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. Therefore, 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 browsing 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 development of the application is completed and the app is ready to be used.

Acknowledgements

I would like to express my deepest gratitude to my project supervisor, Dr T.W. Chim. His invaluable guidance and insightful feedback help me shapes our app to its current state.

Table of Contents

List of Figures

List of Abbreviations

API	Application Programming Interface
Blob	Binary Large Object
Bytea	Byte Array
CDN	Content Delivery Network
EAS	Expo Application Services
GPS	Global Positioning System
JWT	JSON Web Tokens
OCR	Optical Character Recognition
ORM	Object Relational Mapping
UI	User interface
URL	Uniform Resource Locator

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 allow 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 sites 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. At the same time, these sites are not intended to be used as review platforms so most of the posts are unrelated to public transport reviews and they often lack optimized way to find specific posts. 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 type of public transport, 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 rating metrics, AI reviews summarization 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 project result is presented in section 3. Future plan are outlined in section 4. At the end, the conclusion will be presented in section 5.

2. Methodology

This chapter 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), database (Section 2.4) and the file system (Section 2.5) will be explored. Next, the methods and services used to support the text recognition (Section 2.6) and location tracking feature (Section 2.7) will be discussed. Next, the language model (Section 2.8) used will be discussed. Afterward, the authentication and authorization methods will be explained (Section 2.9). Finally, this section will explore the Deployment methods (Section 2.10), UI design tool (Section 2.11) and version control method to be used (Section 2.12).

2.1 System Architecture Overview

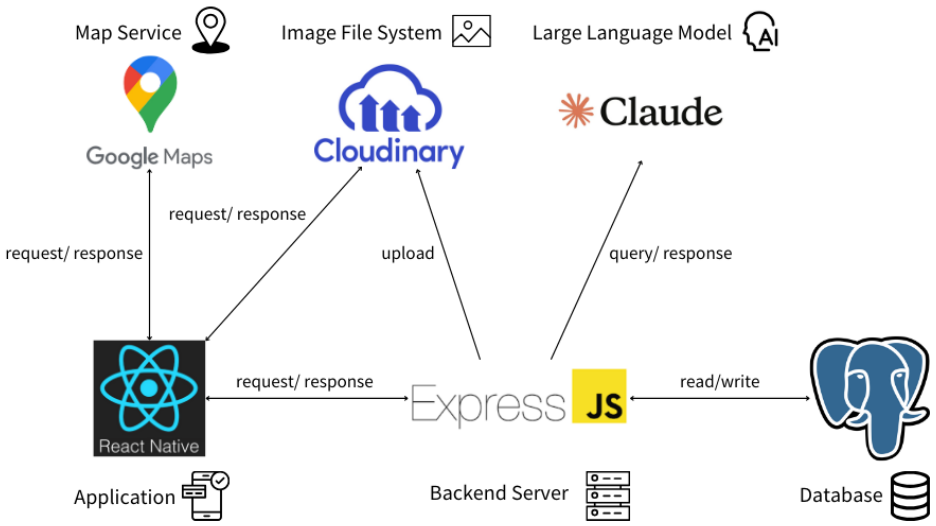

Figure 1: System Architecture

The system architecture comprises three major components: the frontend application, the backend server and the database (see figure 1). The frontend mobile application serves as the main user interface which enable the interaction with the system. It will communicate with the backend using RESTful API to fetch and upload review data. The frontend is also connected to Google Maps for its location-based services and Cloudinary for image retrieval in reviews. 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 while also uploading image files to the image file system, Cloudinary. Furthermore, the backend updates review summaries using Claude 3.7 Sonnet model through the Anthropic API. 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 are 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 is 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 is 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 allows 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 is used for the database and Prisma is 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 is 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 reduces 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 Image File System

An external image file system is used to store the images attached to each review. Image files are unstructured so the backend server needs extra time to convert the images to compatible formats [7] like blob or bytea. Images are also much bigger in storage size comparing to other data so the database size will dramatically increase when more images are being uploaded. It can significantly increase the read/write operation load and the database will become highly inefficient. For these reasons, an extra file system is often used to store unstructured objects like images.

For this project, Cloudinary, was selected as the image file system. Cloudinary is a online platform that provides cloud media management services. The uploaded images can be accessed via URLs. Then, the database only needs to store the URLs of the images and front-end can directly fetch the required images from the file system using the corresponding URLs. It can massively alleviate the workload of the database during image retrieval process. Cloudindary also offers image transformation services such as automatically resizing of the uploaded images [8]. It reduces the file size of large images, and the time to fetch images from the file system will be lower. Therefore, the use of an extra image file system can improve the scalability and performance of image data management.

2.6 Text Recognition

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

To support this feature. Google ML Kit is used. It provides packages that utilize machine learning for mobile devices. Text recognition is one of the features in the package and it is 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 [9]. Allowing the feature to work reliably without an Internet connection. Secondly, it enables real-time processing [10] so the recognition result will be available in a very 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 number of frames received will be recognized using the text recognition module in 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.

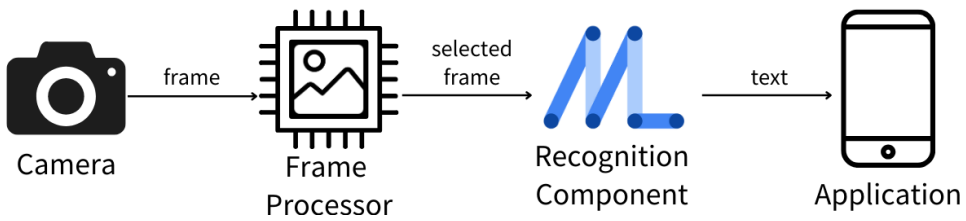

Figure 3: Workflow of text recognition function

2.7 Location Tracking

Users are able to attach a path to each review to show their entire journey. Which is helpful for providing more context for the review.
The location tracking features makes use two components which are the device's GPS service and Google Maps. The GPS service is used to get the current geolocation of the device and Google Maps is used to render the path in the app. Google Maps is used because it is one of the most popular map services and it has many robust features that are 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 [11]. It can also show information like distance and time of travel [11] which will be very useful for the users that want to know more about their journey.
The workflow the location function is as follows. After activating the tracking function, the app will launch a foreground service that will periodically read the GPS location of the device. A foreground service is used because it will not be killed when low on memory so the tracking will not be stopped out of users' control. The array of locations collected will be saved in the device's memory and the application can illustrate the current path using Google Map.

2.8 AI Summary

This app also employs artificial intelligence to summarize reviews for every route. It allows users to quickly understand the common opinion of each route without checking reviews one by one. Whenever a new review is submitted to the backend, the server will prompt the model with the new review and generate a new summary. Subsequently, the summary will be stored in the database and the users can fetch the new summaries when they visit the summary page.

To support this feature, Claude 3.7 Sonnet is used as the large language model. The model has some of the best languages ability among top models, ranking in the top three for reasoning, language, and data analysis on LiveBench benchmark [12]. This ensures that the model can accurately condense the general consensus on each route while weighting other factors like the post date and the amount of upvotes and downvotes. Therefore, accurate summaries that reflects the current user experiences of each route can be provided.

2.9 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 the tokens. JSON Web Tokens is an open standard for sending data securely with JSON payloads [13].

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 (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.

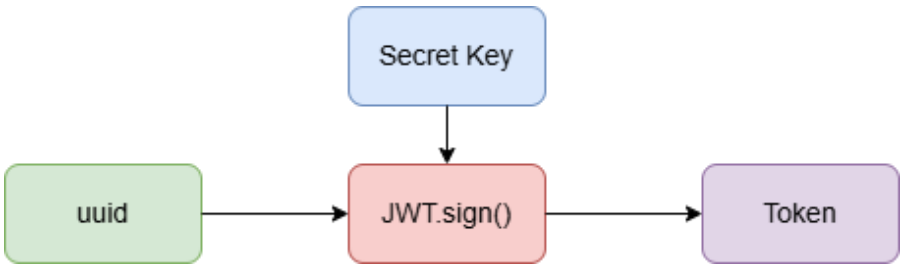

Figure 4: JWT signing process

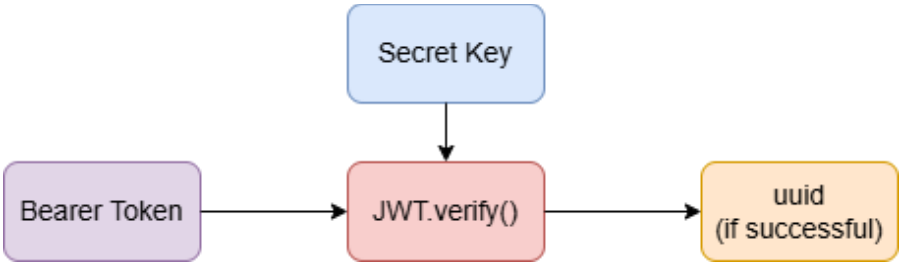

Figure 5: JWT verification process

2.10 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) to the repository. 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.

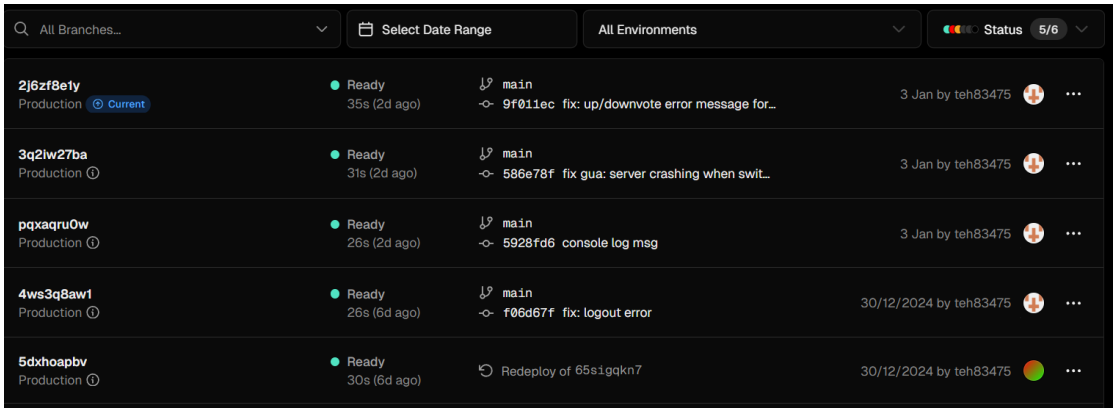

Figure 6: List of deployment deployed by Vercel automatically

2.11 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 is 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 [14] make Figma the ideal tool for UI design.

2.12 Version Control

Since this application is 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 is 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 [15]. 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. Result

In this chapter, the result completed is discussed. Each section of this chapter will describe the functions of the application.

3.1 Home Page

The home page will show the current popular reviews submitted by users (see figure 7). 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. As the result, the application will not load unnecessary data from reviews that the user does not plan to check in detail.

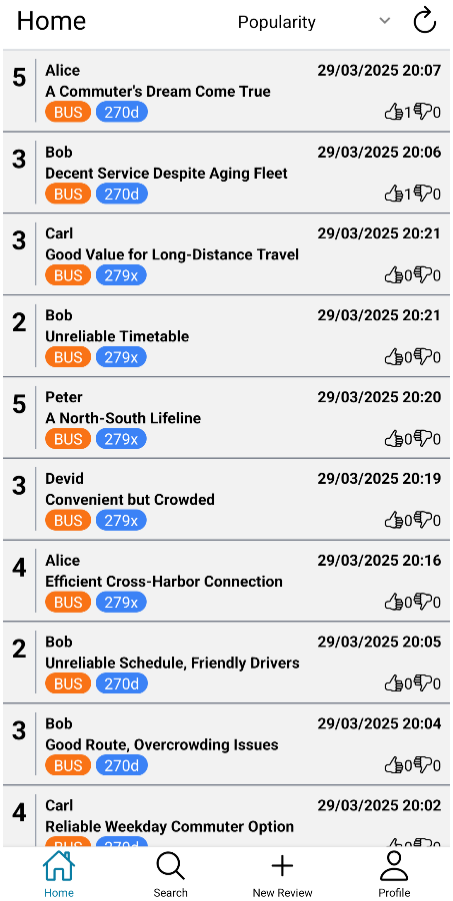

Figure 7: Review list

Each row of the list will show the overview of a review including information like the title, rating, route name, the type of transport and the number of upvotes and downvotes (see figure 8). To view each review in detail, users can tap on each review item in the list and the app will navigate to the review detail page afterward.

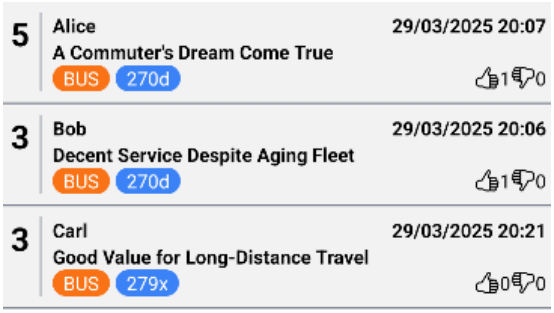

Figure 8: Example of reviews in the review list

Additionally, a refresh button and a sorting option is available at the top of the screen (see figure 9). The refresh button allows the app to get the latest reviews from the server. The sorting options can be used to change the order the list. Details of the sorting options will be explained in Review Summary section.

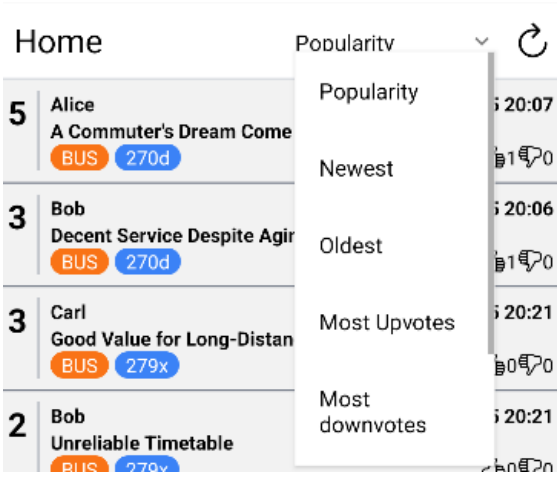

Figure 9: Refresh and sorting options

3.2 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 full detailed information about this review and a list of comments posted under this review (see figure 10).

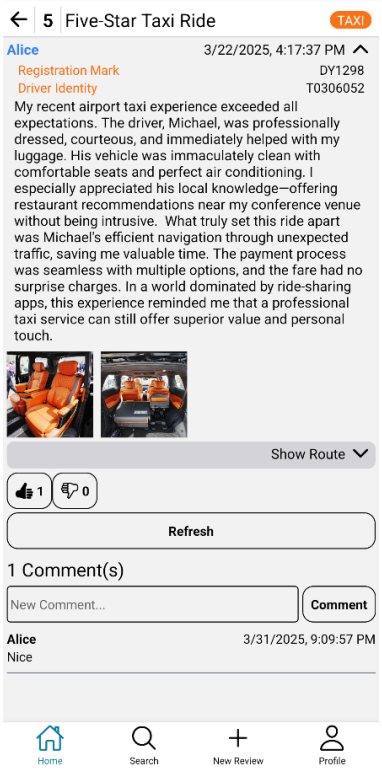

Figure 10: Review details

The list of images previews is shown below the main content of the review. Tapping on each image will bring up the full screen display of the image (see figure 11). Users can also swipe left or right to switch between each image in this view.

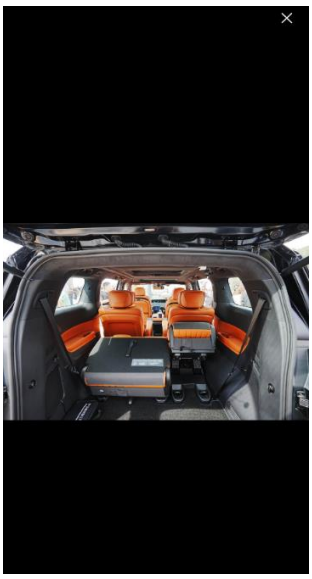

Figure 11: Image full screen display

If the review poster have attached a route to the review. The user can check the map by tapping the show route option. The full route will be illustrated on the map with a green marker indicating the starting point and a red marker indicating the ending point. And every tracked location will be marked as white dots along the path (see figure 12).

Figure 12: Route Display

Users can tap on each marker to reveal more information like the speed and timestamp at that point (see figure 13). Additionally, the average speed and top speed is shown under the map. This feature can be helpful to identify detours of a taxi ride or highlight speeding at specific spots during the trip.

Figure 13:Route Marker

Users can give feedback to the review using the upvote/downvote system. By tapping on the upvote button, an upvote will be given to the review. The icon in the button will turn solid and the upvote count will increase by one (see figure 14). If the user has downvoted the review previously, their downvote will be removed and gives an upvote instead. The user can also cancel their upvote by tapping the button again. This operation is the identical for downvoting.

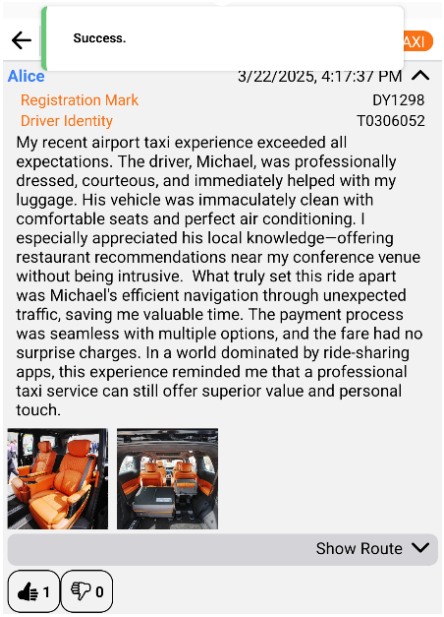

Figure 14: Upvoting a review

User can also express their opinion in the comment section located at the bottom of the screen. An input field is provided for comment submission (see figure 15). 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.

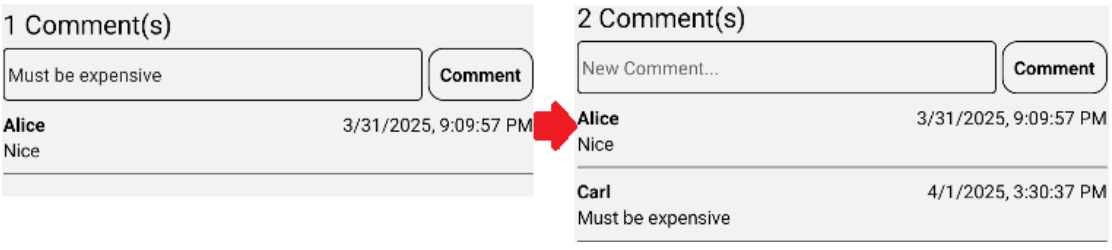

Figure 15: Submitting a comment

3.3 Review Submission

Users can use the review submission function to post a review about public transport. Reviews will be submitted after entering necessary information using the input fields and pressing the create button. Then the review will be sent to the server and saved in the database. Afterward, the review will appear in the review list and can be seen by all other users (see figure 16).

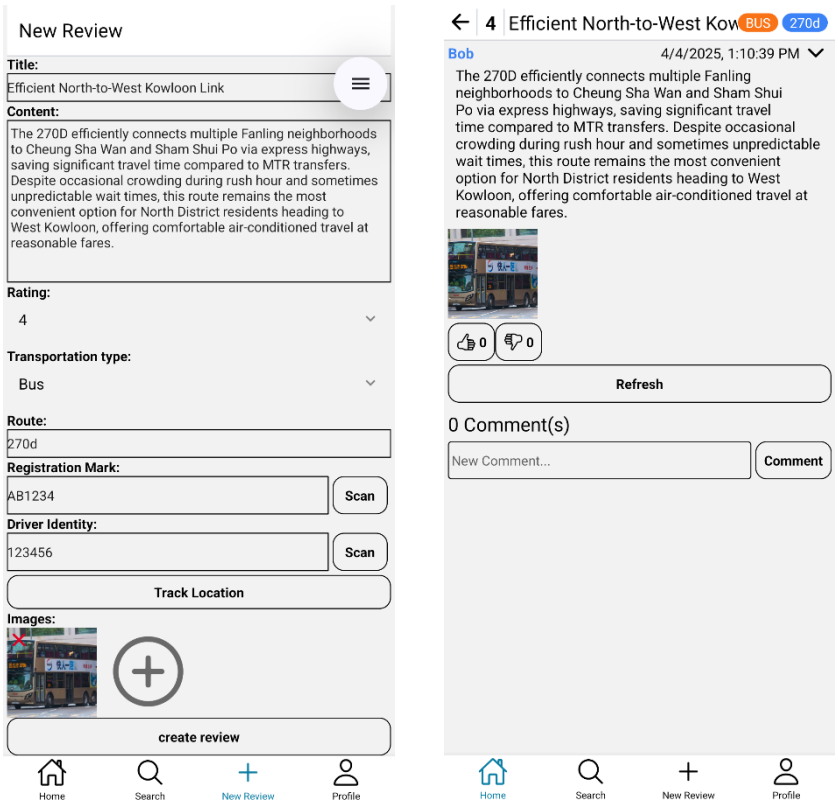

Figure 16: Creating a review

This apps also support attachment of images in reviews. Images can be added by tapping the plus button under the image section. Each image can be removed by pressing the remove button in the image preview. At most 8 images are allowed to be attached to each review.

To facilitate the process of entering registration marks or location data easier. A text recognition and a location tracking feature are provided. These features will be discussed in detail in the following sections.

3.4 Location Tracking

The location tracking feature is provided to allow users to record the entire pathing in their journeys. This feature can be accessed by tapping the location tracking button in the review submission page.

Figure 17: Route track page

After the users press the start tracking button, the app will record their GPS location and update their path on the map in real time (see figure 18). The starting point is marked as green, and the current point is marked as red. Users can stop tracking by pressing the stop tracking button at the bottom.

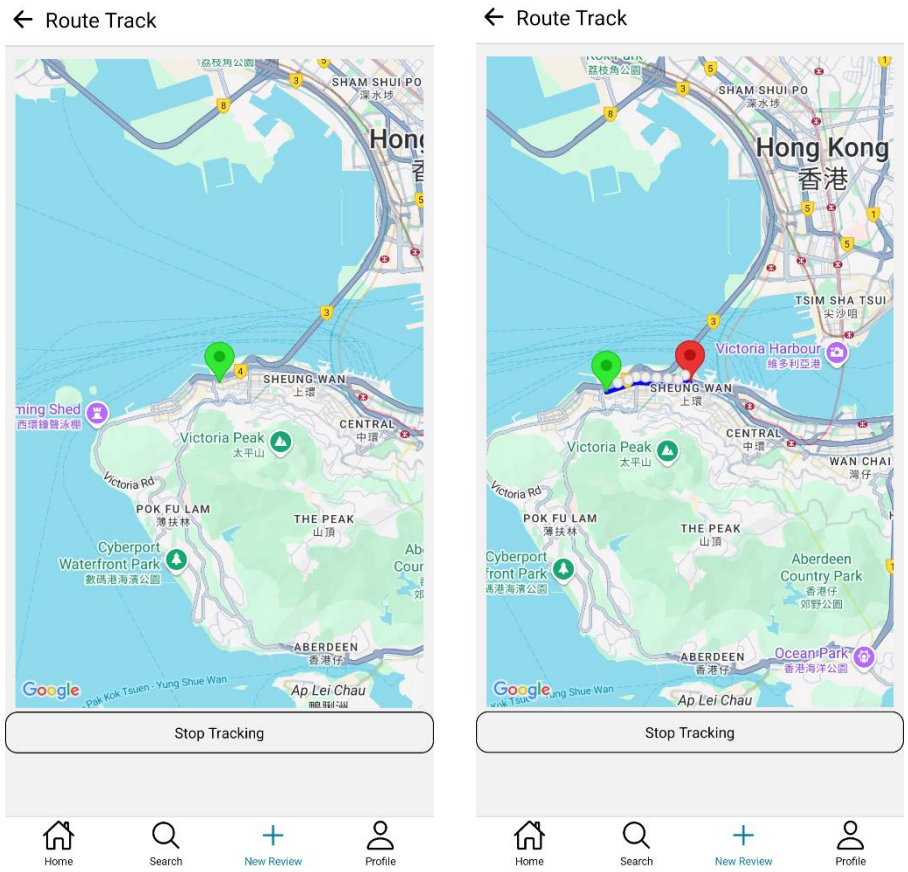

Figure 18: Real time location tracking

Foreground services are used for the tracking so the tracking will continue even when the app is closed. Therefore, it is not required to open the app at all time and users can use other apps in their phone when tracking their journey.

3.5 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 19).

Figure 19: Text recognition capture page

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. 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 result editing screen. This screen contains the captured frame and the full scan result (see figure 20). In the captured frame, the recognized text blocks will be highlighted to show the location of the recognized text.

Figure 20: 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 21). The green blocks indicate that the text is selected and the grey blocks indicate that it is not selected. The result preview in the text input field will reflect the changes accordingly. To reset the result back to the original state, users can use the reset button at the bottom. Then, all text blocks will be selected and highlighted in green again.

Figure 21: 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 figure 22).

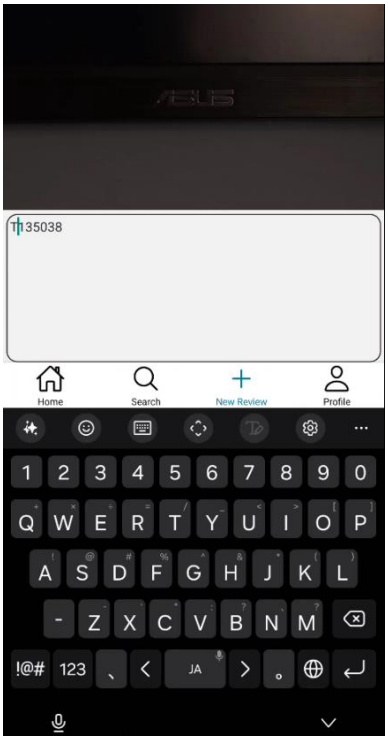

Figure 22: Scan result text editing

When a satisfying result is obtained, users can use the confirm button to use the result in the review creation page (see figure 23).

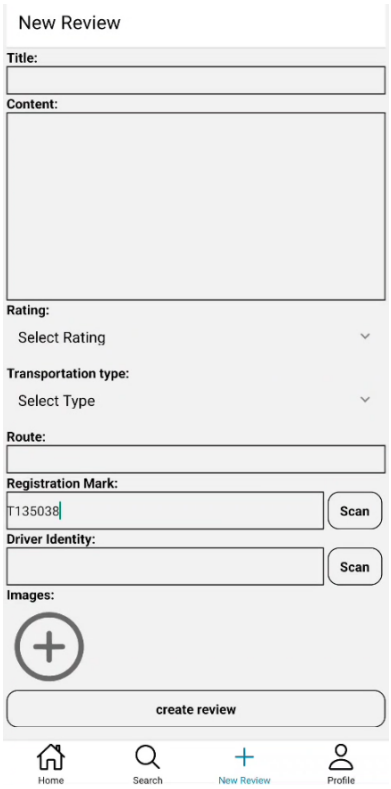

Figure 23: Returned recognition result

3.6 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 24). After entering the full name or part of the name in the search field, the app will automatically search for 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 25). A tag 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.

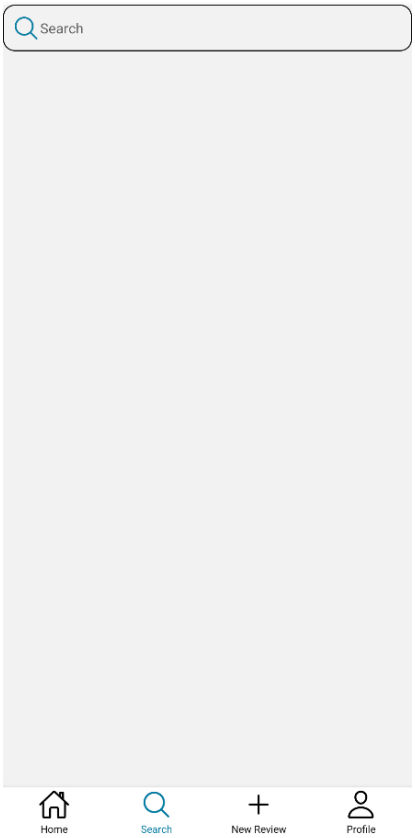

Figure 24: Route search page

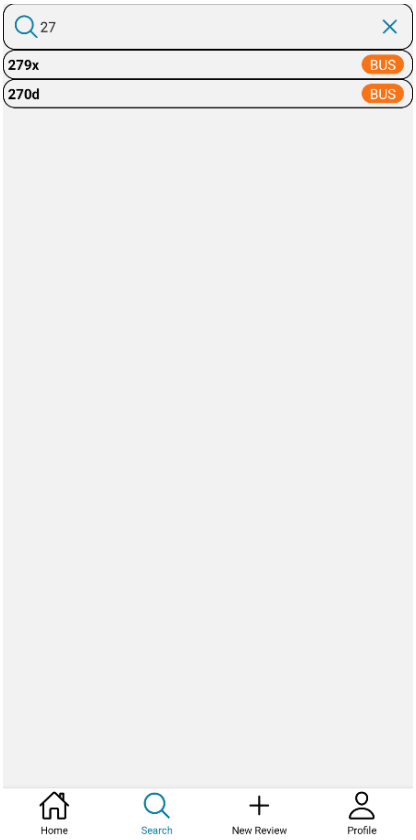

Figure 25: Searching for routes

Upon selecting a route from the search result, users will be sent to the summary page of that route (see figure 26). A graph that shows the rating metrics is featured at the top of the screen. The radial bar chart on the left shows the average rating and the total amount of reviews. The bar chart on the right shows the distribution of each rating score. This section allows users to quickly check the overall satisfaction level towards this route and reveals interesting patterns like spikes in a particular score.

The route summary page also features an AI-powered review summary(see figure 26). It will summarize the reviews about this route in a short paragraph. Allowing users to effectively gain insight into the past experience of the route without reading each review one by one. The summary will be updated whenever there is a new review submitted about the route.

Furthermore, the top reviews about the route are displayed at the bottom of the page (see figure 26). Unlike the home page, the app will show a section of the content in each review here. So, users can directly view each review by scrolling through this page. If the user wants to view the review in full detail, they can also tap on the review to enter the review detail page.

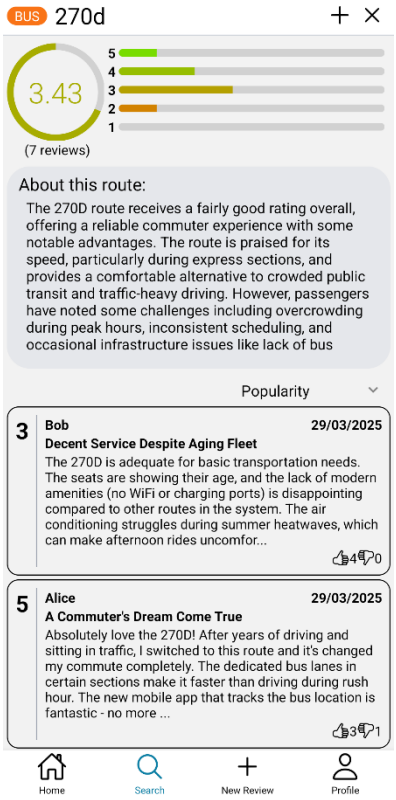

Figure 26: Route summary page

Sorting options are also provided to change the order of reviews in this page. There are total of 7 sorting methods based on popularity, time, upvotes or downvotes and sort by rating (see figure 27). Users can use this option to explore both positive and negative reviews in different time easily. Reviews are sorted based on popularity by defaults. Ensuring the most relevant review to be shown first.

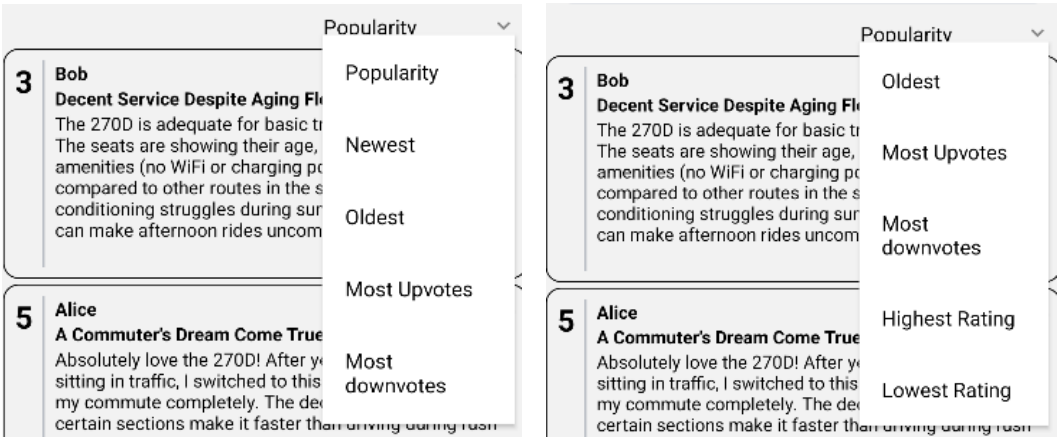

Figure 27: Sorting options

To show the top reviews that represent the popular 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 28). 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 28: Route sorting algorithm

Additionally, the app allows comparison of routes. After selecting a route in the search page, users can tap the add button in header to search for another route. Then, two routes will be displayed side-by-side which enable users to compare routes in the same page without switching back and forth (see figure 29). To remove a route, users can tap the close button in the respective route header to close it.

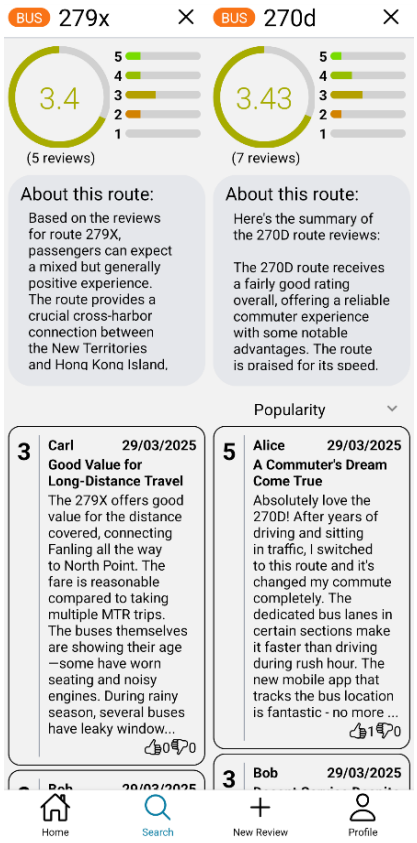

Figure 29: Route comparison

3.7 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 30). Users can register an account in the register page if they don't have one (see figure 31).

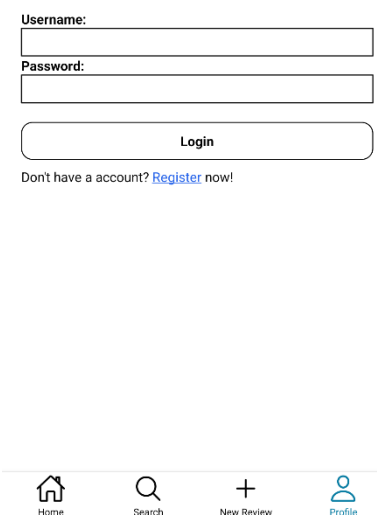

Figure 30: Login Page

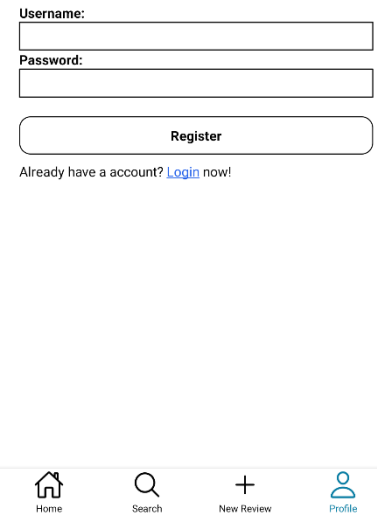

Figure 31: Register Page

The register workflow is as follows (see figure 32). After the application sends the login request to the server, the server will check whether the username is already used. If it is not used, it will create an account and send a success message back to the client. If the username is used, the server will simply return a failure message to the app.

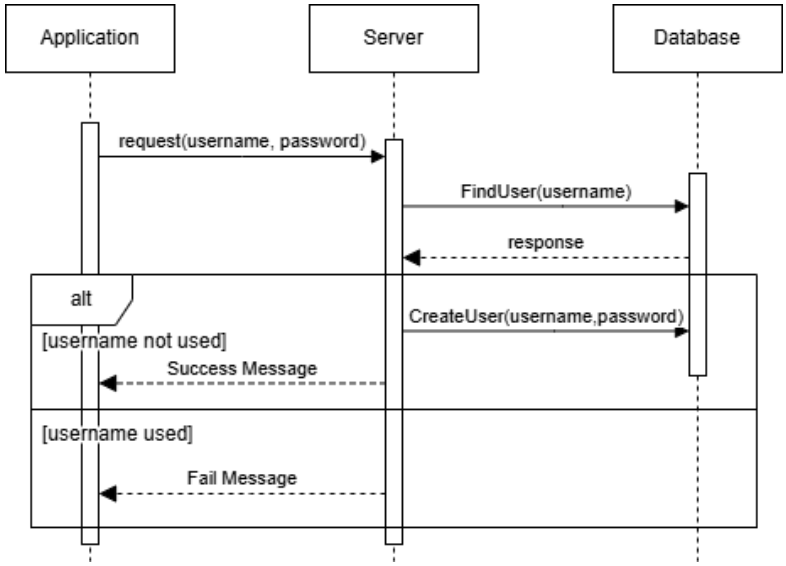

Figure 32: Register workflow

The login workflow is as follows (see figure 33). 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.

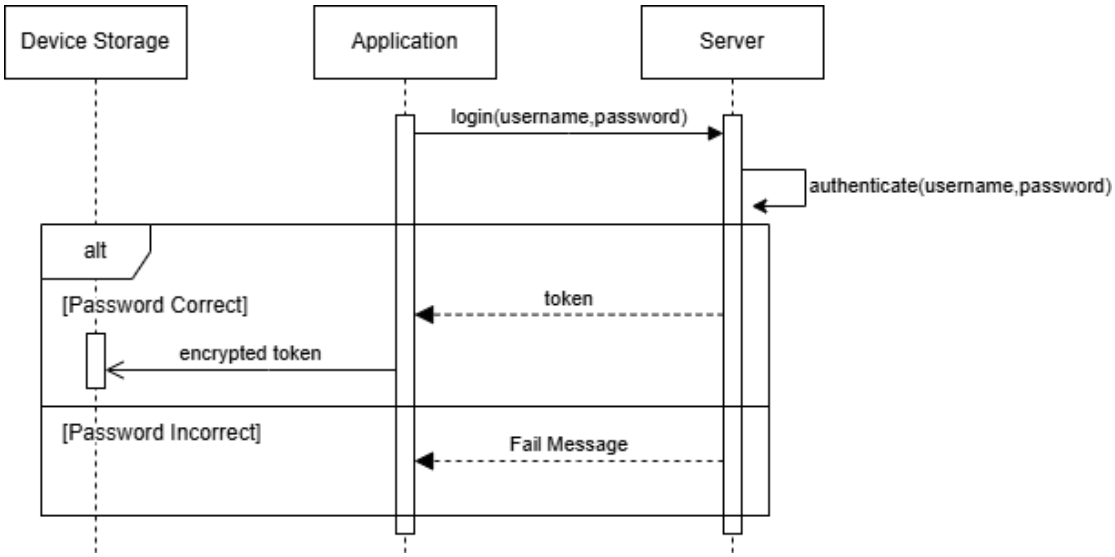

Figure 33: Login workflow

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 34). 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.

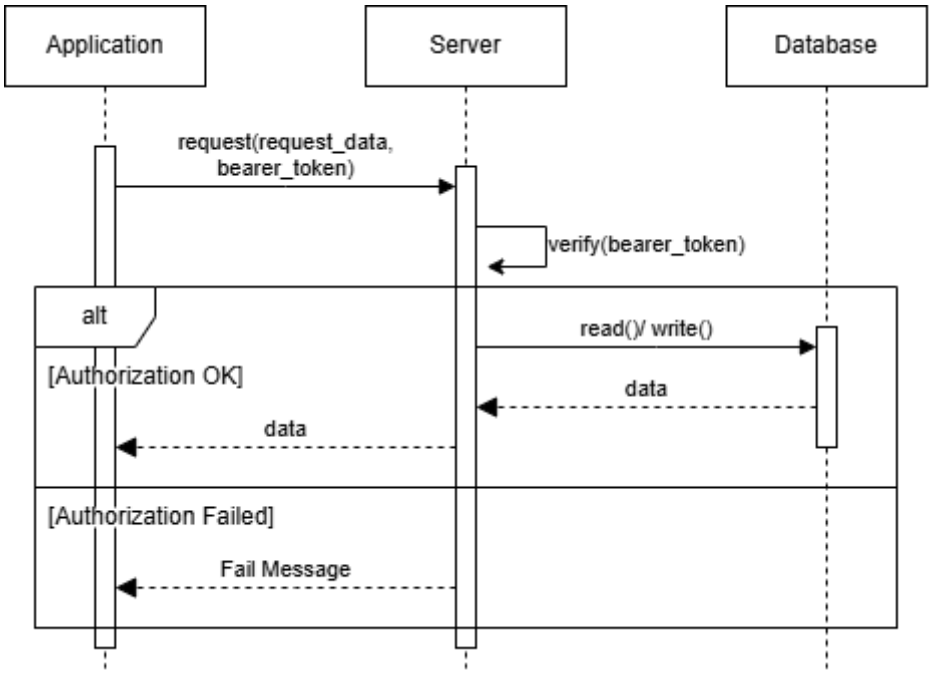

Figure 34: Authorization workflow

3.8 Profile

Users are able to manage their account in the profile page. users can logout using the logout button (see figure 35). They can also view all their reviews in a list in a similar view as the home page (see figure 36). In addition, they can also change the password by entering the old and new password (see figure 37).

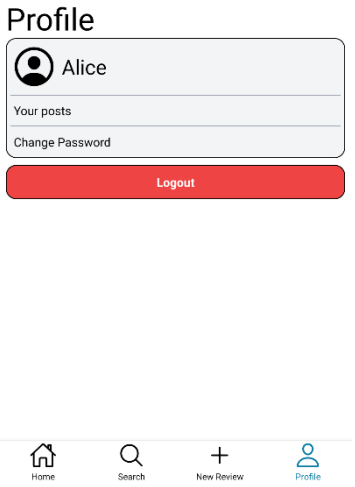

Figure 35: Profile page

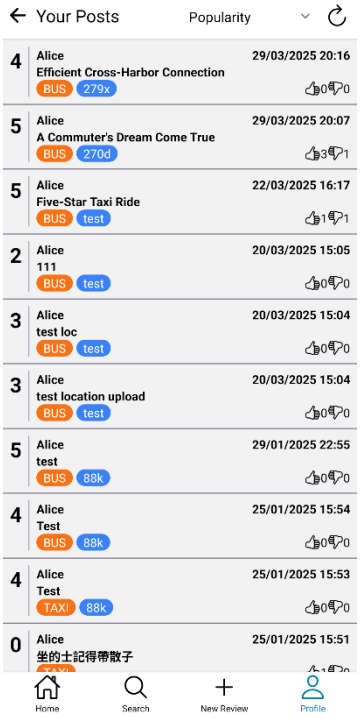

Figure 36:User's posts

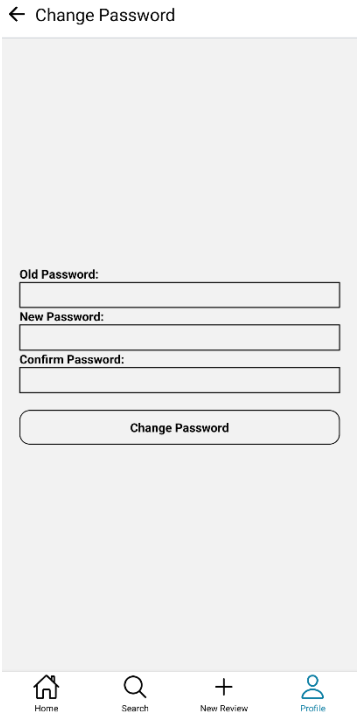

Figure 37: Change password page

4. Future Plan and Limitations

In this chapter, the future plan of this project is discussed. While all planned core functionalities of the application are completed, there is still a big room for improvement for the current app. Also, some new features ideas are not added due to time constraints.

4.1 Location Tracking History

One limitation of the application is the lack of options to save previously tracked journeys. All tracked paths must be used immediately or the data will be lost after the app is closed. This is very inconvenient for the users who want to write review in a later time. There should be a save function for the tracked path and allow users to attach any previous paths in the past.

4.2 Moderation Support

As an online review platform where users can submit whatever they want. Moderation is needed to ensure no uncorrelated or inappropriate content are being uploaded. The current app lacks administration functionalities to manage each review like deleting reviews or banning users. Moderation feature should be added in the future to prevent inappropriate use of the application.

4.3 Additional AI features

Currently, language models are only used to summarize reviews for each route. But there can be more ways to incorporate the use of AI into the app to improve the experience of using the app. For example, AI summary can also be used to highlight the difference between two routes when comparing routes. Then, users will be able to grasp the differences between routes more quickly. Beyond that, one way to massively improve user experience is to directly suggest routes using AI with the current reviews as reference. Then, users can get the optimal route with minimal research.

4.4 IOS Support

Originally, this application is intended to be released to both IOS and Android. Because of the absence of Apple hardware, there is currently no way to test the app on IOS platform to ensure that there is no bug or other issues. Also, testing and releasing applications on IOS platform requires apple developer subscription. These constraints limit the developments out of the Apple ecosystem and testing are only possible in Android devices. Therefore, IOS support is currently on hold.

5. Conclusion

To summarize, because of the lack of a dedicated public transport review platform, people face difficulties comparing different route options or collecting customer feedback. This project aims to fill this gap by developing a public transport review app that makes the process of both searching and writing public transport review easier.

The application is developed using React Native as the front-end interface, Express.js as the backend server and PostgreSQL as the database. Google Machine Learning Kit and Google Maps are integration for optical text recognition and location services. Claude 3.7 Sonnet is the language model used to provide summary of reviews.

Currently, all planned features are implemented successfully, and the application are fully operational. The available features include basic reviews browsing and submission, automatic input functions that trivialize the information input process and summary feature that accurately reflect the general opinion about each route. These features allow users to find the best route options while enabling public transport service providers to collect users' feedback for continuous service improvements.

Although this app reached complete state, many areas still need improvement. Future works of this project includes refining existing features including the location tracking functionality and implementing new features like content moderation systems and more AI powered features to further enhance the user experience.

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] Y. Douslimi. (2021, Mar. 29). You shouldn’t store your images/videos in a database [Online]. Available: https://medium.com/ensias-it/you-shouldnt-store-your-images-videos-in-a-database-6a78ffa277b2 [Accessed Mar. 12, 2025].

[8] Cloudinary. (n.d.) Resize transformations [Online]. Available: https://cloudinary.com/documentation/dev_kickstart_transform_resize [Accessed Mar. 12, 2025].

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

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

[11] 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].

[12] LiveBench. (n.d.). A Challenging, Contamination-Free LLM Benchmark [Online]. Available: https://jwt.io/introduction [Accessed Mar. 12, 2025].

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

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

[15] 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].