

# **Software Requirements Specifications Smart Home Integration Technology**

SkyNet Inc.

# Table of Contents

<b>Revision History</b> .....	6
<b>1. Introduction</b> .....	7
1.1. Purpose.....	7
1.2. Project Scope.....	7
1.3. Glossary of Terms.....	7
1.4. Overview.....	8
<b>2. Overall Description</b> .....	8
2.1. Product Perspective.....	8
2.2. Product Features.....	9
2.3. User Classes and Characteristics.....	10
<b>2.3.1. Owner</b> .....	<b>11</b>
<b>2.3.2. Registered User</b> .....	<b>11</b>
2.4. Operating Environment.....	12
<b>2.4.1. Mobile Environment</b> .....	<b>12</b>
<b>2.4.2. Home Integration System</b> .....	<b>12</b>
2.5. Design and Implementation Constraints.....	12
2.6. Assumptions and Dependencies.....	13
<b>2.6.1. Assumptions</b> .....	<b>13</b>
<b>2.6.2. Dependencies</b> .....	<b>13</b>
<b>3. System Features</b> .....	13
3.1. Login.....	13
<b>3.1.1. Description and Priority</b> .....	<b>13</b>
<b>3.1.2. Stimulus/Response Sequences</b> .....	<b>14</b>
<b>3.1.3. Functional Requirements</b> .....	<b>14</b>
<b>3.1.3.1. Prevent Unauthorized Access</b> .....	<b>14</b>
<b>3.1.3.2. Multi-Step Authentication</b> .....	<b>14</b>
<b>3.1.3.3. Log and Alert – Logins from New Device</b> .....	<b>14</b>
<b>3.1.3.4. Log and Alert – Logins from Largely Different Locations</b> .....	<b>14</b>
3.2. Add a New Device.....	15
<b>3.2.1. Description and Priority</b> .....	<b>15</b>
<b>3.2.2. Stimulus/Response Sequences</b> .....	<b>15</b>
<b>3.2.3. Functional Requirements</b> .....	<b>15</b>
<b>3.2.3.1. Prevent Non-Privileged Users from Adding Devices</b> .....	<b>15</b>
<b>3.2.3.2. Prevent Unrecognized Devices from Being Added</b> .....	<b>15</b>
<b>3.2.3.3. Failsafe for Connection Issues when Adding a Device</b> .....	<b>15</b>

3.3. Remove a Device.....	16
<b>3.3.1. Description and Priority.....</b>	<b>16</b>
<b>3.3.2. Stimulus/Response Sequences.....</b>	<b>16</b>
<b>3.3.3. Functional Requirements.....</b>	<b>16</b>
<b>3.3.3.1. Prevent Non-Privileged Users from Removing Devices.....</b>	<b>16</b>
<b>3.3.3.2. Failsafe for Connection Issues when Removing a Device.....</b>	<b>16</b>
3.4. Configure a Device.....	16
<b>3.4.1. Description and Priority.....</b>	<b>16</b>
<b>3.4.2. Stimulus/Response Sequences.....</b>	<b>17</b>
<b>3.4.3. Functional Requirements.....</b>	<b>17</b>
<b>3.4.3.1. Prevent Non-Privileged Users from Configuring Devices.....</b>	<b>17</b>
<b>3.4.3.2. Validation of Devices' Configuration Settings.....</b>	<b>17</b>
<b>3.4.3.3. Configuration of Powered Off or Unreachable Devices.....</b>	<b>17</b>
3.5. Configure the System.....	17
<b>3.5.1. Description and Priority.....</b>	<b>17</b>
<b>3.5.2. Stimulus/Response Sequences.....</b>	<b>18</b>
<b>3.5.3. Functional Requirements.....</b>	<b>18</b>
<b>3.5.3.1. Prevent Non-Privileged Users from Configuring the System.....</b>	<b>18</b>
<b>3.5.3.2. Emergency Updating of System Configuration.....</b>	<b>18</b>
3.6. Add a New User.....	18
<b>3.6.1. Description and Priority.....</b>	<b>18</b>
<b>3.6.2. Stimulus/Response Sequences.....</b>	<b>18</b>
<b>3.6.3. Functional Requirements.....</b>	<b>19</b>
<b>3.6.3.1. Invitation Expiration.....</b>	<b>19</b>
<b>3.6.3.2. Granting of Permissions.....</b>	<b>19</b>
<b>3.6.3.3. Enabling of Location Services.....</b>	<b>19</b>
3.7. Manage Privileges.....	19
<b>3.7.1. Description and Priority.....</b>	<b>19</b>
<b>3.7.2. Stimulus/Response Sequences.....</b>	<b>19</b>
<b>3.7.3. Functional Requirements.....</b>	<b>19</b>
<b>3.7.3.1. Adding or Removing Privileges.....</b>	<b>19</b>
<b>3.7.3.2. Removing Users.....</b>	<b>20</b>
3.8. Update Software.....	20
<b>3.8.1. Description and Priority.....</b>	<b>20</b>
<b>3.8.2. Stimulus/Response Sequences.....</b>	<b>20</b>
<b>3.8.3. Functional Requirements.....</b>	<b>20</b>
<b>3.8.3.1. Opting Out of Updates.....</b>	<b>20</b>
<b>3.8.3.2. Running on Out of Date Software.....</b>	<b>20</b>
<b>3.8.3.3. Emergency Updates.....</b>	<b>21</b>
3.9. Machine Learning.....	21

3.9.1. Description and Priority.....	21
3.9.2. Stimulus/Response Sequences.....	21
3.9.3. Functional Requirements.....	21
3.9.3.1. Collection of Data.....	21
3.9.3.2. Mining of Data.....	22
3.9.3.3. Automation Based on Recognized Patterns.....	22
4. Use Cases.....	22
4.1. Use Case Diagram.....	22
4.2. Use Case Documents.....	23
4.2.1. Login.....	23
4.2.2. Add a New Device.....	23
4.2.3. Remove a Device.....	25
4.2.4. Configure a Device.....	25
4.2.5. Configure the System.....	26
4.2.6. Add a New User.....	27
4.2.7. Manage Privileges.....	27
4.2.8. Update Software.....	28
5. External Interface Requirements.....	29
5.1. Hardware Interfaces.....	29
5.1.1. Lazy Bot Devices.....	29
5.1.2. Communication Protocols.....	29
5.2. Software Interfaces.....	29
5.2.1. External API.....	29
5.2.2. External Cloud Services.....	30
5.2.3. Software Development.....	30
5.2.4. Tools.....	30
5.2.5. Data Encryption.....	30
5.3. Communications Interfaces.....	30
5.3.1. Communication.....	30
6. Other Non-Functional Requirements.....	30
6.1. Performance Requirements.....	31
6.1.1. Low Power Mode.....	31
6.1.2. Response Time Limitation.....	31
6.1.3. Power Constraint Reached.....	31
6.2. Safety Requirements.....	31
6.2.1. Smart Sensors.....	31
6.2.2. Smart Security Systems.....	32

6.3. Security Requirements .....	32
6.4. Software Quality Attributes .....	33
<b>6.4.1. Availability</b> .....	<b>33</b>
<b>6.4.2. Interoperability</b> .....	<b>33</b>
<b>6.4.3. Usability</b> .....	<b>33</b>
<b>6.4.4. Testability</b> .....	<b>33</b>
<b>6.4.5. Robustness</b> .....	<b>33</b>
<b>6.4.6. Flexibility</b> .....	<b>33</b>
<b>6.4.7. Maintainability</b> .....	<b>33</b>

## Revision History

Name	Date	Reason for Changes	Version
SkyNet Inc.	Jan. 24, 2017	Initial Specification	1.00
Rickus Senekal	Jan. 25, 2017	Section Additions	1.01
Michael Demone	Jan. 25, 2017	Added basic features and users	1.02

# 1. Introduction

## 1.1. Purpose

The product covered in this requirements specification document is the Lazy Bot home integration system. This specification can be applied to the entire Lazy Bot home integration system and is not limited to a single feature. Upon product release, the version number corresponding to this specification will be 1.0.

## 1.2. Project Scope

SkyNet Inc. has been tasked with the creation of a fully integrated software system that allows for the automation of processes within homes or businesses. Objectives of the software system include switching lighting systems on and off, locking doors when a user leaves a defined radial boundary, controlling entertainment systems, and controlling compatible household appliances. The software system will interface with mobile devices, allowing users more accessibility and physical freedom than offered by traditional appliance interfaces. Deep learning algorithms developed by our engineers will allow the system to learn and adapt to routines and trends performed by the user. Along with a decrease in monthly spending due to more efficient device power consumption, Lazy Bot's corporate goals also include user accessibility and compatibility.

## 1.3. Glossary of Terms

Term	Description
API	Application program interface. A program interface that provides a set of subroutine definitions, protocols, and tools used for use by other software systems or components.
Appliances	Any hardware that the Lazy Bot home integration can control. Examples are: Door locks, entertainment systems, home appliances, and lighting systems.
AWS/EC2	Amazon Web Services, a cloud services platform. The Elastic Compute Cloud.
Deep Learning	A subfield of machine learning that uses large neural networks to model high level abstractions in data.
Gradle	A Build Automation tool.

GUI	A graphical user interface that allows users to visually navigate the application through graphical icons.
HTTP(S)	HyperText Transfer Protocol (Secure), is a standard protocol used for communication.
Jenkins	Allows for any code changes to be delivered to the server in a seamless process.
Repository	Used for code version control, stores a specific code base.
Smart Device	A device that is connected to a network or other devices via wired or wireless protocols.
SSL	Secure Socket Layer, an extra layer added underneath HTTP to encrypt communication.

Table 1. A list of all glossary terms and definitions

## 1.4. Overview

This requirements specification details the features and requirements of the Lazy Bot home integration system. Section 1 introduces the purpose, project scope, glossary of terms, and overview. Section 2 is a high level description of the entire system, including product perspective, product features, user classes and characteristics, operating environment, constraints, and assumptions and dependencies. Section 3 specifies and describes the essential features of the system and details their stimulus sequences and requirements. Section 4 contains external interface requirements including user interfaces, hardware/software interfaces, and communication interfaces. Lastly, sections 5 and 6 outline the remaining non-functional requirements such as performance, safety, security, and software quality attributes.

## 2. Overall Description

### 2.1. Product Perspective

The Lazy Bot home integration system is a new, self-contained system which will integrate all of the products on Lazy Bot's current smart home appliance line, including third-party acquired smart appliances. The system consists of two main components: the mobile app and the home integration system. Connected devices will communicate over Wi-Fi using encrypted protocols to the home integration system. The overall system will interact with the devices in three ways:



- Controlling the devices based on events from the system's machine learning component.
- Allowing privileged users to control their appliances through an app on their mobile device.
- Pulling data from appliances for use in the mobile app and for machine learning purposes.

The system will not interact with any other third party systems. All data, such as power usage, will come directly from the appliances themselves.

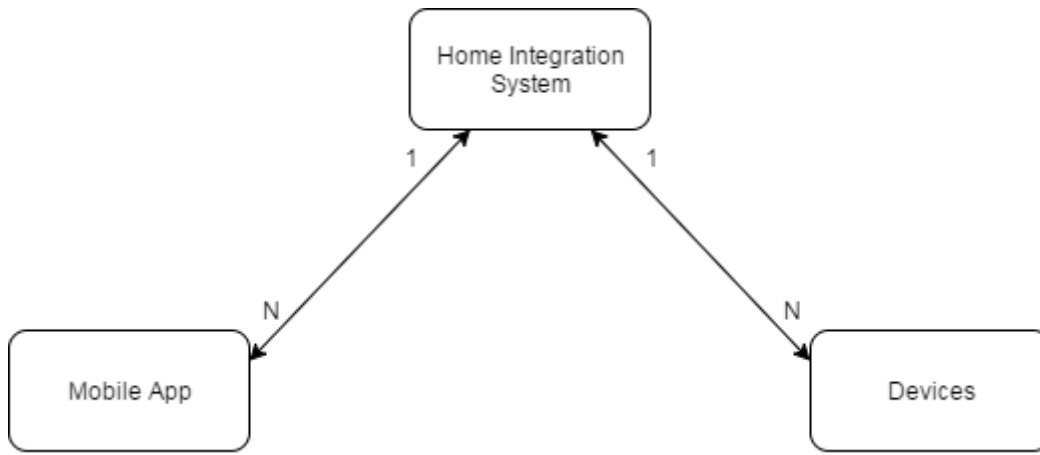


Figure 1. A high level diagram showing the three major components in the overall system.

As shown in Figure 1, there will be a single instance of the Home Integration System per household. This system is capable of integrating as many devices as needed, and the system can be controlled by any number of authenticated instances of the mobile application.

## 2.2. Product Features

The Lazy Bot home integration system provides features to the user and to the home or business it is installed in. The system will provide the following features:

Feature	Description
Login	The system will provide a way for users to log in to their home integration system using the mobile

Add Device	The system will allow privileged users to add new devices to their home integration system.
Remove Device	The system will allow privileged users to remove devices from their home integration system.
Configure Device	The system will allow privileged users to configure their integrated devices.
Configure System	The system will allow privileged users to configure their overall home integration system.
Add New User	The system will allow owners to add new users to their home integration system.
Manage Privileges	The system will allow owners to manage the privileges of the users in their system.
Update Software	The system will allow privileged users to update their system if there are updates available.
Machine Learning	<p>The system will continuously collect non personally identifiable information from integrated devices.</p> <p>The system will periodically mine all stored data to determine the movement and usage patterns of its users and devices in order to increase automation and power efficiency.</p>

Table 2. List of features

## 2.3. User Classes and Characteristics

Each Lazy Bot home integration system will have up to two types of users. An owner who administrates the system at their home or business and any number of registered users who the owner has granted access to some of the home integration system's features.

### 2.3.1. Owner

An owner owns a home with the integration system, or is the administrator of the system at a business where the integration system is installed. An owner will be using the Lazy Bot home integration system and Lazy Bot automated devices to make their home or business more user-friendly, comfortable, and efficient.

Frequency	Daily use.
Subset of Features Used	All features within the mobile application.
Technical Expertise	Familiar with mobile devices and simple touch interfaces.
Security or Privilege Levels	Full access to all features.

Table 3. Key attributes of the Owner class

Many of the characteristics of the owner depend on whether the system is installed in a home or at a business. If the system is installed in a home, the owner may not have a technical background or knowledge of the product domain. If the system is installed at a business, the owner is the administrator of the system and is proficient with using computer systems.

### 2.3.2. Registered User

The registered user is a person who lives in the home or works at the business where the system is installed. A registered user can have a wide array of different privileges depending on the amount of control the owner of the system grants them.

Frequency	Daily to weekly use.
Subset of Features Used	Mostly login and configuration (control) of devices. Can consist of other features depending on privilege level.
Technical Expertise	Familiar with mobile devices and simple touch interfaces.
Security or Privilege Levels	Varies based on privilege level granted by owner.

Table 4. Key attributes of the Registered User class

Many of the characteristics of the registered users vary widely as this is a large set of users compared to the Owners class. The level of technical background and computer skills of the registered users will range widely since anybody familiar with mobile devices and simple touch interfaces can be a registered user.

## **2.4. Operating Environment**

Since the Lazy Bot home integration system will consist of two components with different operating environments. The following two sections will cover their individual operating environments.

### **2.4.1. Mobile Application**

The mobile application component of the Lazy Bot home integration system will function on smartphone and tablet hardware platforms with their respective up-to-date operating systems, including iOS, Android, and others. The mobile application must operate with other software components installed on the mobile device.

### **2.4.2. Home Integration System**

The Lazy Bot home integration system will operate on a cloud environment where a new instance of the system will be generated for each home or business. The cloud environment will consist of several Amazon AWS EC2 systems running more than one subcomponent of the home integration system. All of the Amazon AWS EC2 systems will be running Amazon Optimized 64 bit Linux. Thus, all components of the home integration system will need to be compatible with this operating system. Furthermore, the home integration system and all of its sub components must operate alongside supporting software running on the same operating system.

## **2.5. Design and Implementation Constraints**

The system will be constrained by the fact that any future devices added will be Lazy Bot Automations standard approved. Due to the use of AWS, the system does not have any memory or physical constraints. All communication will be done through HTTPS and SSL for encryption. The system is also constrained by the availability of the Lazy Bot Automations API. This means that although the system may be up and available, if the Lazy Bot Automations API is not accessible, the system will appear down to the user. Lastly, the software for the system must be developed in C++ so that it can be mobile cross platform as well as AWS supported.

## 2.6. Assumptions and Dependencies

Since the Lazy Bot home integration system is mostly self-contained it does not have dependencies outside of the AWS. The system interacts with a number of Lazy Bot's other products, and the assumptions illustrate the assumed functionality of those products.

### 2.6.1. Assumptions

The Lazy Bot home integration system is based on several assumptions:

- The mobile application is the only user-facing interface. There will be no browser or desktop application.
- Future devices will adhere to the Lazy Bot Automations standard and the standard will not be modified.
- The API of Lazy Bot smart devices is available.
- The Lazy Bot smart devices are capable of sending JSON objects via web API.
- Lazy Bot Automations will provide specifications for the API of each current and planned future device on their product line for integration into the system.
- The system will not have to function offline, unless the user is on the same network.
- AWS will be available for the lifetime of the system, along with any third party tools or libraries.

### 2.6.2. Dependencies

The Lazy Bot home integration system depends on AWS to provide support for the lifetime of the system. Other than that, the system is self-contained.

## 3. System Features

For the features mentioned in this section, the priorities have been set based on judgement and can be modified based on the client need.

### 3.1. Login

#### 3.1.1. Description and Priority

Logging into the system through the mobile application will allow the user to access the rest of the features within the system. This feature is of high priority.

### **3.1.2. Stimulus/Response Sequences**

When the user opens the mobile application, the system will prompt them for a username and password.

Once the user enters their credentials, the system will validate the input.

If the user has two-step authentication enabled, they will be prompted for their next steps where they will provide biometric data and/or start the process of using a third party two-step authentication system.

Once the user has successfully passed all steps of authentication, the system will provide them with access to the main page of the mobile application.

### **3.1.3. Functional Requirements**

#### **3.1.3.1. Prevent Unauthorized Access**

If the user fails to provide valid credentials, biometric data, or fails to complete the third party two-step authentication at any stage of the login process, the system will not allow them to access the other features of the application.

#### **3.1.3.2. Multi-Step Authentication**

If the user has enabled two-step authentication using biometric data and/or a third party authentication system, the system will prompt the user to enter the required information after the initial password login.

#### **3.1.3.3. Log and Alert – Logins from New Devices**

If the user is logging into the system from a new device, the system will log their device information and alert the owner of this event.

#### **3.1.3.4. Log and Alert – Logins from Largely Different Locations**

If the user is logging into the system from a largely different geographical location, the system will log their device information/location and alert the owner of this event.

## **3.2. Add a New Device**

### **3.2.1. Description and Priority**

Adding a device will allow the system to configure the device and get data from the device. This function is high priority.

### **3.2.2. Stimulus/Response Sequences**

When the user prompts the system to search for new devices, the system will search the network for new devices.

When the system recognizes a new Lazy Bot Automations device while searching it will add the device to a list of available to devices.

When the user selects the new device the system will prompt them to add the new device to the system.

When the user selects add the new device to the system, the system will then be able to configure and gather data from the device.

### **3.2.3. Functional Requirements**

#### **3.2.3.1. Prevent Non-Privileged Users from Adding Devices**

If the user does not have the proper permissions, the system will not allow them to add a new device.

#### **3.2.3.2. Prevent Unrecognized Devices from Being Added**

If the device does not have a recognized interface, the system will prompt the user that the device cannot be added.

#### **3.2.3.3. Failsafe for Connection Issues when Adding a Device**

If the system loses access to the new device while trying to connect to the new device, the system will prompt the user that the device could not be added.

## 3.3. Remove a Device

### 3.3.1. Description and Priority

Removing a device will prevent the system from configuring the device and getting data from the device. This functionality is medium priority.

### 3.3.2. Stimulus/Response Sequences

When the user selects a device while using the system, the user will have the option to remove a device.

When the user selects remove the device, the system will prompt them to confirm that they want to remove the device.

When the user confirms that they want to remove the device, the system will remove the device from the system.

### 3.3.3. Functional Requirements

#### 3.3.3.1. Prevent Non-Privileged Users from Removing Devices

If the user does not have the proper permissions, the system will not allow the user to remove the device.

#### 3.3.3.2. Failsafe for Connection Issues when Removing a Device

If the system loses access to the device while trying to remove the device, the system will prompt the user that the device could not be removed.

## 3.4. Configure a Device

### 3.4.1. Description and Priority

Configuring a device will allow the system to make changes to the device settings and update received data from the device. This functionality is high priority.

### 3.4.2. Stimulus/Response Sequences



When the user selects a device while using the system, the system will present the option to configure device settings.

When the user selects configure device settings, the system will prompt them with a list of settings that can be changed on the device.

When the user finishes configuring a device, the system will allow the user to confirm or cancel their changes.

When the user selects confirm, the system will change desired settings and update the data received from the device.

### **3.4.3. Functional Requirements**

#### **3.4.3.1. Prevent Non-Privileged Users from Configuring Devices**

If the user does not have proper permissions to change the desired device settings, the system will not allow modifications.

#### **3.4.3.2. Validation of Devices' Configuration Settings**

If the system configurations exceed regulated limits of the device, the system will prompt the user that the desired input setting cannot be accepted.

#### **3.4.3.3. Configuration of Powered Off or Unreachable Devices**

If the device is powered off, and thus connectionless, the system will prompt the user that the device is unavailable for configuration.

## **3.5. Configure the System**

### **3.5.1. Description and Priority**

Configuring the system will allow system administrators to make changes to machine learning and automation settings for the entire system and all of its connected devices. This functionality is medium priority.

### **3.5.2. Stimulus/Response Sequences**

While the mobile application interface is active, the system will present the option to configure the system.

When the user selects to configure the system, the system will prompt them with a list of machine learning and automation settings that can be changed.

When the user completes their configuration of the system, the system will allow for the user to confirm or cancel their changes.

When machine learning or automation settings are manipulated and confirmed, the system will adjust its behavior to fit the new configuration.

### **3.5.3. Functional Requirements**

#### **3.5.3.1. Prevent Non-Privileged Users from Configuring the System**

If the user does not have the proper permissions, the system will not allow the user to configure the system.

#### **3.5.3.2. Emergency Updating of System Configuration**

If emergency conditions are detected, the system must strongly advise the user to change their system settings to resolve the situation.

## **3.6. Add a New User**

### **3.6.1. Description and Priority**

Adding a new registered user to the system will allow the added user access to all devices currently connected to the system that the administrator gives them permission to use. This function is medium priority.

### **3.6.2. Stimulus/Response Sequences**

When an administrator prompts the system to add a new user, the system will prompt the administrator with an invitation form to add the user and set their permissions.

When the new user receives the invitation, accepting it must allow them to create or add an account to the system.

### **3.6.3. Functional Requirements**

#### **3.6.3.1. Invitation Expiration**

The system will ensure that any administrator invitation that is older than 24 hours will no longer be valid.

#### **3.6.3.2. Granting of Permissions**

Once the user creates or adds an account, the system must grant them only the permissions specified by the administrator.

#### **3.6.3.3. Enabling of Location Services**

The system must prompt the user to enable location services on their device if they are not already enabled.

## **3.7. Manage Privileges**

### **3.7.1. Description and Priority**

This feature will allow owners of the system to manage the privileges of other system users. This feature is medium priority.

### **3.7.2. Stimulus/Response Sequences**

Once a user is added, the owner may prompt the system to edit the permissions of that user.

### **3.7.3. Functional Requirements**

#### **3.7.3.1. Adding or Removing Privileges**

The system must allow the owner to grant additional privileges to a user so they can access additional devices and/or controls. The system must allow the owner to remove privileges that a user currently has.

#### **3.7.3.2. Removing Users**

The system must allow for the owner to remove a user's access to the system altogether.

## **3.8. Update Software**

### **3.8.1. Description and Priority**

This feature will allow for the owner or administrator of the system to receive notifications of available updates through the mobile application. The owner or administrator will then be able to initiate the update process from their mobile device if they wish to do so.

This feature is medium priority because it doesn't directly impact the use of the system but is necessary for the maintenance of the system after deployment.

### **3.8.2. Stimulus/Response Sequences**

When the development team has updated the version of the system or mobile application, the system will notify the owner or administrator that an update is available.

If the owner or administrator chooses to accept the update, the applicable system and/or components will be updated without further involvement from the owner.

If the owner chooses to not update their system, the system should continue to function normally on its current version.

### **3.8.3. Functional Requirements**

#### **3.8.3.1. Opting Out of Updates**

The mobile application must allow for users to opt out of getting updates.

#### **3.8.3.2. Running on Out of Date Software**

If a user has opted out of an update, the system should continue to function normally on its old version.

#### **3.8.3.3. Emergency Updates**

In the case of an emergency update for security reasons, the user must be strongly advised to update the application.

## **3.9. Machine Learning**

### **3.9.1. Description and Priority**

Machine learning will allow the system to recognize patterns in the movement of people and the usage of devices within the home or business, which the system can then use for automating processes. The main goal of this feature is to increase usability while also improving efficiency. This feature is a low priority as it does not directly block any other features described in previous sections.

### **3.9.2. Stimulus/Response Sequences**

Data from devices and sensors placed throughout the home or business will collect usage and movement data among other things and send it to the system.

The system will periodically mine the new datasets in an attempt to find patterns in the movement of the users within the home or business along with patterns in the usage of devices.

The system will then use these patterns to automate actions such as the changing of temperature, and turning on/off of lights.

### **3.9.3. Functional Requirements**

#### **3.9.3.1. Collection of Data**

If the owner has configured the system with machine learning off, the system will only collect data necessary for normal functionality.

If the owner has configured the system with machine learning on, the system will only collect data that is not personally identifiable.

#### **3.9.3.2. Mining of Data**

If the owner has configured the system with machine learning off, the system will not mine any of the data retrieved from the home or business.

### 3.9.3.3. Automation Based on Recognized Patterns

If the owner has configured the system with machine learning or automation off, the system will not automate the process of any devices within the home or business.

## 4. Use Cases

### 4.1. Use Case Diagram

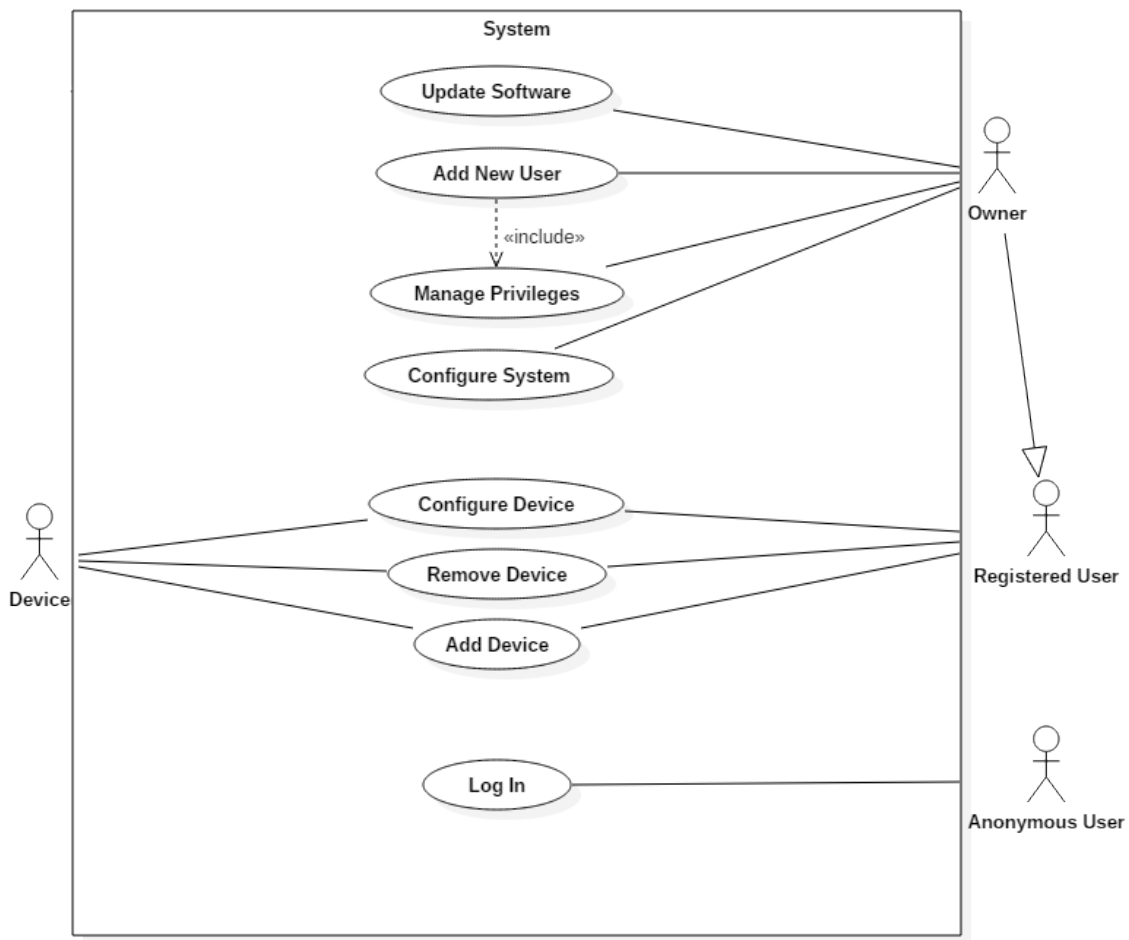


Figure 2. Use Case Diagram for the features of the system.

### 4.2. Use Case Documents

#### 4.2.1. Login

Use Case name

**Login**

Description

**This use case involves users logging into their account**

Actors

**Registered user**

Pre-conditions

**Users must have an account with the system.**

**Users must have a system installed in their home.**

Main flow

1. **<Enter Credentials>** User enters their credentials into the system.
2. **<Confirm Credentials>**User confirms their credentials.
3. The system authenticates the user's credentials.
4. The system displays a success message.
5. The use case ends.

Post-conditions

**The user is now logged in**

Alternative flows

- A. At **<Confirm Credentials>** if credentials are incorrect
  1. User is notified that their credentials were invalid.
  2. The system displays the field for entering the credentials.Return to **<Enter Credentials>**.

#### **4.2.2. Add a New Device**

Use Case name

**Add device**

Description

**The use case for a user adding a new device to their home automation network**

Actors

**Registered user**

#### Pre-conditions

**The new device is plugged in and connected to the internet.**

**The user is logged in.**

#### Main flow

1. The user chooses to search for new devices.
2. The system searches for new devices.
3. The system displays the new device candidates.
4. **<Select device>** The user selects the new device they wish to add.
5. **<Connect device>** The system attempts to connect to the device and verifies its type and functions.
6. System indicates that the device has been connected successfully.
7. The use case ends.

#### Post-conditions

**New device should be able to interact in the same way as devices previously connected to the home automation network.**

#### Alternative flows

- A. At **<Select device>** if the new device does not appear, then
  1. System prompts user to search for new devices.  
Return to **<Select device>**.
- B. At **<Connect device>** if the connection cannot be established, then
  1. The system indicates that the device has not been connected successfully.  
Return to **<Select device>**.
- C. At **<Connect device>** if the user does not have the required privileges to connect devices, then
  1. The system informs the user that they do not have the required privileges to connect the device.  
Return to **<Select device>**.

### **4.2.3. Remove a Device**

#### Use Case name

**Remove device**

#### Description



**This use case involves a user removing a device from their home automation network**

Actors

**Registered user**

Pre-conditions

**The device has previously been linked to the home automation network.**

**The user is logged in.**

Main flow

1. The user navigates to the current device.
2. **<Select device>** The user selects the device in question.
3. The system displays available service options for the device.
4. **<Remove Device>** The user selects the command to remove the device from the system.
5. The system requires the user to confirm their decision to remove the device.
6. The user confirms.
7. The system removes the device.
8. The use case ends.

Post-conditions

**The device should now be successfully removed from the system.**

Alternative flows

- A. At **<Remove Device>** if the user does not have the required privileges to remove devices,
1. The system informs the user that they do not have the required privileges to remove the device.
- Return to **<Select device>**

## **4.2.4. Configure a Device**

Use Case name

**Configure Device**

Description

**This use case involves users configuring specific device settings in context of the system**

Actors

**Registered users**

Pre-conditions

**Users must have an account with the system.**

**Users must be logged into the system.**

**Users must have added at least one device to the system.**

Main flow

1. <**Select Device**> User selects a device.
2. <**Configure Device**> User modifies current device settings.
3. System queries the user for confirmation.
4. User confirms their selection.
5. The use case ends.

Post-conditions

**The device's settings are now configured to the user's specifications.**

Alternative flows

- A. At <Configure Device> if the user does not have the required privileges to configure devices, then
  1. The system informs the user that they do not have the required privileges.Return to <Select Device>.

## **4.2.5. Configure the System**

Use Case name

**Configure System**

Description

**This use case involves the system administrator configuring the system settings**

Actors

**System Administrator**

Pre-conditions

**The administrator must be logged into the system**

Main flow

1. The admin selects the setting that they wish to modify.
2. The admin provides the new value for the setting being modified.
3. The system queries the admin for confirmation.
4. The admin confirms the new value.
5. The use case ends.

Post-conditions

**The system settings must now be modified to the newly configured settings**

## **4.2.6. Add a New User**

Use Case name

**Add New User**

Description

**This use case involves the system administrator adding a new registered user to the system**

Actors

**System Administrator**

Pre-conditions

**The administrator must be logged into the system**

Main flow

1. The admin selects the option to add a new registered user to the system.
2. The system queries the admin for the new user's username and password.
3. The admin confirms the username and password.
4. The system queries the admin for the initial privileges of the new registered user.
5. The admin confirms the initial privileges.
6. The system creates the new registered user.
7. The use case ends.

Post-conditions

**The new registered user must now be registered with the system**

## **4.2.7. Manage Privileges**

Use Case name

**Manage Privileges**

Description

**This use case involves the system administrator setting privileges for a registered user**

Actors

**System Administrator**

Pre-conditions

**The administrator must be logged into the system.**

**There must be registered users in the system.**

Main flow

1. Admin selects the registered user they wish to manage privileges for.
2. Admin selects the privilege level for the registered user in question.
3. The system requires the admin to confirm their decision.
4. The admin confirms.
5. The use case ends.

Post-conditions

**The registered user must have their privileges modified according to what the admin selected.**

## **4.2.8. Update Software**

Use Case name

**Update software**

Description

**This use case describes how a developer updates the application's software to introduce new functionality or fix existing bugs**

Actors

**Developers**

Pre-conditions

1. **The users of the application must be logged into the system and grant access/update privileges to the developers.**
2. **The developers must be employees of SkyNet Inc. or Lazy Bot Automations.**
3. **The developers send a prompt to the customer for a software update.**

Main flow

1. The system displays the prompt to the user.
2. **<Install update>** The user selects the option: install update.
3. The system sends a confirmation to the customer.
4. The system begins to update.
5. The system restarts once the update has completed.
6. The application automatically re-opens and is now updated.
7. The use case ends.

### Post-conditions

**The system is now updated and contains all the new functionality/bugs have been fixed.**

### Alternative flows

- A. At <Install update> if there is no internet connection or some settings are wrong, then
  1. The system displays a message saying “Update failed”.
  2. The system prompts the user to try again.  
Return to <Install update>.

## **5. External Interface Requirements**

### **5.1. Hardware Interfaces**

#### **5.1.1. Lazy Bot Devices**

The system must communicate with current Lazy Bot hardware systems via the Lazy Bot API. The API will reformat commands for each Lazy Bot hardware system. The system does not interact directly with the hardware.

#### **5.1.2. Communication Protocols**

The system must send commands to the Lazy Bot API in the form of JSON objects. The system must also be able to receive and parse JSON objects returned from the API.

### **5.2. Software Interfaces**

#### **5.2.1. External API**

The system must interface with the API supplied by Lazy Bot Automations to connect to the Lazy Bot Automations devices. This API will allow the system to control, configure, and set the values of the Lazy Bot Automations’ smart devices.

#### **5.2.2. External Cloud Services**

The system must be run using AWS including but not limited to EC2 for any servers, Dynamo DB for the database, Lambda for any extra serverless computing and CloudWatch for

maintenance. Using AWS services will allow the system to scale as more users access its features.

### **5.2.3. Software Development**

Due to the nature of the system's mobile application interface and the use of AWS, the system must be implemented using the cross platform and AWS supported language of C++.

### **5.2.4. Tools**

The server must be maintained using the Jenkins continuous integration service. The codebase must be built using Gradle build automation tools as this will allow for any code changes to be automatically updated on AWS Lambda.

### **5.2.5. Data Encryption**

Once data has been successfully transferred to the system, the data must be fully encrypted using a 2-way salted hash encryption. Encrypted data will be stored using the secure AWS Dynamo DB database.

## **5.3. Communications Interfaces**

### **5.3.1. Communication**

Communications between users, the system, and the Lazy Bot Automations API must be done through HTTPS. This means that normal HTTP communication will be used, however an extra layer will be added for SSL encryption of data during transport.

## **6. Other Non-Functional Requirements**

### **6.1. Performance Requirements**

#### **6.1.1. Low Power Mode**

Energy efficiency will be achieved through the proactive management of occupancy needs; the system will strive to maintain a thermal mass which is either consistent with what the user specifies or with historical trends.

### **6.1.2. Response Time Limitation**

Commands from the user must be processed by the system and sent to the Lazy Bot API within 500ms. The measurement will be from the time that the user initiates an action on the mobile device to the moment when the API of the applicable devices receives the transmission.

### **6.1.3. Power Constraint Reached**

If the user specified energy consumption is met, the system will notify the user and enable full manual control of all connected LBA devices. The user may return control to the system at any time using the mobile application.

## **6.2. Safety Requirements**

The foundational safety requirements will be handled by the API provided by Lazy Bot Automations. The software implemented will include the same safeguards and actions that would appear on Lazy Bot Automation's existing hardware devices. The following sections cover safety requirements that are noted to prevent any breakage of regulations, policies, and damage.

### **6.2.1. Smart Sensors**

- The software system must communicate with the smart thermostats to provide alerts if a home/business loses power or if the temperature in the home falls below or rises above a certain threshold.
- The system must authenticate thermostat temperature limits through Lazy Bot Automation's provided API.
- The system shall allow notifications from water sensors that can detect unwanted water in the home, alerting the user to potential leaks near washing machines, dishwashers, water heaters and other areas.
- The system must interface with the smoke, heat, and carbon monoxide detector system to prevent potential harm to the user.

### **6.2.2. Smart Security Systems**

- The smart lock remote control and monitoring of home entry must be authenticated by the user.
- In the case of a security threat, the system must gather data from the Lazy Bot API and notify the user of any risks or harm.
- The system must alert the user if unexpected sensor activity occurs at or around any device embedded doors.
- If a door is left open or unlocked, the system must deliver alerts and allow the user to control the door remotely.

### 6.3. Security Requirements

These are the security requirements that the system must adhere to:

- The system must provide at least one level of authentication, with the option for the addition of one to two more levels of authentication.
- The system shall promptly notify users if their system has been accessed from another geographical location.
- The system shall promptly notify users if their system has been accessed from a new device.
- The system must allow for multiple levels of permission, from administrators all the way down to the lowest level.
- Administrators must be able to grant permissions to those users beneath them.
- All communication related to the system must be encrypted.
- All data used by the system must be encrypted.
- During maintenance from trusted specialists, the administrators of the system receiving maintenance must be able to grant a specific level of access privilege to the specialist.

### 6.4. Software Quality Attributes



### **6.4.1. Availability**

The system shall have a 98% availability based on the number of hours per year the system is fully unavailable.

The system must be available via the mobile app from anywhere in the world, as long as there is an internet connection.

### **6.4.2. Interoperability**

On release, the system must meet the Lazy Bot Automations standard for inter-device communication.

### **6.4.3. Usability**

The system must provide the same flow of logic each time for when a user adds a new device, regardless of its relation to previous devices.

### **6.4.4. Testability**

The software used within the system must have minimum 70% tested code coverage based on nightly testing.

### **6.4.5. Robustness**

The system must be able to handle and recover from errors or faults during normal operation.

### **6.4.6. Flexibility**

The system will allow its users to have any combination of settings within the application.

### **6.4.7. Maintainability**

The system must be at least 70% bug free to ensure the code base is easy to maintain.

All code within the system will follow predefined styling and coding standards.