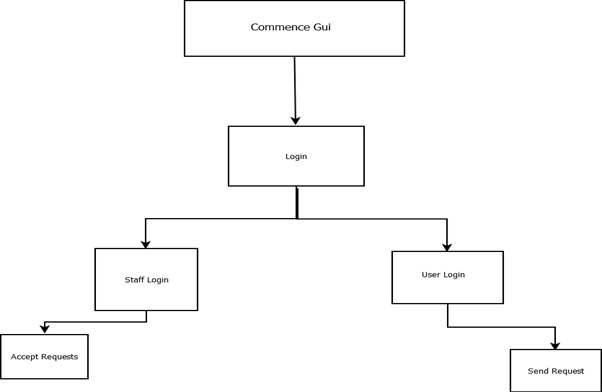
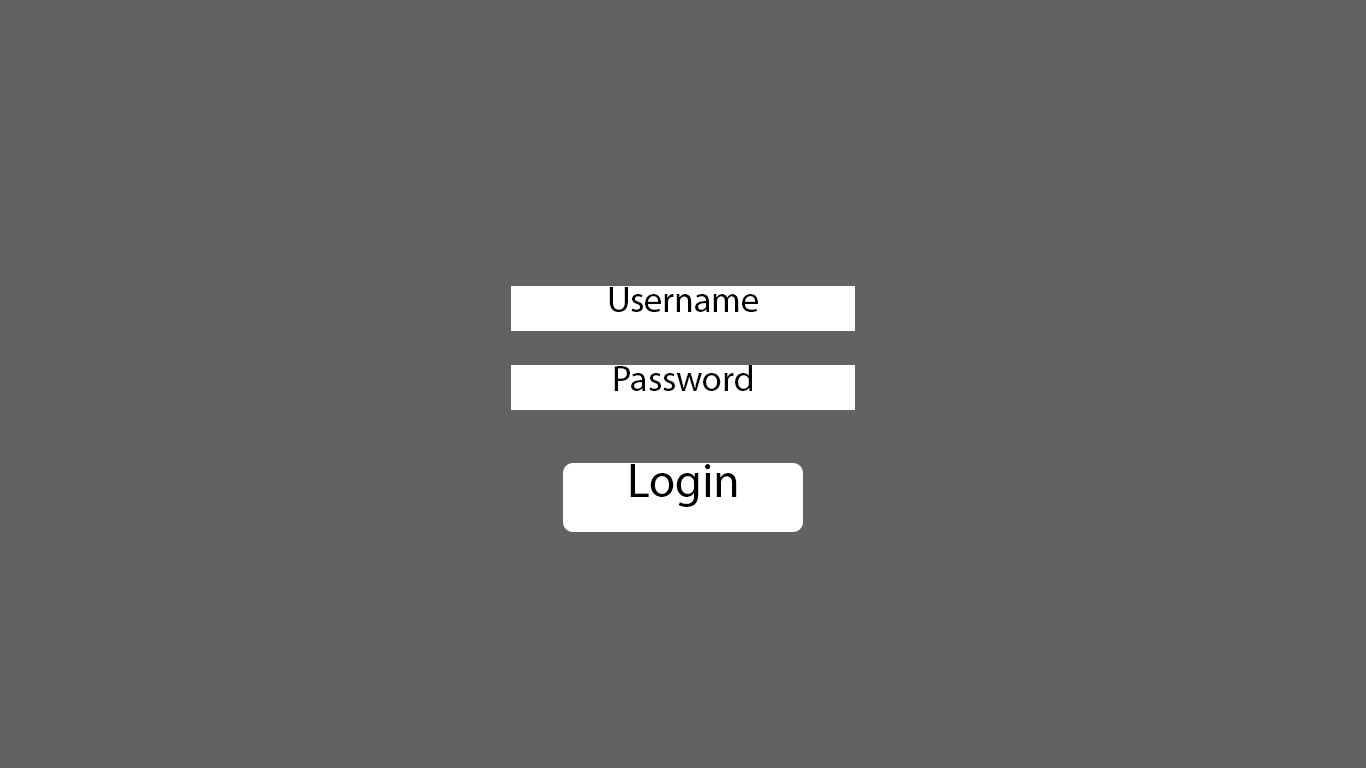
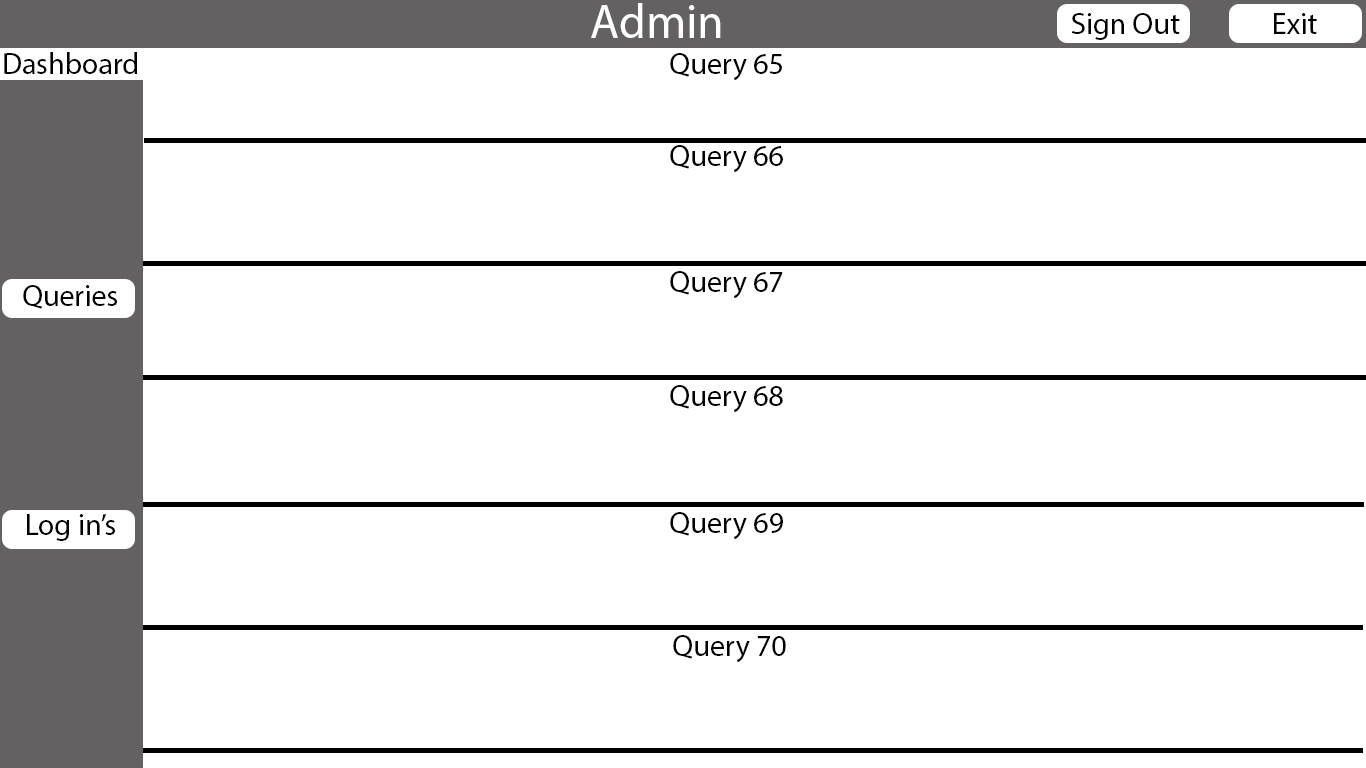
**Property of Architecture Alternative**

**GUI Design**

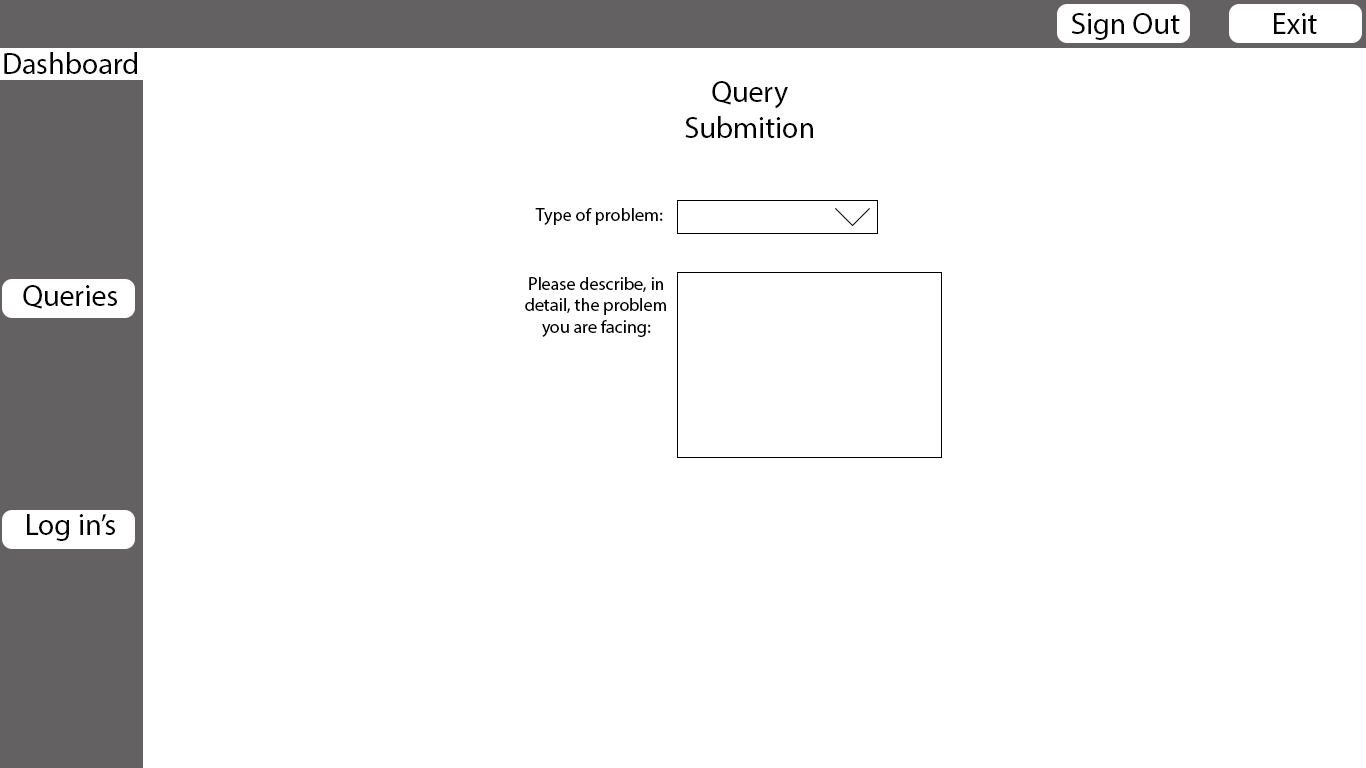
**initial Designs**

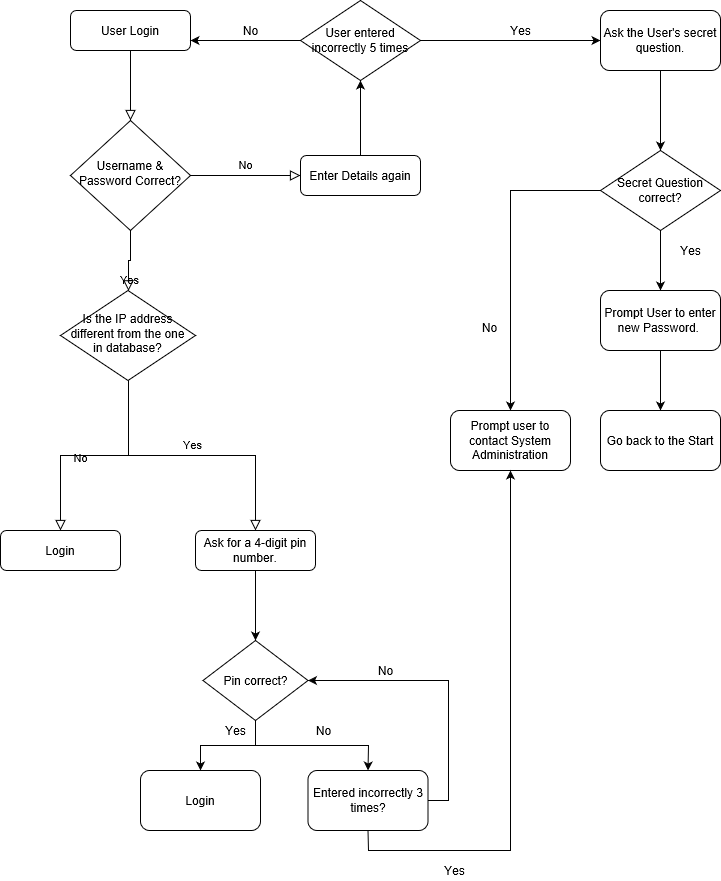


This was our initial GUI flowchart that we decided upon when our team was given the role being the System Admin and after we understood our task of creating a System Admin Support Request GUI. Since the main purpose of the GUI was for users to be able to submit queries to the admins to have them review it and provide the assistance. So, our first step was to allow both user and admin to login to their account to be able to perform different functions depending on the type of account (either user or admin). And thus, we decided to have two separate login menu for the two types of accounts that leads to two different GUI menus.

By the time that phase 1 seminar came around, we were able to design a draft design of what the GUI would look like. This is also where we had made some changes to the plan of the GUI, we have decided to have only one login menu and from a successful login, determine which type of user has logged in and from that, procced to the correct GUI menu. The above is a draft design of the now one login menu. The above is a draft design of the viewing user queries aspect of the admin menu.

The above is a draft design of a new feature that we decided when creating the GUI design, this is an aspect of the admin menu that allows the admin to view a log of every user that has logged in, used to detect any unauthorised activity.

The above is a draft design of the query submission menu that a user will use to submit a query for the admin to review.

Upon reaching phase 2, we were also tasked to research a vulnerability that could affect the security of our GUI and design a system that we could implement that will used as a measure to prevent the chosen vulnerability. We have chosen to research about Social Engineering as we feel that this will have the most impact to our GUI compared to others. Therefore, as a measure to tackle this, we had designed a Knowledge-based Authentication system that checks the IP address of the device on a successful login attempt to determine whether that device has previously logged on to this account. In addition to that, the system will also record the amount failed login attempts and after a certain number of attempts have been made, will proceed to request the answer to a secret question that was chosen upon account creation.

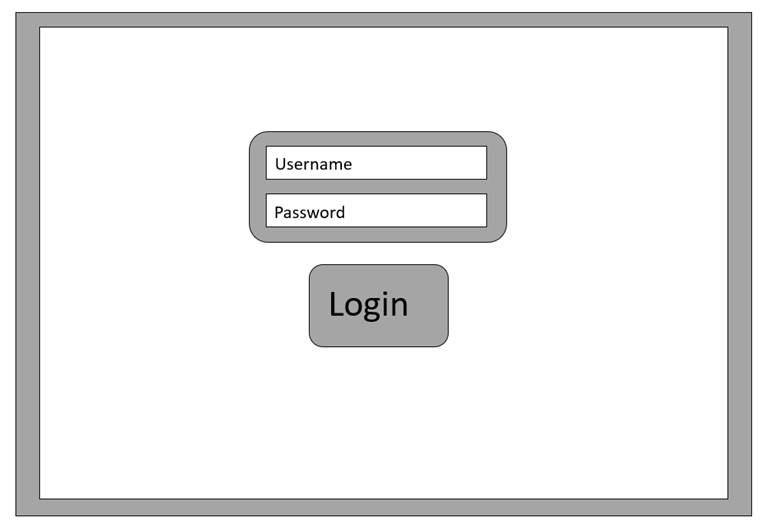
On further introspection, we realise that for future reference, the design of our GUI can be greatly improved. For one, we can implement a more appealing background to the GUI which will be less of an eyesore for the user. To do so, we could have possible used a more appropriate Python module which is not TKInter. On the other hand, we could have taken less of a risk and made the GUI with C# - a language that our development team was more familiar with.

The GUI could use a good layout (several buttons instead of a drop down box) that could be more appealing for the user to navigate with. Python, unfortunately, is not the best language for GUI development. However, without deleting all of our progress, and restarting the development of our GUI from the very beginning, we could instead research different ways with which we can TinKer with the TKInter widgets. This could be distance between widgets (buttons, textboxes, comboboxes, listboxes), size of widgets, using the appropriate colours which can make the GUI more appealing, etc. These are small changed, definitely.

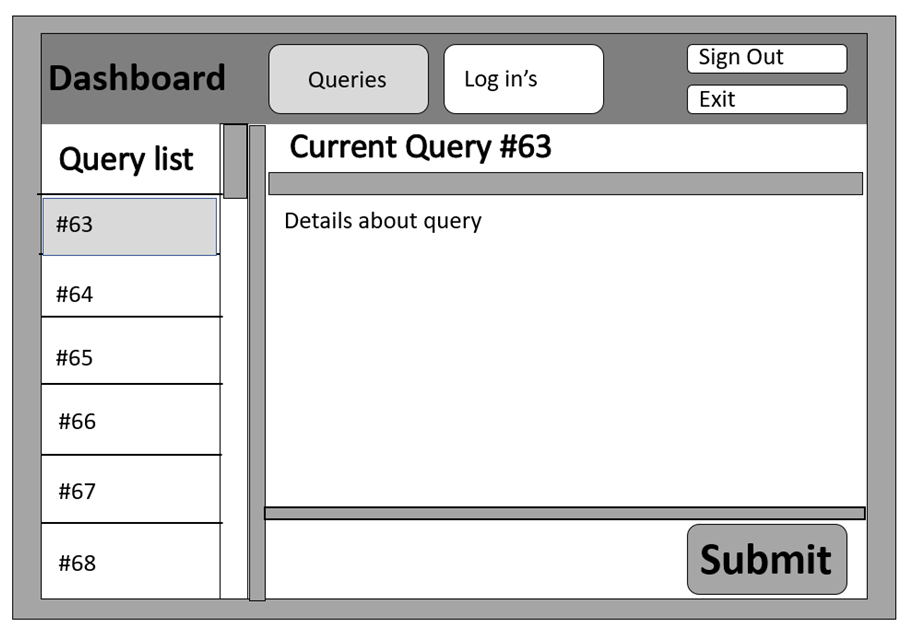
In the long run, however, it would make the product way more appealing to a wider user-base. This is because the average person normally cares more about how their product works, rather than how it functions. They could dismiss the fact that the GUI is milliseconds slower if the design is appealing and easy to navigate.

**Alternate design**

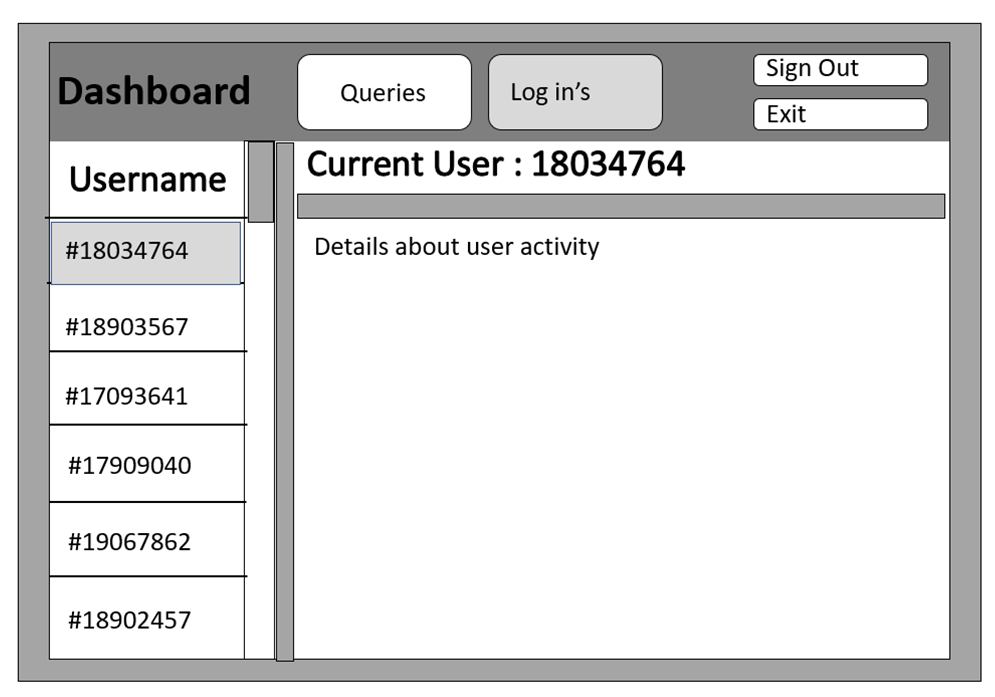
Below is a draft of an alternate design where considering due to complexity we decided to go with a simpler and streamline design. Functionally the user login is no different.



As you can see the query management for admins has had a whole redesign with additional functionality. The admin can check any query they would like to, look at the details and once the problem is fixed, they can submit it as finished.



The same with the Log in’s, the admins can check what users have logging into the GUI and what the activities they have been doing.



Finally, the query submission page for user looks similar and function the same, just matches the new design better.

