Housing Automation

Ben Seiber

Thomas Pionk

Jheryl Lezama

Jeff Wallace

Daniel Wilmot

## Table of Contents

|  |  |
| --- | --- |
| **Housing Automation (Cover Page)** | 1 |
| **Table of Contents** |  |
| **Section 1.0 - Project Vision** |  |
| 1.1 - Project Background |  |
| 1.2 - Business Objectives, Security and Ethical Concerns |  |
| 1.3 - Glossary of Key Terms |  |
| **Section 2.0 - Project Planning & Execution** |  |
| 2.1 - Team Information |  |
| 2.2 - Tools and Technology |  |
| 2.3 - Project Plan |  |
| 2.4 - Best Standards and Practices |  |
| **Section 3.0 - System Requirements Analysis** |  |
| 3.1 - Functional Requirements |  |
| 3.2 - Non-Functional Requirements |  |
| 3.3 - Wireframe Designs |  |
| **Section 4.0 - System Requirements Specifications and Diagrams** |  |
| 4.1 - User Stories, Scenarios |  |
| 4.2 - Use Cases |  |
| 4.3 - System Sequence / Activity Diagrams |  |
| **Section 5.0 - User Interface Specifications** |  |
| 5.1 - Preliminary Design |  |
| **Section 6.0 - Test Cases** |  |
| 6.1 - Unit Architecture and Strategy/Framework |  |
| 6.2 - Unit Test Definition, Test Data |  |
| **Section 7.0 – References** |  |

## Section 1.0 - Project Vision

**1.1 - Project Background**

Home automation or “the internet of things” is not a new concept. However, the benefit of an effective automation and monitoring system is something that cannot be understated. The Home Automation system strives to combine the benefits of housing security, utility monitoring, and cost-saving analytics into one beneficial package. It is our belief that such a system would aid in alerting landlords and tenants to possible issues before a potentially larger incident occurs. This system provides the benefits of receiving live sensor updates regarding water pressure, temperature, humidity, and lighting through the convenience of a website. Additionally, landlords are able to control specific items such a lighting and temperature in hallways and public spaces when the need arises. Furthermore, sensor logs are recorded so that analytics can be later generated in hopes of identifying high-cost utilities. As a result, the home automation system will allow for the automation of a housing facility by using both monitoring and control elements, focusing on building safety and energy savings.

**1.2 - Business Objectives**

* Provide landlords with a convenient method of remotely monitoring property utilities.
* Provide landlords with a convenient method of remotely controlling property utilities.
* Provide landlords with a convenient method of remotely sending alert notifications to the respective residing tenants should there be any utility failures.
* Provide landlords with a convenient method of receiving cost-saving analytics generated from the various sensor readings.

**1.2 - Security Concerns & Ethical Concerns**

* While the system shall allow the remote monitoring & control of building utilities including lighting and temperature, the function to control is strictly limited to public spaces within the facility. These spaces will include, but are not limited to the building’s front doorway, main hallways, front reception area, public laundry rooms, basement, attic and other social spaces as to not invade on individual tenant privacy.
* While the system shall allow the remote monitoring of building utilities through the system’s various sensor data, this feature will include the monitoring of individual tenant’s utilities for the purpose of storing usage data. This feature is to be used in conjunction with the cost saving analytics that could be potentially used in order to better assist the landlord with identifying excessive utility costs. Note that the remote controlling functionality will not extend to the individual tenant spaces.
* The system will hash the password entries during the registration and login processes in order to improve system security. The use of the home automation system shall be limited to end-users that register an account and securely login to the system.

**1.3 - Glossary of Terms**

* **Android** - Refers to the operating system found in the end-user mobile application for use with the Home Automation System.
* **Android Studio -** Refers to the software framework that was used to develop the end-user mobile application.
* **Arduino -** Refers to the open-source microcontroller that houses the logic for the various system sensors including lighting, temperature sensor and security.
* **Asynchronous Javascript and XML (AJAX) -** Refers to the scripting language that was used to dynamically generate HTML elements within the end-user web application.
* **Apache Web Server -** Refers to the server that will host the Home Automation System end-user web application.
* **Cascading Style Sheets (CSS) -** Refers to the scripting language that was used to visually style the end-user web application.
* **End-User -** The intended users of the Home Automation system which are referred to in this document as 1) Landlord and 2) Tenant.
* **End-User Application -** Refers to both the web and mobile applications that function as the user interface for the Home Automated System.
* **Gantt Chart -** A visual timeline chart found in Microsoft Project 2016 © that outlines the project timeframe and various deadlines that was used in the team planning and development of the Home Automation System project.
* **Hypertext Markup Language (HTML) -** Refers to the scripting language that was used to develop the end-user web application.
* **Hypertext Preprocessor (PHP) -** Refers to the scripting language that was used to handle the information passed between the end-user web application and the various web servers.
* **Landlord -** Refers to the building manager/supervisor who also functions as an end-user for the Home Automation system.
* **Microsoft Project 2016 © -** The software framework developed by Microsoft © that was used in the team planning and creation of a project schedule for development of the Home Automation System project.
* **MQTT Server -** Refers to the network server that functions to provide communication between the various system sensors (temperature, lighting, etc.), the system database and the end-user software applications.
* **MySQL Server -** Refers the database that will store 1) end-user account information and 2) sensor readings
* **NodeRed -** Refers to the software framework that supports the communication between the following; MQTT server, MySQL server, NodeRed server, Apache Web server, Arduino microcontrollers and the end-user web & mobile applications.
* **Property -** Refers to the building or residency that is owned by the landlord.
* **Python -** Refers to the programming language that utilized by the Arduino microcontrollers.
* **Raspberry Pi -** Refers to the miniature computer device that will function to host the MQTT server, MySQL server, NodeRed server and the Apache Web server.
* **Sensor/Asset -** Refers to the individual system sensors including - but not limited to - temperature, water pressure, motion detectors, etc. and property lighting.
* **Tenant -** Refers to the residents that reside at a building/property.
* **User Interface (U.I.) -** The interface that the end-user will interact with in order to use the Home Automation System and encompasses both the visual style and layout of the application.
* **Wamp Server -** The software used to host the web application for both development and testing purposes.

## Section 2.0 - Project Planning & Execution

**2.1 - Team Information**

* **Ben Seiber**
  + Project Lead & Designer of the Home Automation system physical model including the CNC laser cutting of the model itself, the electrical wiring of the model, the coding of the Arduino logic and designer of the physical hardware layout.
* **Daniel Wilmot** 
  + Lead programmer of the web application including the HTML, CSS, PHP and AJAX code and co-designer of the web application user interface design. In addition, responsible for implementing the database using PHP MyAdmin and also troubleshooting & testing website code.
* **Jeff Wallace**
  + Lead designer of the Home Automation System user interface and in addition to designing the various system schemas including database design, application page layout & orientation and wireframe schematics. Also responsible for assisting with the coding of the web application.
* **Jheryl Lezama**
  + Project Planner and also responsible for documenting the development of the Home Automation System. Created the use cases, user stories, functional and nonfunctional requirements in addition to the Activity & Sequence Diagrams for each. Also responsible for assisting with the coding of the web application.
* **Thomas Pionk**
  + Lead designer of the back-end system network including the server configurations and various powershell scripts that allowed for communication between the various hardware elements. Also designed the system U.I. dashboard used for controlling the various system sensors and configuring the NodeRed networking schema.

**2.2 - Tools and Technology**

The tools utilized within this project were broken into three categories, 1) Automation Hardware, 2) Networking Hardware and 3) End-User Hardware. The software application frameworks that were used to code each hardware element are listed in the “Software Application Framework” section and lists the framework next to the respective device.

1. The physical Home Automation system hardware includes the following…
   1. LED Lighting - Responsible for simulating a building’s internal lighting system.
   2. Temperature Sensor - Responsible for simulating a building’s internal heating and cooling thermostat system.
   3. Humidity Sensor - Responsible for monitoring the humidity content of a simulated building space.
   4. Arduino Microcontrollers - Responsible for interfacing with the various Home Automation system sensors (lighting, temperature, humidity, etc.).
2. The networking hardware used in the Home Automation System includes the following…
   1. MQTT Server - Responsible for collecting the sensor data from the arduino microcontrollers and placing them within the Raspberry Pi.
   2. Raspberry Pi - Responsible for hosting the MQTT server, MySQL Server and the Node-Red framework architecture.
   3. MySQL Server - Responsible for storing the numerous sensor values that are used as part the cost saving analytic calculations.

1. The software application frameworks utilized in this project are as follows…
   1. Arduino Scripts
      1. C-Programming
   2. Database
      1. Command Line
      2. MySQL Workbench
      3. PHP MyAdmin
   3. Mobile Application
      1. Android Studio
   4. MQTT Server Configuration
      1. Command Line
      2. Node-RED
   5. Project Scheduling
      1. Microsoft Project 2016 **©**
   6. Raspberry Pi
      1. Command Line
      2. Python
      3. Node-RED
   7. Team Collaboration
      1. Github
      2. Google Drive
      3. Google Hangouts
      4. GroupMe
   8. Web Application
      1. Sublime Text Editor
      2. Wamp Server
2. The End User Hardware that is required to run the Home Automation system is as follows…
   1. Laptop or Desktop for use with web application
   2. Mobile Android OS Device for use with mobile application

**2.3 - Project Plan (Needs Updating)**



**2.4 - Best Standards and Practices**

* The website will hash the password entry during both the login and registration process for improved system security.
* The MySQL database will store passwords as hashes as to improve system security.
* The servers will only be accessible on a private network.
* The end-user account information will be stored within the database with no data loss.

## Section 3.0 - System Requirements Analysis

**3.1 - Functional Requirements**

1. **Requirement 1 - Account Registration:** The system shall allow landlords the ability to register an account using either the web or mobile application.
   1. **Actor(s):** Landlord
   2. **Description:** This will allow the landlord to register a new account should one not already exist via the web or mobile application.
      1. The system will use a MySQL database to store the account information.
      2. The system will save a landlord’s account information with no data loss.
      3. The system will hash the password entry as to improve system security.
2. **Requirement 2 - Account Login:** The system shall allow landlords the ability to login to their own account via the web or mobile application.
   1. **Actor(s):** Landlord
   2. **Description:** This will require that the landlord use either the web or mobile application to authenticate their login information.
      1. The web and mobile application will use PHP to fetch and authenticate the landlord login credentials stored within the MySQL Database.
      2. The system will hash the password entry on the login screen to improve system security.
3. **Requirement 3 - Remote Monitoring:** The system shall allow landlords the ability to remotely monitor sensors/assets including lighting, temperature, humidity, etc. via the web or mobile application.
   1. **Actor(s):** Landlord
   2. **Description:** This will allow the landlord with the means to remotely monitor the various sensor readings via the web and/or mobile application.
      1. The ability to remotely monitor the various system sensors shall only be granted to users who have both registered and securely logged into the system via the web or mobile application.
4. **Requirement 4 - Remote Control:** The system shall allow landlords the ability to remotely control the various sensors/assets including lighting, temperature, humidity, etc via the web or mobile application.
   1. **Actor(s):** Landlord
   2. **Description:** This will provide the landlord with the means to remotely control the various sensor readings from within the web and mobile application.
      1. The ability to remotely monitor the various system sensors shall only be granted to users who have both registered and securly logged into the system via the web or mobile application.
5. **Requirement 5 - Storing Sensor/Asset Log:** The system shall store sensor information within the back-end database via the web or mobile application.
   1. **Actor(s):** Landlord
   2. **Description:** This will provide the landlord with the means to create a stored log of sensor readings that can then be used to generate cost saving analytics.
      1. The system will use a MySQL database to store sensor readings.
6. **Requirement 6 - Retrieving Sensor/Asset Log:** The system shall provide landlords the ability to retrieve sensor information from the back-end database via the web or mobile application.
   1. **Actor(s):** Landlord
   2. **Description:** This will provide the landlord with the means to retrieve/access the saved log of sensor readings that can then be used to generate cost saving analytics.
      1. The web and/or mobile application will provide the means to interface with the MySQL Database in order to retrieve the stored sensor readings.
7. **Requirement 7 - Generating Cost Saving Analysis:** The system shall provide landlords with cost saving analytics based on stored sensor readings that could be used to lessen utility expenses.
   1. **Actor(s):** Landlord
   2. **Description:** This will provide the landlord with the access to cost breakdown analysis using the log of the various sensor readings.
      1. The web and/or mobile application will act as the interface for viewing the cost saving analysis.
8. **Requirement 8 - Registering Properties:** The system shall provide landlords with the ability to register multiple properties under their own account with the options to edit or delete the property post registration.
   1. **Actor(s);** Landlord
   2. **Description:** This will provide landlords with a means to keep track of their various properties in addition to the the sensors that occupy those buildings.
      1. The property information will be stored in a MySQL database.
      2. Each property registered will be unique to that landlord’s account (other landlords will not be able to see the property of another landlord, but instead can only see their own registered properties).

**3.2 - Non-Functional Requirements**

1. **Requirement 1 - Web Registration**
   1. **Description:** This will allow building administrators to create a personal account from the registration page.
      1. The system will use a secure MySQL database to store user login information (email, password, etc.).
      2. The system will use a secure MySQL database to store sensor data (lighting, temperature, etc.).
      3. The system will save user login information (email, password, etc.) within the database with no lost data.
      4. The system shall hash user passwords to improve system security.
2. **Requirement 2 -** **Networking Framework**
   1. **Description:** The system shall implement a MQTT Server to provide the back-end networking functionality.
      1. The MQTT Server will be hosted on a raspberry pi.
      2. The networking software for the back-end framework will be handled via NodeRed.
3. **Requirement 3 - MQTT Server Ability:** The server shall be available for 99.98% of the time.
   1. **Description:** This will allow the building administrator to have access to the database 99.98% of the time in order to use the Home Automation system.
      1. The server allows for the back-end network communication between the MySQL server and the system sensors.

**3.3 – Wireframe Designs**



****

Section 4.0 – User Stories, Scenarios and Use Cases

**4.1 – User Stories**

|  |  |  |
| --- | --- | --- |
| **Identifier** | **User Story** | **Size** |
| ST-1 | As a landlord, I want a secure registration process that safely stores my personal information within a database. | 3pts |
| ST-2 | As a landlord, I want a secure login process that grants me access to my own unique account separate from others. | 3pts |
| ST-3 | As a landlord, I want a way to store and retrieve sensor readings as gathered from my properties. | 9pts |
| ST-4 | As a landlord, I want a way to register and manage my properties in addition to the sensors that occupy each one. | 6pts |
| ST-5 | As a landlord, I want to see generated cost saving analytics that could help me save money in regards to utility expenses. | 6pts |

**4.1 – Scenarios**

(Not sure as to what these scenarios refer to, plan to ask Patel)

**4.2 - Use Cases**

|  |  |  |
| --- | --- | --- |
| **UC-#01** | | **Account Registration** |
| Related Requirements: | | REQ1 |
| Intiating Actor: | | Landlord |
| Actor’s Goal: | | Register an account within the database for use with the Home Automation system |
| Participating Actors: | | MySQL Database, Web Application, Mobile Application |
| Preconditions: | | System doesn’t already contain any account information regarding the landlord |
| Postconditions: | | System stores landlord account information within the MySQL database |
| **Flow of Events for Main Success Scenario:** | | |
| 🡪 | 1. | Landlord selects register new account on the login page |
| 🡨 | 2. | The system responds by displaying an account registration form for the landlord to enter personal account information |
| 🡪 | 3. | The landlord enters personal account information into the account registration form |
| 🡪 | 4. | The landlord presses the submit button once they have completed filling out the account registration form |
| 🡨 | 5. | The system responds by first checking that all of the entry fields have information entered |
| 🡨 | 6. | The system responds by checking the provided information against the information stored within the database to insure that there exist no duplicate account information |
| 🡨 | 7. | **(Exit Condition)** The systems responds with visual message confirmation that an account has been successfully registered within the database |
| **Flow of Events for Main Success Scenario (Alternate Scenario):** | | |
| 🡨 | 5a. | The system responds with a visual error message that all of the entry fields have not been filled out with the landlord’s personal information |
| 🡨 |  | The system does not create an account until all of the account registartion form fields contain provided landlord information |
| 🡨 |  | Same as in Step 3 above |
| 🡨 | 6a. | The system responds with a visual error message that the information entered within the account registration form already exists within the database |
| 🡨 |  | The system does not create a new account given the existence of duplicate account information within the database |
| 🡪 |  | Same as in Step 3 above |
| 🡨 | 6a. | The systems responds with a visual error message that there was an error connecting to the database |
| 🡨 |  | The system does not create a new account within the database |

|  |  |  |
| --- | --- | --- |
| **UC-#02** | | **Account Login** |
| Related Requirements: | | REQ1, REQ2 |
| Intiating Actor: | | Landlord |
| Actor’s Goal: | | Login to a created account that is already registered within the database |
| Participating Actors: | | MySQL Database, Web Application, Mobile Application |
| Preconditions: | | System already contains a landlord account |
| Postconditions: | | Landlord logs into their unique account |
| **Flow of Events for Main Success Scenario:** | | |
| 🡪 | **1.** | Landlord enters login account information on the login page |
| 🡪 | 2. | Landlord presses the login/submit button after entering the required information |
| 🡨 | 3. | The system responds by checking the provided login account information against the information stored within the database |
| 🡨 | 4. | The system logs the landlord into their unique account |
| 🡨 | 5. | **(Exit Condition)** The system further responds with visual confirmation message that the landlord has successfully logged into their account |
| **Flow of Events for Main Success Scenario (Alternate Scenario):** | | |
| 🡨 | 3a. | The system responds with a visual error message that the login account information provided was incorrect |
| 🡪 |  | Same as in Step 1 above |

|  |  |  |  |
| --- | --- | --- | --- |
| **UC-#03** | | | **Remote Monitoring** |
| Related Requirements: | | | REQ8 |
| Intiating Actor: | | | Landlord |
| Actor’s Goal: | | | To remotely monitor the sensor/asset information within a particular property via the web or mobile application |
| Participating Actors: | | | Sensor/Asset, MQTT Server, Web Application, Mobile Application |
| Preconditions: | | | Landlord has selected view sensor/asset information from property screen |
| Postconditions: | | | Landlord is taken to a page that displays the sensor/asset information dashboard for that property |
| **Flow of Events for Main Success Scenario:** | | | |
| 🡪 | | 1. | Landlord selects a registered property |
| 🡨 | | 2. | The system responds by visually displaying the desired property information |
| 🡪 | | 3. | The landlord selects the option to view sensor/asset information for that property |
| 🡨 | | 4. | The system receives the request to view sensor/asset information and forwards the request to the MQTT Server |
| 🡨 | | 5. | The system then receives the sensor/asset information data from the MQTT Server |
| 🡨 | | 6. | **(Exit Condtition)** The system visually displays the sensor/asset information dashboard to the landlord |
| **Flow of Events for Main Success Scenario (Alternate Scenario):** | | | |
| 🡨 | 2a. | | The system responds with a visual error message that the property information cannot be retrieved from the database |
| 🡪 |  | | (Same as in step 1) |
| 🡨 | 4a. | | The system responds with a visual error message that the property sensor/asset information cannot be retrieved |
| 🡪 |  | | (Same as in step 3) |

|  |  |  |
| --- | --- | --- |
| **UC-#04** | | **Remote Control** |
| Related Requirements: | | REQ4, REQ8 |
| Intiating Actor: | | Landlord |
| Actor’s Goal: | | To remotely control a property sensor/asset via the web or mobile application (ex: turn off the lights) |
| Participating Actors: | | Sensor/Asset, MQTT Server, Web Application, Mobile Application |
| Preconditions: | | Landlord has selected control sensor/asset from property screen |
| Postconditions: | | Landlord has changed a sensor/asset setting |
| **Flow of Events for Main Success Scenario:** | | |
| 🡪 | 1. | Landlord selects a registered property |
| 🡨 | 2. | The system responds by visually displaying the desired property information |
| 🡪 | 3. | The landlord selects the option to control a utility for that property |
| 🡨 | 4. | System responds by sending the request to the MQTT server |
| 🡨 | 5. | System further responds by visually displaying a system control dashboard to the landlord |
| 🡪 | 6. | Landlord inputs the necessary changes to the sensor/asset dashboard |
| 🡪 | 7. | Landlord submits the change to the sensor/asset information |
| 🡨 | 8. | System displays the change on the sensor/asset dashboard |
| **Flow of Events for Main Success Scenario (Alternate Scenario):** | | |
| 🡨 | 2a. | The system responds with a visual error message that the property information cannot be retrieved from the database |
| 🡪 |  | (Same as in step 1) |
| 🡨 | 4a. | The system responds with a visual error message that the request could not be sent to the MQTT Server |
| 🡪 |  | (Same as in step 3) |

|  |  |  |
| --- | --- | --- |
| **UC-#05** | | **Storing Sensor Log** |
| Related Requirements: | |  |
| Intiating Actor: | |  |
| Actor’s Goal: | |  |
| Participating Actors: | |  |
| Preconditions: | |  |
| Postconditions: | |  |
| **Flow of Events for Main Success Scenario:** | | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **UC-#06** | | **Retrieving Sensor Log** |
| Related Requirements: | |  |
| Intiating Actor: | |  |
| Actor’s Goal: | |  |
| Participating Actors: | |  |
| Preconditions: | |  |
| Postconditions: | |  |
| **Flow of Events for Main Success Scenario:** | | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **UC-#07** | | **Generating Cost Saving Analytics** |
| Related Requirements: | |  |
| Intiating Actor: | |  |
| Actor’s Goal: | |  |
| Participating Actors: | |  |
| Preconditions: | |  |
| Postconditions: | |  |
| **Flow of Events for Main Success Scenario:** | | |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **UC-#08** | | **Register Property** |
| Related Requirements: | | REQ8 |
| Intiating Actor: | | Landlord |
| Actor’s Goal: | | To register a new property within the database |
| Participating Actors: | | Web Application, Mobile Application, MySQL Database |
| Preconditions: | | The landlord has selected the option to register a new property |
| Postconditions: | | The property information has been successfully registered within the database |
| **Flow of Events for Main Success Scenario:** | | |
| 🡪 | 1. | Landlord selects “add a property” feature from the main page |
| 🡨 | 2. | The system responds by receiving the request and displays a form for the landlord to enter property information |
| 🡪 | 3. | The landlord enters the property information into the form fields |
| 🡪 | 4. | The landlord presses the “add” button after entering the property information into each of the form fields |
| 🡨 | 5. | The system responds by first checking that all of the entry fields have information entered |
| 🡨 | 6. | The system then stores the property information within the database |
| 🡨 | 7. | The system then returns the landlord back to the main page |
| 🡨 | 8. | **(Exit Condition)** The system then displays the newly registered property information on the main page |
| **Flow of Events for Main Success Scenario (Alternate Scenario):** | | |
| 🡨 | 5a. | The system responds with an error message that “all of the required form fields have not been completed” |
| 🡪 |  | (Same as in step 3) |
| 🡨 | 8a. | The system displays the message that “no properties have been registered” |
| 🡪 |  | (Same as in step 1) |

|  |  |  |
| --- | --- | --- |
| **UC-#08** | | **Edit Property** |
| Related Requirements: | | REQ8 |
| Intiating Actor: | | Landlord |
| Actor’s Goal: | | To edit listed property information stored within the database |
| Participating Actors: | | Web Application, Mobile Application, MySQL Database |
| Preconditions: | | The landlord already has a registered property, Landlord has selected the option to edit property information |
| Postconditions: | | The property information has been successfully updated within the database |
| **Flow of Events for Main Success Scenario:** | | |
| 🡪 | 1. | The landlord selects the “manage” option on the property task bar for the property they wish to edit |
| 🡨 | 2. | The system responds by receiving the request and opening the property management page for that specific property |
| 🡪 | 3. | The landlord selects the “edit” option on the property management page |
| 🡨 | 4. | The system responds by displaying the property registration form that was completed during the registarion process which contains the information that was used to generate the property information |
| 🡪 | 5. | The landlord makes the necessary changes to the respective form fields regarding the information they wish to change |
| 🡪 | 6. | The landlord presses the “add” button to confirm the changes they have made to the property information |
| 🡨 | 7. | The system responds by first checking that all of the entry fields have information entered |
| 🡨 | 8. | The system then returns the landlord back to the main page |
| 🡨 | 9. | **(Exit Condition)** The system then displays the newly edited property information on the main page |
| **Flow of Events for Main Success Scenario (Alternate Scenario):** | | |
| 🡨 | 7a. | The system responds with an error message that “all of the required form fields have not been completed” |
| 🡪 |  | (Same as in step 5) |

|  |  |  |
| --- | --- | --- |
| **UC-#08** | | **Delete Property** |
| Related Requirements: | | REQ8 |
| Intiating Actor: | | Landlord |
| Actor’s Goal: | | To delete property information stored within the database |
| Participating Actors: | | Web Application, Mobile Application, MySQL Database |
| Preconditions: | | The landlord already has a registered property, Landlord has selected the option to delete property information |
| Postconditions: | | The property information has been successfully deleted within the database |
| **Flow of Events for Main Success Scenario:** | | |
| 🡪 | 1. | The landlord selects the “manage” option on the property task bar for the property they wish to edit |
| 🡨 | 2. | The system responds by receiving the request and opening the property management page for that specific property |
| 🡪 | 3. | The landlord selects the “delete” option on the property management page |
| 🡨 | 4. | The system receives the request and responds with a dialogue box asking if they wish to confirm the deletion of the property |
| 🡪 | 5. | The landlord confirms the action to delete the property information using the dialogue box |
| 🡨 | 6. | The system then returns the landlord back to the main page |
| 🡨 | 7. | **(Exit Condition)** The system then removes the property information from the main page |
| **Flow of Events for Main Success Scenario (Alternate Scenario):** | | |
| 🡪 | 5a. | The landlord does not confirm the property deletion from the dialogue box and cancels the descision to delete the property information |
| 🡨 |  | The system then returns the landlord back to the main page |
| 🡨 |  | **(Exit Condtiion)** The property information is still displayed on the main page and has not been deleted |

Section 5.0 - User Interface Specifications

* Preliminary Design

Section 6.0 - Test Cases

* Test Case 01

Section 7.0 - References

* Reference