Execution

- 3.2.2.2.1. *UI Independence* – The platform shall provide separate user interface for the user to create, update, and browse of their projects. Interface interventions among different users are allowed.

3.2.2.3 Research UI – Replay and Simulation Mechanisms

- 3.2.2.3.1. *Log Storage* – The platform shall keep all the log information for each researcher, including the system context information. The platform shall provide a user interface to setup the log storage options.
- 3.2.2.3.2. *Offline Capacity* – The platform shall allow the user to use his/her user interfaces in the offline statuses.
- 3.2.2.3.3. *Simulation Capacity* – The platform shall provide user interface to allow the user to run the simulation for all the designated devices.
- 3.2.2.3.4. *Data Access Capacity* – The platform shall provide simple user interface, for example, command line interface, to access the data he/she want.

3.2.3 Feature - Application Development UI

This section details the needed requirements for the Application Development UI. The Application Development UI consists of two main subfeatures relevant to the UI: creation of new applications and compatibility. The requirements for these subfeatures comprising the Application Developments UI feature are detailed in the following subsections.

3.2.3.1 Application Development UI – Creation of New Applications

- 3.2.3.1.1. *Application Access Control* - The UI shall only show the applications that the user has created or has access predefined read/write access rights.
- 3.2.3.1.2. *Application Control* - The UI shall provide a mechanism to allow a user to start and stop an application.
- 3.2.3.1.3. *Application Control View* - The UI shall provide a screen to view all currently running applications for only that user.
- 3.2.3.1.4. *Application Control View* - The UI shall provide a screen to view the time duration of all currently running applications for that user.

- 3.2.3.1.5. *Adding an Application* - The UI shall provide a mechanism to add an application to the Safe Home Framework.
- 3.2.3.1.6. *Adding an Application View* - The adding an application screen within the UI shall allow a user to select the application to be added using a browse-select file/folder mechanism similar to a MS-Windows file management client.
- 3.2.3.1.7. *Updating an Application* - The UI shall provide a mechanism to update an application to the Safe Home Framework.
- 3.2.3.1.8. *Updating an Application View* - The updating an application screen within the UI shall allow a user to select the application to be updated using a browse-select file/folder mechanism similar to a MS-Windows file management client.
- 3.2.3.1.9. *Removing an Application* - The UI shall provide a mechanism to remove an application to the Safe Home Framework.
- 3.2.3.1.10. *Removing an Application View* - The removing an application screen within the UI shall allow a user to select the application to be removed using a browse-select file/folder mechanism similar to a MS-Windows file management client.
- 3.2.3.1.11. *Archiving an Application* - The UI shall provide a mechanism to archive an application to the Safe Home Framework.
- 3.2.3.1.12. *Archiving an Application View* - The archiving an application screen within the UI shall allow a user to select the application to be archived using a browse-select file/folder mechanism similar to a MS-Windows file management client.
- 3.2.3.1.13. *Sharing an Application* - The UI shall provide a mechanism to share an application, tested components or UIs that have been developed independently with other users of the Safe Home Framework.
- 3.2.3.1.14. *Sharing an Application View* - The sharing an application screen within the UI shall allow a user to select the application, tested component or UI to be shared using a browse-select file/folder mechanism similar to a MS-Windows file management client and specify to which user(s) to share with.

3.2.3.2 Application Development UI – Compatibility

- 3.2.3.2.1. *Data Interchange* - The UI shall provide a mechanism to define how data shall be stored in the database in a universal data format (CSV, XML, etc.).
- 3.2.3.2.2. *Device Output Data* - The UI shall provide a mechanism to define how to output device data in a universal data format (CSV, XML, etc.).
- 3.2.3.2.3. *Device Output Data View* - The UI shall provide a mechanism to view how output data from a device is being stored and produced (CSV, XML, etc.).

3.2.4 Feature - Administration UI

This section details the needed requirements for the Administration UI. The Administration UI consists of three main subfeatures: access control, user settings and logging. The requirements for these subfeatures comprising the Administration UI feature are detailed in the following subsections.

3.2.4.1 Administration UI – Access Control

- 3.2.4.1.1. *Defining Access Control* - The UI shall provide a mechanism to assign each user, or groups of users, access rights (similar to windows based admin, power user, user, guest, etc.).
- 3.2.4.1.2. *Layered Access Control* - The Safe Home Framework shall define layers of access rights to types of users for access to data, applications and devices.
- 3.2.4.1.3. *Device Access Control* - The Safe Home Framework shall have an access level assigned to adding a smart device.
- 3.2.4.1.4. *Device Access Control* - The Safe Home Framework shall have an access level assigned to removing a smart device.
- 3.2.4.1.5. *Device Access Control* - The Safe Home Framework shall have an access level assigned to configuring a smart device.
- 3.2.4.1.6. *Application Access Control* - The Safe Home Framework shall have an access level assigned to adding an application.
- 3.2.4.1.7. *Application Access Control* - The Safe Home Framework shall have an access level assigned to removing an application.
- 3.2.4.1.8. *Application Access Control* - The Safe Home Framework shall have an access level assigned to configuring an application.

- 3.2.4.1.9. *Application Access Control* - The Safe Home Framework shall have an access level assigned to archiving an application.
- 3.2.4.1.10. *Access Rights* - The UI shall allow a user to define the functions (in terms of read/write/executable access to data and applications) a given user of a certain access privilege available to them.
- 3.2.4.1.11. *Access Rights View* - The UI shall provide a screen to search a user by name or login to view their access rights.
- 3.2.4.1.12. *Access Rights View* - The UI shall provide a screen to list all users and access rights pairs.
- 3.2.4.1.13. *Access Rights View* - The UI shall provide a screen to sort all users and access rights pairs by either username or access rights in ascending or descending order.
- 3.2.4.1.14. *Access Rights View* - The UI shall provide a screen to filter users and access rights pairs by the type of access rights.

3.2.4.2 Administration UI – User Settings

- 3.2.4.2.1. *User Management* – The platform shall provide user interface for create new users.
- 3.2.4.2.2. *User Management* – The platform shall provide user interface to update existing user’s information (for example, user configurations).
- 3.2.4.2.3. *User Management* – The platform shall provide user interface to delete existing users.
- 3.2.4.2.4. *User Management View* - The UI shall provide a screen to search a user by name or login to view their user settings.
- 3.2.4.2.5. *User Management View* - The UI shall provide a screen to list all users and user settings.
- 3.2.4.2.6. *User Management View* - The UI shall provide a screen to sort all users and access rights pairs by either username in ascending or descending order.
- 3.2.4.2.7. *User Management View* - The UI shall provide a screen to filter users and access rights pairs by the type of user settings.

3.2.4.3 Administration UI – Logging

- 3.2.4.3.1. *Logging Control* - The UI shall provide the ability to turn on/off all logging functions.
- 3.2.4.3.2. *Logging Control* - The UI shall provide the ability to turn on/off all logging functions for a particular user or group of users.
- 3.2.4.3.3. *Logging Settings* - The UI shall provide the ability to set the logging settings.

3.2.4.3.4. *Archiving Logs View* - The administrator shall be able to select the user(s) whose records are to be archived using a browse-select file/folder mechanism similar to a MS-Windows file management client.

3.2.5 Feature - Device Operation UI

This section details the needed requirements for the Administration UI. The Administration UI consists of one main subfeature: change settings. The requirements for this subfeature comprising the Device Operation UI feature are detailed in the following subsections.

3.2.5.1 Device Operation UI - Change Settings

3.2.5.1.1. *Changing Device Settings* - The UI shall provide a mechanism to override/change settings for executing control routines on a smart device.

3.2.5.1.2. *Changing Grouped Device Settings* - The UI shall provide a mechanism to override/change settings for executing control routines on a predefined group of smart device.

3.2.5.2 Device Operation UI – Emergencies

3.2.5.2.1. The UI shall flash "Emergency" whenever an emergency event has occurred within the Safe Home Environment.

3.2.5.2.2. The UI shall provide an Emergency override to cease the flashing "Emergency" in the event of a false alarm or a non-emergency.

3.2.5.3 Device Operation UI – Failure

3.2.5.3.1. The UI shall give a warning/indication whenever a failure has occurred in any of the smart devices.

3.3 PERFORMANCE REQUIREMENTS

This section lists the performance requirements placed upon the Safe Home Framework UI.

3.3.1 *UI Transitions* - The transition between any Safe Home Framework UI screens shall not take any longer than 5 seconds under normal network traffic conditions.

3.3.2 *UI Screen Loading* - All screens of the Safe Home Framework shall load within 10 seconds.

3.4 DESIGN CONSTRAINTS

This section lists the design constraints placed upon the Safe Home Framework UI.

- 3.4.1 *Java Development Constraint* - The Safe Home Framework shall be developed in the Java programming language (see also 2.4.1.1).
- 3.4.2 *Component-Object Paradigm Constraint* - The Safe Home Framework shall be designed using the component-object design paradigm (see also 2.4.1.2).
- 3.4.3 *COTS Software Constraint* - The Safe Home project shall employ as much COTS software as possible (see also 2.2.1.3).
- 3.4.4 *Real-Time Functionality Constraint* - The Safe Home Framework shall have the ability to handle real-time operating systems.
- 3.4.5 *Windows NT Operating System Constraint* - The Safe Home Framework shall use the Windows NT operating system as one of its target operating systems.
- 3.4.6 *Web-Based UI Constraint* - The Safe Home Framework shall use a web-based UI for the Safe Home Framework UIs.

3.5 SOFTWARE SYSTEM ATTRIBUTES

This section describes various software system attributes including attributes pertaining to reliability, availability, security, maintainability and portability placed upon the Safe Home Framework UIs described in this document.

3.5.1 Reliability

This section lists the reliability requirements desired of the Safe Home Framework UI.

- 3.5.1.1 *Mean Time Between Failures* - The Safe system shall not, on average, fail more than once per every two weeks.
- 3.5.1.2 *Simulation Data Accuracy* - The experiment simulation feature within the Research UI shall accurately simulate the smart devices for all experiments (i.e. the simulated smart devices shall act exactly as the actual smart devices).
- 3.5.1.3 *Replay Data Accuracy* - The experiment replay feature within the Research UI shall accurately replay the stored experiments data and actions (i.e. the experiment replay shall exactly mirror the original experiment).
- 3.5.1.4 *User Setting Accuracy* - The specified user settings feature within the Administration UI shall accurately represent the current user settings for the currently selected user.
- 3.5.1.5 *Log Accuracy* - The Safe Home Framework's logging feature within the Administration UI shall accurately represent the actions taken by each user.

3.5.2 Availability

This section lists the availability requirements desired of the Safe Home Framework UI.

- 3.5.2.1 *System Maintenance Notification* - In the event that the Safe Home Framework or the Safe Home Framework UIS will undergo foreknown system maintenance lasting longer than 1 hour, all users of the system shall be notified via email.
- 3.5.2.2 *UI Availability* - The Device Integrator/Upgrader UI, Research UI, Application Development UI and Administration UI shall be available 90% of the time unless previously announced for scheduled maintenance or backup.
- 3.5.2.3 *Device Operation UI Availability* - The Device Operation UI shall be available 95% of the time unless previously announced for scheduled maintenance or backup.
- 3.5.2.4 *Degraded Mode of Operations* - In the event of the unavailability of any UI within the Safe Home Framework, the system users shall see a temporary unavailable screen in place of the UI detailing the reason why the system is down and the estimated time to repair.

3.5.3 Security

This section lists the security requirements desired of the Safe Home Framework UI.

- 3.5.3.1 *User Login Security* - All users logging into the Safe Home Framework shall use a username and password combination to gain access to any Safe Home Framework UI.
- 3.5.3.2 *Account Access Security* - Each account within the Safe Home Framework system shall only be accessible to the associated user.
- 3.5.3.3 *Password Security* - Passwords used by users to access the Safe Home Framework shall be hidden to all other users of the system.
- 3.5.3.4 *Password Security* - Passwords shall be stored within a secure database.
- 3.5.3.5 *Password Security* - Passwords shall be stored in an encrypted form.
- 3.5.3.6 *Password Security* - Passwords shall be sent over a network in an encrypted form.
- 3.5.3.7 *User Data Access* - After logging in, users shall only be able to access their own experiments and data.

3.5.4 Maintainability

This section lists the maintainability requirements desired of the Safe Home Framework UI.

- 3.5.4.1 *Safe Home Framework Code Maintainability* - The programming code developed for the Safe Home Framework application shall conform to programming style standards and shall be thoroughly commented as determined by a code review team.
- 3.5.4.2 *Smart Device Maintainability* - A researcher, developer and administrator shall be able to add new smart devices to the Safe Home environment via the Device Integrator/Upgrader UI.
- 3.5.4.3 *Smart Device Maintainability* - A researcher, developer and administrator shall be able to remove smart devices to the Safe Home environment via the Device Integrator/Upgrader UI.
- 3.5.4.4 *Smart Device Maintainability* - A researcher, developer and administrator shall be able to configure devices to the Safe Home environment via the Device Integrator/Upgrader UI.
- 3.5.4.5 *Smart Device Maintainability* - A researcher, developer and administrator shall be able to trouble-shoot smart devices to the Safe Home environment via the Device Integrator/Upgrader UI.
- 3.5.4.6 *Control Routine Maintainability* - A researcher, developer and administrator shall be able to add new control routines for the smart devices of the Safe Home environment via the Research UI.
- 3.5.4.7 *Grouped Control Routine Maintainability* - A researcher, developer and administrator shall be able to add new control routines for group control of smart devices of the Safe Home environment via the Research UI.

- 3.5.4.8 *Control Routine Settings Maintainability* - A researcher, developer and administrator shall be able to adjust the control routines for the smart devices of the Safe Home environment via the Research UI.
- 3.5.4.9 *Grouped Control Routine Settings Maintainability* - A researcher, developer and administrator shall be able to adjust the control routines for defined groups of smart devices of the Safe Home environment via the Research UI.
- 3.5.4.10 *Control Routine Scheduling Maintainability* - A researcher, developer and administrator shall be able to schedule control routines for the smart devices of the Safe Home environment via the Research UI.
- 3.5.4.11 *Grouped Control Routine Scheduling Maintainability* - A researcher, developer and administrator shall be able to schedule control routines for grouped smart devices of the Safe Home environment via the Research UI.
- 3.5.4.12 *Replay Experiments Maintainability* - A researcher, developer and administrator shall be able to delete replayed experiments via the Research UI.
- 3.5.4.13 *Access Control Maintainability* - An administrator shall be able to assign access rights to users of the Safe Home Framework via the Administration UI.
- 3.5.4.14 *Access Control Maintainability* - An administrator shall be able to assign different levels of access rights (such as administrators, students, professors, etc.) to users of the Safe Home Framework and specify the system rights they have via the Administration UI.
- 3.5.4.15 *Access Control Maintainability* - An administrator shall be able to assign different rights to users for adding devices to the Safe Home environment via the Administration UI.
- 3.5.4.16 *Access Control Maintainability* - An administrator shall be able to assign different rights to users for adding applications to the Safe Home environment via the Administration UI.
- 3.5.4.17 *Access Control Maintainability* - An administrator shall be able to assign different rights to users for adding control routines to the Safe Home Framework via the Administration UI.

- 3.5.4.18 *Access Control Maintainability* - An administrator shall be able to assign different rights to users for experiments to the Safe Home Framework via the Administration UI.
- 3.5.4.19 *User Settings Maintainability* - An administrator shall be able to create user settings via the Administration UI.
- 3.5.4.20 *User Settings Maintainability* - An administrator shall be able to configure user settings via the Administration UI.
- 3.5.4.21 *User Settings Maintainability* - An administrator shall be able to delete user settings via the Administration UI.
- 3.5.4.22 *Log Maintainability* - An administrator shall be able to adjust the log settings via the Administration UI.
- 3.5.4.23 *Occupant Device Settings Maintainability* - An occupant shall be able to override/change settings for executing control routines on the smart devices within the Safe Home environment via the Device Operation UI.
- 3.5.4.24 *User Settings Maintainability* - An administrator shall be able to create user settings via the Administration UI.

3.5.5 Portability

This section lists the portability requirements desired of the Safe Home Framework UI.

3.5.5.1 *UI Web Browser Portability* - All Safe Home Framework UIs shall be portable to, at a minimum, the following web browsers: Internet Explorer, Mozilla Firefox and Netscape Navigator.

3.5.5.2 *UI Web Browser Version Portability* - All Safe Home Framework UIs shall be portable to, at a minimum, the web browsers listed in 3.5.5.1 with version release dates within the last year as of December 2004.

3.6 OTHER REQUIREMENTS

[None]