Network Programming Project Report

IQueue123
排隊系統

Group 7:

林興隆 D0708837
雷國亮 D0726433
陳肇海 D0869929

- **Project Goal (設計目標)**

1. Our first project goals are to get the profit from publishing or selling this application to another company, for example:
   - Banks
   - Government offices
   - Restaurants
   - Hospitals, etc.
2. To manage the queue systems more easily

   If we have this queueing management system, it can help us to manage customers' queue experience, and waiting time is an essential part of delighting your customers. The purpose of this page is to give you a central understanding of what a queue management system is, why it's important and how to choose a solution.
3. Develop the system of the counter queue

   By developing these systems, we can get a lot of benefits from it. A queue management system enables you to practically manage customers throughout their interactions with your organization and make that journey as comfortable and smooth as possible.
   - Improve access to services with appointment scheduling
   - Increase the staff mobility
   - Match the right competence to each case
   - Improve service quality

- **System Architecture (系統架構)**

1. We use the Android as our Client
   - Java: we use java for application development, so we can code our android apps using the Java API.
   - Client Socket:
     i. Initialize socket
     ii. Call connects the socket referred to by the file socket descriptor to the specified address
     iii. Handshake
     iv. Send command / receive commands from/to the Server
     v. Close socket and terminate app
   - OpenAPI:
     1. Create the program
     2. Create the openAPI
     3. Connect the program with OpenAPI

4. Server socket:
   - Initialize socket
   - Assigns the address specified by address to the socket referred to by the file descriptor
   - Extracts the first connection request on the queue of pending connections for the listening sockets
   - Handshake
   - Send/Receive command(add queue, call queue, broadcast emergency, etc.) and data(which counter, which number, etc) from/to the Clients
   - Close active sockets to terminate queue services

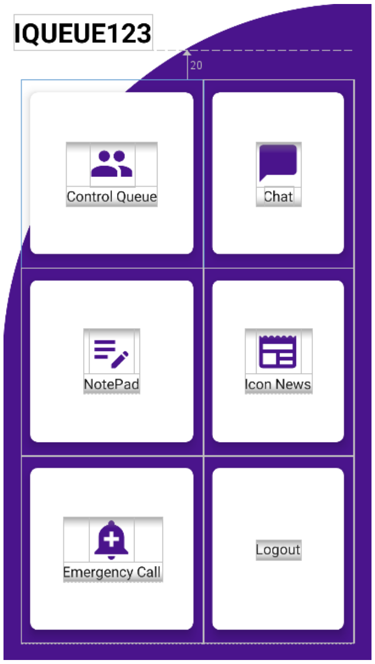
2. Linux Server:

   We use a Linux based server(location-based in Singapore) to serve SQL databases for our app services such as authentication service and storing database service to store and access data. We applied PHP in our app so it can store/fetch databases from/to our server.

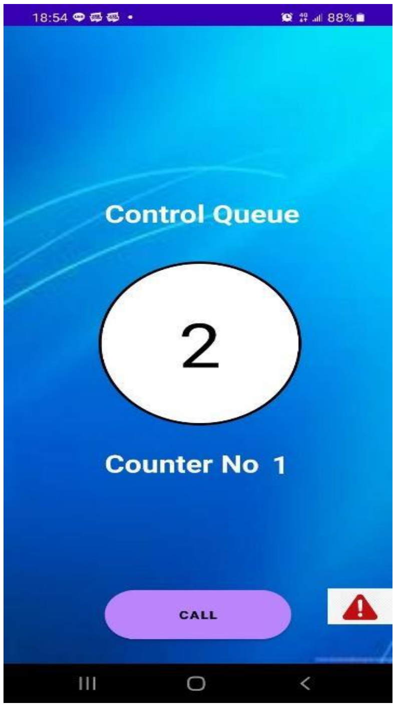
   - In conclusion, we applied cloud-based databases.

3. Desktop (Server)
   - Java
   - Multithreading:
     i. Connecting all clients and threads to 1 Server
     ii. Monitor all the activities file of the clients and send it to main server
     iii. Whenever a new client joins the server, it will be assigned automatically to the server
   - Respond

- **Application Scenario (應用情境)**



This is the homepage/dashboard for our application, and from this. We can directly press the button and it will go to the designed app that we need.



As for the Control and Admin Queue, after we press the button, it will queue directly and it will place us to the counter that the admin has assigned us.

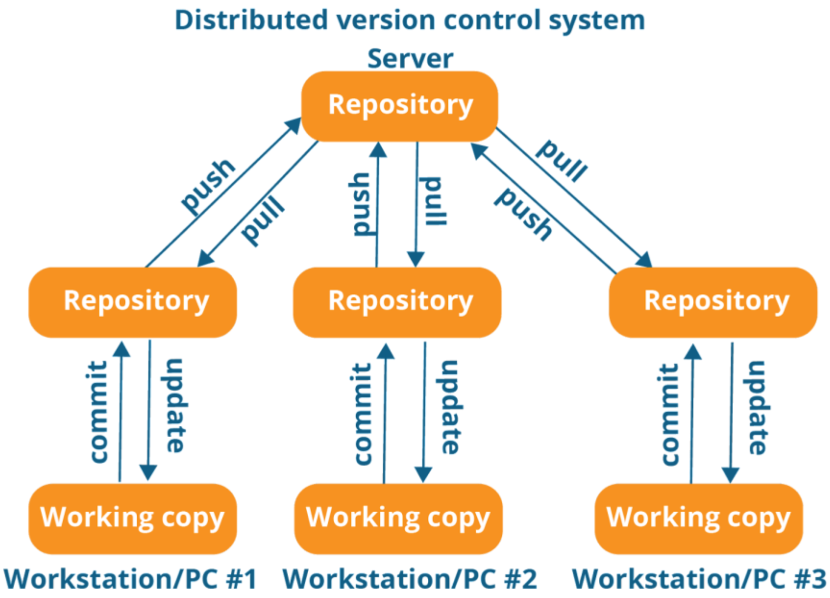
- **Development Environment (開發環境)**

  So, for the development we use a different type of application to make the application.

1. IntelliJ Idea:

   We use IntelliJ to design:

   - Java
   - Socket creation
   - GUI design IDE
2. Android Studio
   - We design the Android UI from this application
   - Backend IDE
3. JetBrains Space
   - This app is like GitHub
   - Provide repositories for push and pull
   - Merge services



So, as we can see from the picture from each workstation, we can do it from anywhere, then we must commit & push the data and put it on the server. After that we can pull & update our data and resume the work.

- **Division of Labor Work (分工項目)**

1. For 林興隆:

   - Client and Server socket (Debug)
   - Authentication
   - Android Backend (Client)
   - Presentation Slide
   - OpenAPI

2. For 雷國亮:

   - Project Initiator
   - Algorithm guider

  - Client and Server socket (Creation)
  - Server (Desktop) GUI and Backend
  - Linux data server provider
  - Presentation Slide

3. For 陳肇海:
  - Android UI Design
  - Presentation Slide
  - Project Report
  - Speaker

- **Project Timeline (執行時程)**

1. Project Planning

We start the project planning on 17 November 2021, we discuss what project we are going to make and distribute the workload based on each member's abilities.

2. Starting the Project Creation & Structure Building

At this point we start it on 21 November 2021 until 28 November.

After we finish distributing the work, we start to make everything that we already discuss, for example, making the Android UI from the easiest part to the hardest part, and setting up the JetBrains Space (Similar to GitHub).

3. Starting the Client & Server

From 1 December 2021 until 8 December 2021.

We start to connect our Client & Server together, from a lot of trial and error, we manage to connect the UI to the Client and Server, and it runs very well.

4. Making the GUI Server

From 7 December to 11 December 2021

雷國亮 start to make the GUI Server for the Desktop and make it Runnable.

5. Finishing the UI (Editing) and Making the Open API

From 17 December 2021 until 29 December 2021

After arriving at this point, we start to Edit the UI to make it look better and make the OpenAPI for the Chat Application that we have.

6. Debugging & Testing

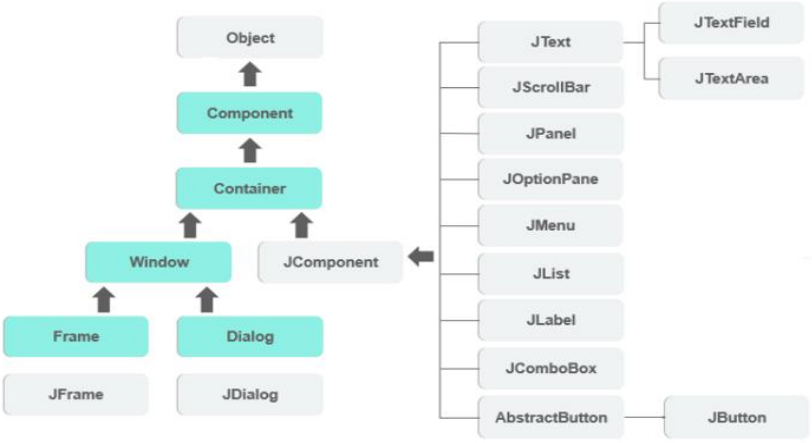
From 29 December 2021 until 1 January 2022

At last, at this point we only try to see if there is still any error from our project and see if the program and application all run well without any error or not.

• Graphical User Interface (圖形化使用者介面)

For the GUI (Server) we use Java Swing

Java Swing is a GUI Widget toolkit for java – an API for providing a GUI for Java Programs. It reduces the burden that we have because in this Application we already have GUI components like buttons, checkbox etc.



- **Basic Functions (基本功能)**

We have 5 Types of basic functions:

1. Login Page

For the Login Page we divide 2 types of ID, the first letter “A” is for Admin and “C” for Client, and after we already logged in to the application it will directly read our ID and transfer to the UI which should.

*Note: we did not provide a register account function because this program is intended for commercial companies, only admins can add a new user from the databases

2. User Profile

For the User Profile the Admin and User can change the password according to what they want.

3. Control Queue

- **We implemented Java LinkedList Queue in our server.**

- **For the Control Queue we also divide it into 2 types of it:**

  - **Admin Queue**

    In the Admin Queue, the admin can call the User and see what the queue number is right now and which counter is it right now. The admin also can Reset the queue or make it backward if they accidently press the wrong button.

  - **User Queue**

    In the User Queue, the user can press the button and it will automatically give the queue number for the user and see which number of the queue is now.

4. Notepad

For the Notepad the admin can write if there is anything or that must be updated daily and put it there.

5. Emergency Call

For the Emergency Call the Admin and User can press it if there is a problem occurring right now in the queueing.

- **Advance Functions (創意功能)**

  We have 4 types of Advance Functions:

1. Chatbot

For the Chatbot, this can help us to directly answer the problem or question about our application without disturbing the Admin.

2. Chat Application (Admin)

For the Chat App, the admin can directly report to the superior directly if they encounter any problem that they can’t solve on their own.

3. News Search

We also applied News API for news search engines, News API is a simple JSON-based REST API for searching and retrieving news articles from all over the web. Using this, one can fetch the top stories running on a news website or can search top news on a specific topic (or keyword).

News can be retrieved based on some criteria. Say the topic (keyword) to be searched is ‘Geeksforgeeks’ or might be concerned to a specific channel.

4. Alpha Vantage API

We applied Alpha Vintage API to receive stock, ETF, forex, technical indicators, and cryptocurrency data on our app, so the users (bank clerk) don’t need to switch to another app to check stock market value, etc.

- **Market Research (市場調查)**

  After doing some research we can see that we can get:

1. Benefit:

- ### Improved Employee Efficiency

  A queue system eliminates the time-wasting distractions that add up over the course of the day managing customers, ultimately allowing more customers to be seen and freeing up resources for other tasks. Happier customers also generally make for less frazzled employees, who then stay more focused on efficiency.

- ### Improve Service Quality

  Employees who can focus on one customer at a time and not worry about the throng of people visibly waiting in line are better able to serve that one customer. They don't feel the need to rush the interaction and are more inclined to go the extra mile because there is less pressure to get to the next customer.

  A queue management system can also collect information about customers and their needs while they are waiting. That intel becomes instantly available to the staff, who can tailor service or designate a specialist for each customer who reaches the front of the virtual queue.

- ### Reduce Wait Times

  A virtual queue's combination of improved employee efficiency and shorter, more productive interactions decreases how much time customers spend waiting for their turn. Moreover, because they aren't tethered to a physical line and can move freely around (or outside) the store, their perception of how long they're waiting also decreases. In other words, what was once a 20-minute wait may take only 15 minutes and can feel like 10.

- ### Better Customer Data

  The digital backbone of a virtual queue management system offers something traditional queues struggle with. Advanced reporting capabilities of the best solutions can measure wait times, peak and slow periods, customer satisfaction, and more. This easily gathered and retrievable data can help inform strategy as well as improve efficiency and, subsequently, the customer experience.

- ## Increased Customer Loyalty

Customers appreciate not only great service but also little things that make their shopping experience and their lives easier. A queue management system may not reduce wait times significantly for every customer, but even just giving people a few minutes back or not making them stand in a line or sit in a crowded waiting area—particularly during a pandemic—shows that you care and improves the overall customer experience.

2. Mostly Used in:

- ### Banking & Hospital Service

Customers often must endure long wait times before they're attended to. This negatively affects the service perception. A queue management system lets customers see their status in a queue, thus minimizing anxiety and frustration.

- ### Government and Public services

The council's own service guidelines laid out their queue management needs, in a desire for a welcoming and safe environment that was accessible and clearly signed. They wanted appropriate self-serve facilities and to greet guests within three minutes of arrival. We answered these wishes with a bespoke queue management system, including ticket dispensers that were specially developed for the council. This is a great example of how queue management can result in much-improved customer experience.

- ### Booking Restaurant and Retailers

A well-known quick service restaurant and coffee retailer came to us simply to modernize their queue management.

Our solutions helped address concerns like how to avoid clutter and organize merchandise, keeping walkways clear for guests. Staff had also noticed that customers would not leave the line if they'd already joined it, meaning they didn't pick up certain products.