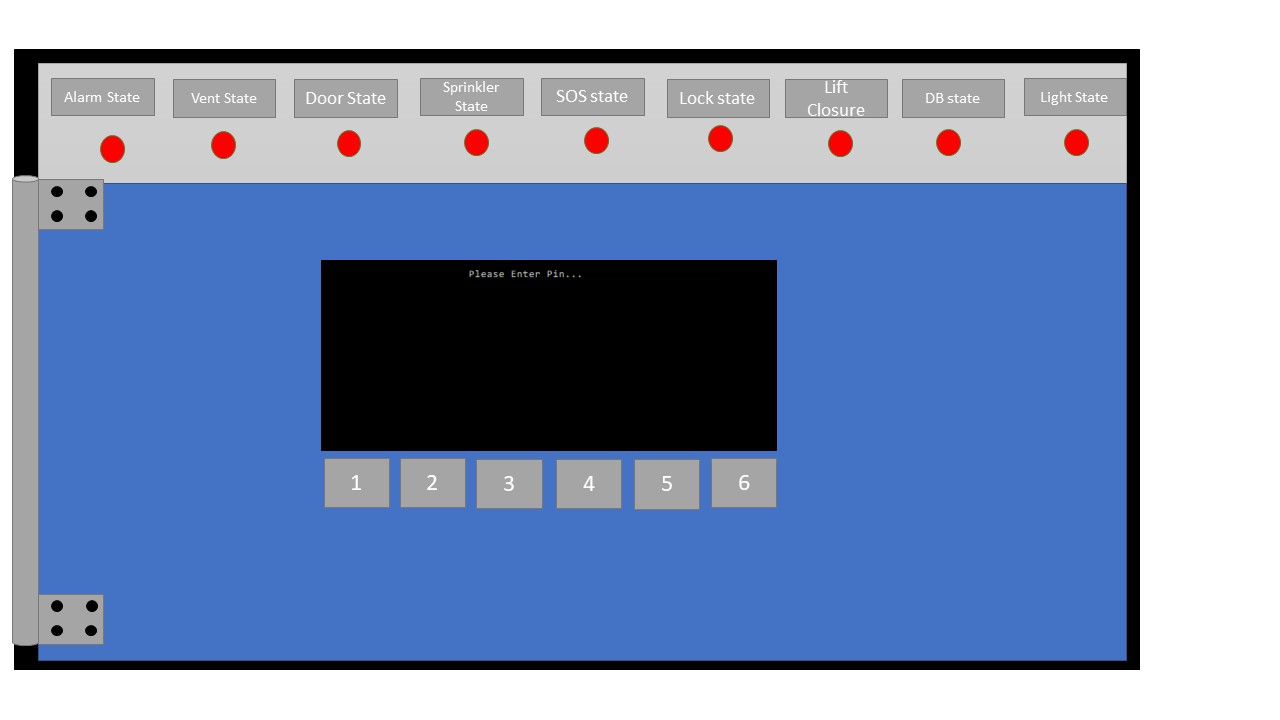
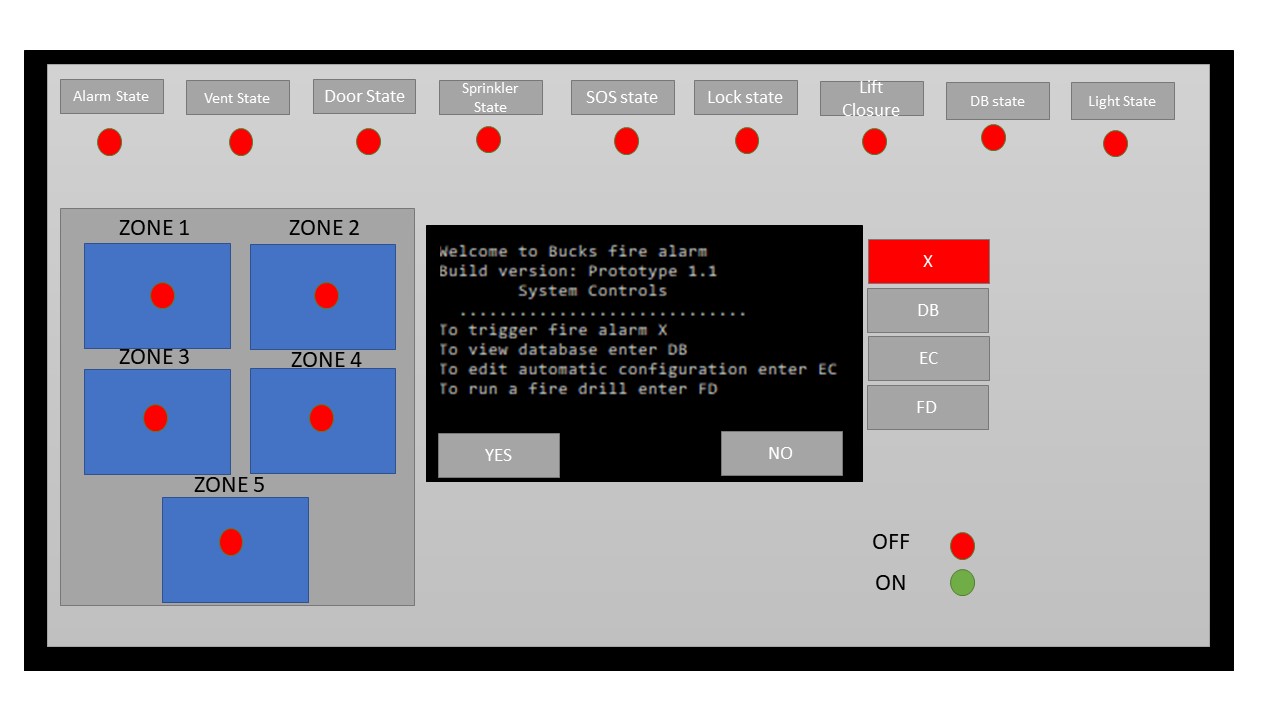
FireSystem Explaination

User Interface Prototype

Locked Fire Panel

This is a mock-up of what the fire panel will look like in its state located on the wall. From the outside the user can see which settings have been configured so if there was to be a fire alarm set off they can understand which procedures will follow. To get further access the user must enter a pin – this is to stop anybody who is unauthorized to use the fire alarm system administration panel from triggering an alarm or change any settings.

We decided on having a pin rather than a key due to the nature that having the key always on hand is not the smartest idea – for instance if fire settings need to be changed and the administrator does not the key or has lost it they will need to get a new one – but if they just need a pin they don’t need to carry anything around.

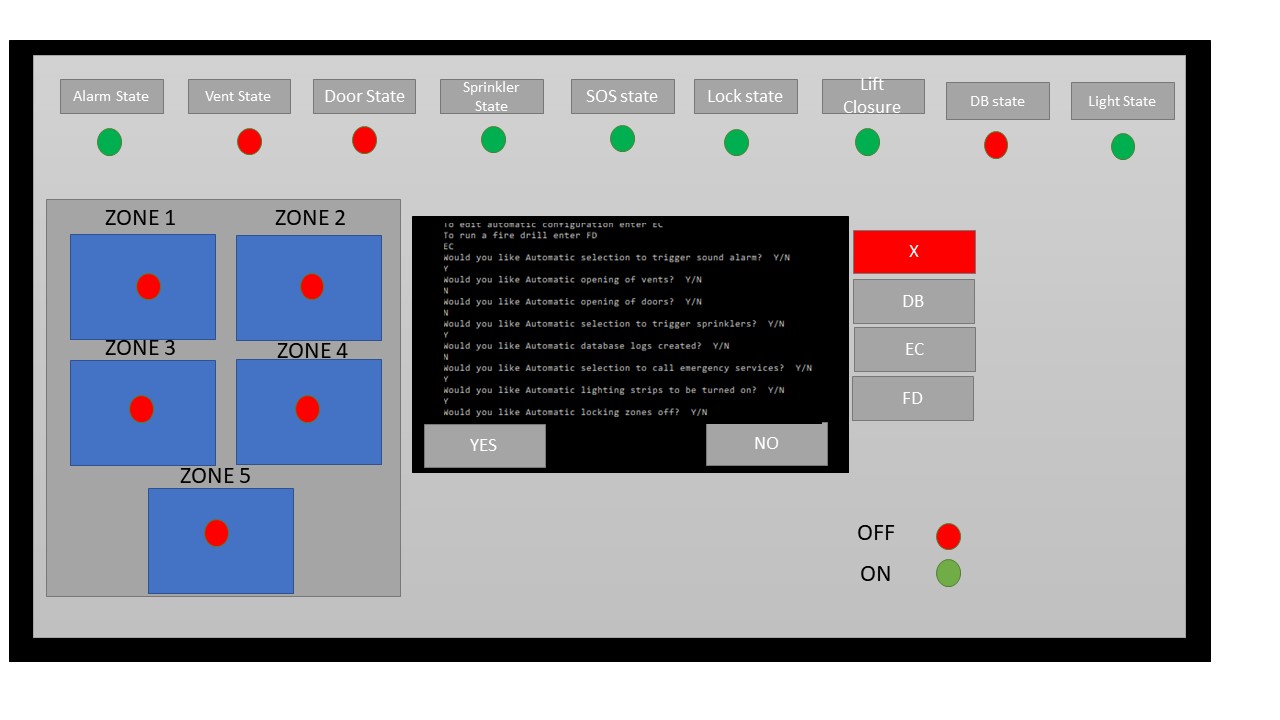
Unlocked Fire Panel

This is what the user will see when they unlock the fire panel by entering the pin, the hinges will unlock, and the configuration settings can be accessed. The top row indicates if a procedure is disabled or enabled, to change the configuration the user must select the “EC” button on the right of the user interface. They then will be greeted with all the potential settings where they can turn them off or on. The user has access to view the database (DB), run a fire drill (FD) or run an alarm manually (X). On the left of the panel the user can see all the zones inside of the building, the lights will indicate if there is an alarm active in the zone – the user can also decide if different zones have different settings.

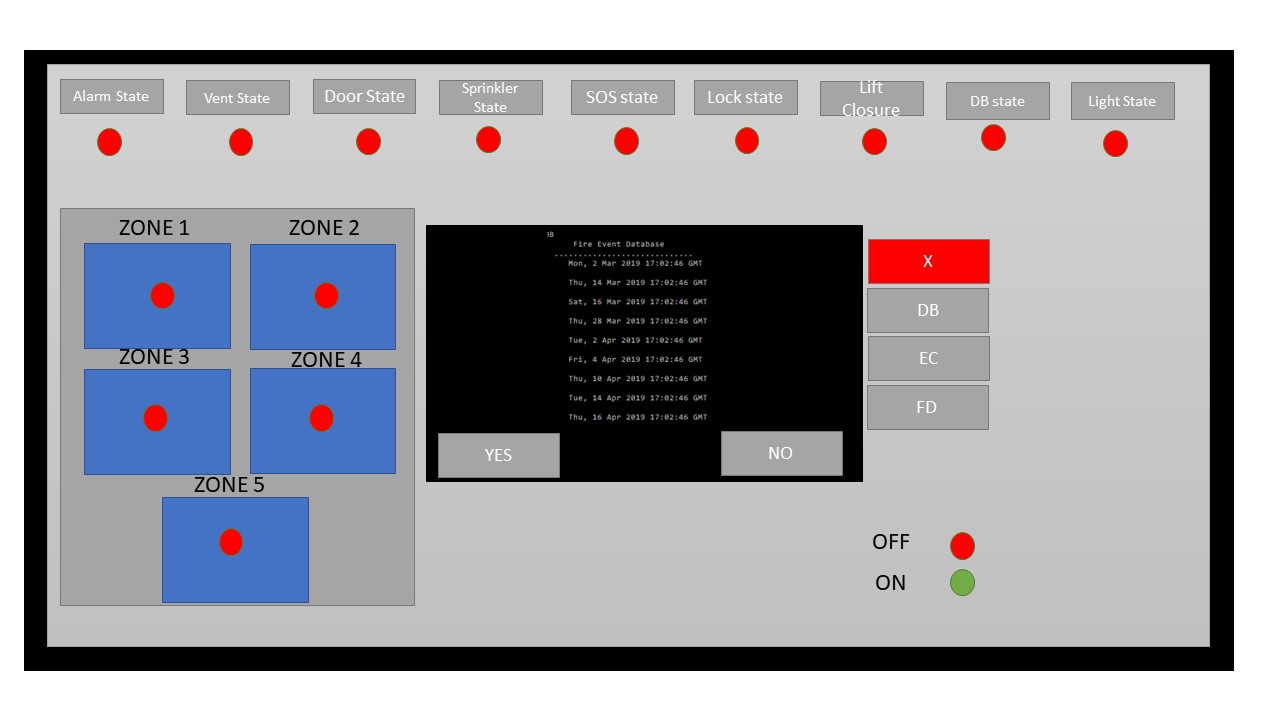
Fire Alarm Triggered

This is what the user interface will look like if an automatic manual alarm has been triggered or the fire system has detected a potential fire (via smoke detectors, etc) this mock-up is assuming that the user has all settings in their active state.

The user can press the red “X” button to disable the fire alarm.

Edit Configuration

This is what the user will see if they select “EC” or edit configuration. This function allows the user to cycle through which procedures they want to happen if they trigger an automatic fire alarm. If the setting is turned off the light underneath the setting will change go to red.

View Database

This is what the user will see if they enter “DB” to view the database. Here they can see the

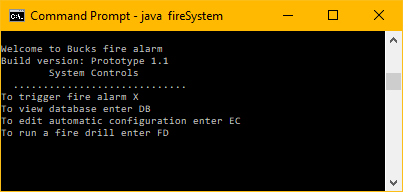
dates of all fires that have occurred with an optional reasoning. If the user has the “Log DB”

function as active after every automatic fire the date will be added to the database.

Establishing Objectives

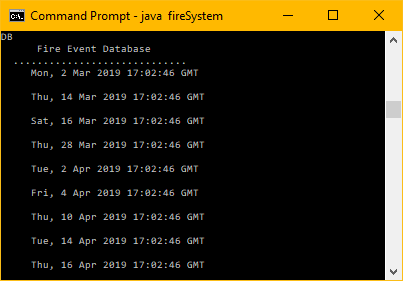
The client in this scenario requested a ‘classic’ fire alarm system with a few additional features such as: Unmanned fire alarms, Sprinkler systems that would not activate if there are people underneath/electrical equipment/directional indicators, automatic door looking, and alarm confirmation. I have addressed each of these factors within my prototype and have added a few additional features that I felt would compliment the current system.

Welcome screen



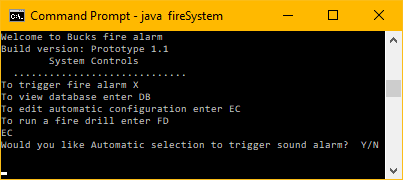
This is the terminal run of the fire system welcome screen – here the user is greeted and can view the initial commands

Fire Database



This is the fire database (“DB”) here the user can see a database of all fires that have occurred since last reset.

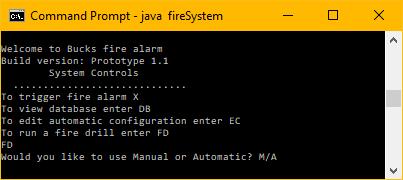
Edit Configuration



This is the edit configuration function (“EC”) here the user can cycle through all the function and decide if they would like them on their automatic fire alarm system.

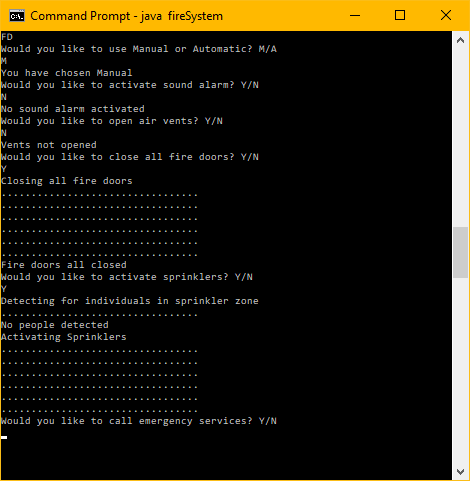
The functionality of this works by using Boolean statements, if the user selects Y on a specific function the variable will change to true.

Fire Drill



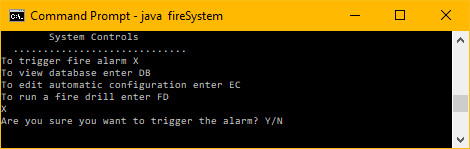
This is the fire drill function “FD” here the user gets to decide whether they would like to run an automatic fire drill (which runs all chosen functions selected from EC function) or a manual drill – where they can cycle through each function and decide to trigger it or not

Manual

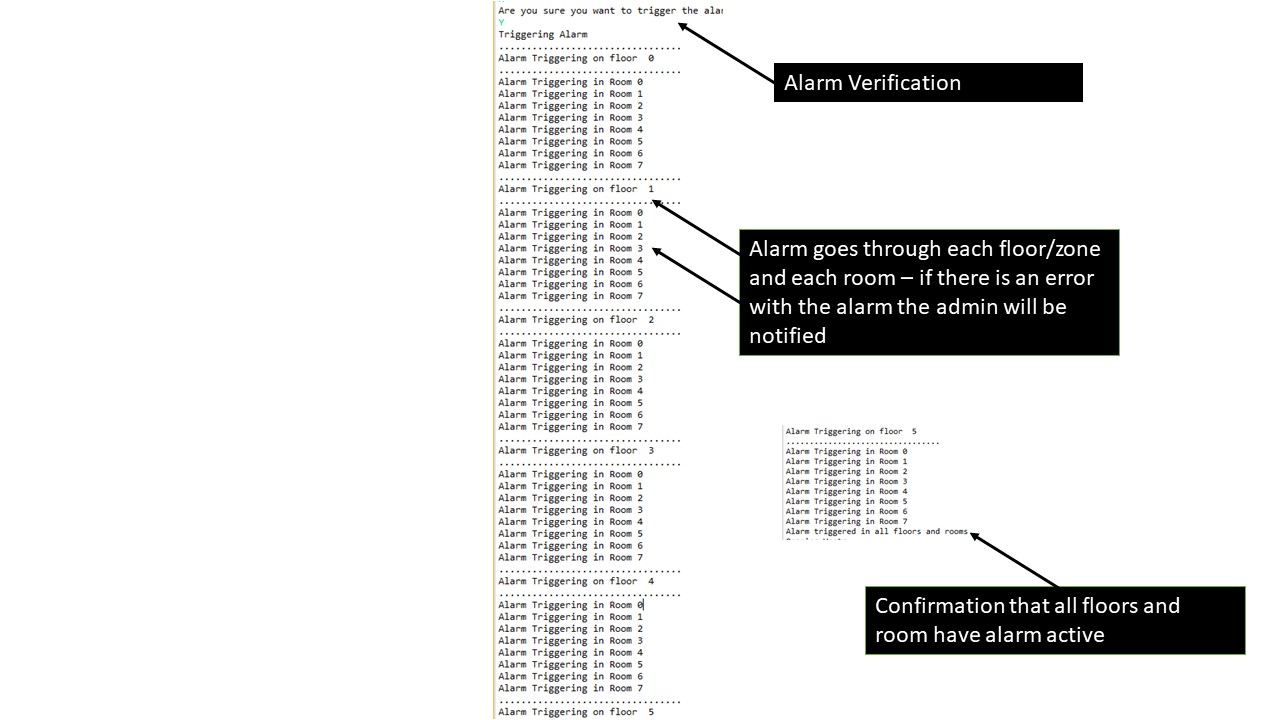


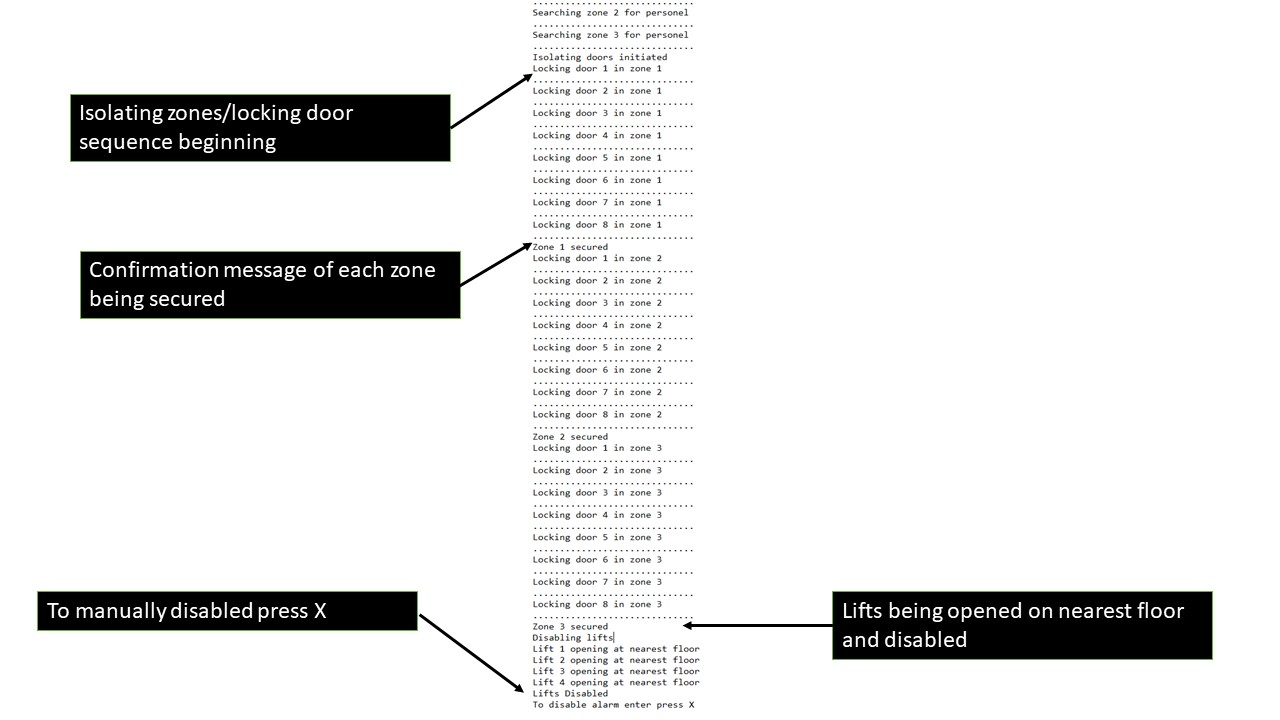
The program cycles through each function and asks the user if they want to active it. – functions such as zone locking off and the sprinkler system have sensors on them – they will not activate if they detect that individuals are inside the premises.

Fire alarm



If the user manually starts a fire alarm they will be greeted with a confirmation message – if they select Y a fire drill will be ran with all automatic functions they have selected.

Fire alarm analysis



What is the goal of the prototype?

The goal of this prototype is to give the customer an idea of what the product that they will be receiving will be like, this is an early stage prototype drawn up and programmed with little resources, so the client can see what the developers have in mind. If the client wants some changed to the UI or the general usability of the prototype they can request changes without any budget/time consequences at this time.

The UI is subject to change if the client’s requirements may differ from the assumed requirements – for instance a pin may not be required if the admin panel is in a secure place that only high privileged certain people can access.

By creating a prototype, the total cost of development has decreased as the features and requirements have been broken down and represented in a mock-up of a final design. The goal of this prototype is to present a concept of the fire alarm system that can demonstrate functionality and layout to the customer and even the design team.

If the customer is happy with this design the development team can use this mock-up as a layout, which would further increase development speed as they will have a general layout/template of what to follow.

This prototype is a throwaway prototype, it is written in Java in a procedural programming format. It is written this way and was created as a throwaway as this takes little as possible development time. The software running on the alarm system is all text based – none of the functions are functional (for instance the trigger alarm doesn’t trigger an alarm), it is built this way just to give the customer a suggestion of what the product will look like and the development team a template of what the product potentially could look like.

Features

The prototypes current list of features are as follows:

* Secure locking system with pin entry
* Visual representation of state of procedures
* Visual representation of each zone with status
* Graphical user interface with description of state
* Emergency button

Features of current proposed fire system

Please note that each feature can be disabled or enabled – on Automatic sequence enter “EC” for edit configuration to disable/enable specific features, or on manual cycle through and choose each feature.

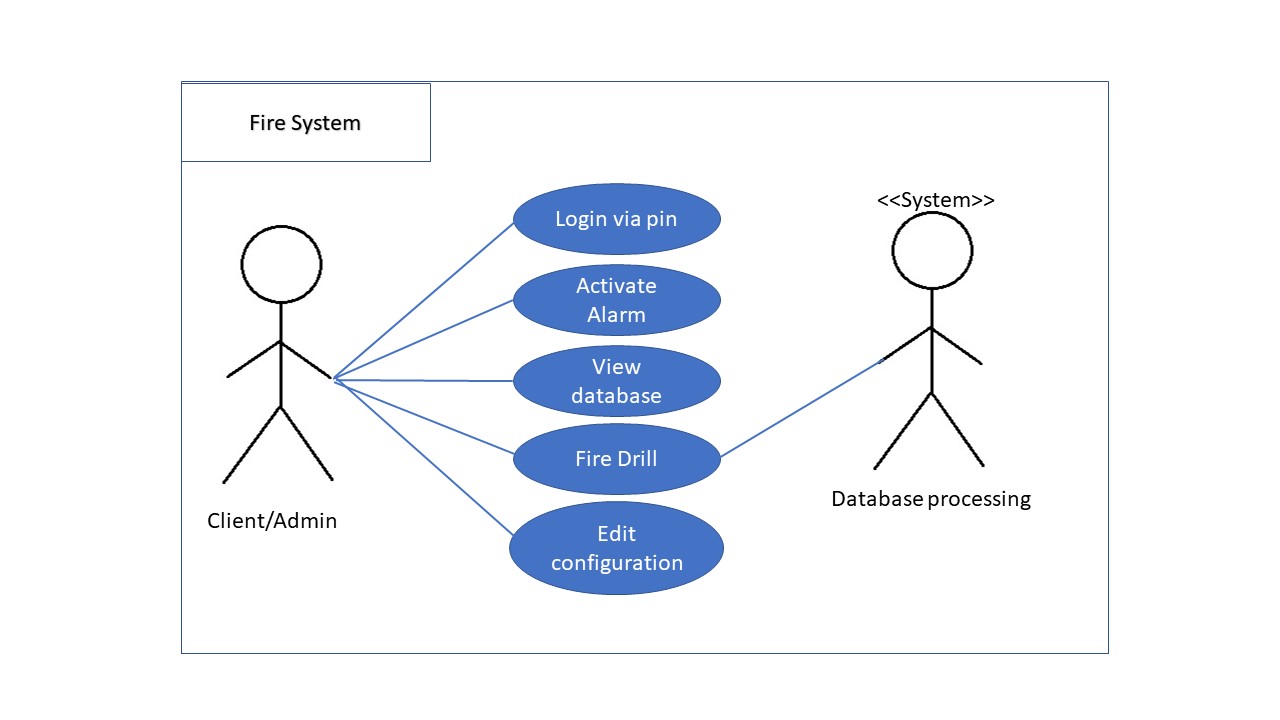
* Alarms in each room
  + Text based & Visual tracking of such in each room and zone
  + Verification of alarm in each room and zone
  + Error checking if vent opening sequence fails
* Vents on each floor
  + Text based tracking of vent condition and state in each zone
  + Verification of each vent in every zone
  + Error checking if each vent is not closed
* Automatic closing of doors in each room
  + Text based tracking of door state
  + Verification of each door closing in every zone
  + Error checking if doors are not closed
* Sprinkler system in specific rooms
  + Text based tracking of sprinkler state
  + Automatic searching for individuals in sprinkler zone
  + Sprinkler system activate tracking
  + Error checking if sprinklers are not activated
* Emergency services calling (SOS)
  + Text based tracking of SOS system
  + Automatic calling of emergency services
  + Information such as address/time is sent to emergency services
  + Confirmation message when information is sent
  + Error checking system
* Zone locking off
  + Text based tracking of zone locking sequence
  + Automatic searching for individuals in each zone
  + If zones are clear, doors will lock – will be automatically unlocked if personnel detected
  + Error checking if zones are not locked
* Lifts closing
  + Text based tracking of lift closing sequence
  + Lifts will be opened at closest floor
  + Lifts will be out of use
  + Error checking if there is an issue with a lift
* Database logging
  + Text based tracking of database sequence
  + Date/Time and reason (optional) will be logged in database
  + Error checking if there is an issue with database
* Directional lighting
  + Text based tracking of lighting sequence
  + Directional lights leading to closest fire exit will be turned on
  + Error checking if there is an issue with a specific light

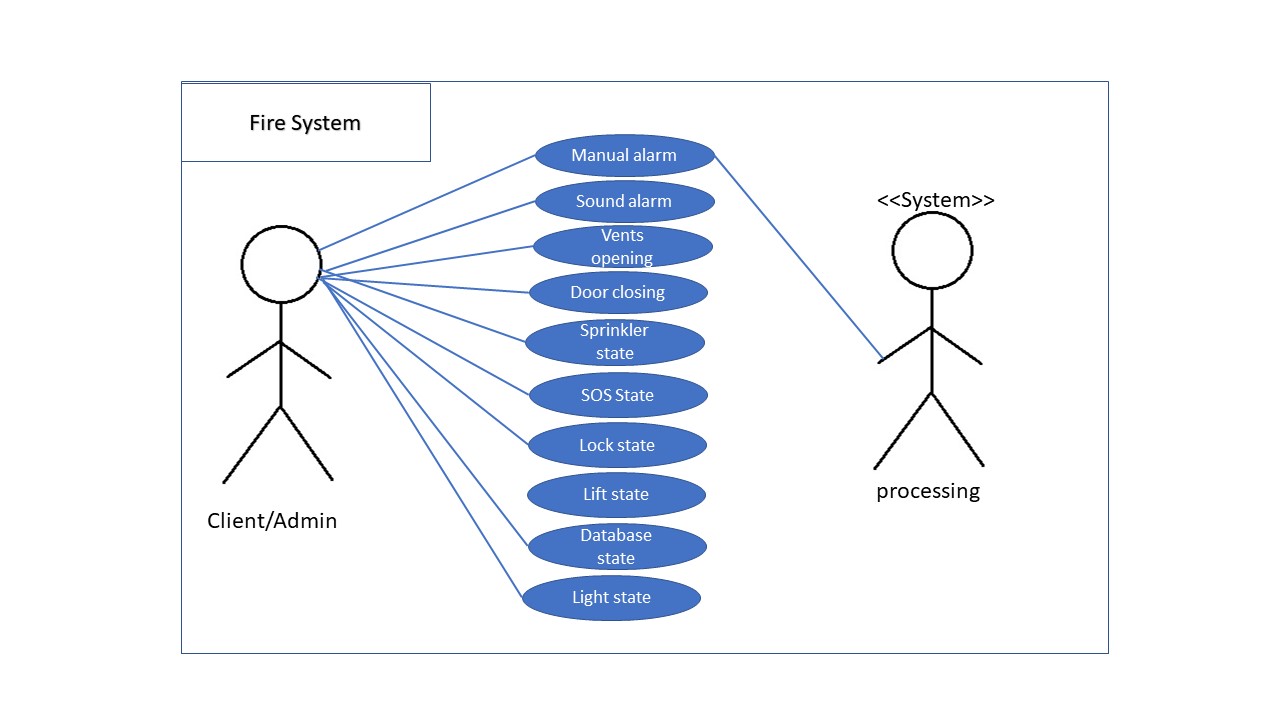
The current proposed fire system is a user-friendly device and will be available 24/7.

There will be a reset function if the customer forgets their pin.

A mobile phone application will be possible if the user requests it.

Use case diagrams



User stories

Student: As a student at Bucks New University I would like confirmation if a fire is a drill or an actual fire – that way I can be sure if it’s a good idea to pick up my laptop before I exit class.

Proposed solution: Text message verification sent to students

Fire Marshal: My main priority is the safety of the people inside the buildings, I feel that if some doors are left open in some classrooms (by chairs at the doors etc) it may lead to the spread of the fire.

Proposed solutions: Mandatory closing of all doors inside the building

Signs on door (Fire door: Keep shut)

Lecturer: I am worried about the safety of my car in the event of a fire I may have no access to the car park via the main entrance

Proposed solution: Back entrance to carpark opened

System Admin: I feel that a pin system may not be the right system in the case of an emergency as I may not be on site and someone else may need to access the fire panel

Proposed solutions: Leaving the case open

No pin system installed

System Admin: If there is a power cut and the fire panel is out of service I want another way of accessing it in emergencies

Proposed solutions: Fire system app

Emergency battery as secondary power source

IT support: If I leave some electronics such as a laptop in a room and then a fire is detected my laptop may be destroyed by the sprinklers

Proposed solution: detecting for electrical devices

Don’t leave laptop in unattended room

Development process

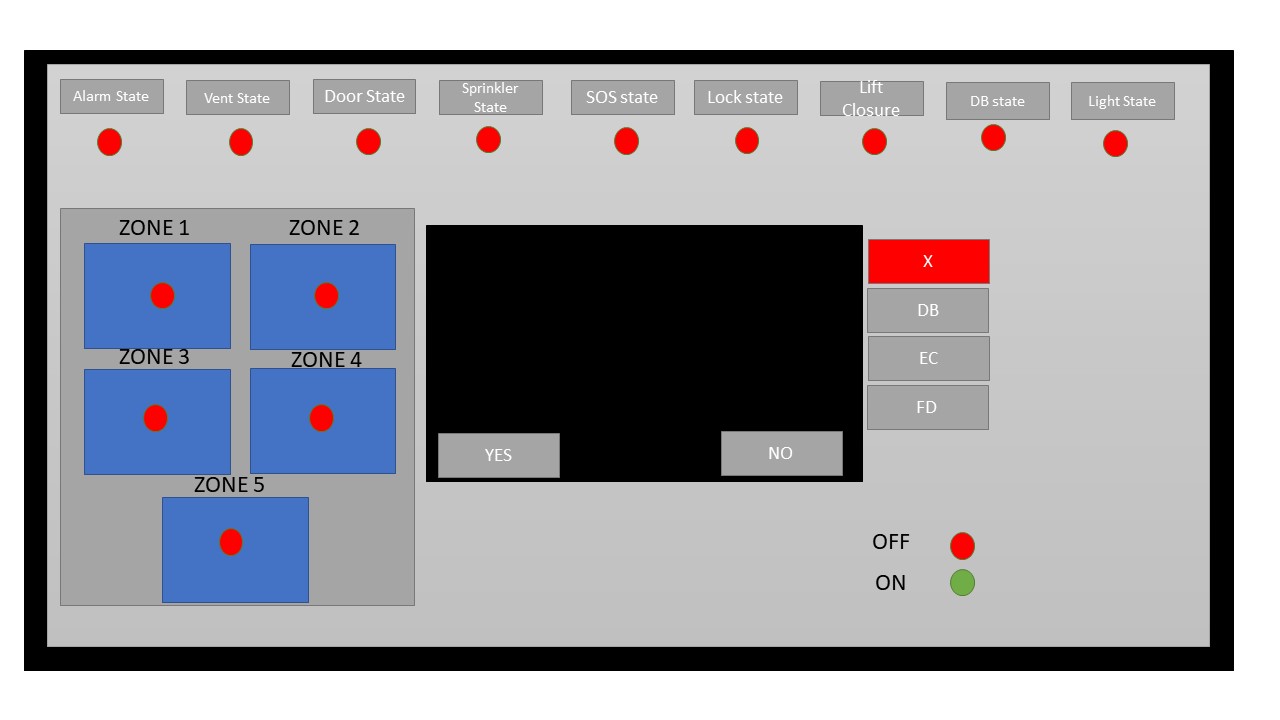
To begin the development process, I read through the client’s description of what they wanted and drew up a diagram of everything that I thought would be necessary for this to be please the client’s requirements.

I then researched what features are necessary for a fire alarm system to function efficiently and safely, I thought about what fire alarm systems in places I have been such as university and workplaces were like.

List of features:

* Alarms in each room
* Vents on each floor
* Automatic closing of doors in each room
* Sprinkler system in specific rooms
* Emergency services calling (SOS)
* Zone locking off
* Database logging
* Directional lighting

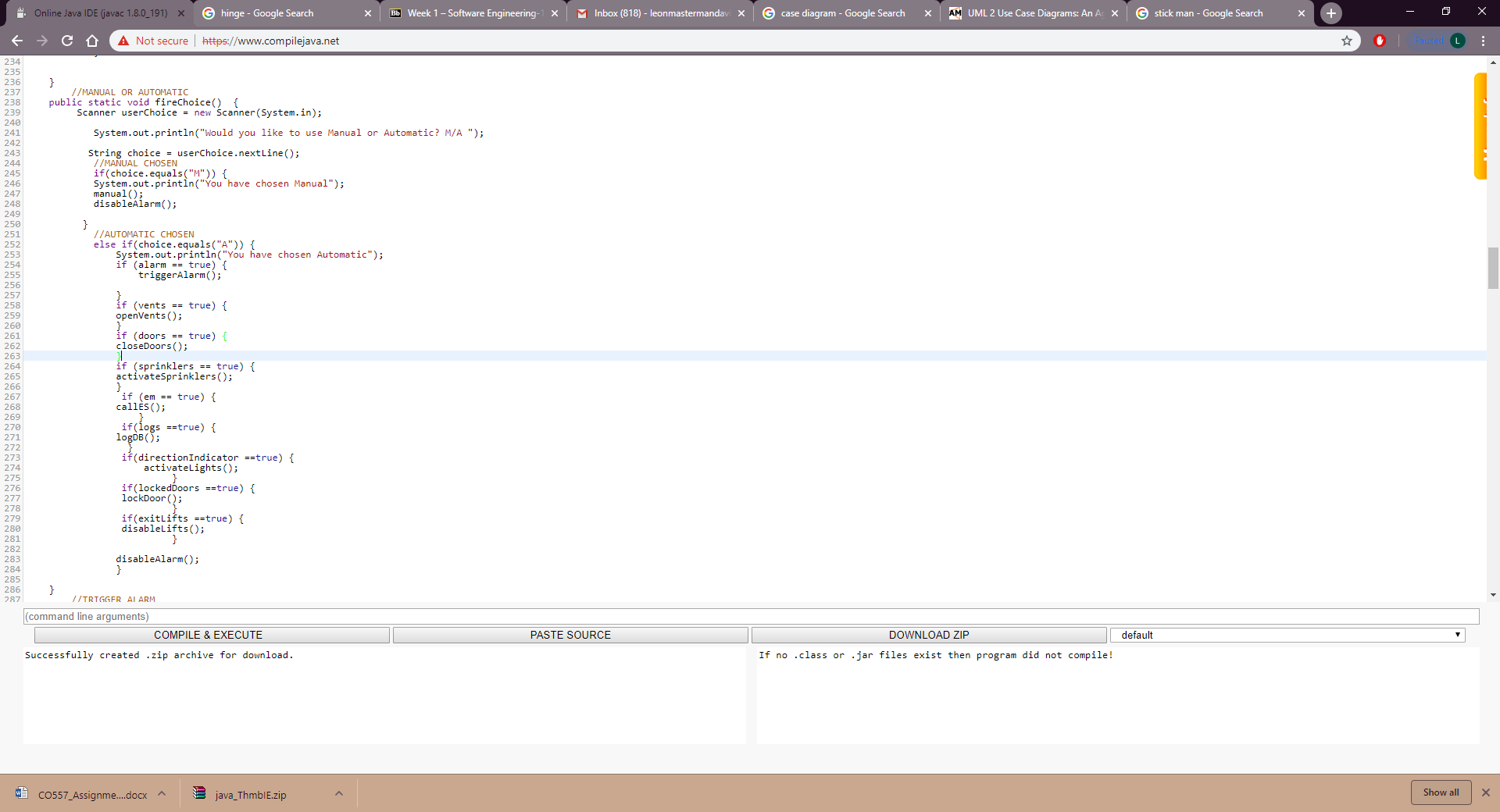
I then drew up a user interface of what I thought the fire panel should look like – this is what I came up with



I wanted the user to be able to see which zones the alarm is trigged in and the status of each function – whether it is switched on or off.

I wanted the user to not be overwhelmed with buttons, so I decided to only implement 4 buttons that I thought were necessary for navigation of the terminal as well as a YES and NO button.

I decided to make the alarm button red to stand out to the user – once pressing a confirmation messaged is displayed where the user will have to press YES if they wish to activate the alarm.



Above is a sample of the code I wrote for the prototype. I made sure to write comments after each function and change if course to make is easier to understand if I come back to it or if somebody else wants to understand my code.

I wrote the code in Java as a I felt that it is the language that I am the fastest at programming in. I did not want to waste resources by spending too much time programming, therefore I also chose to write procedurally.

I reused my code several times for each function – as It is all text based not much had to be changed for each function.

In the finished product I may chose to develop the program in C if the client wishes for us to manufacture the alarms/sensors/sprinklers as well – this way I can create them with Arduino boards and program them accordingly myself.

Evaluation

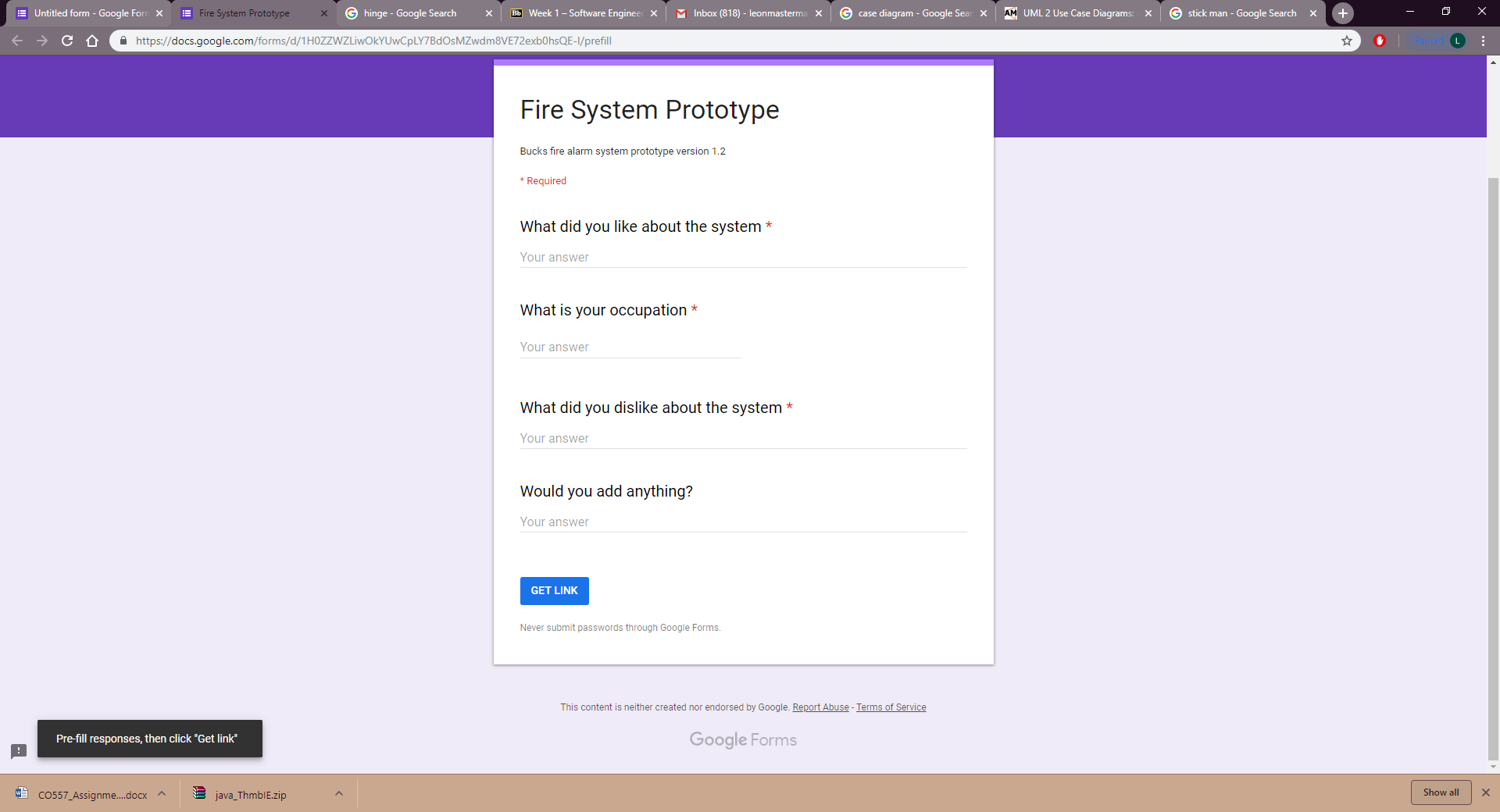
Personal thoughts: I feel that the customer will be happy with the proposed prototype as it meets all their requirements plus some extras that I thought would compliment theirs. The user interface is easy to understand and navigate and the programming will make sense even if the client does not understand Java.

Criticisms: I feel that the customer may not like the order of sequence in the automatic selection, calling emergency services may be the priority but this can easily be changed in the final design.

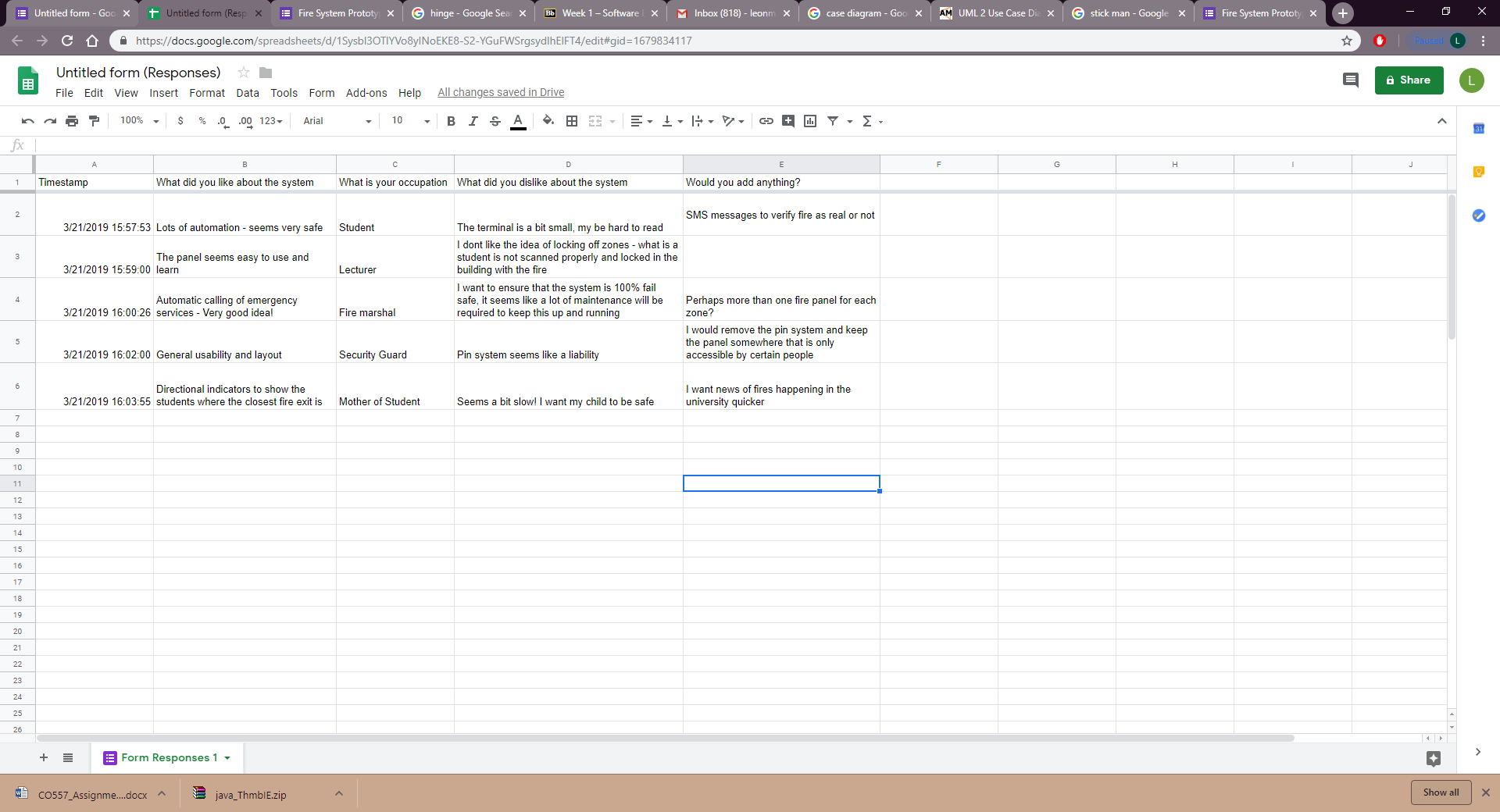
The customer may not be like the pin system to unlock the panel as this could be a potential liability.

Questionnaire:

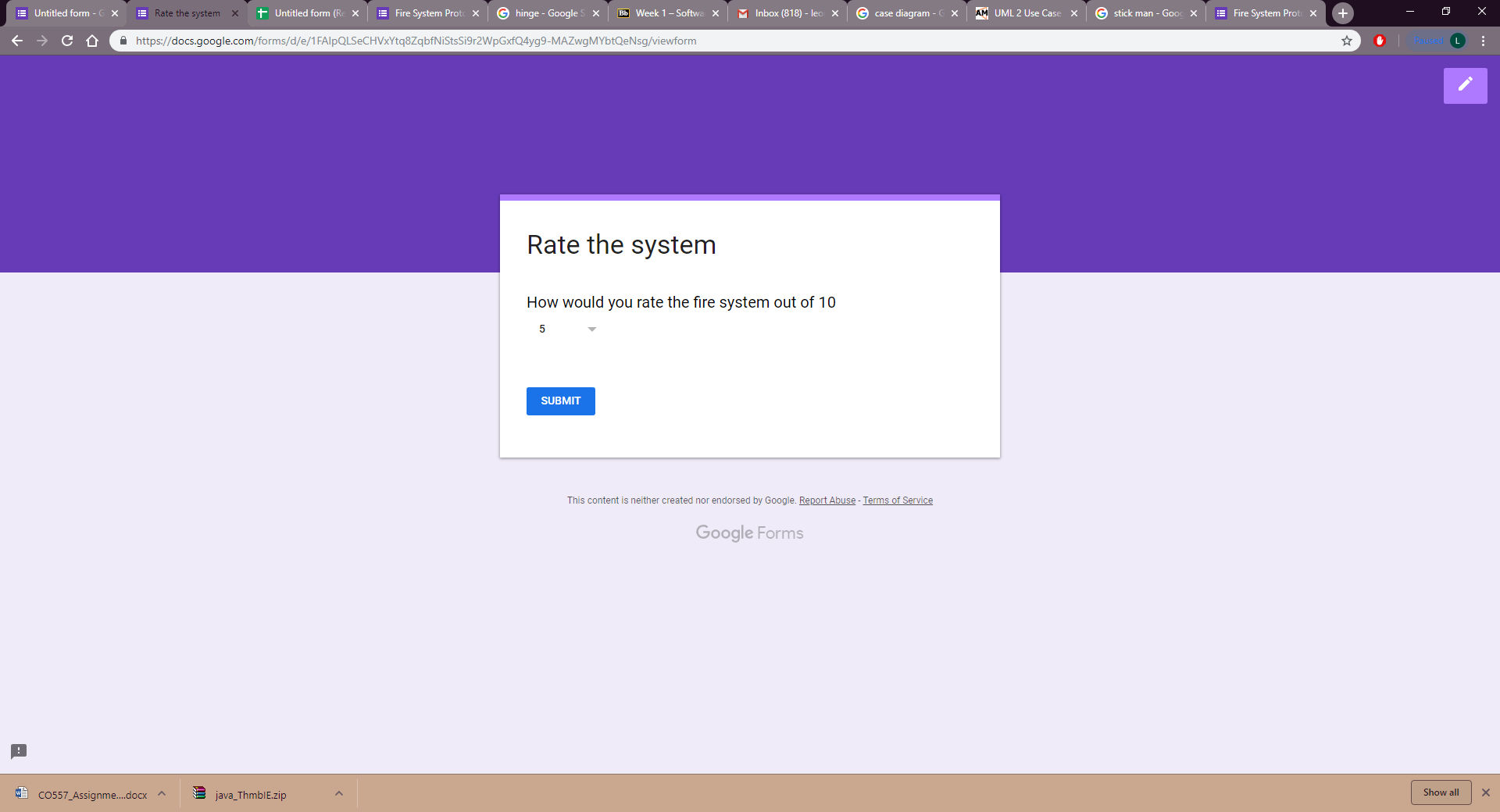
A questionnaire was given out to stakeholders that have seen the system and would be affected by the fire alarm system such as students and lecturers



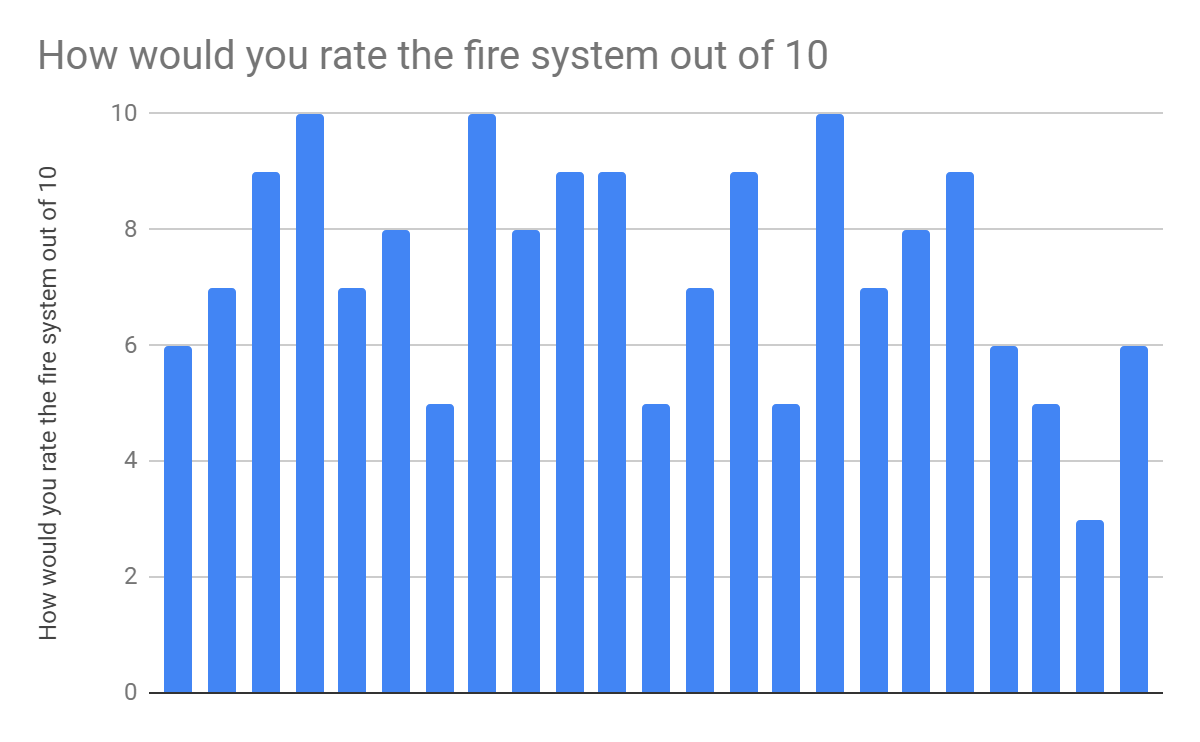
Results



Questionnaire 2:



Results



Average rating: 7/10