Group Members: KEI Yat-long

Lai Chun Yin

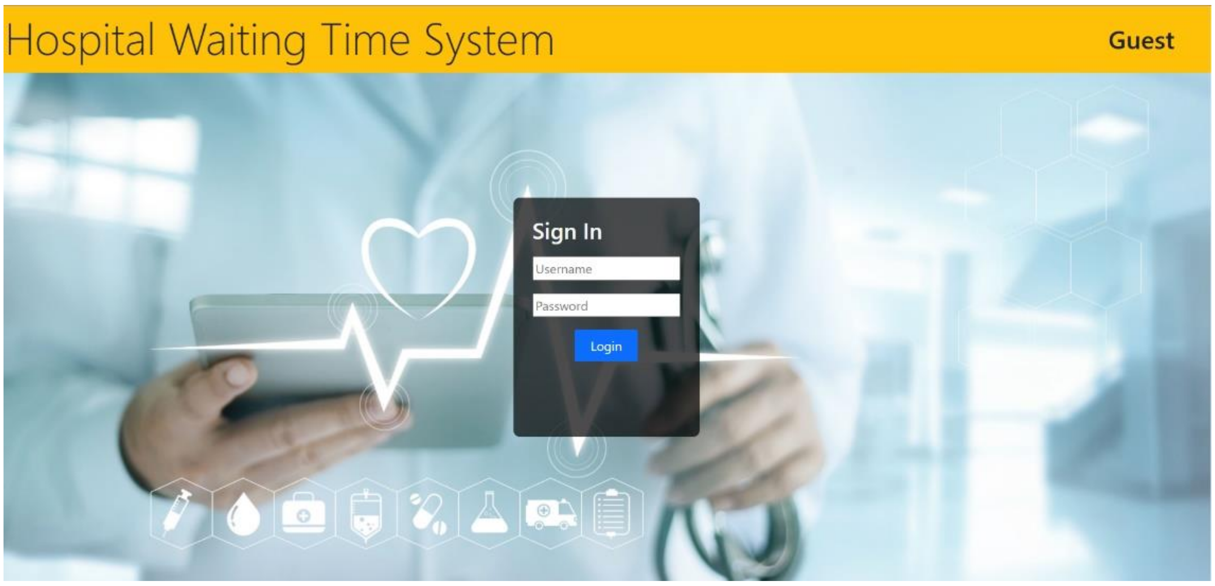
Leung Yun Ki

Wong Yik Hin

Philip Tarrantino Limas

Abstract

Our application is for checking accident and emergency waiting time of different hospitals in Hong Kong. We will be using React for the front end, Node.js and Mongoose with express framework for the back end. The application will allow logging in with administrative right or as a normal user. Users can check the comments of different places by clicking on the map provided in the web page. They can also find their favorite places that they have added. Moreover, sorting and searching functions are available on the website while admin can manipulate users and places information.



Methodologies

We used React class component instead of functional component because of the familiarity and had incorporated all class within a single file – App.jsx. We did not build the app on top of the create-react-app as for a small-scale project such as this, it avoids us wasting time in its complex configuration as well as its opiniated setup process. The degree of complexity of the code doesn't really require us to use create-react-app as it is still relatively easy to debug. Hence, we went for the more practical alternative.

We had also incorporated all the REST API routing into a single file for the back end – Project_server.js. Bcrypt is used for password hashing and JWT tokens are used for remembering login information. We also used the light-weight module node-fetch as an alternative for '*windows.fetch'* code.

The data schema used is detailed below:

1. PlaceSchema

field	**type**	**required**	**unique**	**reference**
name	String	✓	✓	
latitude	Number	✓		
longitude	Number	✓		
address	String			
telephone	String			
comments	[ObjectId]			Comment

2. *UserSchema*

field	**type**	**required**	**unique**	**reference**
name	String	✓	✓	
password	String	✓		
administrator	Boolean	✓		
favPlaces	[ObjectId]			Place

3.*CommentSchema*

field	**type**	**required**	**unique**	**reference**
user	ObjectId	✓		User
message	String	✓		

There are a few things to be noted in our design of schemas. First, name was chosen to be the primary key of our database, so the names in both the place schema and the user schema were set to be required. Second, we made favPlaces to be a field in the user schema so that a user could easily add or remove a place to his favorites and we could easily display all the favorite places of a user. Third, we made the comment schema as an independent schema so that we could easily link each comment to a user and each place to multiple comments left by the users. If we made the comment as a field of the place schema, it would be too clumsy to display all the users' comments of a place.

Comparison table of chosen platform and technologies comparing to others

	Advantages	Disadvantages
React vs Vue / Angular	1. React is more efficient since the Virtual DOM used by React only renders the real DOM when needed. Also, it only edits the original DOM instead of rewriting the whole thing when it renders the DOM. This is especially good to our project since we are writing a SPA, so we only need to change a certain part of the page and keep the remaining part fixed when the user is browsing our website. It is not necessary to update the whole DOM. 2. The React developer community is more popular than that of Vue. It is easier to seek help for React if we run into trouble [2].	1. Changes in the DOM of the Angular are bound in two-way, which means if one of the DOM or the application changes, the other one changes as well [1]. This may make handling form data more convenient, especially we have a lot of forms to deal with in the application, such as forms for updating and adding users and places. 2. Having a higher complexity makes React harder to use compared to Vue. For example, the segregation between functional and class components mean that programmers need to learn both to understand codes written by others. On the other hand, Vue does not have this problem [2].
Mongoose vs SQL DB	1. Mongoose is a NoSQL database, which is horizontally scalable. This means instead of using a more powerful computer as server, we could use more computers as servers to handle larger traffic [5]. 2. Mongoose, a NoSQL database provides a higher flexibility than the SQL database since the schema for the SQL database has to be predetermined and structured.	1. Relational SQL database can create different view for different users easily. 2. SQL database is easier to use. Plugins for accessing and manipulating SQL databases are available in many languages and platforms such as Node.js and Linux. On the other hand, different NoSQL databases may have different syntax for manipulating the databases. And we may need to use middlewares for connecting the databases if we want to facilitate our process such as the Mongoose for MongoDB [3].
NodeJS vs PHP	1. Makes server-side programming easier, only need to know how to write JavaScript. 2. Asynchronous code and quicker server-side response time, which improves the performance. 3. Can work well with NoSQL databases like MongoDB.	1. Not many people understand server-side programming with Node.js. 2. Not as portable as PHP that can runs on most platform with Apache, Internet Information Services, as well as a supported database system.

References

[1] A. Gazta. (2019, July). Understanding the pros and cons of Angular Development. *Programming* [Online]. Available: https://www.greycampus.com/blog/programming/good-and-bad-of-angular-development

[2] DDI development. (2020, December). The Good and the Bad of Vue.js Framework [Online]. Available: https://ddi-dev.com/blog/programming/the-good-and-the-bad-of-vue-js-framework-programming/

[3] M. Smallcombe. (2020, May). SQL vs NoSQL: 5 Critical Differences. *Data Intgeration* [Online]. Available: https://www.xplenty.com/blog/the-sql-vs-nosql-difference/#:~:text=SQL%20databases%20are%20relational%2C%20NoSQL%20are%20non%2Drelational.&text=NoSQL%20databases%20have%20dynamic%20schemas,graph%20or%20wide%2Dcolumn%20stores

[4] Boardinifinity. MERN V/S MEAN Stack – Difference Between [Online]. Available: https://blog.boardinfinity.com/mern-vs-mean-stack-explained/

[5] A. Delessert. (2012, October). Node.JS with NoSQL or SQL? [Online]. Available: https://stackoverflow.com/a/12756025

Appendix

Workload Distribution

This appendix consists of the workload distribution for the course project. The user interface design and implementation were done by Lai Chun Yin and Wong Yik Hin. The functionality for the user action was a collaborative effort by Lai Chun Yin, Philip Tarrantino Limas and Leung Yung Ki. The functionality for the admin action was done by Kei Yat Long and Leung Yun Ki, the charting of the historical data by Wong Yik Hin and Kei Yat Long and the non-user actions by Philip Tarrantino Limas and Leung Yun Ki. The Back-end code was a collaborative effort by Lai Chun Yin, Leung Yun Ki, and Wong Yik Hin. The project outline and report were drafted and written by Philip Tarrantino Limas and Kei Yat Long.