COMP4801

Final Year Project

Public Transport Reviewer

Group: fyp24003

Group Member:

Tai Yu Yeung (3035928639)

Wong Wai Ching (303606946)

Supervisor:

Dr. T. W. Chim

Abstract

Public transport is the most popular way of transportation in Hong Kong. Therefore, the quality of their service has a great impact on Hong Kong people's quality of life. On the other hand, reviews are often used by companies to collect customers' feedback for service improvement purposes. However, there lacks a corresponding review platform for public transport. Creating a review platform for public transport can be beneficial to the public. This project focuses on the development of Public Transport Reviewer, a mobile app review platform dedicated to public transport reviews. The main features of this app are review viewing and creation, input automation, comment and discussion feature, and review summary. The system of this app consists of a frontend application, a backend server and a database. Currently, the system of the app has been set up successfully. The server and database have been deployed and a functioning prototype app is downloadable now. Most functions have been implemented but some core functionality is not completed. In the second semester, the development will continue and the testing stage will begin.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

API	Application Programming Interface
EAS	Expo Application Services
JWT	JSON Web Tokens
OCR	Optical Character Recognition
ORM	Object Relational Mapping
UI	User interface

1. Introduction

Public transport is an integral part of most Hong Kong citizens' everyday life. With around 90% of all passenger travel using the public transportation service [1], it can be said that public transport dominates the current travelling market. Simultaneously, there exists a wide range of public transportation options, including railways, buses, minibuses, and taxis. Each consists of a great number of distinct routes. The diverse variety of public transport provides a great number of route combinations that can let people travel from one place to another. Therefore, deciding on service to be used can be very difficult. On the other hand, online reviews are often an important factor that many people consider when they buy products or services. Companies also use online reviews to learn more about the customer demands and improve their services based on the feedback. Despite the importance of reviews to both customers and companies, there is currently no platform that is dedicated to public transportation.

The absence of a dedicated public transport review platform can cause a lot of issues. One problem is that it is not easy to find reviews about a specific service. Currently, the reviews are distributed in various places like Facebook groups or individual posts in different social media. Finding reviews about different public transport services will require browsing multiple sites which can be very time consuming. Consequently, it is very difficult for customers and public transport service providers to understand the general opinion of the service. Additionally, the process of writing a review is challenging because of the lack of features that streamline input on the existing social media platforms. The current platforms are not designed for review posting and people are required to input every detail manually. This can discourage people from writing reviews.

To address the shortage of a review platform for public transport, a dedicated public transport reviewing mobile application called Public Transport Reviewer is proposed in this project. Public Transport Reviewer is a mobile application that will allow users to post and view reviews about public transport. In this app, users can submit reviews by entering related information like the service provider, route number, rating and comments. To facilitate the reviewing process, automatic input functions including

OCR and location tracking will also be available to enter information like location data and licence plate automatically. Additionally, users can view and comment on reviews posted by other users. This allows more people to share their opinion in a more in-depth discussion. Furthermore, a review summary page will be provided for each route. In this page, an overview of that service or route will be available including the basic information, average rating and the top reviews. Users will be able to gain a greater understanding of that service in a short period of time. The objective of this app is to assist the public in selecting services for their trips and enable companies to enhance service quality based on customer feedback.

The rest of the report is structured as follows. The methodology used to develop the app is covered in section 2. Then, the current progress is presented in section 3. The problem encountered is discussed in section 4. Future plan is outlined in section 5. At the end, the conclusion will be presented in section 6.

2. Methodology

This section delves into the details of the technical components in this project. It starts with an overview of the system architecture (Section 2.1). Then, the frameworks used for frontend (Section 2.2), backend (Section 2.3) and database (Section 2.4) will be explored. Next, the methods and services used to support the text recognition (Section 2.5) and location tracking feature (Section 2.6) will be discussed. Afterward, the authentication and authorization methods will be explained (Section 2.7). Finally, this section will explore the Deployment methods (Section 2.8), UI design tool (Section 2.9) and version control method to be used (Section 2.10).

2.1 System Architecture Overview

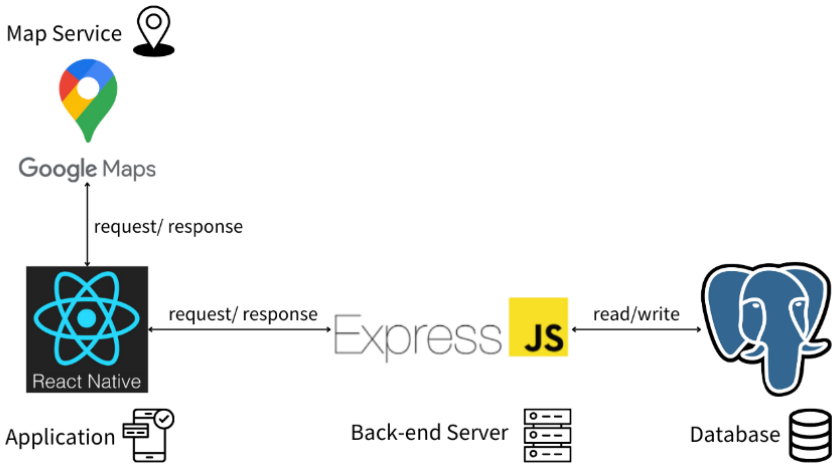

Figure 1: System Architecture

The system consists by three major components: frontend application, backend server and the database (see figure 1). The frontend mobile application will be the interface which the users will be interacting with. It will communicate with the backend to fetch and upload review data using RESTful API. The frontend is also connected to Google Maps using Google Maps API to display user's journeys in the app and provide more information about their journey. The backend server will handle all user requests like reviews submission and summary fetching. This is achieved by querying or updating the data from the database. It is also responsible for authentication and authorization operations to prevent unauthorized actions being done by users. The Database will be responsible for storing all data that are needed for this application including user information and review data.

2.2 Frontend

The front-end mobile application provides a user interface to allow users to interact with the platform. There will be multiple sections in this front-end application. The home page will show the recent popular reviews. The search page allows users to find any public transport route and get a detailed summary. The create page lets users write new reviews about their experiences of using public transport. And the profile page can be used to manage account information.

The frontend app will be developed using the React Native framework. React Native is a mobile app UI framework which is using JavaScript/Typescript language [2]. There are multiple reasons for choosing this framework. React Native can be used to develop mobile apps for both IOS and Android platforms under the same code base. It eliminates the need of maintaining multiple codebases for different platforms and make the development process easier. Flutter is another framework that is similar to React Native which can also be used to develop multi-platform mobile apps. But React Native provides more benefits to the development of the app. React Native uses native components from each operating system to render the app [3]. Allowing the app to provide similar experience to those native apps in their respective platforms and promotes learnability of the app. Also, Flutter uses Dart as its programming language which is a language of one framework. While React Native uses JavaScript which is more universal. Moreover, React Native has one of the biggest mobile app development communities with over 120,000 stars on its GitHub page. It means there is a wide range of libraries available so lower-level components like the UI and camera component can be imported directly. Then, it is not required to build those components from scratch and saves more development time. And it is easier to find help from other developers when difficulties are encountered.

2.3 Backend

The backend is responsible for handling user requests and performing authentication and authorization. To build the backend server that supports these operations, Express.js will be used. It is a backend framework that is based on Node.js. The main reason for using Express.js is its performance. It is fast and minimal so the servers written using Express.js will be highly efficient [4]. Then, the speed of client-server interaction will be maximized and the waiting time of each action will be reduced. It also supports middleware features that enable authentication to be implemented [4]. It will prevent unintended access to database data and improve the security of the server. Furthermore, Express.js is mainly written in JavaScript or TypeScript which is the same for React Native. It will allow both front and back end to be developed using the same language without constantly switching. Therefore, efficiency and productivity of this project can be increased.

2.4 Database and ORM

PostgreSQL will be used for the database and Prisma will be used as the Object–Relational Mapping (ORM). PostgreSQL is chosen because it is an object relational database where the data can be stored as objects with properties [5]. It will be required for storing and managing objects such as user's journeys which are arrays of locations.

In addition, a layer of ORM is used on top of the database. ORM is used because it allows the database to be accessed using functions like an object instead of using SQL (see figure 2). It removes the need of writing SQL codes and makes the process of writing queries easier. It will also reduce the risk of SQL injection [6] and fortifies the security of the database. Prisma is chosen to be the ORM because it offers automatic SQL migration generation. Database will be generated or updated automatically based on the schema defined by the developer. It makes database management easier and allows developers to focus on implementing the features.

```
const review = await prisma.review.findFirst({
  where: {
    id: id
  },
  include: {
    upvotedBy: {
      select: {
        username : true,
        uuid : true,
      }
    },
  }
})
```

Figure 2: An example query using function

2.5 Text Recognition

Text scanning is another feature that will be available in the app which will allow users to quickly enter important information like vehicle's registration mark and driver's identification.

To support this feature. Google ML Kit will be used. It provides packages that utilize machine learning for mobile devices. Text recognition is one of the features in the package and it will be used in the app. There are two reasons for choosing Google ML Kit. Firstly, all processes of ML Kit are done locally on the device without using an external server [7]. Allowing the feature to work reliably without an Internet connection. Secondly, it enables real-time processing [8] so the recognition result will be available in a short time after receiving the frame input. It allows the app to show the result immediately on screen after the camera captures any recognizable object. Then, the user will be able to see real-time results and adjust the camera without having to retake photos. Therefore, the time to get an accurate result can be massively reduced.

The workflow of the text scanning function is illustrated (see figure 3). After the user launches the scan camera in the app, each frame captured by the camera will be sent to the frame processor of the app. To prevent putting too many loads on the CPU, only a limited amount of frames received will be recognized using the text recognition module every second. The list of recognized text will then be returned and displayed on screen. Finally, the returned result will be shown on screen. The user can use the output when they see a satisfying result.

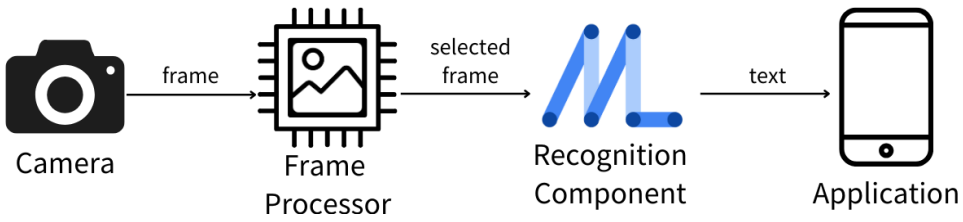

Figure 3: Workflow of text recognition function

2.6 Location Tracking

A location tracking function will be provided to input position data easily. After activating the tracking function, the app will capture the location of the device periodically. The array of locations collected can later be connected to form a path that can be shown on the map.

To get the path information, Google Maps will be used as the map service. Google Maps is used because it is one of the most popular map services and it has many robust features that will be very useful for the app. The location information recorded by the device can be entered into Google Maps and it will draw the path on the map [9]. It can also show information like distance and time of travel [9] which will be very useful for the users that want to know more about their journey. Additionally, Google Maps will suggest paths to users [9] so users can compare the recommended path to the actual path they have taken.

2.7 Authentication and Authorization

In this application, only logged in users are authorized to use all features. Each logged in client will receive a token that can be used to verify the identity and the log in status of the user. JSON Web Tokens (JWT) is used to generate. JSON Web Tokens is an open standard for sending data securely with JSON payloads [10].

The process of signing and generating JWT tokens is illustrated. When a user logs in successfully, a JWT token is generated using the secret key of the server and the uuid of the user below (see figure 4). This token will then be sent back to the user. Each JWT token will expire after one week. When a logged in user sends other requests with the JWT token as the bearer token. This token will be verified using the secret key of the server (see figure 5). If the verification is successful, the uuid of the user will be retrieved and the server will know the identity of the user.

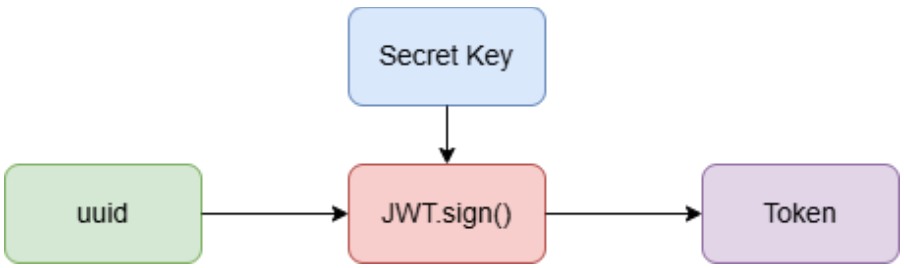

Figure 4: JWT signing process

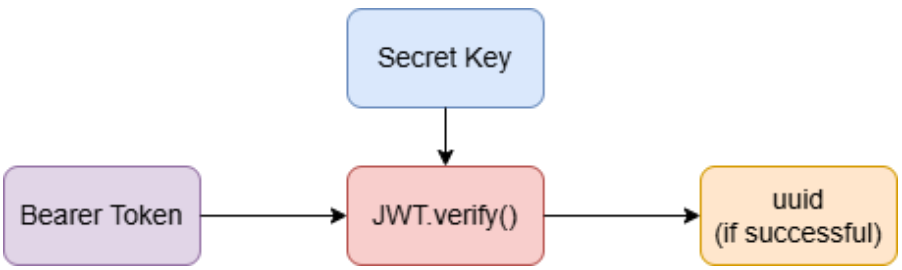

Figure 5: JWT verification process

2.8 Deployment

The deployment method selected for this project is Vercel. Vercel is a cloud platform that provides free web applications hosting service. A huge advantage of Vercel is its ease of use. After connecting the GitHub repository to Vercel, the server can be built and deployed automatically. Vercel will also redeploy the updated server when a new change is committed (see figure 6). This will save a great amount of server configuration time whenever there is any update to the server. Rollback function is also provided so the server can revert to any previous working deployment if there is any problem in the new deployment. It ensures that the backend services will always work.

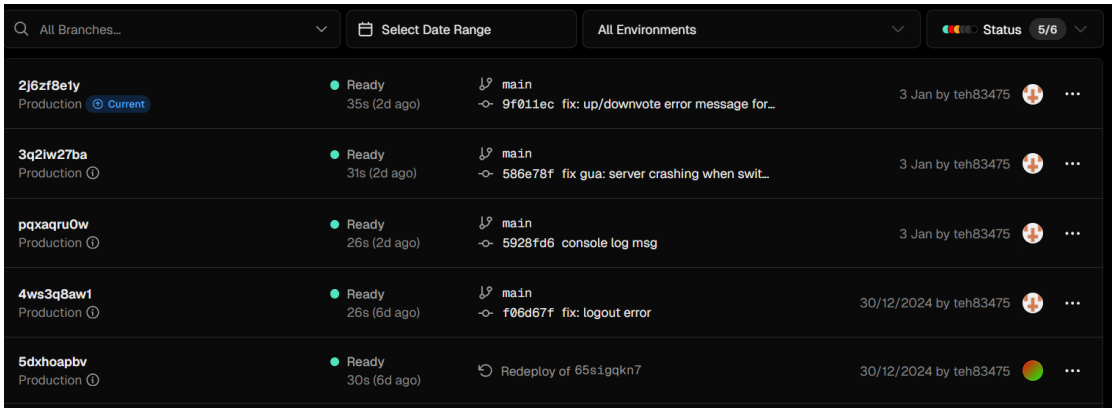

Figure 6: List of deployment deployed by Vercel automatically

2.9 UI Design

A good UI design is vital to the user experience of any application. To create the best UI possible for this project, Figma will be used. Figma is a collaborative tool for building UI prototypes which can enable teams to explore different design choices easily. The ability to make realistic prototypes without writing any code and converting the design into usable codes [11] make Figma the ideal tool for UI design.

2.10 Version Control

Since this application will be developed by multiple members, a proper version control is needed to synchronize the changes made by each member. The version control platform to be used in this project will be GitHub. GitHub is a cloud-based platform that allows users to store and track their codes in an online repository and collaborate with other developers on the same repository [12]. The reason for adopting GitHub in this project is that it is the most popular version control system,

and most developers have experienced using this platform. Also, by using GitHub, new branches can be created to create new features without affecting the main branch. This can accelerate the development through allowing different members to work on different areas of the app simultaneously.

3. Current Progress

In this section, the result completed to date is discussed. Section 3.1 will describe the setup and deployment of the system. Section 3.2 will show the implemented functions of the application.

3.1 System Setup

3.1.1 Server and Database

The server and database have been deployed to Vercel (See figure 7). Previously, the server and database are local hosted on personal computers and only emulators on the same computer can connect to the server. Presently, the backend is hosted on Vercel which is accessible via the Internet. Therefore, the frontend will be able to connect to the backend regardless of the network location.

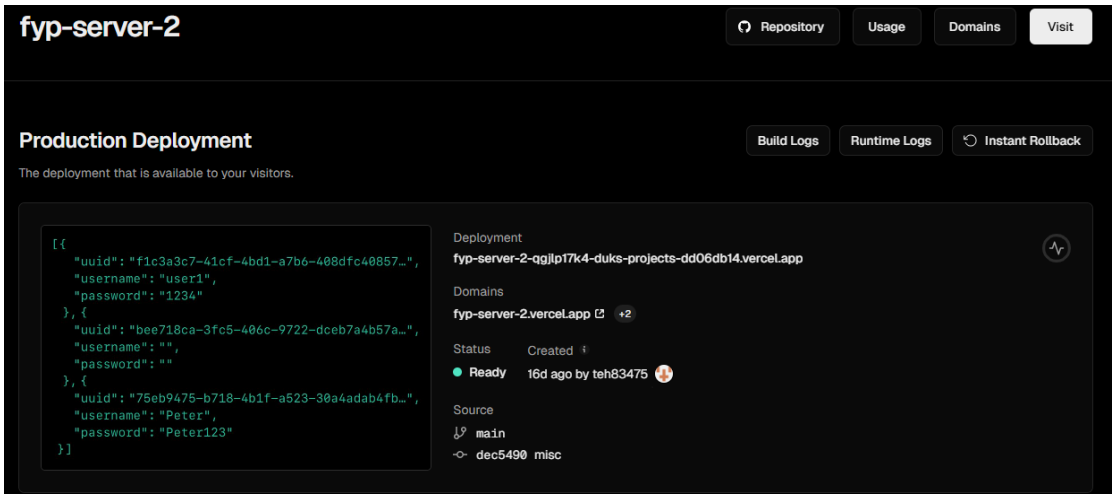

Figure 7: Server deployment page

3.1.2 Application

The development build of the application is available now and new features can be tested on physical devices (See figure 8). The front-end application is developed with React Native with a layer of Expo framework on top. Expo allows the application to be built using EAS. It will allow the development build of the app to be downloaded in real devices. It enables the testing of text recognition functionality which requires the use of real a camera.

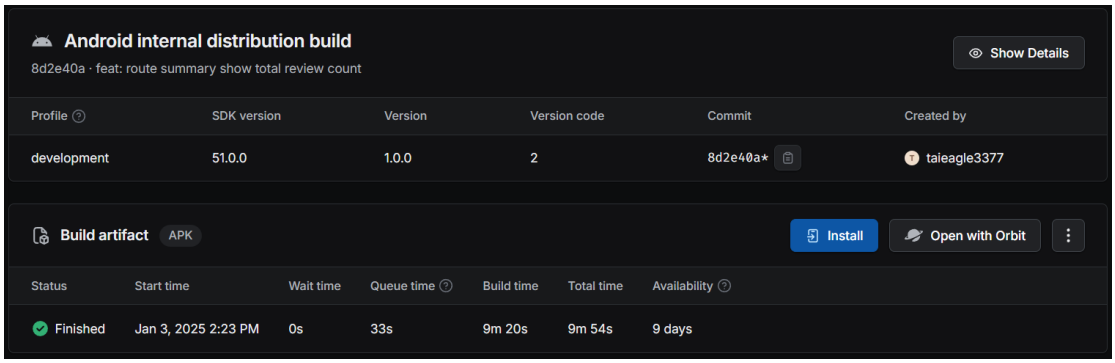

Figure 8: Front end application development build download page

3.2 Implemented Functions

The implementation of the functionalities of the app have started. Currently, most of the fundamental functions are implemented. Which includes review viewing, review submission, review summary, text recognition and login system.

3.2.1 Review List

The review list will show all recent reviews submitted by users (see figure 9). Each row of the list will show the overview of a review including information like the title, poster's name, postdate and rating. When entering this screen for the first time, the app will send a request to the server to get the review list data. The data received only contains partial information that are required for displaying this list. So, the application will not load unnecessary data from reviews that the user does not plan to check in detail. To view each view in detail, users can tap on each review item in the list and the app will navigate to the review detail page afterward. Additionally, a refresh button is available for refreshing this list and getting the latest reviews from the server.

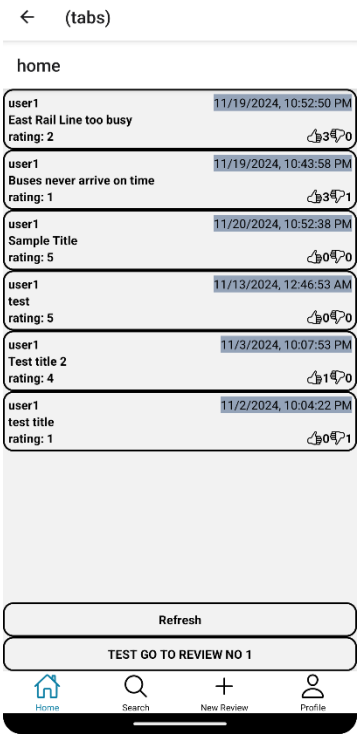

Figure 9: Review list

3.2.2 Review Detail

After selecting a review in the review list, the app will fetch the full review data and switch to the review detail page. This page will display the detailed information about this review and a list of comments posted under this review (see figure 10). In this page, users can express their opinions about this review using the upvote/downvote function and commenting system.

Users can upvote/downvote the review using the respective button on screen. If the user hasn't upvoted this review, the user will be added to the upvoted user list. The icon in the button will become filled and the count of upvotes will increase by one to show that the user has upvoted this review (see figure 11). Otherwise, the user will be removed from the list and the count will decrease by one. This operation is the same for downvoting.

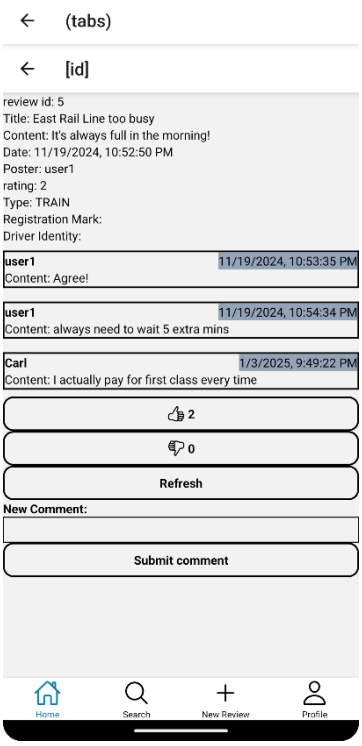

Figure 10: Review details

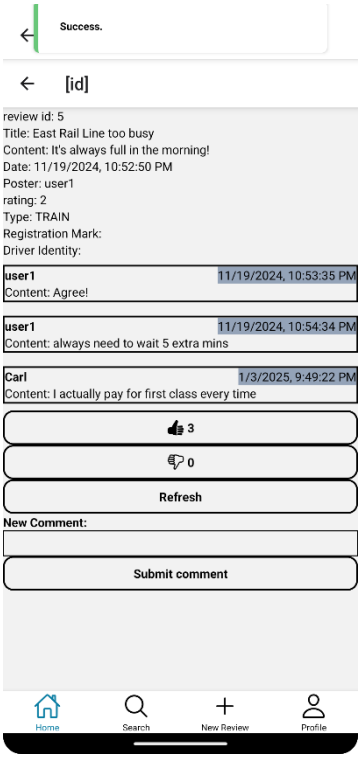

Figure 11: Upvoting a review

An input field is provided at the bottom of this screen for comment submission (see figure 12). After tapping the submit button, the comment will be sent to the server and saved in the list of comments under this review in the database. The comment will appear in the comment list in this page if the request is successful.

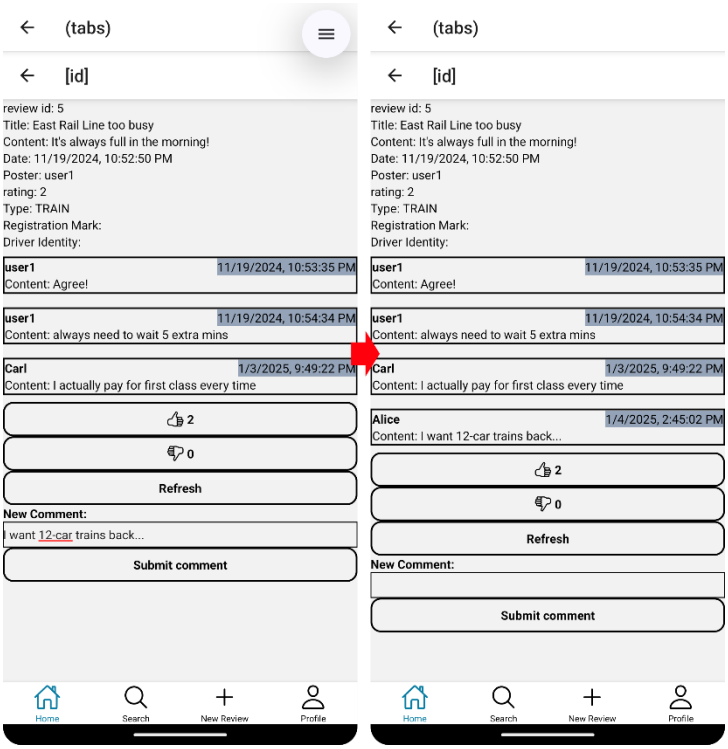

Figure 12: Submitting a comment

3.2.3 Review Submission

Users can use the review submission function to post a review about public transport (see figure 13). Reviews will be submitted after entering necessary information using the input fields and pressing the create button. Then the review will be saved in the database and added to the user's review submission list. Afterward, the review will appear in the review list and can be seen by all other users (see figure 14).

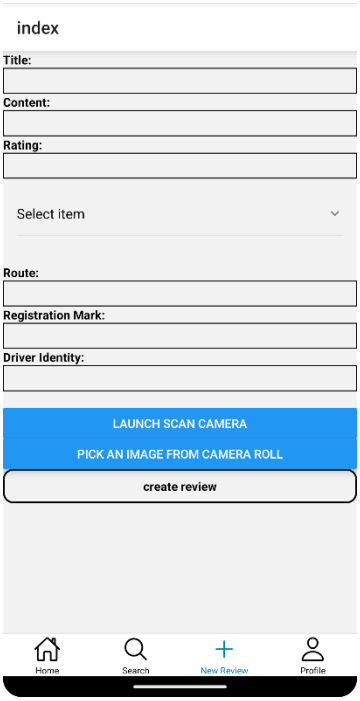

Figure 13:Review creation page

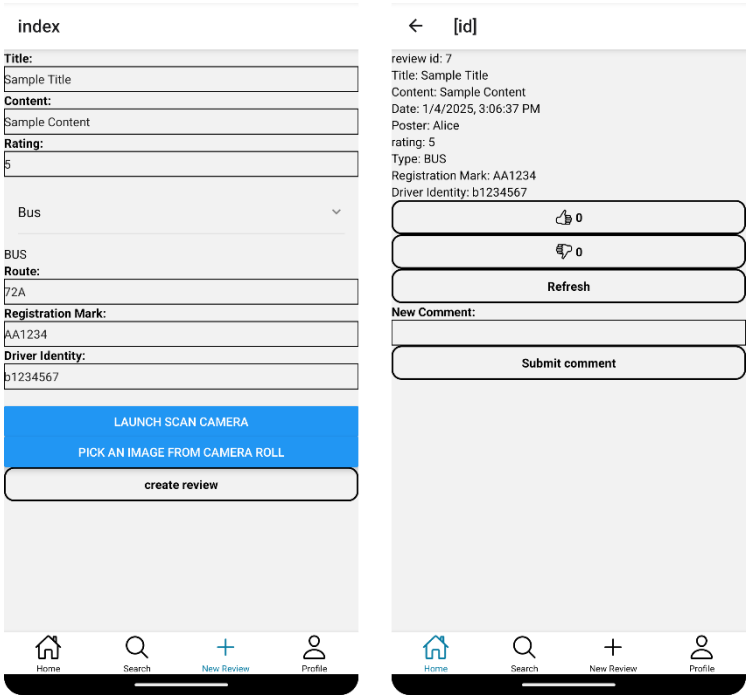

Figure 14: Creating a review

To facilitate the process of entering registration marks or driver's id easier. A text recognition feature is provided. This feature will be discussed in detail in the next section.

3.2.4 Text Recognition

The text recognition feature makes use of OCR to scan text captured by the camera to accomplish quick and accurate text input. If the camera is pointed at recognizable text, the real time scan result will be shown at the bottom of the screen (see figure 15). The user can focus on a specific spot by tapping on the screen if the camera is not focusing on the object to be recognized.

Figure 15: Text recognition capture page

The zoom level can also be changed using the pinch and zoom gesture to keep only the target on screen. Then, the text on the target object will be bigger and improve the accuracy of the recognition. It can be used to focus on the target objects such that there will be less unrelated objects that may interfere with scan results.

After pressing the use result button, the user will be sent to the confirmation screen. This screen contains the full scan result and the captured frame (see figure 16). In the captured frame, the recognized text blocks will be highlighted to show the location of the recognized text.

Figure 16: Text recognition result page

Users can choose the blocks to be included in the final result by tapping on each block to select or deselect it (see figure 17). The green blocks indicate that the text is selected and the grey blocks indicate that it is not selected.

Figure 17: Scan result block selection

Sometimes, the block may include some extra bits of unwanted text. Users can also edit the result manually using the text field (see figure18). Then, the user can decide whether they will use this result by pressing the confirm button or scan again using the back button.

Figure 18: Scan result text editing

3.2.5 Route Summary

The route summary function allows users to check the summary of any public transport routes. Users can search the route in the route search page (see figure 19). After entering the full name or part of the name in the search field, the app will automatically fetch all the routes with a name that matches the query from the server. The matching routes will be displayed under the search field (see figure 20). A label is shown on the right-hand side of each route to indicate the type of public transport. In the search field, a “clear” button is also provided to clear the query and reset the search results below the search field.

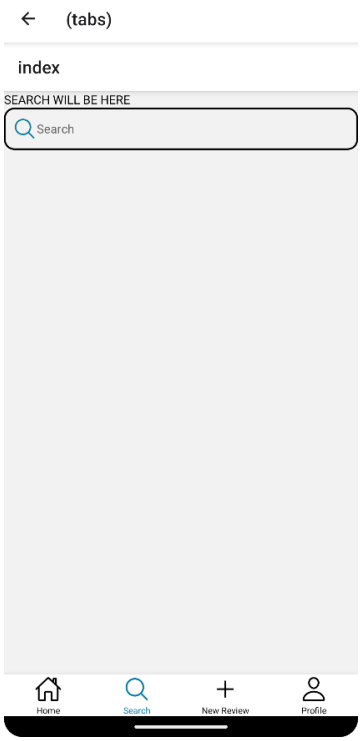

Figure 19: Route search page

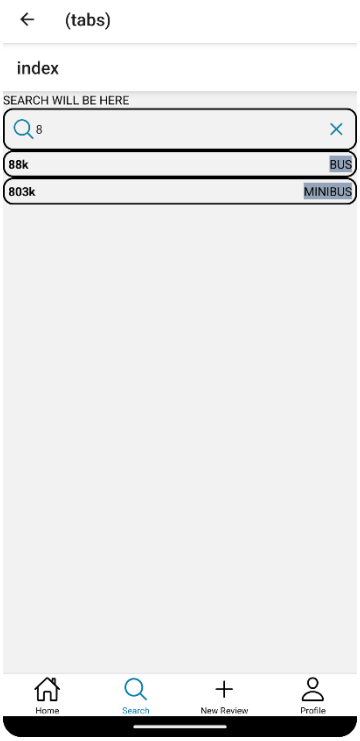

Figure 20: Searching for routes

After tapping on a route in the search page, users will be sent to the summary page of that route. In the page, the average rating and top reviews of the route will be shown (see figure 21). If the user wants to view the review in full detail, they can also tap on the review to enter the review detail page.

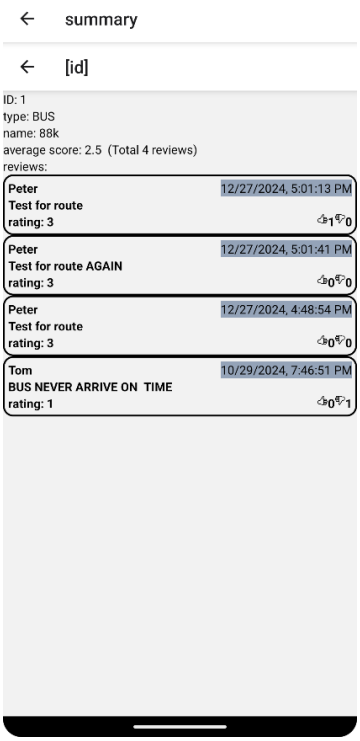

Figure 21: Route summary page

To show the top reviews that represent the general opinion about the route. The reviews in the summary page are sorted according to the score given to each review that represent their popularity (see figure 22). Generally, a review is more popular if it has more upvotes, fewer downvotes and is posted recently.

```
const reviewsSorted = route.reviews.sort((a, b) => {
  const diff_a= Math.abs(a.postedAt.getTime() - dateNow.getTime()) / (1000 * 3600 *24);
  const diff_b= Math.abs(b.postedAt.getTime() - dateNow.getTime()) / (1000 * 3600 *24);

  const score_a = a.upvoteCount-a.downvoteCount-diff_a*0.5;
  const score_b = b.upvoteCount-b.downvoteCount-diff_b*0.5;
  return (score_a > score_b ? -1 : 1);
});
```

Figure 22: Route sorting algorithm

3.2.6 Authentication and Authorization System

Some functions of the app are only available to the logged-in users, including review posting, commenting, upvoting and downvoting. To use these functions, users need to log in to the platform by entering their username and password in the profile page (see figure 23).

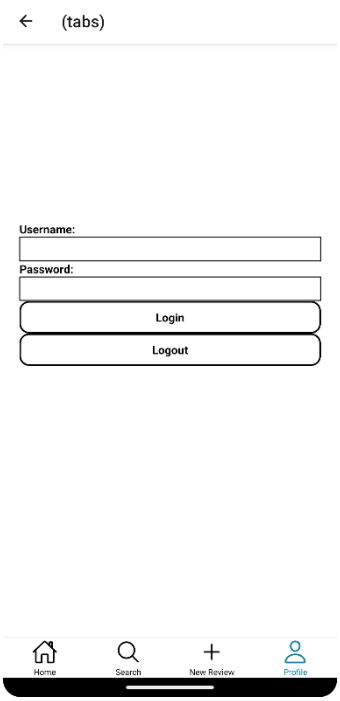

Figure 23: Profile page

The login workflow is as follows (see figure 24). After the application sends the login request to the server, the server will authenticate the request by checking the username and password. If the password is correct, the server will sign a new JWT token and send this token back to the application. This token is used to verify the identity of the user. The app will encrypt and save the JWT token in the device's storage. Then, the token can be retrieved when launching the app next time. If the password is incorrect, the server will simply return a failure message to the app.

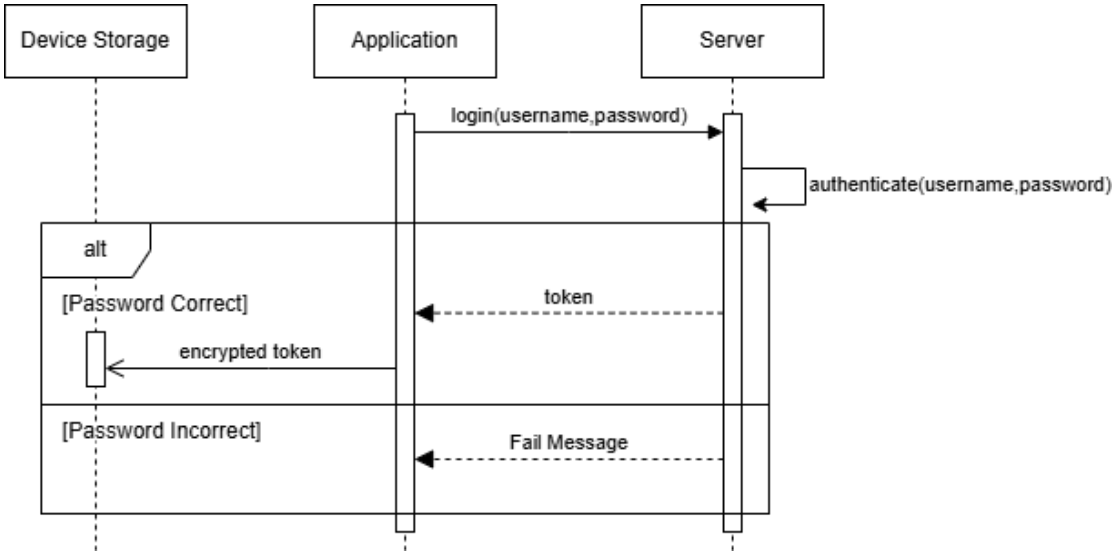

Figure 24: Login workflow

After the user logs in successfully, in every subsequent request sent by the user, the JWT token received previously will be attached as a bearer token (see figure 25). When the server receives the request, it will verify whether the token is valid. If the verification is successful, the server will handle the request. Otherwise, the request will be declined and a failure message will be returned to the user.

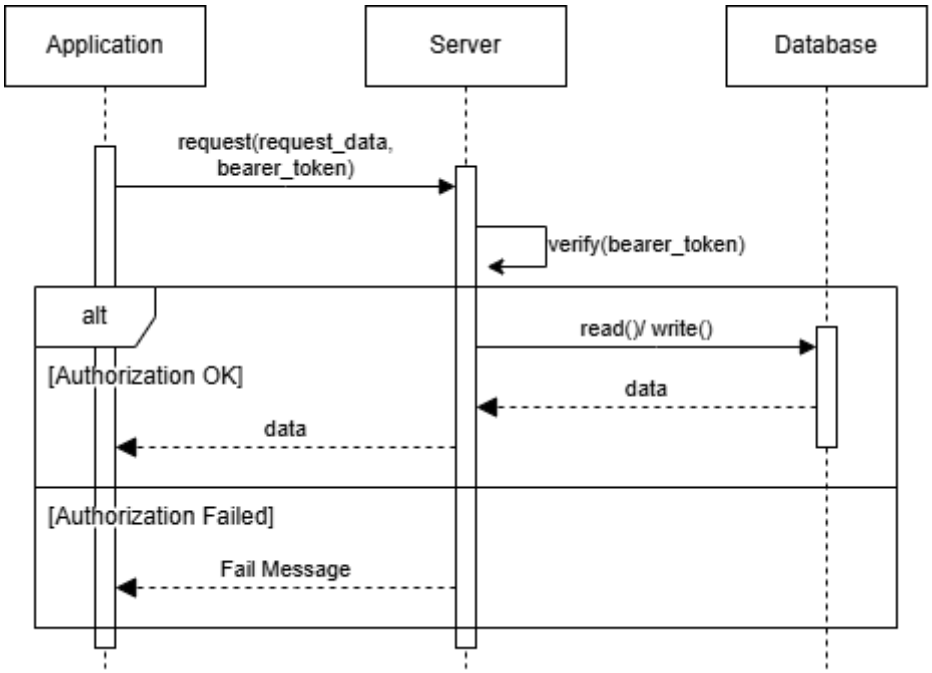

Figure 25: Authorization workflow

4. Difficulties encountered

Thus far, there are not many problems encountered during the development. The only challenge is the performance issue of a certain part of the application. Detailed information will be illustrated below.

The current performance of the existing text scanning implementation is undesirable. Specifically, the application will become very laggy and unresponsive when using this functionality. It is suspected that the lag is caused by the huge workload needed to process and recognize text on every captured frame. The current mitigation method is limiting the number of frames to be processed down to one per second. However, the app still experiences micro stutters every second during frame capture and processing. A possible solution to further alleviate the load of processing will be resolution reduction. The current frame resolution is 3000*4000, decreasing it may make the workload more manageable. However, it may also negatively affect the accuracy of the text recognition result. Therefore, more testing is required in the future to strike a balance between performance and accuracy.

5. Future Plan

This section outlines the plan and work to be undertaken in the future. Section 5.1 will cover the remaining development works. Section 5.2 and section 5.3 will outline the UI interface and the optimization work to be done. Section 5.4 will illustrate the full schedule of the project.

5.1 App development

While most of the fundamental functionality are implemented, some functions are still not completed and many of them are unpolished and need further improvement.

5.1.1 Location tracking feature

At present, one of the core features, route tracking is yet to be implemented. It should be built as soon as possible to allow the tracked locations and routes to be attached to a review.

5.1.2 Review sorting

The review list in the home page and the route summary page is currently ordered by popularity using the algorithm stated in previous chapters. It will be beneficial to provide more soring options like sort by date, most upvoted or most downvoted to provide a more comprehensive overview of the opinions about a route.

5.1.3 Route comparison

In many situations, users may want to compare different route options side by side. But the current route summary page only supports showing one route at a time. A compare function should be added to the summary page for users to compare routes more easily.

5.1.4 More detailed route summary

Currently, users need to scroll through all the reviews in the route summary page to learn about the opinion of other users. It can be time consuming if users need to check multiple routes in a row. It is suggested to add an AI-powered summary function that automatically summarize all reviews of a route so users can check the summary more quickly.

5.2 UI design

Previously, the development focused on the implementation of functionalities but not the user interface. Therefore, the UI of the existing prototype is minimal. A more user friendly and visually appealing UI interface should be designed and implemented.

5.3 Optimization

At the same time, the app needs to be optimized to improve the user experience. Considering the functions are subject to change, the UI implementation, optimization and testing work will begin after all functions are finalized.

5.4 Project Schedule

The full schedule of this project is illustrated below (see table 1).

Period	Work Description	Progress
Sep – Oct	• Confirm project topic and scope • Detailed project plan • Create Project Website	Completed
Nov – Dec	• Design system architecture • Test feasibility of app functions • Implement basic features • Develop Text Recognition feature	Completed
	• Develop Location Tracking feature	In Progress
Jan – Feb	• Design and finalize user interface • Refine existing function	In Progress
Mar – Apr	• Optimization • Complete app development • Testing and debugging • Clean up code	Scheduled

Table 1: Project Schedule

6. Conclusion

Because of the lack of a dedicated public transport review platform, it can be challenging to compare different route options or collect customer feedback. This project aims to fill this gap by developing a public transport review app that not only delivers the basic reviews functionalities. But also provides automatic input functions that trivialize the information input process and summary feature that accurately reflect the general opinion of the users.

Presently, the whole system has been deployed and it allows the app to be tested in a real environment instead of an emulated environment. Most functions of the app are implemented. The only unfinished feature is the location tracking feature. But there are still a lot of improvements that can be made to the implemented functionalities.

The development of Public Transport Reviewer is behind schedule currently. The development pace needs to be accelerated to catch up to the original schedule. For the remainder of January and February, the main focus is finishing the development of app features and UI implementation. In March and April, the focus will shift to optimizing the functionalities and software testing.

References

[1] Transport Department. (2024, Apr. 5). Intelligent Transport Systems (ITS) Introduction [Online]. Available: https://www.td.gov.hk/en/transport_in_hong_kong/its/introduction/index.html [Accessed Sep. 30, 2024].

[2] M. Budziński. (2024). What Is React Native? Complex Guide for 2024 [Online]. Available: https://www.netguru.com/glossary/react-native#react-native-your-go-to-framework-for-cross-platform-mobile-app-development [Accessed Sep. 30, 2024].

[3] React Native. (2024, Aug. 15). Core Components and Native Components [Online]. Available: https://reactnative.dev/docs/intro-react-native-components [Accessed Sep. 30, 2024].

[4] GeeksforGeeks. (2024, Jul. 15). Express.js Tutorial [Online]. Available: https://www.geeksforgeeks.org/express-js/ [Accessed Sep. 30, 2024].

[5] Amazon Web Services. (n.d.). What's the Difference Between MySQL and PostgreSQL? [Online]. Available: https://aws.amazon.com/compare/the-difference-between-mysql-vs-postgresql/?nc1=h_ls [Accessed Jan. 6, 2025].

[6] I. Abba. (n.d.). What is an ORM – The Meaning of Object Relational Mapping Database Tools [Online]. Available: https://www.freecodecamp.org/news/what-is-an-orm-the-meaning-of-object-relational-mapping-database-tools/ [Accessed Sep. 30, 2024].

[7] ML Kit. (n.d.). Machine learning for mobile developers [Online]. Available: https://developers.google.com/ml-kit [Accessed Sep. 30, 2024].

[8] ML Kit. (2024, Jul. 10). Text recognition v2 [Online]. Available: https://developers.google.com/ml-kit/vision/text-recognition/v2 [Accessed Sep. 30, 2024].

[9] Google Map Platform. (n.d.). Build awesome apps with Google's knowledge of the real world [Online]. Available: https://developers.google.com/maps [Accessed Sep. 30, 2024].

[10] JSON Web Tokens. (n.d.). Introduction to JSON Web Tokens [Online]. Available: https://jwt.io/introduction [Accessed Jan. 6, 2025].

[11] Figma. (n.d.). Figma Design [Online]. Available: https://www.figma.com/design/ [Accessed Oct. 14, 2024].

[12] GitHub. (n.d.). About GitHub and Git [Online]. Available: https://docs.github.com/en/get-started/start-your-journey/about-github-and-git [Accessed Oct. 14, 2024].