

## **Documentation**

School of Computing and Information Technology, University of Technology, Jamaica

CIT3009: Advance Programming

Abbygay Stewart, Winston McLeod, Mikhail Webb, Daniel Eccleston, Andre Grant

Wednesdays 7pm- 9pm & Thursdays 6pm - 8pm (Tutorials)

Mr. O'neil Charles (Lecturer/Tutor)

July 19, 2023

## **Learning trails for each group member:**

### **Challenges faced**

### **Steps taken to overcome the challenges and lessons learned**

**Abbygay Stewart:** The challenge faced in this project was related to the organization and confusion that arose when transitioning from a traditional Model-View-Controller (MVC) architecture with separate files and folders to a single or rather multiple Clientfile and Serverfile setup. In the previous MVC architecture, the code was logically separated into different files (Model, View, Controller, and Connector), which made it easier to manage and understand the project structure. However, in the new approach with only two files, Clientfile and Serverfile, the code became more tightly integrated, and it was difficult to keep track of different sections and functionalities. As a result, adding new code or understanding the existing code became a challenge. Moreover, the absence of specific classes and the use of ClientClasses and ServerClasses separately made it challenging to adapt the UML class diagram as needed. This lack of clear class structure hindered the diagram's alignment with the actual code implementation. To overcome this challenge, I asked team members who were already working on specific sections of the project about the purpose and functionality of existing code to understand where new code should/can be added.

**Mikhail Webb:** The client/server set up is a bit confusing, still trying to grasp the concept as it relates to database management. The hibernate CRUD operations were also tedious, and kept returning errors, for this reason SQL was used. The lesson learned from all this was that it is always better to have an outline of what is expected such as

projects and their respective packages, with their respective classes and attributes. Without a layout, mistakes are most likely to be made.

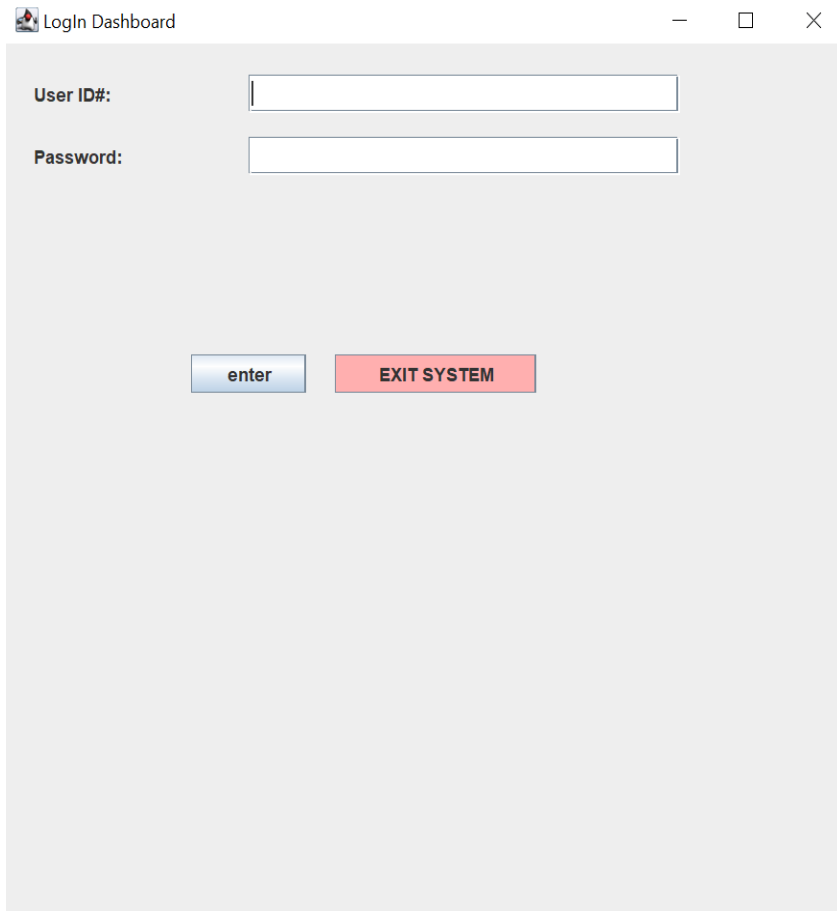
**Winston McLeod:** One of the many challenges faced was developing a real-time live chat functionality for technicians and representatives to communicate with customers. Another challenge faced was creating a dynamic and interactive dashboard interface for both customers and employees. The implementation of a secure authentication process was very difficult with regards to the different user roles i.e. customers, representatives and technicians. In our attempt to overcome these challenges we went through endless rounds of trial and error, extensive research and targeted each feature of the project exclusively.

**Daniel Eccleston:** One of the challenges faced was sending objects across from the client to the server. The slightest type of out of place detail would lead to trials of errors that are time consuming to fix since java error messages are oftentimes very vague and not very helpful at all. The only way to overcome this is a watchful eye and trial and error.

**Andre Grant:** One of the challenges faced in this development was that of configuring the GUI to fit what we had destined. Some layouts were not working and we wanted to add some special features. However we had done our research and were able to resolve each matter with a learning eye.

## User Manual

### Log in Screen



A screenshot of a login window titled "Login Dashboard". The window has a standard OS title bar with minimize, maximize, and close buttons. Inside the window, there are two input fields: "User ID#:" and "Password:". Below these fields are two buttons: a blue "enter" button and a red "EXIT SYSTEM" button.

Login Dashboard

User ID#:

Password:

Fig. 1. Allows the user of the system to log in. Or exit the window if they wish not to continue

## Customer Dashboard

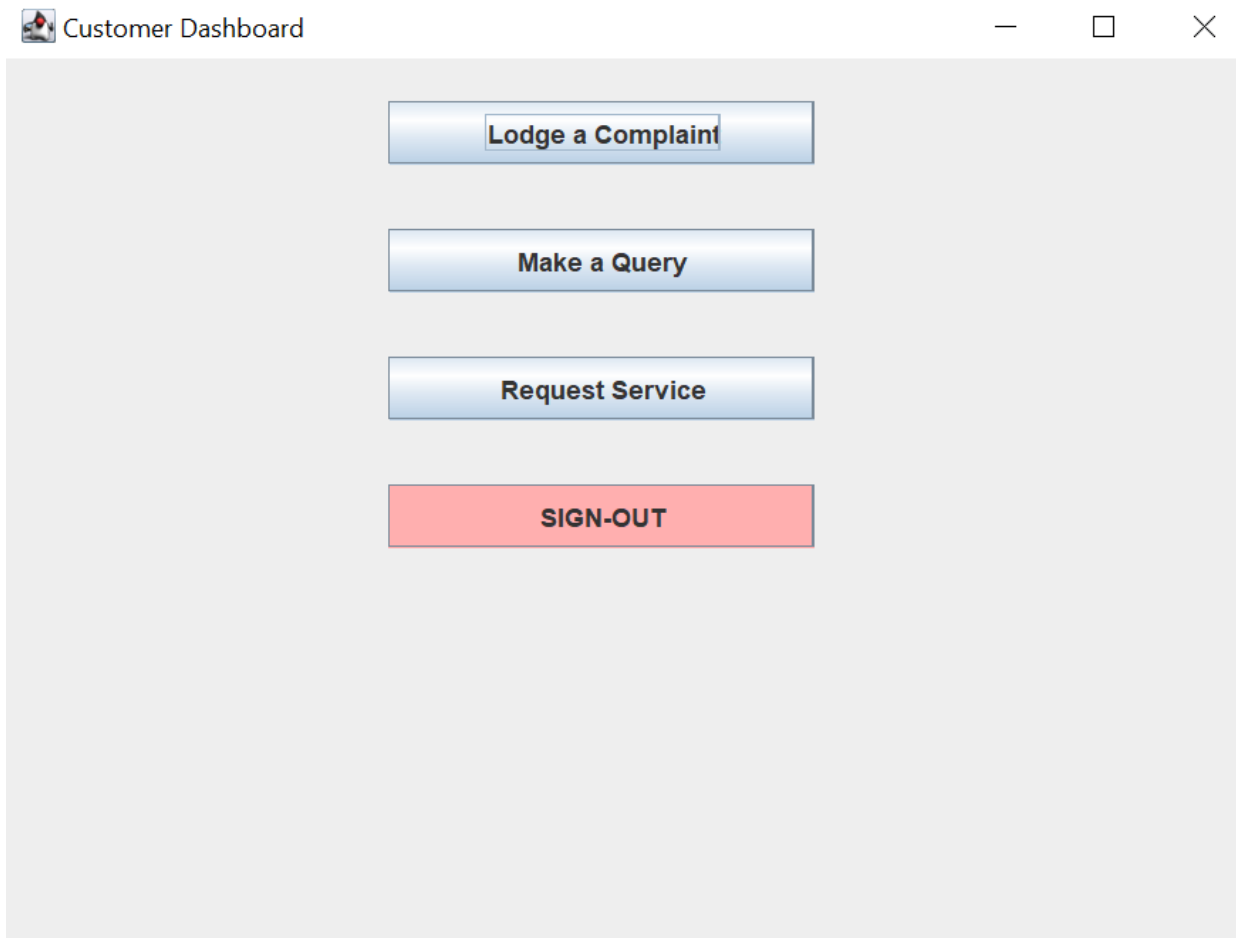
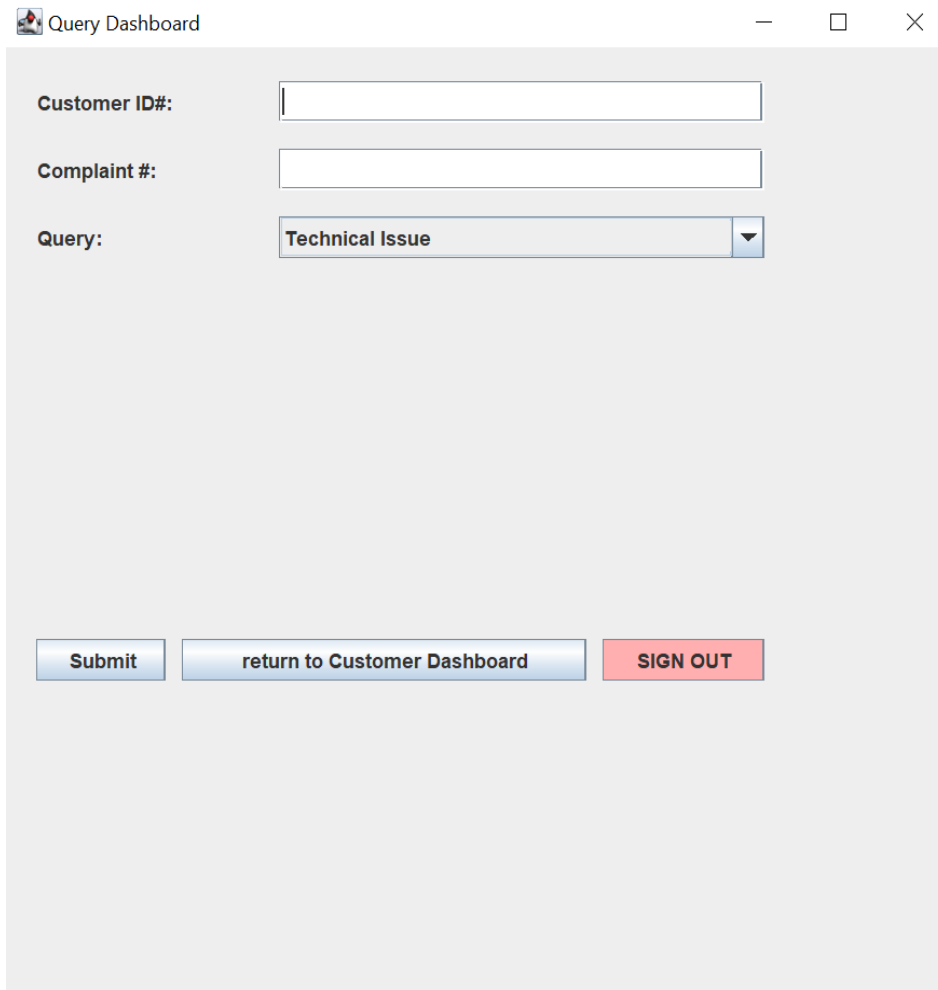


Fig 2. The customer can choose to speak LIVE with a technician using the Request Service feature. Or check on a query, or make a new complaint. If the user wishes to end their session, they can simply sign out.

## Query Dashboard



The image shows a window titled "Query Dashboard" with standard window controls (minimize, maximize, close) in the top right corner. The window contains three input fields: "Customer ID#" (a text box), "Complaint #" (a text box), and "Query:" (a dropdown menu with "Technical Issue" selected). At the bottom, there are three buttons: "Submit" (blue), "return to Customer Dashboard" (blue), and "SIGN OUT" (red).

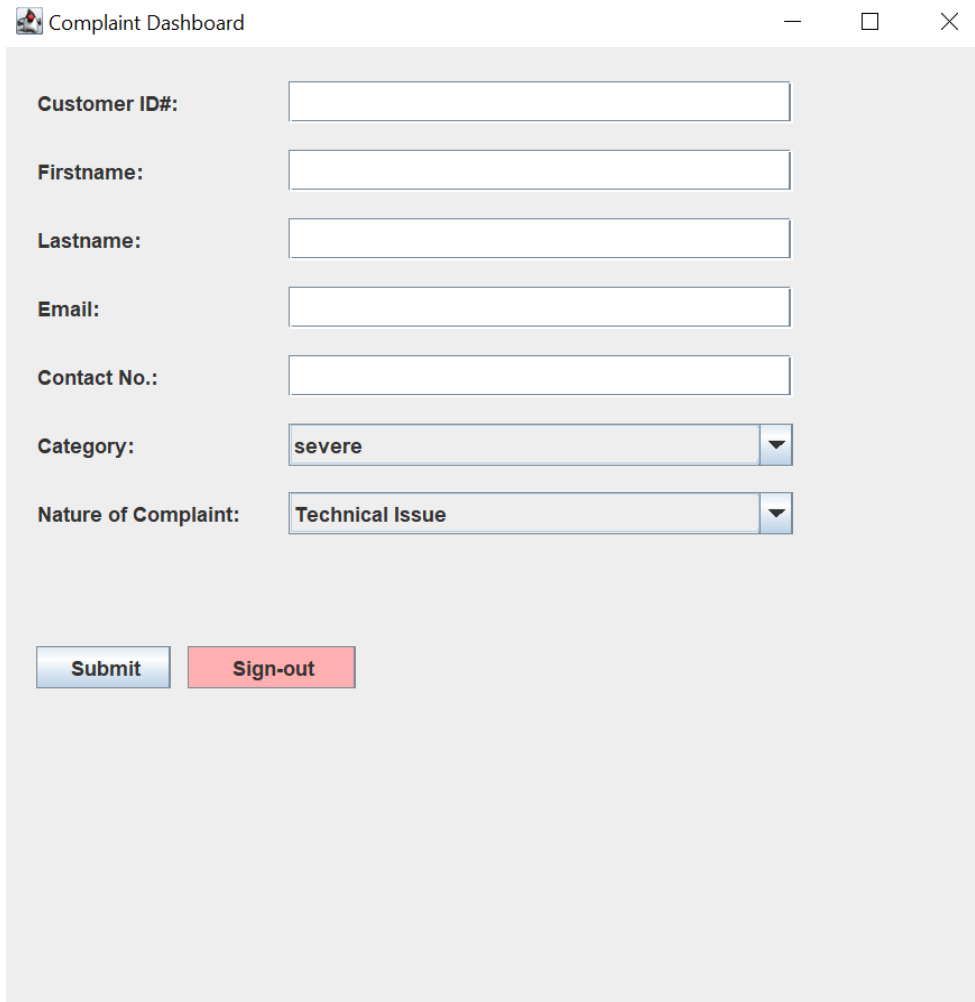
Customer ID#:

Complaint #:

Query:

Fig 3. Previous complaints made by the customer can be later checked on here, by entering the complaint id number that was given to them by a customer care or technician. If the user wishes not to make a query, they can simply return to the customer dashboard.

## Complaint Dashboard



A screenshot of a web application window titled "Complaint Dashboard". The window has a standard title bar with a minimize button, a maximize button, and a close button. The main content area is a light gray form with the following fields:

- Customer ID#:** A text input field.
- Firstname:** A text input field.
- Lastname:** A text input field.
- Email:** A text input field.
- Contact No.:** A text input field.
- Category:** A dropdown menu with "severe" selected.
- Nature of Complaint:** A dropdown menu with "Technical Issue" selected.

At the bottom left of the form, there are two buttons: a blue "Submit" button and a red "Sign-out" button.

Fig 4. Allows the customer to make a complaint or sign-out if they wish not to.

## LIVE Chat

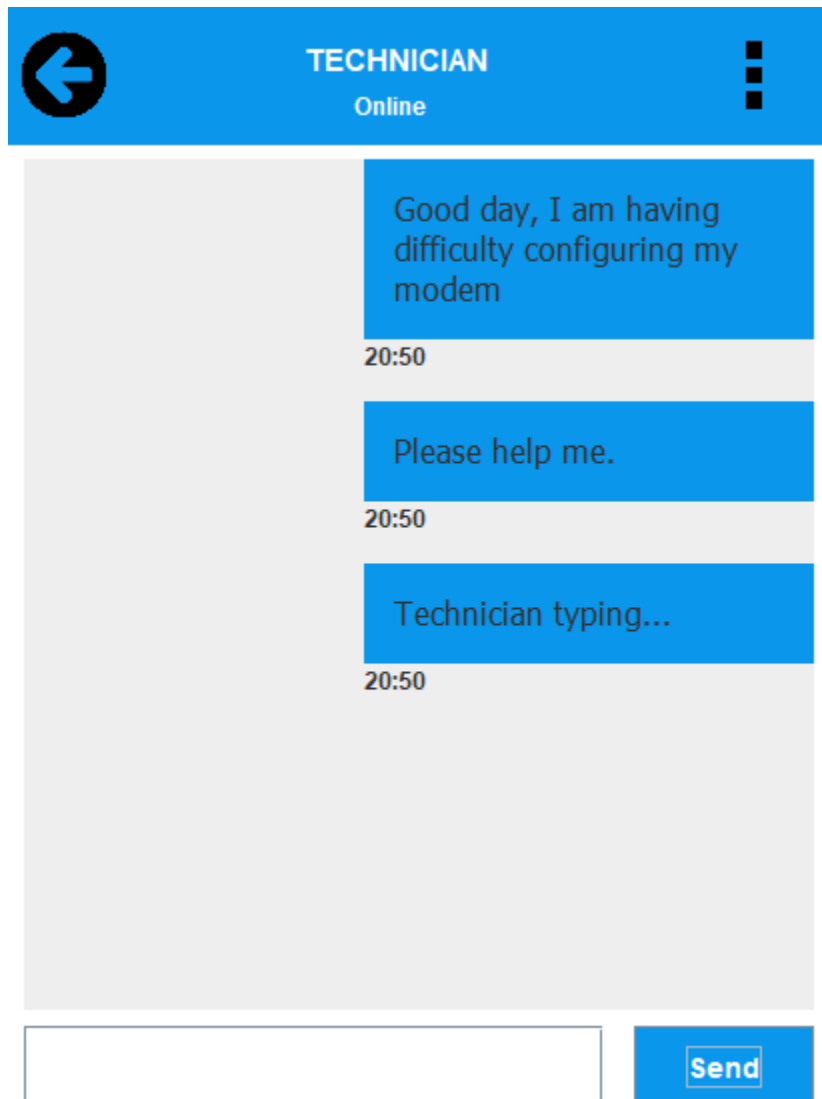


Fig 5. LIVE chat feature between technician and customer



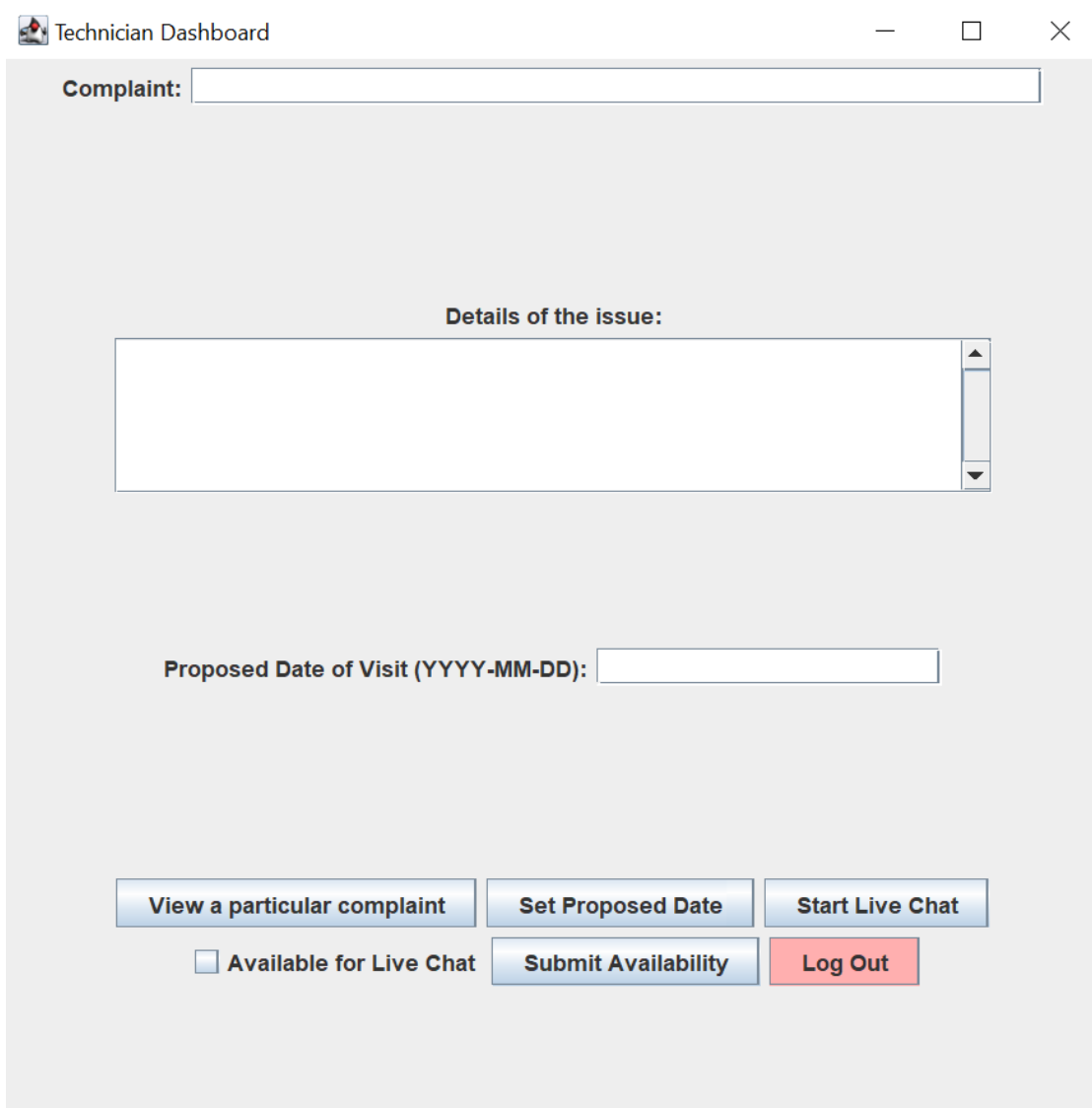
Customer Care Representative Dashboard

 Customer Service Representative Dashboard

View List of Services

View Customer Complaints

## Technician Dashboard



The image shows a web application window titled "Technician Dashboard". It features a "Complaint:" label followed by a text input field. Below this is a section labeled "Details of the issue:" with a large, empty text area. Further down is a "Proposed Date of Visit (YYYY-MM-DD):" label with a date input field. At the bottom, there are five buttons: "View a particular complaint", "Set Proposed Date", "Start Live Chat", "Available for Live Chat" (which includes a checkbox), and "Submit Availability". A "Log Out" button is also present.

Technician Dashboard

Complaint:

Details of the issue:

Proposed Date of Visit (YYYY-MM-DD):

☐ Available for Live Chat

Fig 7. Allows the technician to view a particular complaint and also start LIVE chat