# Group 2 Project Report Dineth Perera (1155098386), Cheng Mo (1155144832), Anju Otsuka (1155086631), Yousif Alebyary (1155116199)

# 1. Abstract

This project creates a single page application for a social network of bus stop locations along ten bus routes. Location information is retrieved via APIs from the “Real-time ‘Next Bus’ arrival time and related data of Citybus” dataset provided by Citybus Limited. The application is built with the MERN stack, with a ReactJS view for each of administrator and user access modes, a NodeJS with Express backend, and a MongoDB database to store user and bus stop information. Ultimately, the application enables administrators to maintain the database, and users to view, search, favourite, and comment on bus stop locations.

# 2. Background

## 2.1 Coursework Relevancy

The project undertaken is relevant to the course lectures and labs as it serves as a hands-on application of the cumulated acquired knowledge that draws together all the concepts, languages, and practices covered in the course. The project is a single page application, taking the benefits of SPAs including efficient use of bandwidth, separation of view and data, and improved user experience without reloading as discussed in the course. The user-interface is designed to be kept simple and intuitive with the frontend implemented using ReactJS which uses HTML, CSS, Bootstrap, and JSX. The database is implemented using MongoDB along with the Mongoose Object Data Modeling library.

## 2.2 Motivation

One of the key reasons for working with the Citybus routes dataset is that many people are dependent on the public bus transportation for their daily lives and therefore accurately knowing the stops/route information as well as the Estimated Time of Arrival (ETA) of buses is vital. This app aims to provide a reliable method to view these data, with the added functionality of allowing a registered user to maintain an account, which can be used to view their favourite stops and post comments.

## 2.3 Similar Works

Within the scope of local transportation applications, several similar works exist on the market of which most notable are the online platforms of Kowloon Motor Bus (KMB), and Citybus & New World First Bus (NWFB). Both offer a similar set of functionalities including the display and search of bus stop location information, and estimated arrival time, with additional functionalities of bus route search, and routing services given a starting and destination point. However, neither require a user login for usage of services. Our application differs from these solutions in that it offers a tailored user experience through the provision of custom user pages coupled with user restricted functionalities such as favoriting and adding comments to locations. Moreover, the project is most comparable to Google Maps which offers an extensive version of the application in this project. That being said, where our application mainly differs from Google Maps is its concentrated focus on bus routes, and furthermore, its functionality to enable users to search and favourite for specific bus routes which is not offered in the current version of Google Maps.

# 3. Methodologies

## 3.1 Platform & Technologies

As aforementioned, the application was developed with a frontend in ReactJS using HTML, CSS, Bootstrap, and JSX, a backend in NodeJS with Express, and a database in MongoDB with the Mongoose Object Data Modeling library. The weighted advantages and disadvantages of the selected platforms and technologies are summarized in Table 3.1. Furthermore, the subsections that follow serve to elaborate on the selected platforms that were used in the project.

Table 3.1 Pros and Cons of Selected Platforms

|  |  |  |
| --- | --- | --- |
|  | **PROS** | **CONS** |
| **ReactJS** | - Beginner friendly: simple to use/learn  - Supports continuous rendering  - Better performance as it uses a virtual DOM  - Allows for the creation of reusable components, which saves time | - Does not provide as much granular control as it is only a library compared to Angular  - Requires other libraries to add more functionality: higher number of dependencies of the project. |
| **NodeJS** | - Offers high performance as Node.js interprets the Javascript code via Google’s V8 JS engine  - Caching in Node.js means that when a request is made for the first module, it gets cached preventing re-execution of code. | - Requires an understanding of asynchronous programming.  - Does not support multi-threading |
| **MongoDB** | - More flexible as it is schema-less (more clear structure)  - Utilises Document oriented Storage in the form of JSON, which has high readability and easy manipulation of data | - Not the best option for complex transaction or multi-row operation.  - High memory consumption. |

3.1.1 ReactJs

ReactJs is a javascript library which can be used for front-end user interface development. When comparing with other frameworks such as Angular, React is Javascript-centric, while Angular is HTML-centric. Furthermore, as ReactJs is library, it is far more lighter than Angular, while giving users the flexibility to import plugins for added functionality (eg. hooks). The main advantage of React is that it supports prompt-rendering, which helps to minimise the number of DOM operations and make rendering more efficient. React utilises JSX, which streamlines the development process as all technologies involved in MERN stack utilise Javascript.

This project employs functional components rather than class components as it makes the code less verbose and gives greater flexibility when handling multiple components. In addition react hooks such as useState and useEffect were used to manage the components.

3.1.2 NodeJs

Node.js is a JavaScript run-time environment. It makes writing and deploying servers easy, compared to technologies such as Apache/Nginx. It also uses non-blocking I/O meaning that there is no requirement to wait for input/output, network operations or other software. Furthermore, NodeJs is also more lightweight compared to other backend technologies as it allows users to import any required module rather than having unwanted tools pre-installed. Modules are handled by the node package manager (npm).

One such module that we utilised was Express, which is a web application framework. The main advantage of the express framework is that it is minimal and it helps to simplify the code required to write a server.

The router middleware is used to handle routing as it improves modularity, allowing easy grouping of routes that handle different sections of the website. In addition middleware such as multer and fast-csv are utilised for implementing the creation of stops using CSV file (multer handles file upload and processing is carried out by fast-csv). Bcryptjs is used for converting plain text passwords to hashed passwords. For ETA calculation, Javascript Date objects are used. A date object is created to represent the current time and another is created based on the timestamp provided by the API. The difference is calculated and upto 3 ETA’s are passed to the frontend to be displayed. The Haversine formula is utilised to implement the ‘search by distance from home location’ feature.

3.1.3 MongoDB

MongoDB is a document-oriented NoSQL database. The project utilises Mongoose Object Data Modeling library, which manages relationships among the data and acts as the bridge between the javascript code and the objects stored in MongoDB. The main advantage of Mongoose is the easy to use syntax. As it is a non-relational database, MongoDB gives greater flexibility in managing the data as the rules are not as strict. However, through the use of Mongoose schema, which has a document data structure, it allows certain field requirements to be specified such as data type, length and required status. This schema helps to make the MongoDB collections more uniform.

3.1.4 Docker

During the development phase, Docker container was used to maintain a standardised testing environment. Docker containers allow all dependencies of an application to be packed together, ensuring that the application will run exactly the same in different environments. The standardisation provided by docker improves the productivity of the developers, makes bug fixing easier and makes the code more maintainable. It makes collaboration easier as there are fewer environment specific issues and helps to reduce the time taken to set up the environment.

## 3.2 Data Schemas & Models

The User model maintains the email, username, password, comments and a list of favorite stops. The Stop model maintains the ID, list of corresponding routes, name, latitude, longitude, comments and number of favourites of a bus stop location. The Route model maintains the route id, origin, destination, a list of stops along the route, and the direction for a single route. The data schemas and models can be summarized in the ER-diagram in Figure 3.2 below.

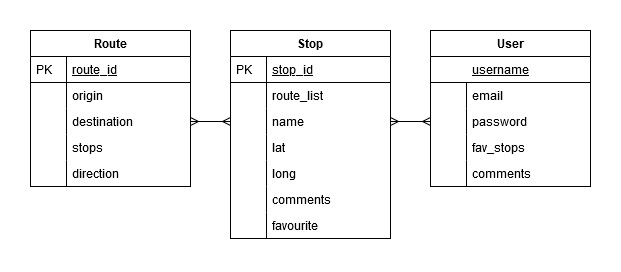
****

Figure 3.2 ER-Diagram

# 4. Conclusion

This social network of bus stop locations web application utilises the location and bus information retrieved via APIs from the “Real-time ‘Next Bus’ arrival time and related data of Citybus” dataset provided by Citybus Limited.

The application displays a list of stops that are found in 10 bus routes, which can be sorted based on the fields displayed. They are also shown on a map for easy reference. Users are able to create an account which allows them to maintain a list of favourite stops as well as post comments about stops. Charts are generated to show top 5 stops with most favourites as well most comments. The user can also set up his/her home location and search stops by distance from home location. Admin users have the ability to perform CRUD operations on the route/stop/user data, with the option of creating new stops using a CSV file. Admin can also view charts that depict the top 5 users with comments and favourites.

# 

# 5. References

[1]C. Chau, "Node.js with Express", The Chinese University of Hong Kong, 2020.

[2]J. M, "ReactJS vs Angular Comparison: Which is Better? | Hacker Noon", *Hackernoon.com*, 2016. [Online]. Available: https://hackernoon.com/reactjs-vs-angular-comparison-which-is-better-805c0b8091b1. [Accessed: 22- May- 2020].

[3]"What is Node.js and Why Should I Care? - DZone Web Dev", *dzone.com*, 2020. [Online]. Available: https://dzone.com/articles/what-nodejs-and-why-should-i. [Accessed: 22- May- 2020].

[4]"Real-time “Next Bus”arrival time and related data of Citybus. | DATA.GOV.HK", *Data.gov.hk*, 2020. [Online]. Available: https://data.gov.hk/en-data/dataset/ctb-eta-transport-realtime-eta. [Accessed: 22- May- 2020].

[5]H. Development, "Everything You Need to Know about the MERN Stack", *HyperionDev Blog*, 2018. [Online]. Available: https://blog.hyperiondev.com/index.php/2018/09/10/everything-need-know-mern-stack/. [Accessed: 22- May- 2020].

[6]N. Karnik, "Introduction to Mongoose for MongoDB", *freeCodeCamp.org*, 2018. [Online]. Available: https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/. [Accessed: 22- May- 2020].

[7]"Express Tutorial Part 4: Routes and controllers", *MDN Web Docs*, 2020. [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express\_Nodejs/routes. [Accessed: 22- May- 2020].

[8]"Using docker in software development by ING, Spotify & ADP | Apiumhub", *Apiumhub*, 2020. [Online]. Available: https://apiumhub.com/tech-blog-barcelona/top-benefits-using-docker/. [Accessed: 22- May- 2020].

[9]"Haversine formula to find distance between two points on a sphere - GeeksforGeeks", *GeeksforGeeks*, 2020. [Online].Available:https://www.geeksforgeeks.org/haversine-formula-to-find-distance-between-two-points-on-a-sphere/. [Accessed: 22- May- 2020].

# 

# 6. Appendix

## 6.1 Workload Distribution

Anju Otsuka

Implemented charting statistics for users and admin. Additionally assisted in implementation of routing for initial fetching of data from the open dataset.

Cheng Mo

Building database models, basic CRUD api for all the data and data initialization on server start. Furthermore, implemented user management, user positioning, search by distance, uploading csv files for stop creation, update and deletion of stops.

Dineth Perera

Handled Bus stop database model, NodeJs backend routing for stop related queries and ETA calculation. Basic functionality of Admin location creation and ETA view pages.

Yousif Alebyary

Fetching data from the backend and state-management. Basic functionality of list, map, user, login/signup, location and about pages. Implemented navigation and field search.