The University of Hong Kong

Department of Computer Science

2023 - 2024

COMP4801 Final Year Project

Detailed Project Plan

“Instant Messaging Software for Academic Consultation”

Table of Contents:

1. Project Information:

Project Title: Instant Messaging Software for Academic Consultation

FYP Computer Account Name: fyp23039

Supervisor: Dr. Chim, Tat Wing

Student: Fung Ka Ho (3035902635), Chan Chin Hung (3035780879)

Project Webpage: https://wp2023.cs.hku.hk/fyp23039/

2. Introduction:

Communication is the best way for people to exchange information and ideas. With the rapid development of technology, the ways people converse with others have diversified and are no longer limited to face-to-face interaction. Nevertheless, among the numerous approaches, using instant messaging software to communicate with others is the most popular method of communication nowadays.

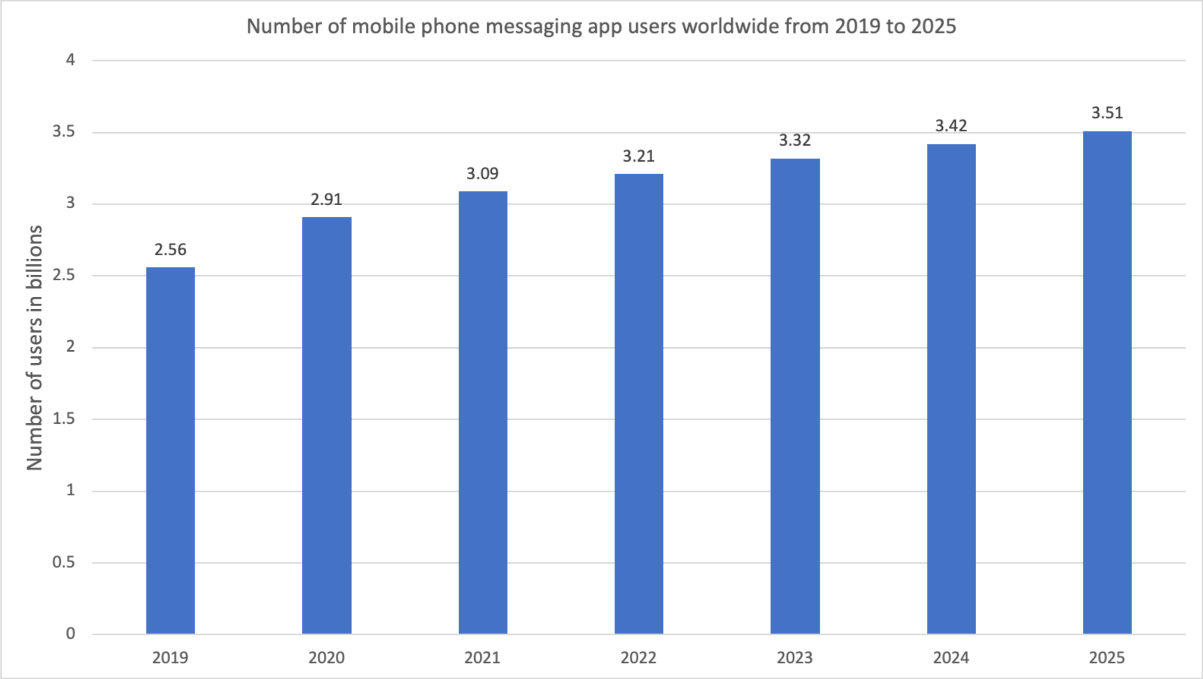

Figure 1: Number of mobile phone messaging app users worldwide from 2019 to 2025[1]

Instant messaging software refers to an application that allows users to perform communication with others in real-time. According to the statistics conducted by L. Ceci in 2023, there were approximately 2.56 billion mobile phone users who made use of instant messaging software for their communication in 2019.[1] This number is still growing, by 2025, the number of users is expected to increase to 3.51 billion.[1]

The convenience and ease are the crucial reasons that attract people to use instant messaging software for their communications. As long as the chat app user holds a computer, tablet or mobile phone, they can deliver their ideas and share information with their friends without any format, time or geographical constraints. Moreover, benefitting from the innovative functionality provided by instant messaging software, apart from interacting through texts, app users are allowed to exchange multimedia elements such as voice messages, emojis or images, which significantly facilitate communication efficiency and productivity.

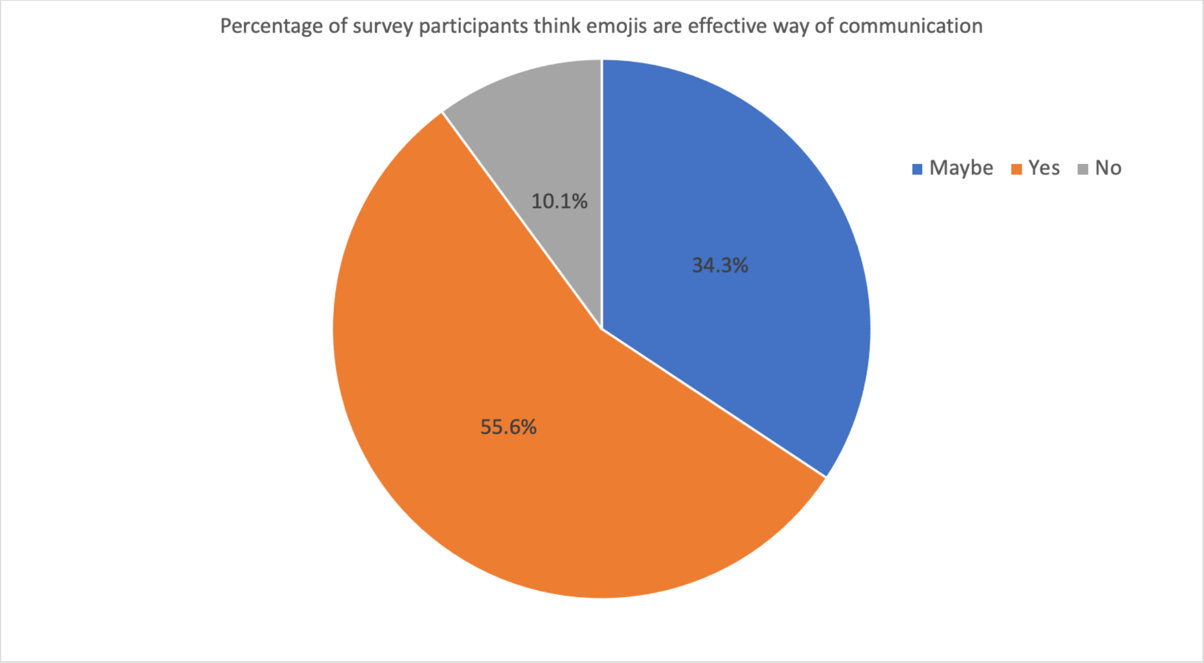

Figure 2: Percentage of survey participants think emojis are effective way of communication[2]

According to the research conducted by Professor Priya Chandran in 2018, over a half of the survey participants agreed that using emojis can effectively facilitate their conversation.[2] This shows that combining non-text elements with text elements in messaging can facilitate a higher level of efficiency and productivity in communication as app users have multiple ways to convey ideas and express their emotions. By using emojis, users are capable of showing their facial expressions which cannot be done solely with text.

3. Motivation:

Nowadays, when students encounter any difficulties in their learning and they wish to consult their teachers, they need to go back to school in person or draft an email written with a serious tone and in formal format. Compared to using instant messaging software, such approaches are relatively inconvenient and time-consuming. Also, the usage of email is not generally perceived to be for instant messaging or casual chatting as it conventionally requires a clearer structure and is used for discussing one specific subject. With instant messaging software, students can consult their teachers in a much convenient and easy manner. This enables communication to be done in a more efficient and productive way as students can speak freely with fewer or no format restrictions.

In addition, there is a lack of interaction between students and teachers in most lessons nowadays.[3] Most of the time during the lesson is taken up by presenting the notes content.[3] Due to such insufficient interaction, student's learning efficiency has been reduced. By adding the feature of creating and answering quizzes in the instant messaging software, it can further engage students in class activities which effectively can improve the learning efficacy of the student.

4. Objective:

1. To provide an applicable graphical user interface for teachers to share interactive quizzes during the lesson to facilitate the interaction between students and teachers during class
2. To provide a consultation platform for students to solving the academic problems
3. Allow students and teachers to have academic discussion privately or in groups
4. Support message storage such that the app users can review their past chat messages

5. Methodology:

5.1 Frontend:

A progressive web app will be built for the project. As it is relatively easy to build, maintain and release to users. A progressive web app has more capabilities than a normal web app, which makes it very user-friendly as it combines the best aspects of web and app experience and works offline or on low-quality network connections owing to service workers that cache the app data locally on the device. [12] Many existing progressive web apps, such as Uber, Pinterest, and Starbucks, provide a responsive and intuitive user interface that is similar to native applications. Therefore, we will build a progressive web app.

In order to convert a normal website to a progressive web application, the service worker and manifest files are a must. The manifest file mainly configures the metadata about the web app, such as its name, icons, theme color, orientation, and start URL. The purpose of the manifest is to enable the web app to be added to the home screen of the user's device, and to customize its appearance and behavior when launched from there. [16] Whereas service worker is a background script which runs in another thread from the browser. Since it is able to act as a proxy to intercept and handle network requests, it allows the website to be visited normally even offline by retrieving network resources stored in browser cache. One important feature is the push notification which allows the user to receive messages even if the browser is not open. [12, 17]

React is a JavaScript library for creating web and native user interfaces that was developed by Facebook in 2011. It uses components to make the code more reusable and maintainable as well as combines complex UI elements that improve the rendering speed with a virtual DOM method. [10-11] Compared with other frontend frameworks, react has a large eco-system that provides detailed documentation and adequate react user interface libraries. Moreover, Vite build tool will be used together with react to speed up the development cycle as it makes the development server faster and real time update of the source code can be achieved. [10,16] React with Vite will be used for the frontend website development. Then, manifest and service worker files will be added to make it a valid Progressive Web App.

5.2 Backend:

The backend server will be built using Node.js, which is a popular and cross-platform JavaScript runtime environment for developing web servers. It uses an asynchronous event-driven model that can run smoothly on various operating systems and handle concurrent connections. [3] In contrast with other server frameworks, its comprehensive and reliable package manager npm provides over two million packages with detailed documentation. [5] It allows us to build complex server logic and reuse the features built by other developers.

Express frameworks will be used on top of Node.js to offer a fast and flexible way to handle and create API requests and routing. Its middleware function allows us to break down the code into smaller modules such as error-handling, logging, authentication, etc. [6] These modules can be reused in different API endpoints and thereby speed up the backend development process.

The WebSocket protocol will also be used for real time communication. Unlike normal http request, it reduces the number of TCP handshake and allows bi-directional communication, providing high efficiency and low latency connection ,[10]which is essential for our application, as we need to handle and transfer a large amount of real time chat data.

5.3 Database:

MongoDB is a documented-based NoSQL database. It stores data in a structure of BSON that is similar to JSON (JavaScript Object Notation), which is a data-interchange format in JavaScript and is simple and readable for humans. BSON has extra features that allow effective indexing and querying of data. [5,9] Therefore, it is a suitable database for Node.js server. Compared with traditional relational databases, it is efficient and flexible when performing data manipulation on huge datasets. [15] Additionally, there are multiple user-friendly packages that support the mongoDB driver connection and facilitate data manipulation, data modeling such as mongoose, monk, etc.

5.4 Deployment:

The server of the project is hosted on Amazon Elastic Compute Cloud (Amazon EC2). As ec2 provides lots of types of virtual machines which allows users to rent according to their needs. Since it offers flexible and scalable virtual machine instances with plenty of built-in functionality and features, the maintenance cost is reduced as physical machines or other hardware are not required. The node.js server will be deployed on a 24/7 ec2 virtual machine instance.

The database is hosted on a cloud service called MongoDB Atlas which provides a multi-cloud database platform. Azure, AWS, Google cloud are the famous cloud service providers that allow us to deploy the database on them. Moreover, it offers real-time analysis of the data and graphic interface as well as data visualization features which accelerate the development and debugging process. [19] However, we have to decide which cloud platform suits our needs best. Since the server is deployed on AWS, our database will deploy on the same cloud platform.

5.5 Testing:

5.5.1 Unit Testing:

Vitest will be used as the testing framework because it is vite-native. Vitest is built on top of the Jest which is the most popular JavaScript testing framework. Jest offers lots of testing features such as snapshot testing, mocking and coverage reports and these API and ecosystem libraries can be accessed by Vitest. [13] In addition, it is relatively easy to configure and build through a single vite.config.js file in contrast with Jest which requires an extra jest.configure.js file. With Vite, Jest has many more limitations. For example, it is not injected into all the modules and it may not work properly for coverage tests. [14]

Vitest will be used by us to test the low-level logic and functions as well as the user interface content. The react testing library, which is a testing utility, will be used together with Vitest. As it allows direct working with the actual DOM nodes. In this way, Code can be written to simulate the user interaction with the DOM and verify the real content that appears on the website. [18] We will test each page by rendering its react DOM and then verify the content. For instance, the text content, the image source URL and the buttons are checked.

5.5.2 API Automated Testing:

Server security is crucial as it holds a lot of business logic and data. Even though the website can restrict the user's input and limit the access of the endpoint, there are still other ways to directly communicate with the endpoint. This poses potential danger to the API. Therefore, the API endpoint security will be tested.

Postman is a well-known tool for backend developers and software testers to test the API. It provides a comprehensive set of tools that cover various aspects such as, testing, design, mock server, API detection. It also supports a wide range of protocols such as gRPC, Rest, GraphQL, etc. [18] We focus on the testing features and writing automation scripts to test all the HTTP API and WebSocket API.

5.5.3 Integration Testing:

Although Vitest has the browser mode feature which allows testing in native browsers, it is still in the early stages of development and there are potentially a lot of bugs and issues in the code. [13] Another tool called Puppeteer is a node js library that can be used to control the chromium browser. It offers lots of API which allows us to perform actions such as clicking a button, pressing the keyboard, navigate to other pages and help us to test the real scenario of the software. Online and offline behavior and the actual user interaction on the software will be tested.

6. Schedule and Milestones:

Items	Date	Task
1	September 2023	● Detailed project plan writing ● Project web page making
2	October 2023	● Graphical user Interface design and related research ● Database design and related research
3	November 2023	● Graphical user interface implementation ● Database setup ● System design and related search
4	December 2023	● Real-time chatting feature implementation (for both server side and client side) ● Interactive quizzes creating and answering system implementation ● Detailed interim report writing ● First presentation preparation
5	January 2024	
6	February 2024	
7	March 2024	● System testing and debugging ● System Optimization
8	April 2024	● Final report writing ● Final presentation preparation

7. Reference:

[1] L. Ceci and A. 29, “Mobile messaging users worldwide 2025,” Statista, https://www.statista.com/statistics/483255/number-of-mobile-messaging-users-worldwide/#:~:text=Global%20number%20of%20mobile%20messaging%20users%202019%2D2025&text=In%202021%2C%20an%20estimate%20of,popular%20mobile%20messenger%20app%20worldwide (accessed Sep. 23, 2023).

[2] P. Chandran, S. Tayade, and N. Pujari, “Survey on usage and influence of emoji in social media communication,” A survey on usage and influence of emoji in social media communication, https://ijcrt.org/papers/IJCRT1802532.pdf (accessed Sep. 23, 2023).

[3] B. Beth Hurst, R. Randall Wallace, and S. B. Nixon, “The Impact of Social Interaction on Student Learning,” *Reading Horizons: A Journal of Literacy and Language Arts*, vol. 52, no. 4, Oct. 2013. https://scholarworks.wmich.edu/cgi/viewcontent.cgi?article=3105&context=reading_horizons (accessed Sep. 23, 2023).

[4] Node.js, https://nodejs.org/en (accessed Sep. 17, 2023).

[5] “Introducing Json,” JSON, https://www.json.org/json-en.html (accessed Sep. 11, 2023).

[6] “NPM,” npm, https://www.npmjs.com/ (accessed Sep. 11, 2023).

[7] Express, https://expressjs.com/ (accessed Sep. 11, 2023).

[8] I. Fette and A. Melnikov, “The WebSocket Protocol,” RFC Editor, https://www.rfc-editor.org/rfc/rfc6455 (accessed Sep. 11, 2023).

[9] “JSON and Bson,” MongoDB, https://www.mongodb.com/json-and-bson (accessed Sep. 11, 2023).

[10] React, https://react.dev/ (accessed Sep. 17, 2023).

[11] “Virtual dom and Internals,” React, https://legacy.reactjs.org/docs/faq-internals.html (accessed Sep. 17, 2023).

[12] Fortunato, D., & Bernardino, J. Progressive web apps: An alternative to the native mobile Apps. In *2018 13th Iberian Conference on Information Systems and Technologies (CISTI)* IEEE, Spina, pp. 1-6.

[13] V. Anthony Fu, Vitest, https://vitest.dev/guide (accessed Sep. 20, 2023).

[14] “Vite-Jest,” GitHub, https://github.com/sodatea/vite-jest/tree/main/packages/vite-jest (accessed Sep. 24, 2023).

[15] T. Crawford, "A comparison of server side scripting technologies," in Proceedings of the International Conference on Software Engineering Research and Practice (SERP), 2017, pp. 69-76.

[16] T. Majchrzak, A. Biørn-Hansen, T. GrØnli. "Progressive web apps: the definite approach to cross-platform development?,", 2018.

[17] “Introduction: Testing library,” Testing Library RSS, https://testing-library.com/docs/ (accessed Sep. 24, 2023).

[18] “API tools,” Postman API Platform, https://www.postman.com/product/tools/ (accessed Sep. 24, 2023).

[19] “MongoDB atlas database: Multi-cloud database service,” MongoDB, https://www.mongodb.com/atlas/database (accessed Sep. 24, 2023).