DESIGN AND DEVELOPMENT OF A FAMILY MOBILE APPLICATION

By

YEO KENJI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONOURS)

COMPUTER ENGINEERING

Faculty of Information and Communication Technology

(Kampar Campus)

FEBRUARY 2026

COPYRIGHT STATEMENT

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Information Technology (Honours) Computer Engineering at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I would like to show my gratitude and appreciation to my supervisors, Ts Dr Chang Jing Jing who has given me guidance and advice throughout the planning of report and development of FYP2. In which motivate me and smoothen the development process.

Furthermore, I would also like to express my thanks to my faculty FICT, for designing numerous intuitive and informative courses throughout the 3-year computer engineering program. Those classes prepare me with adequate information and give me higher advantage when seeking for work in the future.

ABSTRACT

In today's digital era, technology plays a significant role in shaping interpersonal communication and social dynamics within families. With the rise of smartphones and online platforms, family members—especially children—are increasingly exposed to fragmented and superficial communication, which can weaken emotional bonds. This project aims to address this challenge by designing and developing a mobile application tailored specifically for family interaction. The app will be served as a centralized platform combining essential features such as a shared calendar for coordinating family schedules, a secure photo gallery for storing memories, real-time messaging to foster connection, location sharing for safety of member, event creation and reminders for help with daily chores to prevent overuse of mobile phones. Moreover, a Natural Language Processing sentiment text analysis will be integrated to interpret emotions in text messages. The development process will follow the Software Development Life Cycle (SDLC) model to ensure systematic implementation. Flutter and Dart are the development platform and language. User-centred design principles will guide the interface and feature set, ensuring accessibility for both tech-savvy and less-experienced users. The broader goal is not only to develop an app but to contribute to the field of digital parenting and family cohesion through technology.

Area of Study: Mobile Application Development, Artificial Intelligence Development

Keywords: Family Communication, Mobile Application, User Experience, Machine Learning, Artificial Intelligence

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

w_i	Weight of message i
age_i	Age of message i in days
S_e	Total weighted score for emotion e
$score_{i,e}$	Prediction score of emotion e for message i
P_e	Final displayed percentage for emotion e
ΣS_k	Total weighted score of all emotion classes
lat	Latitude coordinate
lng	Longitude coordinate

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
API	Application Programming Interface
AOT	Ahead-of-Time
AR	Augmented Reality
CSV	Comma-Separated Values
FCM	Firebase Cloud Messaging
FICT	Faculty of Information and Communication Technology
FYP	Final Year Project
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
IOS	iPhone Operating System
JIT	Just-in-Time
JSON	JavaScript Object Notation
ML	Machine Learning
NLP	Natural Language Processing
OCR	Optical Character Recognition
ROC	Receiver Operating Characteristic
SDK	Software Development Kit
SDLC	Software Development Life Cycle
TF-IDF	Term Frequency–Inverse Document Frequency
UI	User Interface
URL	Uniform Resource Locator
UX	User Experience
VS Code	Visual Studio Code

CHAPTER 1

Introduction

In this chapter, it presents the problem statement and motivation of the research, the contributions to relevant fields, and the outline of the thesis. By getting the grasp of what the challenges are, a better understanding can be achieved, which can help for later discussion on the application itself and its implementation.

In this rapidly developing era, acquiring smartphones and spending huge amounts of time on them have become inevitable. Many younglings born around Generation Z could have heard the same complaint repeatedly from their parents: “The phone has ruined my child!”. Despite the apparent irony of parents choosing to give phones to their children for convenience instead of spending time with them, they may have similar opinion regarding their parent’s phone usage too. According to [1], about four in ten teenagers have the same concern about their parents of overspending their time on their phones.

With the ever-growth of isolation and lack of effective communication, the possibility of misunderstanding rises and the gap between generations widens. Many Companies sees these trends as opportunities and made platforms to encourage online communication. As stated by [2], the appearance of social media could even date back to 1792. Although these platforms serve their purpose in strengthening relationships, they are usually focus on broader public used rather than family-centered communication, their features often lack family-focused functions and transpire as surface-level products.

Artificial Intelligence (AI) and Natural Language Processing (NLP) sentiment text analysis are beneficial in aiding better comprehension. As mentioned in [3], sentiment text analysis is applied in many sectors, including determining the overall opinion and feelings of the human language. With the help of these technologies, it can “translate” emotion from text messages, which help in interpretating messages correctly.

1.1 Problem Statement and Motivation

Research conducted by [4] with the participation of ten to eleven years old student has shown that out of the 351 students involved, 158 proved to own smartphones, which is approximately 45 percent of the total students. [5] highlighted that the main reasons of this phenomena could be attributed to the declining price of mobile phones, or the availability and accessibility of getting one.

There is actually more good than bad in having a smart gadget, especially in this fast-paced environment, the issue often lies in how social media operates. This is also the similar in allowing children to use smart phone, the action itself is not wrong but the lack of guidance and monitoring from parents are. There are many reasons why children should not have social media accounts, but greedy companies often go through superficial effort and turn a blind eye on age restriction enforcements, such as ability to lie about age. That's the main reason why they could easily register and obtain an account on their platforms.

According to [5], since smartphone access has been normalized, the continuous exposure to online content, over-comparing with others and unrealistic expectations could be obtained through social media. These could lead to negative effects especially on teenagers and kids, where their mindsets are yet to grow completely. If parents continue to ignore, tolerance sanctioning their children to overspend their time by using mobile phones, it could lead to nomophobia, as stated in [6]. Without constant dopamine load from their phone, they could feel anxious, lack of confidence and insecurity.

Moreover, the overwhelming number of apps installed on one device is also a concern to many. To elaborate, they all push notifications in different forms and styles to drive engagements, which is mentioned in [7]. [8] further shows how the notification behaviour directly affects user experiences. This could result in clutters of notification bar and buries important and urgent notifications. Without acknowledging which notifications are significant, confusion will arise and delay in work on communication will happen too.

Growing up from a family of seven, communication is important. As stated in [9], mutually supportive families are significant to shape a heightened well-being and life satisfaction. But nowadays, most parents have to work and leave their kids to local nursery or tutor class, which could lead to less communication among them. [10] support this claim by presenting that problematic smartphone use can affect communication within families.

In addition, clean and easy to use environment is also crucial, as shown in [11], interruptions in app environment have shown to affect performance in every day communication. [12]'s studies on mobile app for older adults have emphasized the importance of readability and clarity, which can be focus on the development too since family members usually covers wide range of age groups.

Thus, the motivation of this thesis is to develop a family-centric mobile application to resolve these issues. By integrating functions such as emotion detection system, messaging system, photo gallery, shared calendar, event creation and location sharing features, the mobile application aspires to provide families with a holistic tool that simplifies daily interactions and planning.

In conclusion, the proposed application is to be a medium that could strengthen bonds and make families' lives easier.

1.2 Objectives

The aim of the thesis is to propose a user-friendly family-oriented mobile application within the final year project 2 timeline, which is around 4 months. Implement and test the usability of core features, conduct test groups and surveys to gather feedbacks. More specifically, the objective can be broken down into two sub objectives: development of the family mobile app and integrating the app with Natural Language Processing (NLP) sentiment text analysis features into its messaging system.

Fundamentally, the project strives to solve the obstacles of misunderstanding while concurrently strengthening family bonds. The application focuses on acting as a medium for communication among families, which is scarce in this era of fragmented apps. By doing so, family members will gain a better understanding of each other and able to communicate more effectively.

Firstly, the development of the family mobile app will have numerous functions, which are messaging systems, gallery systems, shared calendar system, events creation system and location sharing features. With these functions, it will be sufficient and plentiful for a family mobile app. These functions should also be clean and clutter-free to ensure effective communication.

Moreover, an NLP sentiment text analysis features will be integrated into the messaging chat system, where it will analyze the emotion of each family member by their text messages. It will be updated if new messages are added and have a decay function which mimic real life emotion, which can provide accurate emotion of a member in a specific period. In this way, the user can think for a more appropriate and fitting response when replying in the chatroom. Furthermore, an AI assistant powered by Google's Gemini can also provide user with advices in planning and messaging.

In addition, the user interface should follow the rule of "ease of use" from the principle of user experience design. In this case, the app will avoid on multiple window actions and repetitive functions on different place in the app, where they normally cause more confusion and hassles. In short, the application should be easily accessible to its target audience, which ranges from children to elderly. By doing so, the user can have a smoother and more enjoyable experience while using the application. Accordingly, the users will also be less likely to abandon this application, guaranteeing long term loyal user base.

The project does not emphasis on cross-platform compatibilities. Instead, it will only focus on the Android system due to difference in code inputs, app behaviors and other limits that will be discuss in later parts.

1.3 Project Scope and Directions

The scopes of the project include implementation of features mentioned earlier, easy accessibility and intuitive user interface. It should have working application for user testing within a fixed time frame. Instead of raising unreasonable expectations, by defining a scope beforehand, it can ensure that the project can be completed in time and in a servable state instead of coming out with overextended app with incomplete functions.

Furthermore, the feature should be verified whether they work as intended in late development. There is no need for excessive number of features, the focus should be on implementing essential and fully functional core features. Bug testing and iterative correction will also be excessively run through to prevent errors and ensure the app will work as intended. As mentioned before, the app will be focus on Android only. Therefore, a detailed timeline should be predefined before developing the project, achieving a usable prototype before deadline.

The project should be concluded within the February semester, which is around 4 months. By the end of this semester, a functional app should be ready for test and presentation. Ideally, the test groups should consist of families, who will experience the application over the course of at least one week before submitting feedback for future improvements. The project seeks to receive valuable feedback from its target audience. By doing so, reviews can be used to identify problems which are normally overlooked by its developer.

1.4 Contributions

As stated above, the project seeks to solve the difficulty of communicating across long distances, decreases the exposure of children to social media and encourages communication and sharing among family members. The addition of sentiment text analysis is also aimed to aid the dilemma of misinterpretation of emotions.

The main difference of this proposed application to other market available platforms is that it focuses on being an all-around tool, where the user does not have to seek a different application. Instead of requiring users to switch between separate apps, it aims to be a unified one stop and provide users with their needs in communicating with family members.

Besides, another contribution that make difference from conventional platform is its NLP sentiment text analysis feature, which is embedded in its chat messaging system. By doing so, it let the user to be more aware and thoughtful when communicating with their families.

From a development perspective, the project also contributes to a documented app that combines Flutter, Google Firebase services and backend cloud APIs in one student project. These implementation shows future developers how to plan a similar project within this time space and integrate these technologies in a single coherent system.

1.5 Report Organization

To prevent confusion, the details of this report are organized in these following chapters.

In chapter 1, it introduces the project background, problem statement, motivation, objectives, project scope, project direction, contributions and the overall report organization. In chapter 2, it presents literature reviews, which includes related scholar publishes in relevant technologies used, existing family-oriented applications and sentiment text analysis.

In chapter 3, it explains the methodology and overall approach used for the development-based project, which include the system architecture and workflows. In chapter 4, it describes the system design, such as major modules in the app, the database, service interaction and interface logic.

In chapter 5, it records the implementation of the project, including the hardware and software setup, configuration steps, operational screenshots, implementation issues, and the post-testing enhancements added after user feedback. Chapter 6 evaluates and discusses the completed system through functional testing, questionnaire-based user feedback, emotion-classifier performance metrics, project challenges, and objective evaluation. Finally, Chapter 7 concludes the project and provides recommendations for future improvement and possible continuation of the system beyond the current prototype stage.

CHAPTER 2

Literature Review

This literature review aims to explore relevant information in two main directions: technology reviews and existing knowledge or practices related to family-oriented mobile applications. By examining academic research, website resources, and user experiences, the review identifies prevailing features, limitations, and potential areas for innovation and improvements. This comprehensive analysis informs the development of a proposed family mobile application that addresses identified gaps and leverages best practices. In short, this chapter will explain both the technologies applied and literature background related to this project.

2.1 Reviews of the Technologies

The technology reviews are crucial in the development of the app as it provides knowledge on how these technologies should be implemented so that they can operate correctly. In this part, it will be organized in these sections: hardware platform, firmware or operating system, database, programming language, algorithm.

In addition to these technologies that are used directly by the app, the development environments and Ai-assisted tools also play a crucial supporting role in this project. For instances, development tools such as Android Studio and Visual Studio Code help manage source codes, emulator testing, debugging and build the app itself, while AI-assisted tools like Codex and ChatGPT can support code explanation, code implementation and debugging assistance when used under careful human supervision.

2.1.1 Hardware Platform

The intended platform for the project is Android smartphone, which is appropriate and suitable as the project focus on portable and convenient family communication. As

mentioned before by [4] and [5], the accessibility of acquiring a smartphone is easier as time goes on, so this platform will be the easiest to connect to most families. Moreover, smartphone platform is suitable as it fulfils the requirements such as internet connectivity, notifications system, camera and storage access and GPS-based location services. The suitability of smartphones as the hardware platform is also supported by the normalisation of including them in family's daily lives. Research by [10] and [12] prove that how central smartphones has become in everyday family interaction.

Besides functioning as the final platform, the app also requires real device behaviour such as notification receiving, location sharing and camera features. Thus, choosing the smartphone as the main hardware platform is very practical from the testing and implementation perspective.

2.1.2 Firmware/ Operating System

Since the project will be on smartphone platform, Flutter is suitable for cross-platform mobile UI development. According to [13], Flutter is an open-source mobile SDK developed by Google in 2017. It is very convenient and easy in developing Android only mobile app which fulfil the project scope. [14] shows that the project benefits from Flutter's widget-based UI model and hot-reload workflow, which helps in rapid prototyping and iterative refinement during development.

Flutter is a friendly platform to beginners. One of the useful built-in libraries, "material.dart" contains prebuilt classes and features such as widgets, buttons and app bar, which are useful when building a UI from ground up. It also contains "scaffold", which is crucial to build a layout of the application. Not to mention the variety of icons available for use too.

Furthermore, since it was developed by Google, there are plenty of supports and tutorials available on the Internet to learn from. The huge user base of this app can also

help me to solve difficult problems. With both efforts from Google and its users, Flutter is constantly being updated and optimized, which also ensures a healthy life span for the platform, so the skills learned won't be wasted.

Similar to the reason smartphones are chose, Android is an OS with years of supports and developments where it can support the project requirements, which are firebase integration compatible, push notifications, location permissions and basic media handlings. According to [15], Android is dominant in the mobile OS market, which is around 67.46%, as shown in Figure 2.1. Therefore, by prioritising on Android, it can cover most of the user in the market.

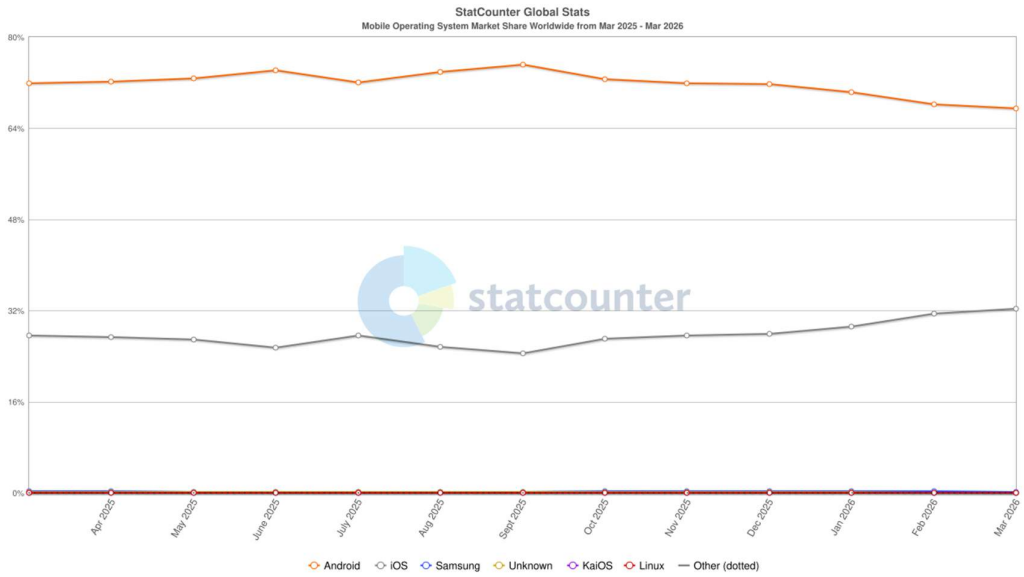


Figure 2.1 Mobile OS Market Share Worldwide

2.1.3 Database, Storage and Authentication

Google's Firebase is used as the database platform in this project. In the firebase console, three functions are implemented, which are Firebase's Firestore, Firebase Storage and Firebase Authentication.

According to [16] and [17], Firebase's Firestore can support real-time data synchronisation across data, such as storing user profiles, group info, message data, event records, notification feed and shared locations records. By [18], Firebase Storage can then be use for storing profile pictures, group pictures, chat pictures and gallery features. [19] says that Firebase Authentication supports account creation and sign in via emails, guest sessions and identity linkage of the app.

By implementing all database into the same platform, it is easier to manage and alter during development. Firebase stability and long-term support by Google are also suitable with the real-time and frequent messaging system.

Moreover, Firebase is suitable as it reduces backend complexity for a student development-based project. Instead of starting all these database functions from scratch, it provides these services in a unified and clear environment. By doing so, this allows the development effort to focus more on application logic and features design.

Lastly, Firebase Cloud Messaging and Cloud Functions helps in notification trigger feature of the app. By using cloud-based notification delivery structure, the notification logic and stability is more reliable.

2.1.4 Programming Language

Flutter uses Dart as its development programming language. According to [14], Dart is suitable for fast UI iteration and reactive mobile interaction. Dart is purely object-oriented, and class based, as stated by [20]. Besides, [21] suggested that it resembles C language and Java, which is purposefully designed with ease of use, familiarity to the vast majority of programmers. The main purpose it was created was to be a better and more efficient version of Java since Java is widely used in application development. To

conclude, it is fast and efficient due to its AOT compilation for production, and JIT for development speed, as refer to [22]. Figure 2.2 represent the differences in code length between Java and Dart.

Python is used for the backend API language that support the NLP sentiment, image categorizing system and AI implementation via FastAPI [23] and Cloud Run. As mentioned in [24], Python is a lightweight and good readability language. It works on many platforms and has a huge support user base. Lightweight JavaScript is used as the backend API output to communicate with the app. With its simple syntax and popularity, there are a lot of support and existing libraries ready to use.

FastAPI is appropriate for serving the trained model because it is a modern framework for building APIs via Python, as stated in [23]. Google Cloud Run is used as a practical deployment for the API so that the backend can run online, without the need of the user hosting the API with their own computer.

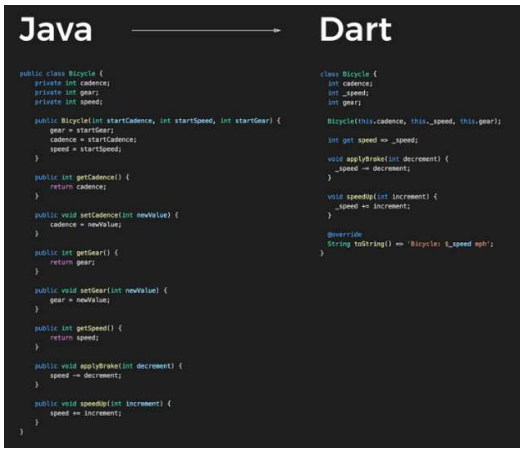


Figure 2.2 Comparison between Java and Dart

2.1.5 Algorithm

The NLP sentiment text analysis uses a machine-learning approach based on text pre-processing, TF-IDF feature extraction and Logistic Regression classification.

According to [3], text pre-processing is important because noise, casing and informal spelling can alter the model quality. [25] and [26] describe that sentiment and emotion detection from text is a mature field yet still evolving with varied datasets and modelling choices. For this project, TF-IDF and Logistic Regression were selected as they are lightweight and practical for a custom dataset, supported by [27] and [28]. Besides, as stated in [26], TF-IDF and Logistic Regression are easier to train repeatedly on datasets and easier to serve through API.

Text pre-processing is also a critical part for the sentiment text analysis as realistic conversational text often contains inconsistent casing, abbreviations of words and spelling variation. [29] and [30] researches claims that social-text pre-processing must consider slang and informal wording rather than classifying them as useless noise. Therefore, it is an important process that needs to balance cleaning and preservation of emotion. For example, repeated letters, uppercase words and slang expression can indicate stronger intensity of emotion, if the pre-processing is too aggressive these emotion cues will be removed and produce inaccurate outcomes.

In this project, this algorithm is applied not only because of its classification accuracy, but also it can be deployed and retrained easily without excessive computational cost. By being light weighted, it can also help in being responsive and easier to host stably.

2.1.6 Development Environments

From the development workflow perspective, Visual Studio Code is the main code editor and builder. This is because it provides many useful features, such as available of many add-ons, debugging tools, terminal integration and a user customizable workspace, as stated in [31] and [32]. Therefore, it is useful in this project since this project includes many different sectors like Cloud Functions and multilanguage files.

Moreover, Android Studio is also used in this project. As mentioned in [33], Android Studio is the official Integrated Development Environment (IDE) for Android app development based on the powerful code editor and developer tools from IntelliJ IDEA. In this project, it is mainly used to download the required Android environment files, which allows Flutter to run properly. It is also used to install emulator devices so that it can help in testing the code more efficiently instead of needing to deploy on the phone every time.

Codex is largely used in this project. [34] stated that it is a cloud-based software engineering agent that can write features, answer code related inquiries, fix bigs and propose code changes in isolated environments. In short, it can assist in exploring unfamiliar project files, drafting implementation logic, explain code structure and suggest debugging steps.

Last but not least, ChatGPT helps giving suggestion and clarifying confusions. As stated by [35], it interacts in a conversational way which can reply follow-up questions, explain problems and provide step-by-step answers. In this project, it is useful in planning report structure, explaining complicated concepts and help the developer reason through choices.

However, the output from either ChatGPT or Codex should not be accepted blindly. They remain support tools rather than completely replacing the need of developers. Therefore, to be an ethical developer, all generated suggestions, code, and report explanation must be reviewed and checked manually to ensure its correctness, relevance and compatibility with the actual project.

2.1.7 Summary of the Technological Review

To summarize, all technologies mentioned are suitable for developing the family-oriented app. Flutter and Dart support cohesive mobile UI development. Google's Firebase supports real-time data synchronization. Python, FastAPI and Cloud Run support the lightweight model and mobile backend API integration. TF-IDF and

Logistic Regression provide an interpretable and deployable baseline for the sentiment text analysis. Visual Studio Code and Android Studio are used parallelly for developing codes due to their efficiency and ease of use. Ai-assisted tools such as Codex and ChatGPT can accelerate planning and debugging of the development.

To summarize, the selected technologies are suitable because they match the objectives and time constraints of this project.

2.2 Review of the Existing Systems/ Applications

2.2.1 Journal Article

After conducting an extensive review of existing journal articles researched, only a limited amount of them were found to be aligned with the direction of the project. Most scholar publications regarding family applications tend to focus heavily on healthcare, medical monitoring, or support for caregivers, rather on family communication or coordination tools. As a result, despite reviewing over ten academic sources, only a select few journal articles were deemed relevant to the proposed development of family mobile applications. These chosen articles provide meaningful insights into design planning, user requirements, and technological approaches that can inform and support the objectives of this project. Table 2.1 shows the list of previous scholar publications that are relevant to this project.

Table 2.1 Lists of Previous Scholar Publications

No	Paper Title	Author (Year)
1	On Developing an Integrated Family Mobile Application [36]	Subhieh El-Salhi et al. (2019)
2	The Methodology of Developing a Mobile Application Design for Creating a Genealogical Tree [37]	Ye. M. Hrabovskyi, Yu. D. Brusiltseva. (2022)
3	A Co-family Managerial Photo Share Mobile App UI Development [38]	Yu-Mee Chung, Jong-Hoon Choeal. (2014)

2.2.1.1 Overview of Family Mobile App in Academic Research

[36], [37] and [38] shows previous work developed by former scholars, they all focus in different functionality, but in the same mobile app platform. While [36] mainly acts as a reminder app for notifying user, [37] focuses on maintaining or improving relationships between family relatives' relationships. Although [38] is entirely in Korean, with the help of translation, information and ideas can be obtained. The reason [38] is listed in the literature review although it is a foreign language is due to its idea and directions the developers shared is valuable and enlightens.

2.2.1.2 Key Features and Trends in Existing Studies

In the application [36] proposed, it also have an interesting function of educating kids with AR. But since there are already many educational mobile applications in the market and many reliable educational channels on YouTube, this function was deemed unnecessary. On the other hand, its location sharing function, task/chores list function was a good direction and inspiration to work with. At the end, it also conducted a survey with 24 families as participants, their opinions on the education part of the application aligned with our previous thought, only 47 out of 120 participants (around 39%) agreed with the function. Thus, this project will not take consider in educational areas to prevent overflooding the app with excessive features.

[37] suggested a more simple and one-dimensional approach. It has a family tree function, where you can observe all the family members and relatives. Apart from that, the user could also leave stories to pass on and share in a space where other members can react and reply. Although this is an interesting function to inheriting the family culture, this project will also not focus in this field to prevent hectic functions in the app. The clean aesthetic of the UI is also worth learning for this project future development. Overall, it acts as an easy, intuitive medium to inherit ancestor culture and maintaining family bonds. Figure 2.2 shows the UI of the app.

Figure 2.3 User Interface of app developed by [37]

Lastly, [38] presented a centralized photo sharing album, which addresses the challenge of fragmented photo sharing and difficult in locating the pictures in personal drives. It operates as a family drive, where every family member can upload pictures to their respective album, which is shown in Figure 2.3. In short, this app provides chat features

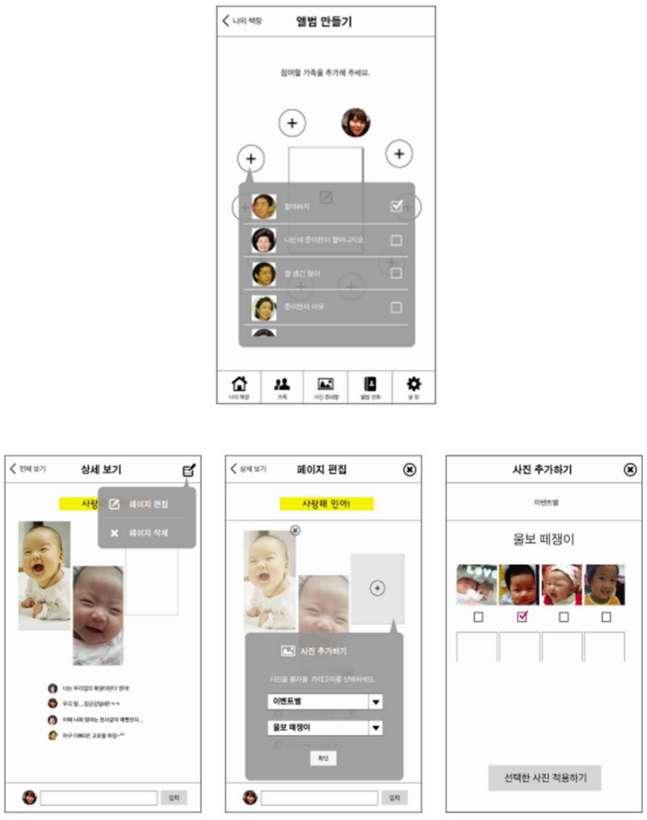


Figure 2.4 User Interface of the app developed by [38]

2.2.1.3 Limitations in Prior Research

Since they are only scholar developments, the applications are discontinued and unable to be found online, therefore the information are purely from the journal article. Without obtaining the user experience for those app, not much details outside of the reports can be acquired.

Besides, ss mentioned before, all these applications put emphasis on different directions. Therefore, they turned out to be one-dimensional, single-functioned tools, which worsen the problem of crowding apps. In short, this project will keep these limitations in mind while taking inspiration from them.

2.2.1.4 Proposed Improvements Based on Journals

The solution is to combine and integrate those features into a single application, taking the calendar feature from [36], fusing the family tree function from [37] and incorporating the album sharing feature from [38]. Thus, the user can unify all their needs in one app instead of having to transfer to other apps.

2.2.2 Website Resources

Due to the limited availability of journal articles that align with the goals of this project, the literature review also places significant emphasis on web-based resources. Blogs and development case studies published by experienced app development companies were selected. By referring to these sharing by experience industry owners, valuable industry insights can be acquired and learnt. This is because they reflect real-world challenges encountered during the development of family-focused mobile applications, they offer practical perspectives that are often absent in academic literatures. Furthermore, their method to make detailed analysis of market demands and user expectations also contributes meaningfully to understanding current trends and feature requirements. To conclude, these industry resources serve as a complement to the scholar publications in guiding the practicality of the application. Table 2.2 show the list of website resources by relevant industries.

Table 2.2 Lists of Website Resources by Relevant Industries

No	Companies	Product
1	Yellow	Famlicious [39]
2	Mbicycle	Family Organizer App [40]

2.2.2.1 Industry Insights and Development Approaches in Family Apps

[39] is a cross-platform family chat app developed by Yellow, while [40] is an interactive family organizer application developed by Mbicycle. They are either discontinued or client-based project that are not available to the public, but their sharing of development process is valuable for this project.

[39] shared about its app feature and development procedure to build an MVP (minimum viable product). It also provides insight to the tools used and challenges faced during the development. [40] also shared similar info while developing its app. To summarize, these industry insights showed real world complications that are not seen from the scholar publishes, so this project can have a better view on what is the market needs and what should be avoided.

2.2.2.2 Feature Integration

Similar to [37], [39] generates a family tree, but with an additional feature of chat rooms. By having an exclusive chatting application for family, it can prevent messages from being lost in spam and work-related messages. [40] is similar to [36], with the functionality of time tracking and task listing. It also has the features to chat and sharing photos. Figure 2.4 shows the user interface of [40].

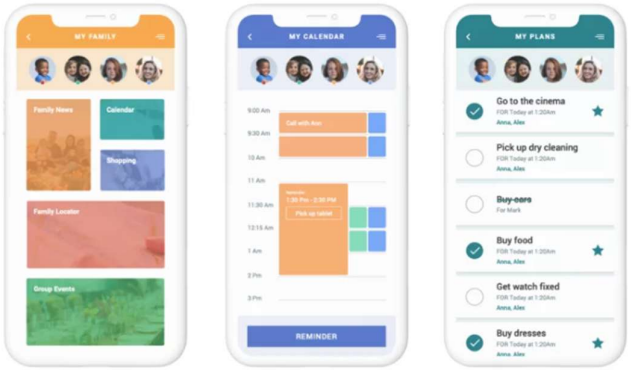


Figure 2.5 User Interface of family organizer application by Mbicycle

2.2.2.3 Challenges in Real-World Implementation

[39] was once available on IOS App Store and Google Play Store, but it seems that they have left the family mobile application market although the application is relatively new. This proved that the idea and product by Yellow is proved not suitable to survive in this competition. On the other hand, [40] was a private project requested by their clients with no intention of releasing publicly to test. They both have a stark difference in directions and showed the obstacles in real-world implementation.

Although the whereabouts and conclusion of these two apps may not seems promising, they showed serious and real-world problem during their development. For example, [39] shared that they have trouble implementing complicated features such as the family tree and chat from scratch, so they ended up using existing platforms and APIs. [40] also shared that they were tasked to develop it for cross platform compatibility (IOS and Android) completely from scratch, which is a time and cost consuming approach.

2.2.2.4 Proposed Solutions Based on Industry Works

Their success and failure proved that to stand a ground in this saturated environment, the idea of a family mobile app must be innovative and unique. Simple app with too few features will not be interested by the mass. Therefore, the cater to user experience and market needs are crucial and have to be in consideration for all time.

Few viable solutions are to make sure which features are rare and have demand, and plan them beforehand. This can prevent the app being stale and boring, while planning in the beginning can help lay down the expectation and ensure the smoothness of future development. Besides. the app can also be frequently maintained and supported by a team, so that it does not give the feeling of abandonment to its user. For instances, it can have seasonal updates to allow user to have customizable interfaces such as changing themes, so the app will feel fresh without the needs to push more unique functions.

2.2.3 User Experience Analysis

As the saying goes, “it is better to try something yourself than merely hear about it”. By personally exploring and experiencing these market-available family-based mobile applications, a deeper comprehension of their features and use experience can be achieved. These hands-on evaluations allow for the identification of limitations from each application, which in turn helps avoiding repeating similar mistakes during the development of the project. Furthermore, it can serve as a source of inspiration, offering good ideas and guiding the planning process. Consequently, ensuring a more thought out, efficient, and user-centred approach to development. Table 2.3 presents the list of market-available family applications that are downloadable in Google’s Play Store.

Table 2.3 Lists of Market-Available Family Applications

No	Company	Application	Features
1	Google	Family Link (2017)	restrict content, approve or disapprove apps, set screen times, location monitoring
2	Microsoft	Family Safety (2020)	calender, family email, activity report, content filter
3	Cozi Inc	Cozi Family Organizer (2010)	event list, grocery list, family calender
4	Family360	Family360 – GPS Live Locator (2021)	real-time location sharing, safety features, traffic updates
5	Tinywork Apps	FamCal (2016)	calender, task list, grocery list, memos
6	Family & Co	Family Wall: Family Organizer (2011)	calender, task list, grocery list, messaging, gallery, location sharing

2.2.3.1 Evaluation of Existing Family Mobile Apps

Apart from Family Link and Family Safety being the products constructed by famous tech companies. The other applications are produced by smaller indie companies. They are all still available and downloadable through Google Play Store. Some of them have similar functions, while some focus solely on one purpose.

Both Google's and Microsoft's app mainly focus on parental control features. With their wide support and huge influence in the tech fields, they have higher downloads counts compare to the others. Therefore, their survivability and probability to update in the future are higher due to their huge user base. The other apps have interesting functions that gives huge inspiration to this project. For instances, they have the calendar feature which is convenient for family planning and some daily life optimisation features that improve day-to-day life efficiency. Family360 app takes an interesting approach, it focuses on family member safety by its real-time location sharing feature.

2.2.3.2 User Feedback and Review

As stated before, Family Link and Family Safety have more resources to develop and maintain, they both also have website versions. Thus, they tend to be more trustworthy and accepted by the public. With their wide influence around all parts of our life, they give users more confidences in their product, so they can focus more on security features where they normally required phone controlling authorisation.

The other apps that focus on everyday efficiency are useful in assisting family planning. Normal social media apps only provide basic functions for planning, but these family-oriented app provides family-focused feature such as event planning, shopping list, task scheduling and more. These functions help improve the communication between family members and make the productivity of daily lives more meaningful.

CHAPTER 3

System Methodology/ Approach

This chapter shows the system methodology used to plan, build, connect and refine the app. It focuses on the overall system design, system architecture, use case description and activity diagram.

Its methodology is based on connecting multiple features, including authentication, profile setup, group chat, emotion classifier, shared gallery, events creation, weekly calendar, AI family advice, live location sharing and notification services. Therefore, it needs a layered architecture where the Flutter app, Firebase services, external backend API and Google Cloud functions to combine as one fully coordinate system.

As a development-based project, this chapter will explain the system from a high-level methodological perspective, such as demonstrates how the main modules are planned and how the services were connected.

3.1 System Requirements

This system does not rely on a single local Android mobile application only. Instead, it integrates a cloud architecture where the mobile app client collects and sends data through Firebase system and external backend services. By this way, the app can have a common cloud-based data storage, which is the essential component for communication app.

The backend API consists of the four main features: sentiment text analysis, image categorization, AI advice and the geocoding. The algorithmic details are briefly noted here and will be explained further in the following chapters. In this project, the system is designed as a layered application. At the upper layer, the user interacts with the Android app. At the middle later, the Flutter client manages UI rendering. At the lower layer, Firebase and external backend services help operate all the actual functions.

No complex mathematical equation is required to develop the entire app as it is mainly a software development system rather than a mathematical calculation model. However, a simple equation is used to represents how recent emotion scores are aggregated along the time. The detail explanation will be at Chapter 4.

Figure 3.1 shows the overall system design diagram of the app.

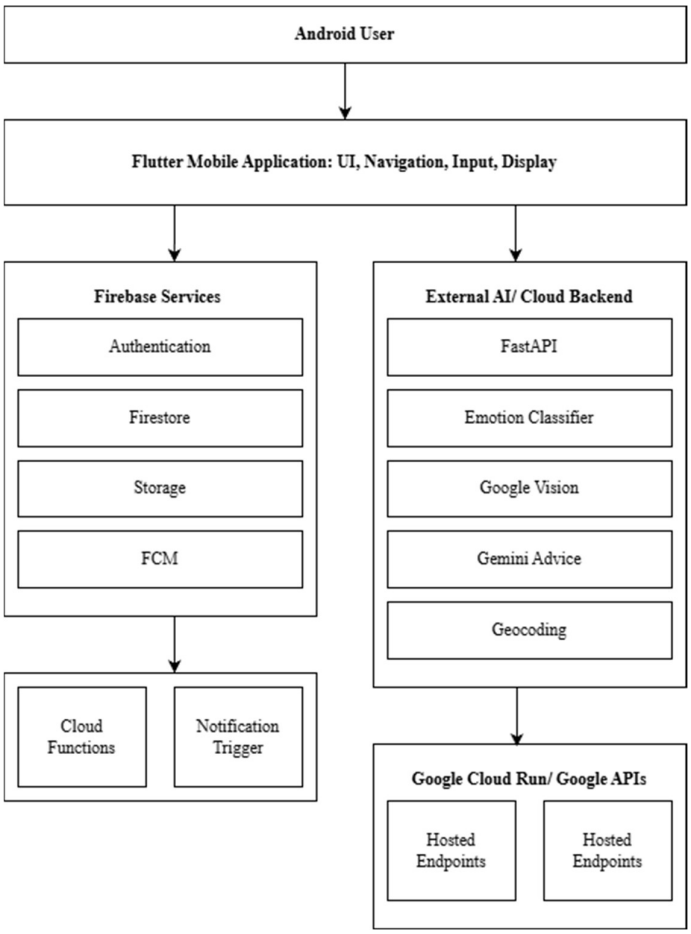


Figure 3.1 Overall system design diagram of the app

Figure 3.1 shows that the how the system operates, from how the user interact with the UI elements of the app, to the app communicating with several backend services. All of the data from the app such as messaging data, event data and location metadata are stored inside the firebase services, while other more complicated features such as image recognition and emotion classifier are unified in a cloud backend service.

3.1.1 Firebase Services Architecture

The firebase services can be dissected into four main categories. Firstly, Firebase Authentication manages account formation, session and identity. When an account is created in the app, the developer can view the account via the Firebase Console. Then, Firestore documented and stored all data from the app. Moreover, Storage saves images data from the app. Lastly, notifications are push out via Firebase FCM features by Firebase Messaging.

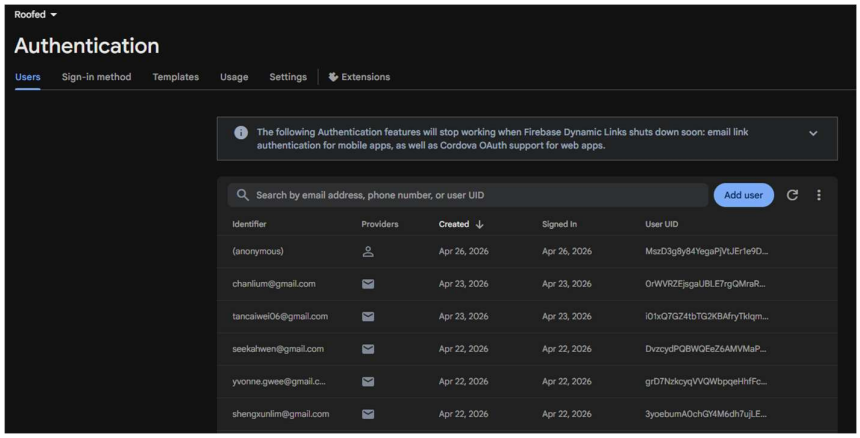


Figure 3.2 Firebase Authentication

As shown in figure 3.2, this is the Authentication interface in the Firebase Console. Different types of accounts can be seen, such as anonymous logins or Email logins. The developer can either freeze or delete the account from the console, but not see the details for the account such as its password.

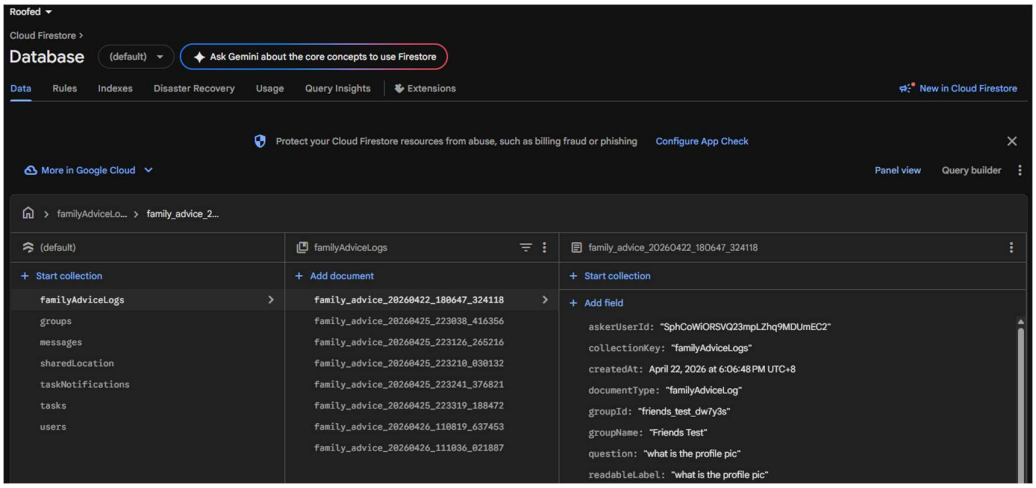


Figure 3.3 Firebase Firestore

Figure 3.3 shows the Firestore interface in the Firebase Console. Inside the database, all data are categorized and stored in different collections for better clarity and readability. In each collection, it stores the data in specific format as written in the code. For example, the family advice data have many strings values such as the question and group names. Information about the collection structure will be explained in Chapter 4.

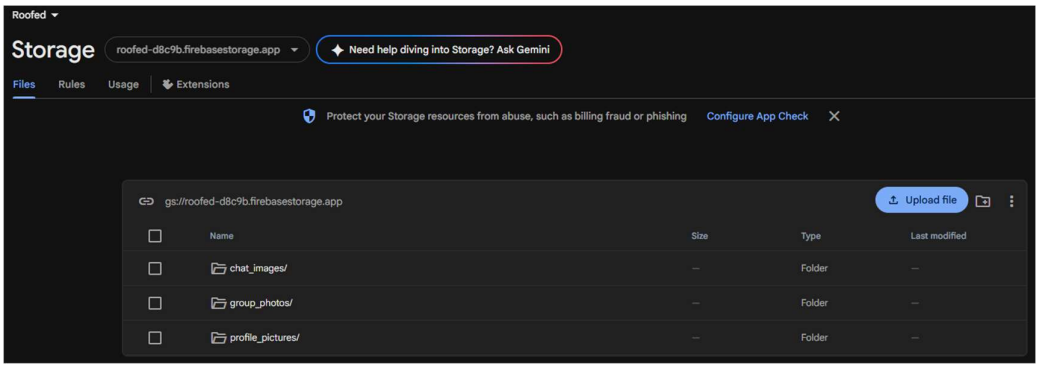


Figure 3.4 Firebase Storage

Figure 3.4 illustrates the Storage interface in the Firebase Console. Since the Firestore can only store limited data such as string values and integer values, image metadata have to be stored in a different way, which in this case, in the Storage. The developer can view all pictures uploaded by user, including profile pictures, group photos and chat images. It is worth noting that most features in firebase console are free to use within a limit, but Storage require subscription to Firebase Blaze Plan.

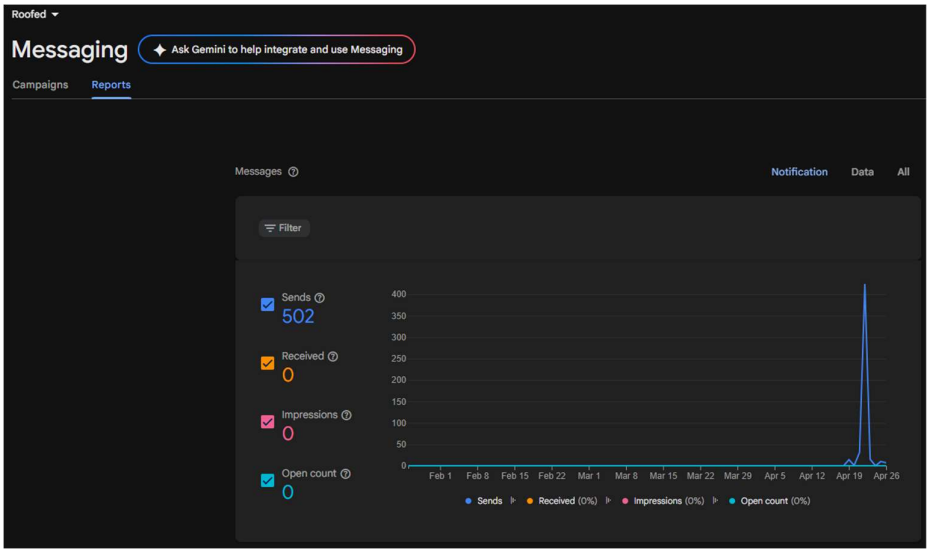


Figure 3.5 Firebase Messaging

Notification feature is a rather easy procedure, when a user does a task like creating new events, send messages or change group info, the cloud function detects and retrieve the recipient's FCM token, then Messaging will push out notification to other users in the group

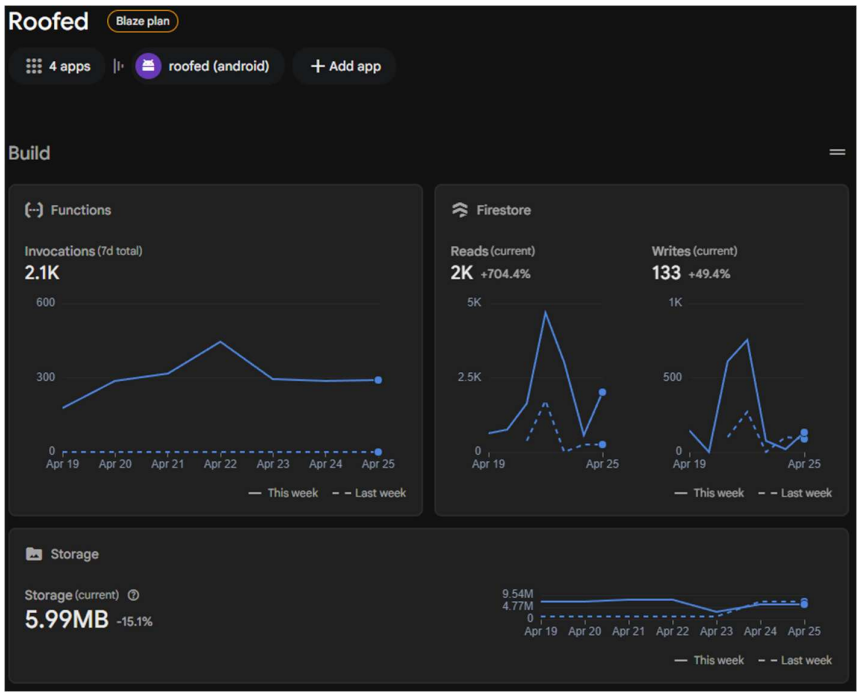


Figure 3.6 Firebase Console

To summarize the firebase services, Firebase Console is a convenient way to save all data from the app, which ensures the app to work properly. With the stable database, the app can receive data correctly and transmit data to it, which is crucial in features such as chat function, where previous data and real-time data are important. Figure 3.6 shows the analytic dashboard provided in the console. The developer can view the data flow rate across the project, which helps in maintaining the app and assure abnormal error can be discover swiftly.

3.1.2 Backend Services

The cloud backend API are run via FastAPI and Google Cloud. The API includes three main functions: emotion classifier using text messages, image categorization and AI adviser feature. These features are written in Python, then deployed online via FastAPI and Google Cloud.

The Geocoding function is not combined together with the Python FastAPI or the firebase services. Instead, it uses one of the Google Cloud API, where it generates an API key to be inserted into the app. If connected successfully, the map will be generated in the app, and simple functions such as searching or pinging locations can be done. It is also free to use within a limit. Figure 3.7 shows the location related functions usage of the app.

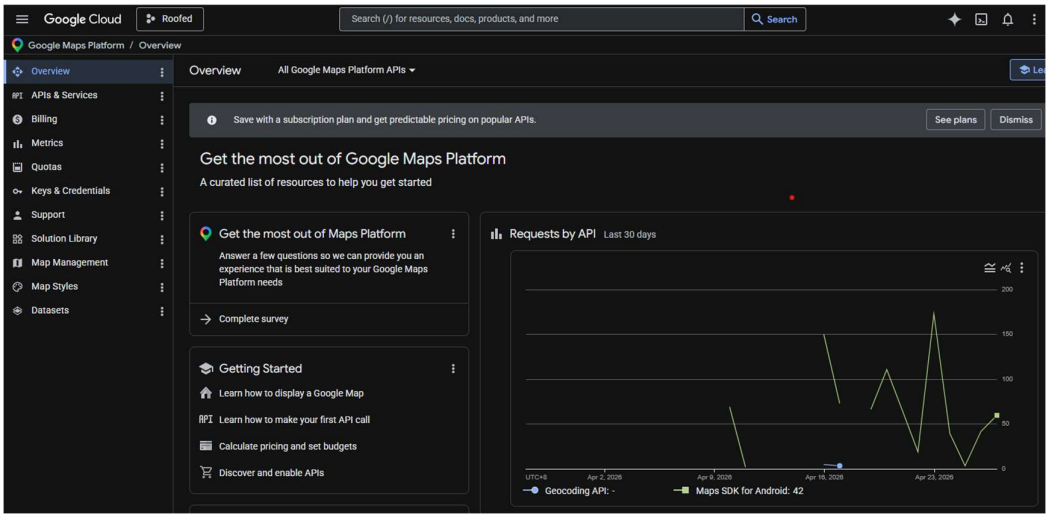


Figure 3.7 Google Cloud Map API

3.1.3 Summary for System Design

The system design can be concluded into the Table 3.1, it is categorized into several layers.

Table 3.1 Layers of System

System Layer	Main Responsibility	Examples in Roofed
Presentation layer	Displays pages, collects input, controls navigation, and shows feedback to the user.	Splash page, login page, profile setup, app shell, home, chat, gallery, calendar, event, location, emotion, AI adviser.
Application logic layer	Controls page state, validates form input, prepares data, and calls service classes.	AuthService, ProfileService, EmotionDetectionService, FamilyAdviceService, GalleryImageClassifierService, LocationSharingService.
Firebase cloud layer	Stores shared data, account state, files, tokens, and notification trigger records.	Firebase Auth, Firestore, Storage, FCM, Cloud Functions.
AI and external service layer	Performs emotion prediction, image recognition, family advice generation, and geocoding.	FastAPI backend, scikit-learn model, Google Vision, Vertex AI Gemini, Google Geocoding.
Device service layer	Provides phone-level capabilities needed for mobile operation.	Location permission, gallery selection, local notification display, map rendering.

3.2 System Architecture Diagram

Figure 3.8 demonstrates how the main components of the app are connected.

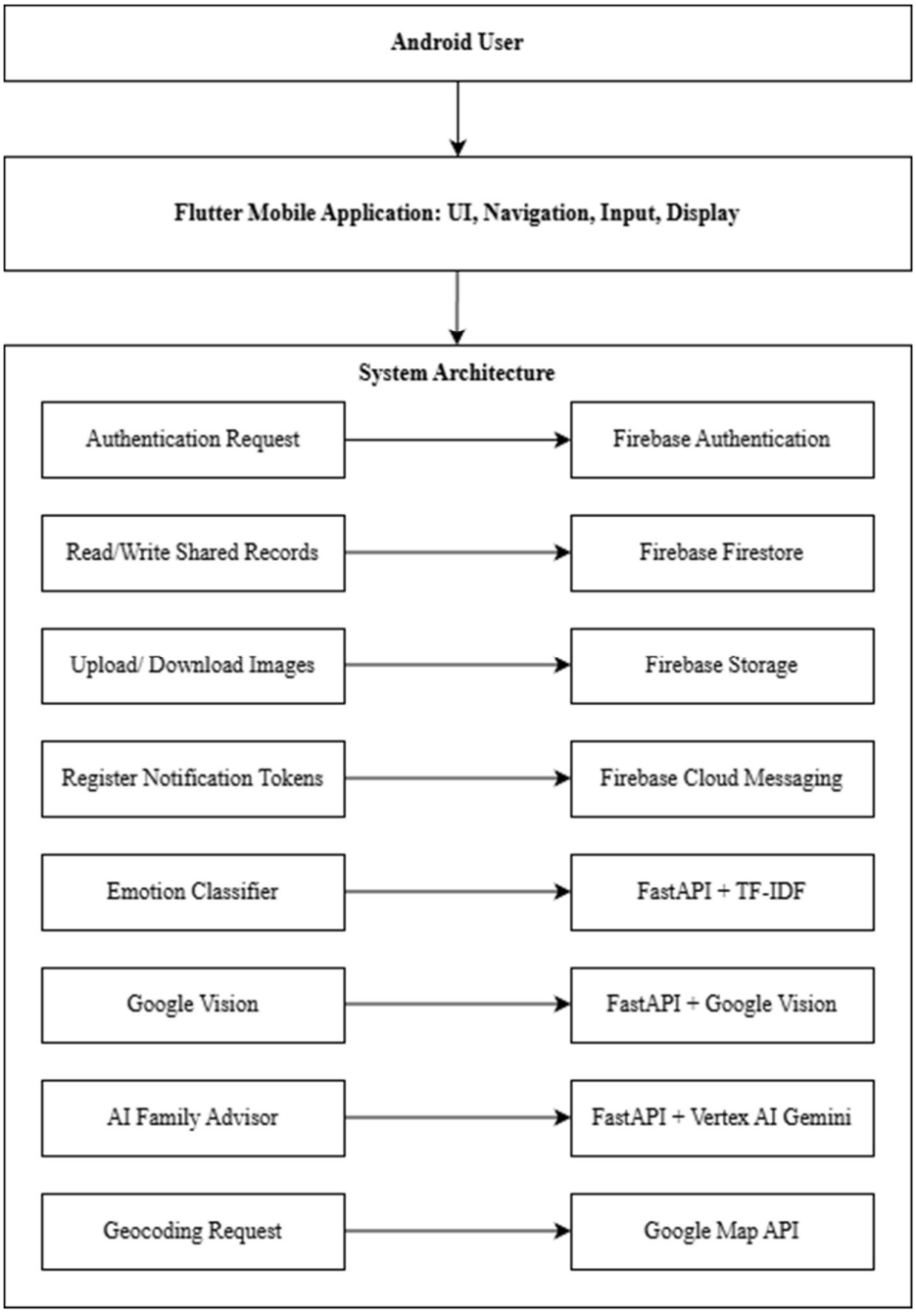


Figure 3.8 System Architecture Diagram

Table 3.2 presents the summary of architecture component. As mentioned before, every component the app is connected to services to operates properly.

Table 3.2 Summary of Architecture Component

Component	Input	Processing	Output
Flutter client	User actions, form fields, selected images, device location.	Validates input, updates UI state, calls Firebase or HTTP services.	Updated screens, feedback messages, stored records, uploaded files.
Firebase Auth	Email, password, username-resolved email, anonymous session request.	Creates or restores authenticated identity.	User ID and session state.
Cloud Firestore	Structured application records.	Stores, streams, queries, and updates shared data.	Realtime data snapshots for pages and services.
Firebase Storage	Profile photos, group photos, chat images.	Stores media and generates download URLs.	Image URL and storage path.
Cloud Functions	Created notification records and scheduled reminder checks.	Filters recipients, checks muted groups, sends push notifications.	FCM push notification and token cleanup.
FastAPI backend	Text, image data/URL, advice request context.	Runs model inference, image recognition, and Gemini advice workflow.	JSON response for the mobile app.

Figure 3.9 illustrates how the emotion classifier works.

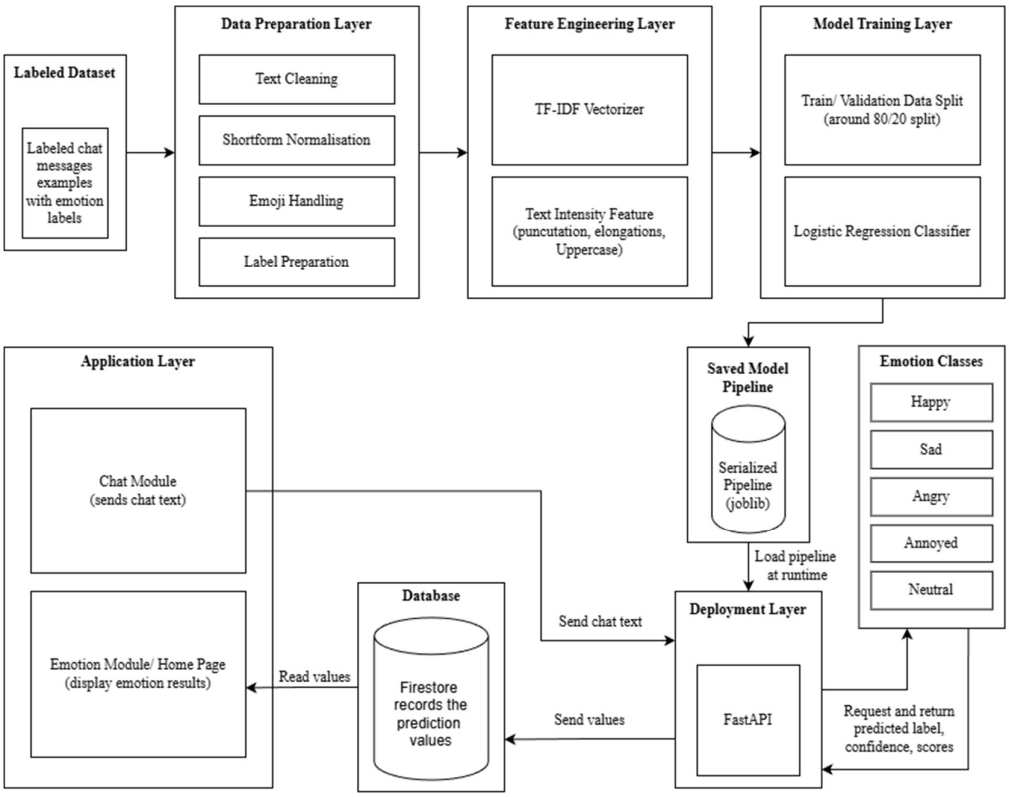


Figure 3.9 Emotion Classifier Architecture Diagram

This architecture diagram shows the training and runtime components of the emotion classifier. During the training stage, a labelled dataset is passed through the data preparation layer for text pre-processing. The text pre-processing layer includes text cleaning, short form normalisation, emoji handling and label preparation. By doing so, it can ensure the data passing through the model are recognizable and predictable. For instances, emoji will be translate into actual words due to the model not capable of reading emoji directly, the label preparation is to make sure the format is correct for the model. The processed text is then transformed into TF-IDF and text intensity features before being used to train the Logistic Regression classifier. After training, the pipeline will be deployed through a FastAPI service.

During runtime, chat message will be submitted to the API for emotion prediction. After prediction, the values are returned and store to the database. Finally, the Emotion Page and Home Page retrieve the stored values from it to display the emotion summaries.

3.3 Use Case Diagram and Description

Figure 3.10 shows the user interaction with the app and how the services under support it, while the Table 3.3 shows the use case description.

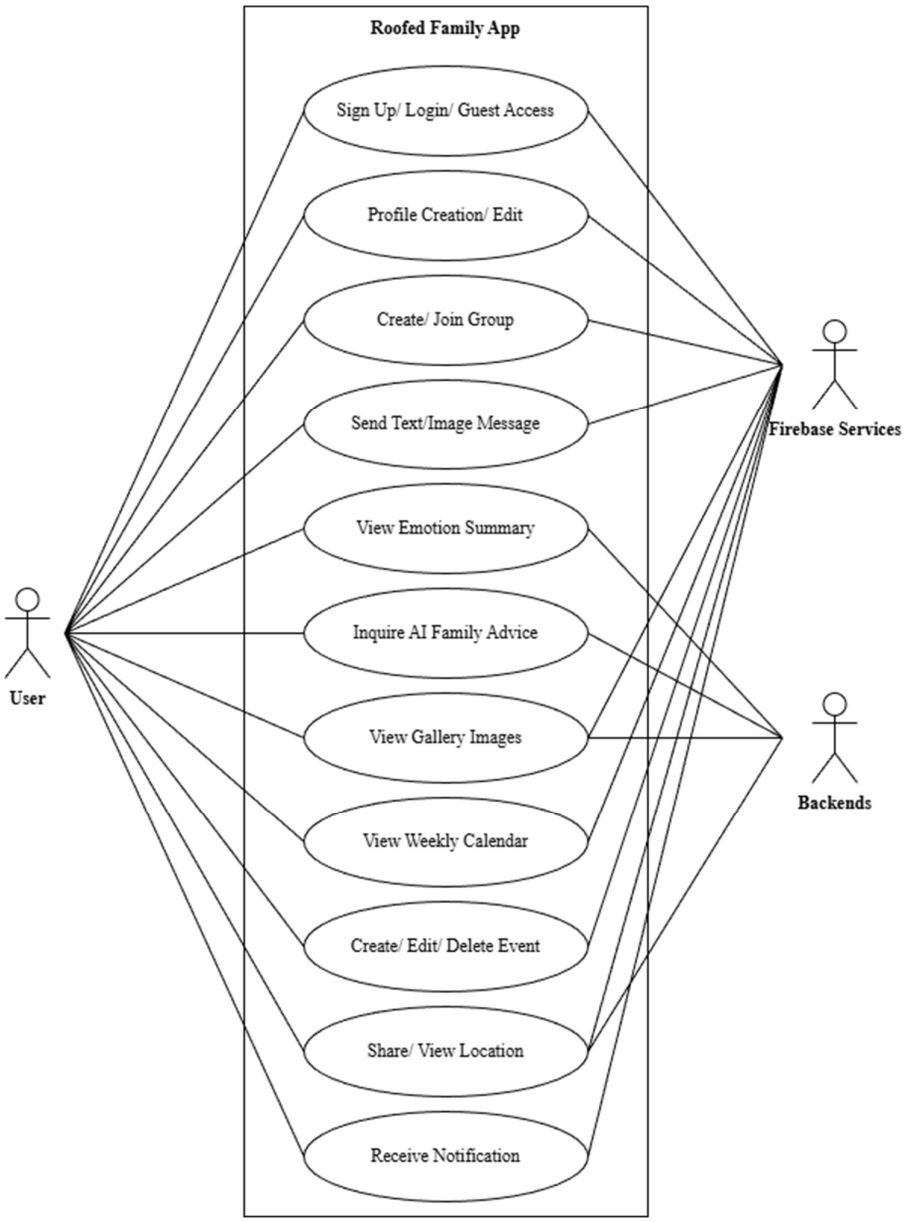


Figure 3.10 Use Case Diagram

Table 3.3 Use Case Description

Actor	Use Case	Description	Related System Component
User	Sign up, log in, or guest access	The user enters the system through account authentication or anonymous access before completing a profile.	Firebase Auth, AuthService, SplashPage, AuthEntryPage.
User	Complete and edit profile	The user provides username, bio, image, interests, food preferences, and lifestyle details for personalization and advice context.	ProfileSetupPage, ProfileService, Firestore, Firebase Storage.
User	Create or join group	The user creates a private group or joins by code so that family members can share a common space.	ChatPage, groups collection, messages collection.
User	Send text message	The user sends a message to the group. The system stores the message, calls emotion detection, and creates notification records.	ChatRoomPage, EmotionDetectionService, messages collection, taskNotifications.
User	Upload image message	The user uploads images through chat. The system stores the image and later displays it in the gallery.	OptimizedImageService, Firebase Storage, GalleryImageClassifierService, GalleryPage.
User	View home and emotion summary	The user views weighted recent emotion summaries for themselves or selected family members.	HomePage, EmotionPage, messages collection.
User	Ask AI family advice	The user asks a question with safe context and receives a short family-oriented response.	FamilyAdvicePage, FamilyAdviceService, familyAdviceLogs.
User	Create and manage event	The user creates standard events, going-out plans, shopping lists, polls, or other items.	TaskPage, AddTaskPage, tasks collection.
User	View calendar	The user views visible group tasks in a weekly calendar structure.	CalendarPage, PlannerTasksStreamBuilder, tasks collection.

Actor	Use Case	Description	Related System Component
User	Share live location	The user enables or disables location sharing and views group member markers on the map.	LocationPage, LocationSharingService, sharedLocation collection.
Cloud Functions	Deliver notifications and reminders	Backend automation reads notification records and sends FCM messages to target recipients.	functions/index.js, taskNotifications, FCM.
FastAPI backend	Predict emotion, classify image, generate advice	External backend returns structured JSON for AI-related functions.	/predict, /classify-image, /family-advice.

3.4 Activity Diagram

In this part, activities diagrams will be shown to explain the sequence of major system actions. Instead of describing every button or page flow, it will emphasis on system-level flow, with more detailed page flows will be explained in Chapter 4.

3.4.1 Application Entry and Authentication Routing

Figure 3.11 illustrates the activity flow of how a user enters the app and the system authentication procedure.

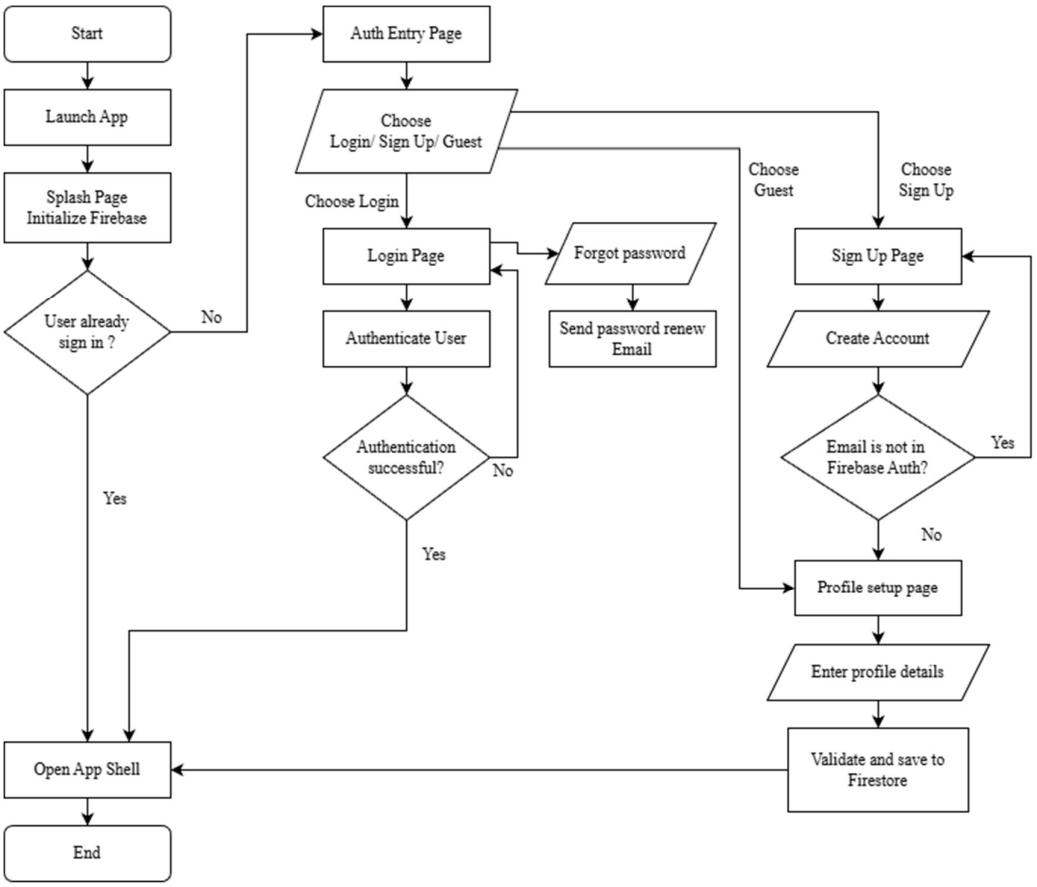


Figure 3.11 Activity Diagram for Application Entry and Authentication Routing

The application starts with a 2 second splash screen while initializing the firebase services, so that the app will load relevant info while the user is waiting for the splash screen to end, ensuring the app will not lag afterwards due to incomplete synchronisation with the firebase. Then, the app will go through authentication logic showed in the figure. Finally, the app will end its authentication process and move forward to the app shell itself.

3.4.2 Chat and Image Processing

Figure 3.12 shows the activity flow for chat and image processing.

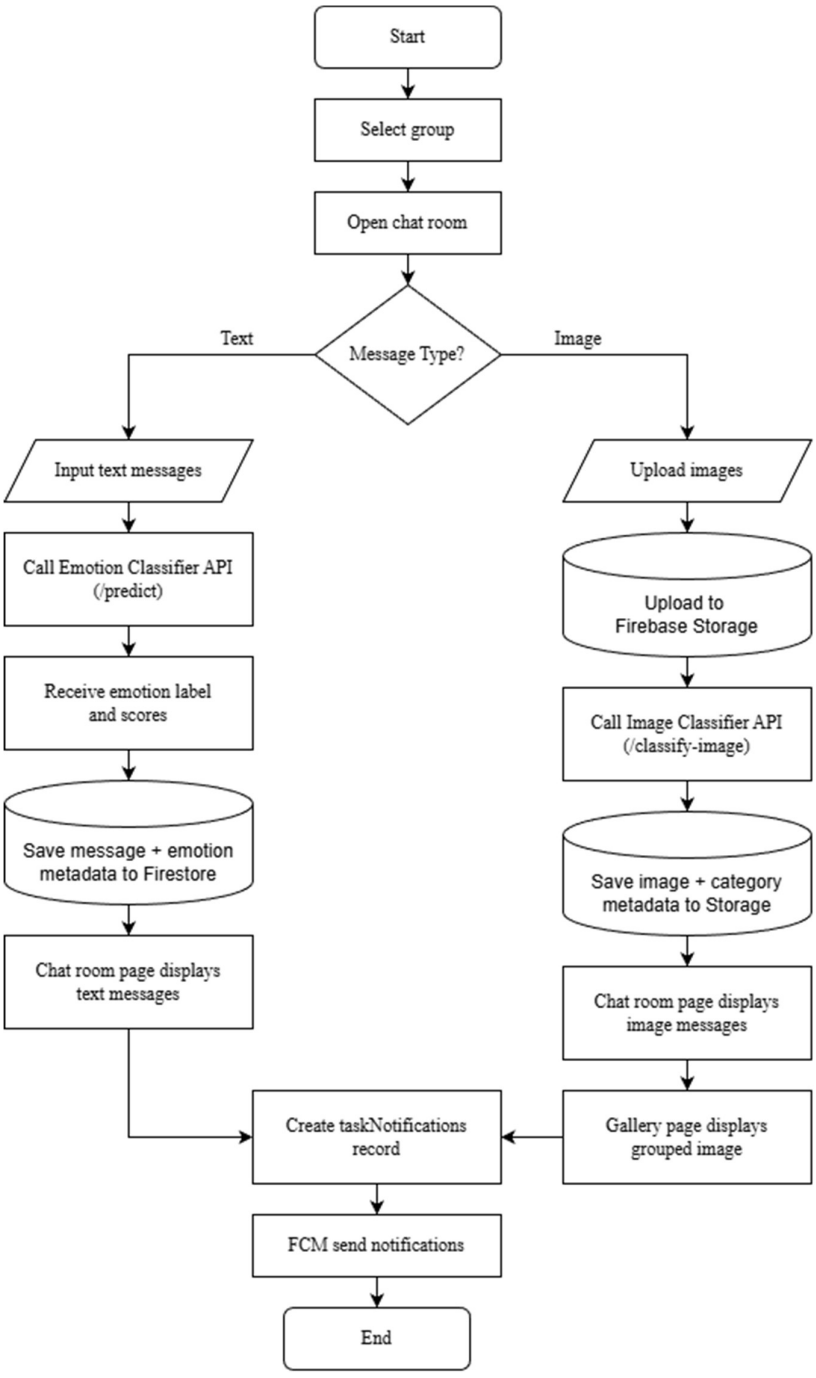


Figure 3.12 Activity Diagram for Chat and Image Processing

As shown above, it demonstrates the activity flow of how the messages in chatroom is processed. Firstly, text messages will be pass through the emotion classifier API, then the score acquired by it will be save back to the firebase. For the image messages, it will be uploaded to the firebase and image classifier API, then the API result will be sent to the firebase. After the process and documentation of messages, it will be displayed in the chatroom, while sending notifications to notify relevant members. Images will also be shown in gallery page, allowing user to categorize it with the result save from the API.

3.4.3 Event Features

Figure 3.13 presents the activity flow for the event features.

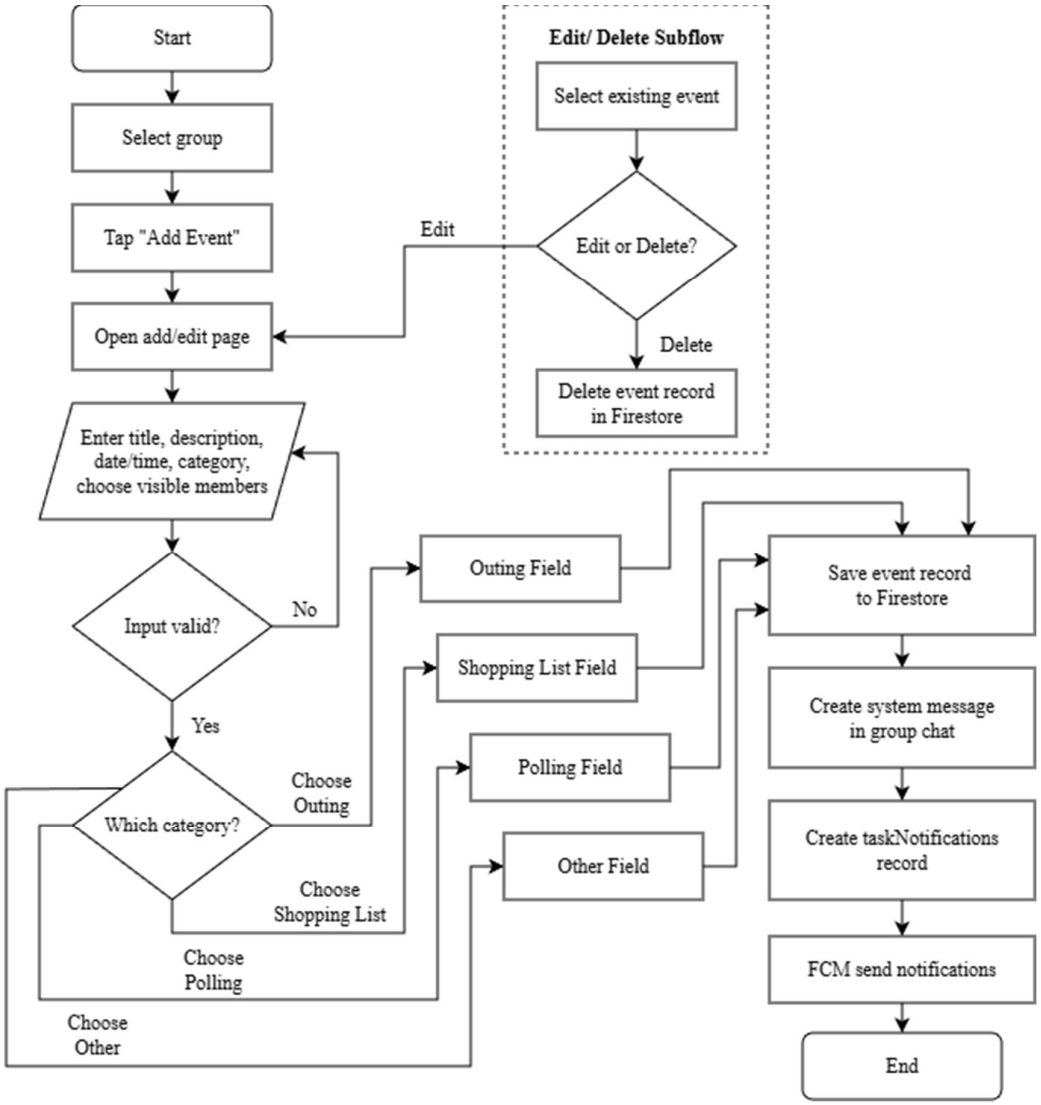


Figure 3.13 Activity Diagram for Event Features

In the event creation logic, users can choose different type of event they want, where each type have their own unique features. Similar to how text messages are saved, it will be documented in to a collection in Firestore, then a system message will be sent to group and notifications will be sent to relevant members. The edit and deletion of event works similarly to the creation logic as shown in the figure.

3.4.4 Location Feature

Figure 3.14 describes the activity flow for location feature.

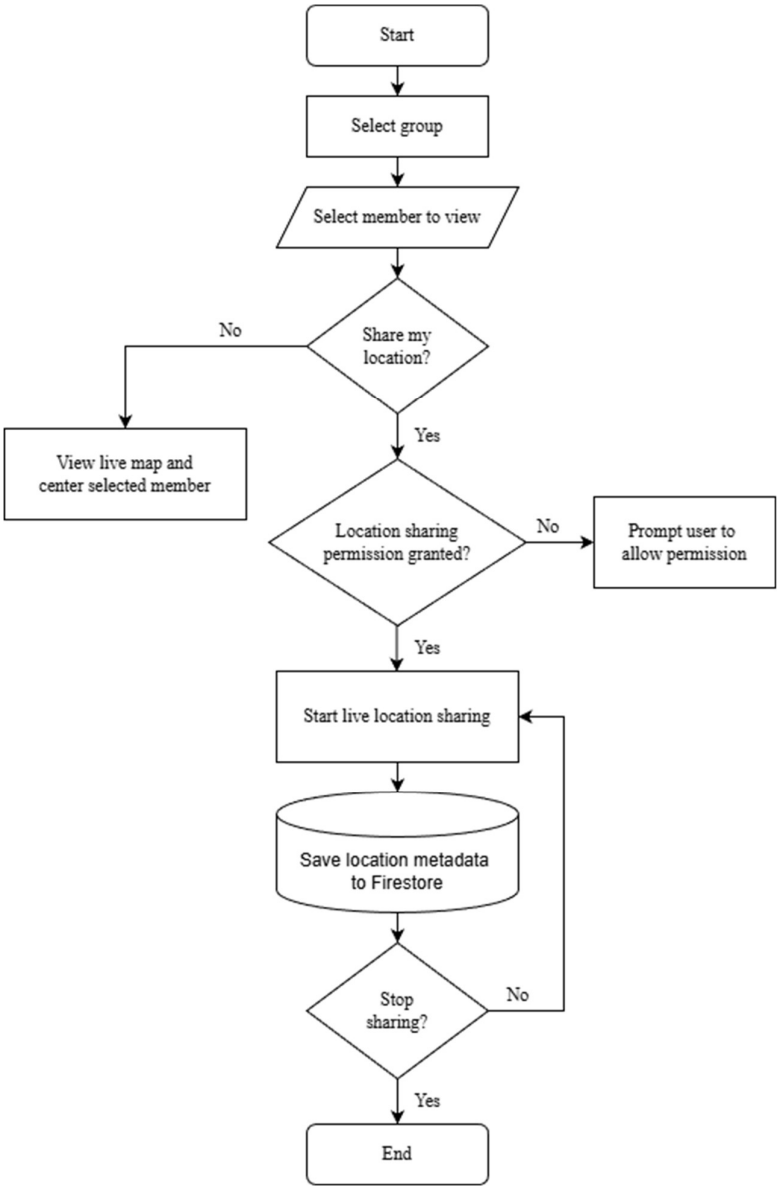


Figure 3.14 Activity Diagram for Location Feature

It begins with user permission. Then, if it is allowed and location sharing is started, the location metadata will be stored to Firestore, until the user choose to stop sharing.

3.4.5 AI adviser

Figure 3.15 shows the activity flow of the AI adviser.

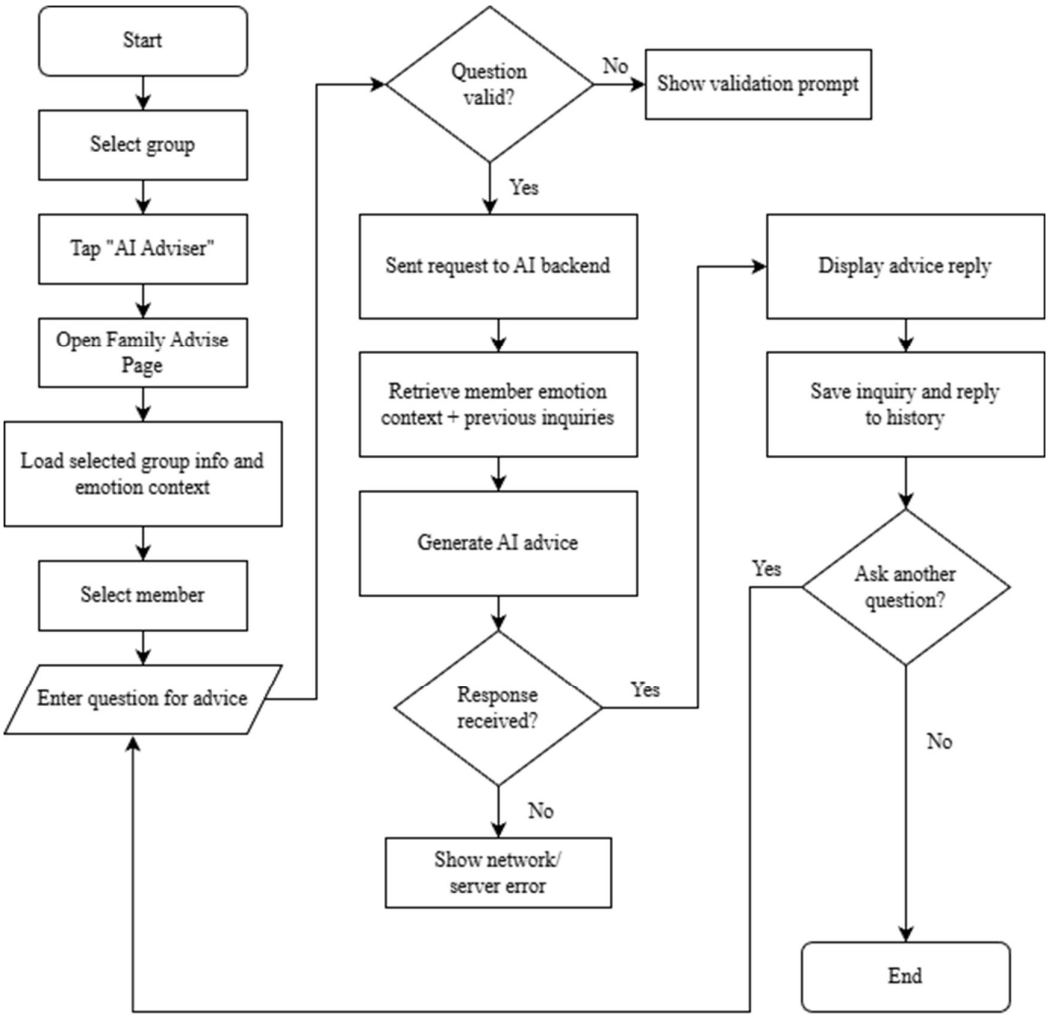


Figure 3.15 Activity Diagram for AI Adviser

Firstly, it will start by loading in the selected group and member emotion context, so that the AI can have context on how to reply based on member recent behaviours. Then, it will check if the question is valid, if yes, the inquiry will be sent to the AI backend and generate the advice to the user. In a nutshell, these activity diagrams provide the insight to how the functions' logics operate.

3.4.6 Emotion Classifier

Figure 3.16 shows the activity flow of the emotion classifier.

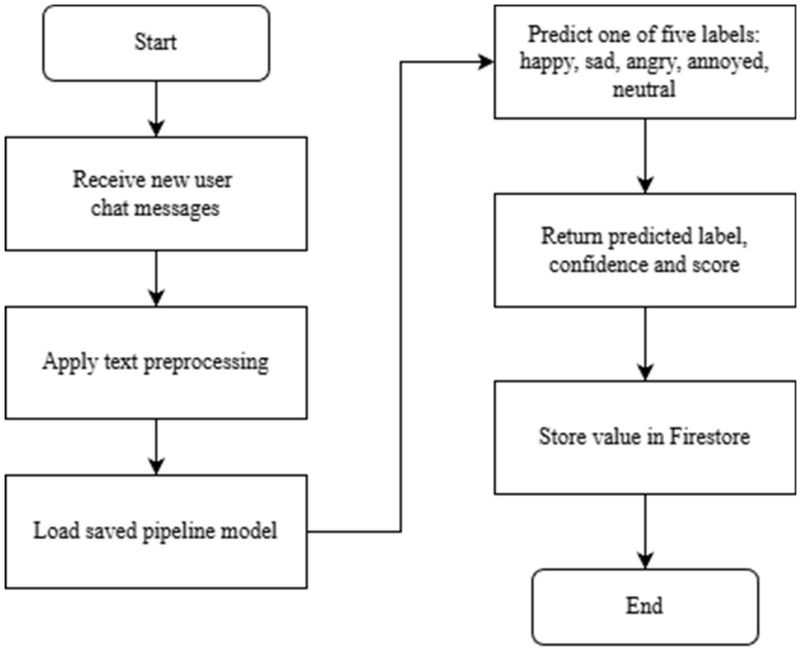


Figure 3.16 Activity Diagram for Emotion Classifier

The emotion classifier is a simple process where it acquires text messages from chatroom, then text pre-processing will be applied to the text. Then, the emotion prediction will be produced by the saved pipeline model. Lastly, all relevant values will be stored in Firestore for future reading, such as emotion page.

3.5 Project Management Methodology

Deciding a project management methodology is indispensable and plays a central role in shaping the software development life cycle (SDLC). There are many popular techniques in managing a project effectively, in this case, the agile project management methodology will be adopted.

As the word “agile” insinuated, this methodology moves quickly, which saves budget and time. Furthermore, it has the ability to respond to the change of requirements mid-developing, as stated in [41]. It also breaks down huge amount of work into small and iterative procedures that can be done swiftly, as mentioned in [42]. The adaptability of this methodology can guarantee the end product are in line with the ever-changing requirements during development, unlike the water fall approach.

In the context of this project, this methodology is suitable because the number of modules cannot be completed in one direct sequence, a lot of repetition and iteration of work flows are required. The future testing, gathering feedbacks and improve also requires back and forward development.

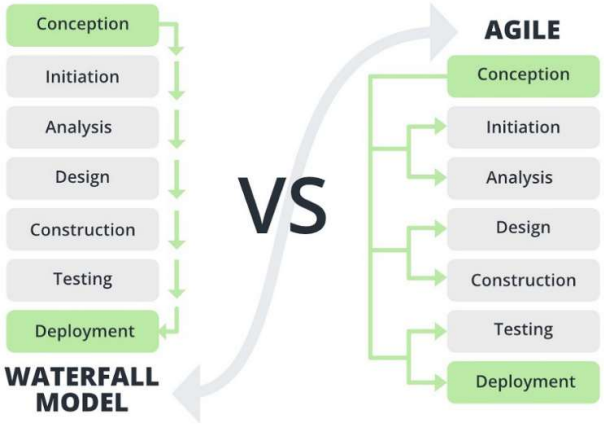


Figure 3.17 Waterfall and Agile Project Management Methodology

3.6 AI Assistance

In this era, the integration and assistance of AI is commonly seen in information technology (IT) related projects. Either they are used for generating code lines, or giving ideas, it is undeniably that they are significant in saving human power, cost and time.

By incorporating and getting help from AI, time can be saved instead of needing personally surfing and reading through learning materials. The only downside to over relying on AI is that they are known for fabrication of information, thus extensive research still must be undertaken to verify the validity of the information.

In this project, multiple AI model will be embodied, including OpenAI's ChatGPT, and OpenAI's Codex. Codex will be used for main development of project and will be further explained in Chapter 4.

To prevent overdependence on these tools, they are strictly used just for suggesting, reviewing and accelerating work process, but important outputs will still be manually checked and corrected by the developer.

CHAPTER 4

System Design

This chapter describes the app design in more details. It will explain project identity, module blocks, page flows, component specifications, database design, API design and component interaction operations. The goal is to provide information to clarify the structure and main functions of the application.

4.1 Project Identity, Logo Meaning and Design Inspiration

The application is named Roofed. Since this is a family-oriented project, the name is inspired by the idea of feeling protected and secured under one roof. It symbolises shelter, safety and belonging of one family. Therefore, even when the family members are physically apart, the applications can act as a medium where they can communicate, coordinate daily routines and understand each other more clearly.

Figure 4.1 App Logo - Roofed

As shown in Figure 4.1, the logo concept also supports the name through several visual design ideas. Firstly, the connected letters of two “O”s from the name Roofed, it is designed like an infinity symbol, which represent immaculate and long lasting family relationships. Besides, a small heart can be seen in one of the letter “O”, which portrays care, affection and emotional support between family members. Lastly, the roof design covering the whole name of the app depicts that every user is being sheltered, similar to the idea of the name itself.

The inspiration of the logo and name comes from one of the objectives of the project, which is being a family-centred app. Therefore, the name and the logo are designed to fulfil this purpose. The summary of this part is listed in Table 4.1.

Table 4.1 Logo Design Elements and Meanings

Logo Element	**Meaning**	**Reason for Using It**
Roof shape	Protection, home, family shelter, and safety.	It directly matches the application name and the family-oriented purpose.
Word “Roofed”	A shared digital space under one roof.	It makes the brand name readable and suitable for both app icon and report presentation.
Connected “O” s letters	Bonding, continuity, and family connection.	It shows that family members are linked rather than isolated.
Heart symbol	Care, affection, and emotional support.	It reflects the emotional-awareness and family-advice direction of the project.

The logo can be seen throughout the app, from the splash screen, authentication pages to the home page. Therefore, it can strengthen user’s impression of the logo.

4.2 System Block Diagram

This section shows the system block design of the app. Figure 4.2 illustrates the system block diagram while Table 4.2 shows the system block description.

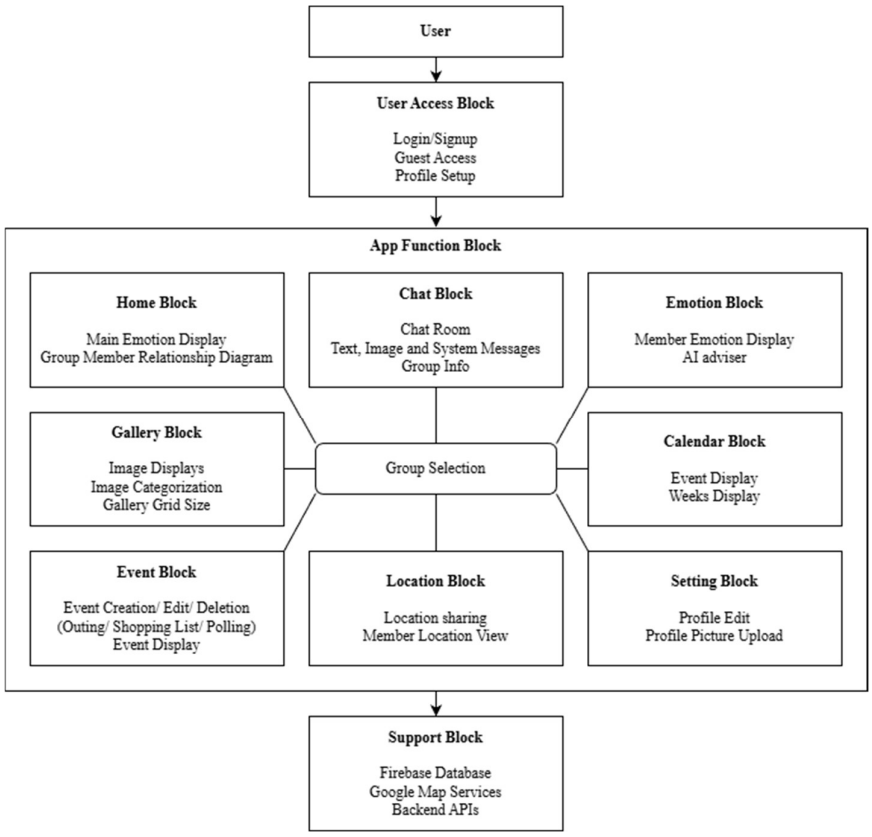


Figure 4.2 System Block Diagram

Table 4.2 System Block Description

Block	Main Pages / Files	Main Function
User Access	SplashPage, AuthEntryPage, LoginPage, SignupPage, ProfileSetupPage	Controls account entry, guest entry, profile completion, profile editing, and user identity display.
Home	HomePage	Display main emotion of user, interactive relationship diagram to show active status of other members
Chat	ChatPage, ChatRoomPage, GroupInfoPage	Allows users to create/join groups, send messages, upload

Block	Main Pages / Files	Main Function
		images, manage group settings, and view group chat.
Emotion	EmotionPage, FamilyAdvicePage	Summarises recent message tone, displays member emotion status, and provides family-oriented advice.
Gallery	GalleryPage, image preview dialog	Displays uploaded chat images in a browsable and categorized family gallery.
Calendar	CalendarPage, PlannerTaskCard	Display weekly viewing, show events
Event	TaskPage, AddTaskPage,	Manages upcoming events, category-specific tasks, shopping lists, polls, and event visibility.
Location	LocationPage	Allows live in-app location sharing within selected family groups.
Setting	SettingsPage, ProfileSetupPage	Allow user to logout, profile editing
Support	backend API services and firebase services	Give functionality to the UI blocks above

4.3 System Components Specifications

This section describes the important components of the app. Since Roofed does not include electronic circuits or intricate physical hardware components, the component will focus on software perspective.

Table 4.3 represents the system component specifications.

Table 4.3 List of System Component Specifications

Component Type	Component Name	Specification / Responsibility
Mobile framework	Flutter and Dart	Used to build the Android mobile application, user interface, navigation, form handling, and service calls.
Application shell	AppShell	Maintains all app pages such as Home, Emotion, Chat, Gallery, Calendar, Event, and Location.
Authentication service	AuthService	Handles email login, username-based login resolution, signup, anonymous login, password reset, and sign out.
Profile service	ProfileService	Reads and updates profile records, uploads profile images, resolves user labels, and updates activity timestamps.
Notification service	TaskNotificationService	Requests notification permission, stores FCM tokens, initializes local notification display, and handles foreground notifications.
Location service	LocationSharingService	Checks location permission, obtains current position, streams location updates, and updates sharedLocation records.
Image upload service	OptimizedImageService	Selects profile/chat images, prepares upload data, uploads files, and returns URLs or progress.
Emotion service	EmotionDetectionService	Calls the external emotion API and normalizes label, confidence, and score output.
Advice service	FamilyAdviceService	Sends question and safe context to the family advice API and returns the generated response.
Image classifier service	GalleryImageClassifierService	Sends image data to external backend and returns category metadata.
Notification delivery	Firebase Cloud Functions	Processes notification trigger records and sends push notifications and scheduled reminders.
Backend API	FastAPI service on Cloud Run	Hosts emotion prediction, image classification, and family advice endpoints.

4.4 Database Design

As mentioned before, Firestore is the database applied to this app. It provides data logging for these data collections: users, groups, messages, tasks, locations, notifications, and advice logs. Firebase Storage is used for storing images files, while Cloud Messaging handle the FCM notification delivery system. The external backend API is connected through HTTP endpoints.

Table 4.4 explains the Firestore database collection design.

Table 4.4 List of Firestore Database Collection Design

Collection	Purpose	Important Fields / Examples	Related Module
users	Stores account-linked profile and notification data.	email, username, displayName, photoUrl, bio, preferences, lastActiveAt, fcmTokens, mutedGroupIds.	Profile, Auth, Notifications, Advice.
groups	Stores shared group identity and membership.	name, code, createdBy, members, photoUrl, nicknames.	Group, Chat, Gallery, Event, Location.
messages	Stores chat, image, system, event, and emotion-related message records.	groupId, senderId, type, text, imageUrl, emotionLabel, emotionScores, imageCategory, eventSystemType.	Chat, Emotion, Gallery, Event preview.
tasks	Stores planner and event records.	groupId, title, category, description, taskDateTime, visibleToAll, selectedMemberIds, eventLocation, shoppingItems, pollOptions.	Event, Calendar, Notifications.
taskNotifications	Stores normalized notification trigger records for backend processing.	type, groupId, taskId, messageId, actorId, recipientIds,	Notifications, Cloud Functions.

Collection	Purpose	Important Fields / Examples	Related Module
		notificationTitle, notificationBody.	
sharedLocation	Stores current in-app location sharing state.	userId, groupId, lat, lng, isSharing, updatedAt, photoUrl.	Location.
familyAdviceLogs	Stores previous family advice questions and replies.	askerUserId, targetUserId, groupId, question, reply, usedEmotionSummary, createdAt.	AI Advice.

4.4.1 User Collection

The user collection is vital for the account system. It records the username, profile setup active status and FCM token related data.

Figure 4.3 and 4.4 show examples of the user collection.

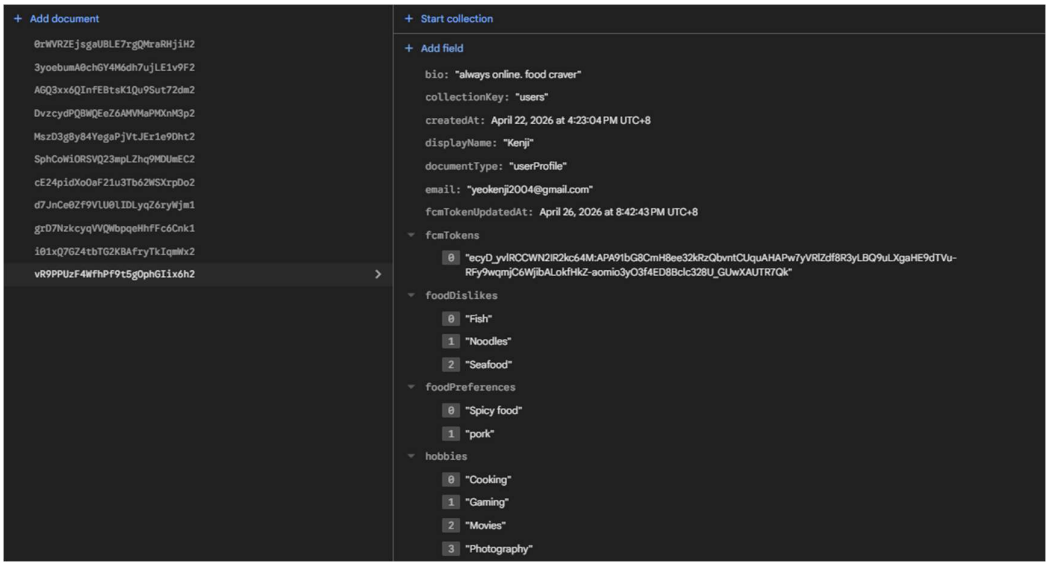


Figure 4.3 Example of User Collection 1

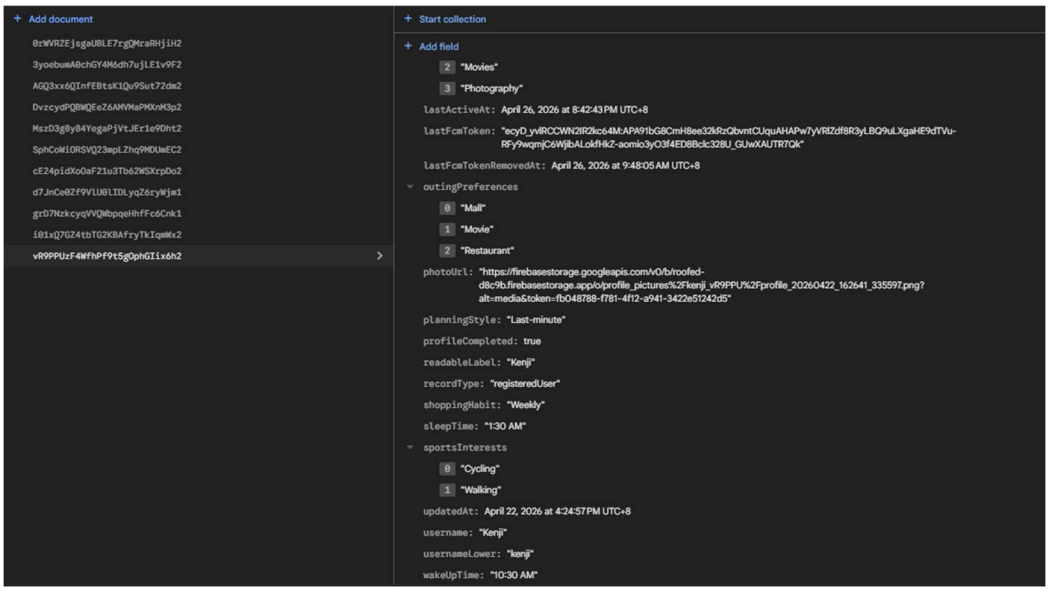


Figure 4.4 Example of User Collection 2

4.4.2 Group Collection

The group collection is pivotal for the chat group system. It registers group info to form a chatroom.

Figure 4.5 shows example of the group collection.

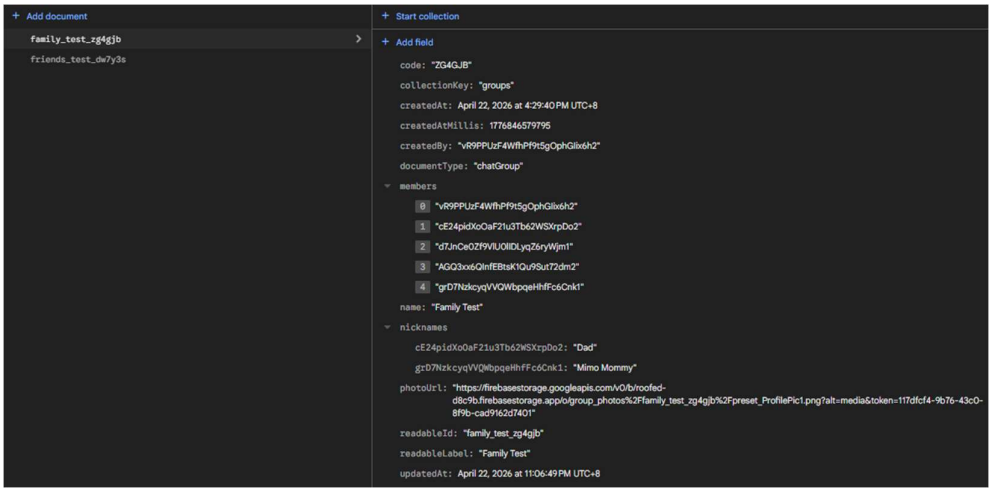


Figure 4.5 Example of Group Collection

4.4.3 Message Collection

The message collection is the significant part for the chat messaging system. It supports multiple different type of message such as text, image and system messages.

Figure 4.6, 4.7 and 4.8 show example of the message collection for text, image and system message respectively.

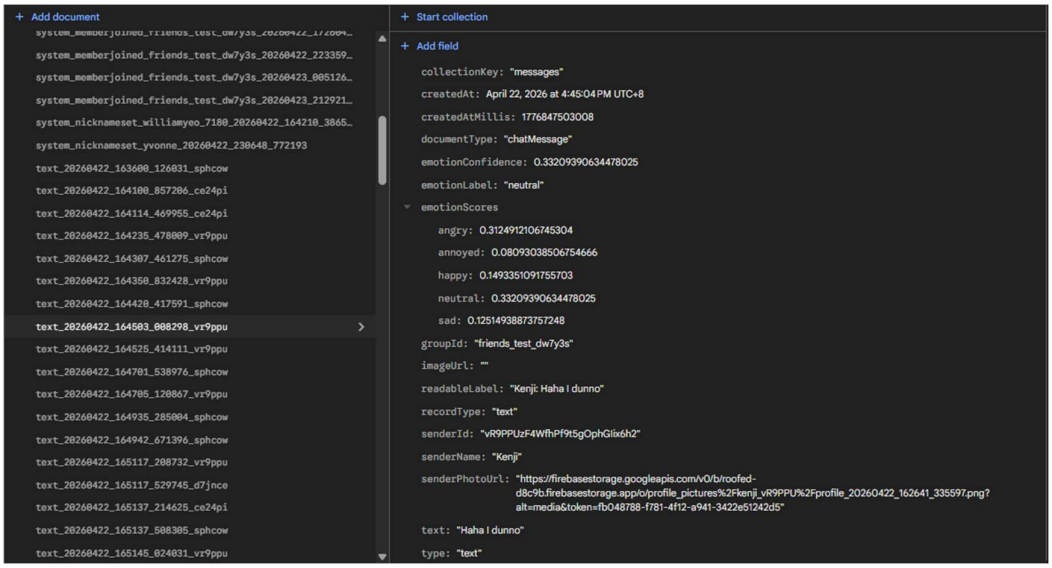


Figure 4.6 Example of Message Collection - Text

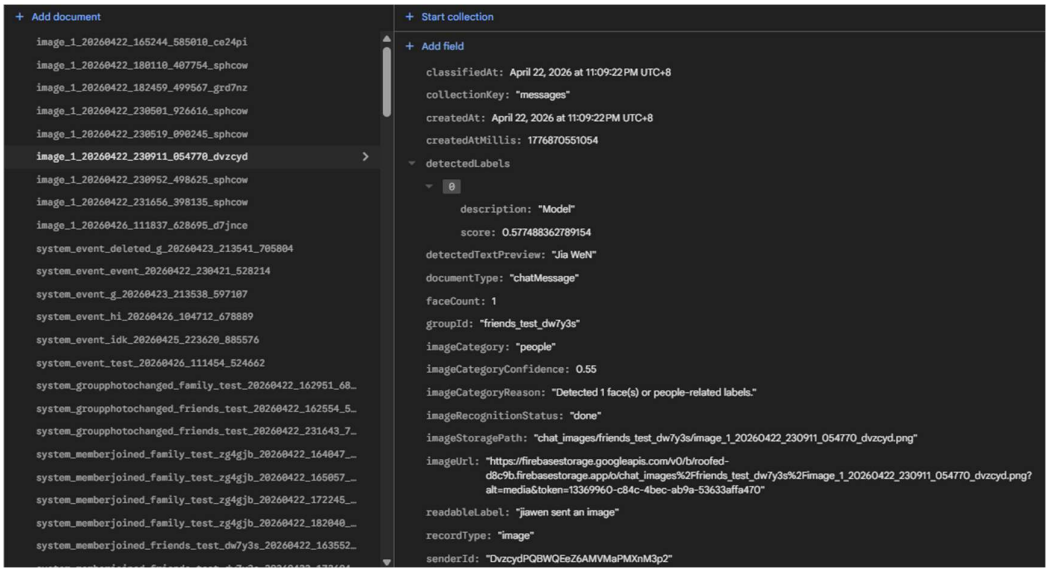


Figure 4.7 Example of Message Collection - Image

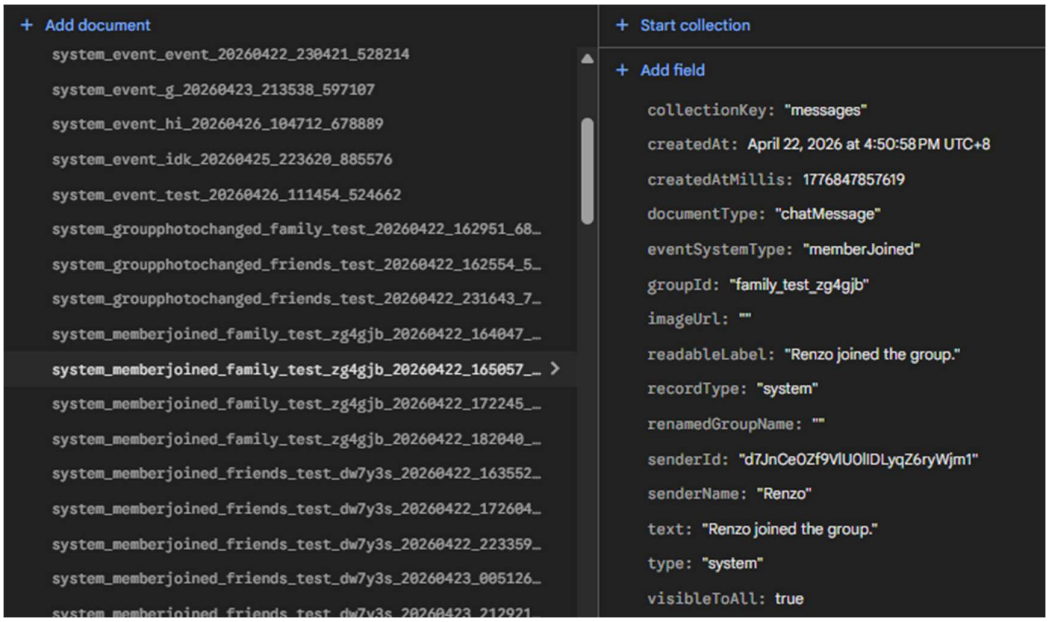


Figure 4.8 Example of Message Collection – System

4.4.4 Tasks Collection

The tasks collection is crucial for the event system. It logs event details.

Figure 4.9 shows example of the tasks collection.

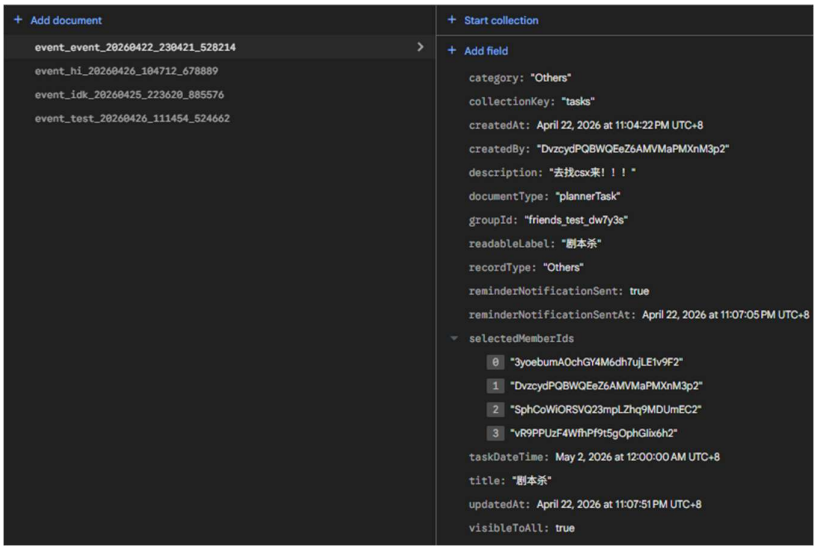


Figure 4.9 Example of Tasks Collection

4.4.5 Task Notifications Collection

The task notifications collection is essential for the notification of every element. It notifies users when message is sent in group, group info is changed and events related actions.

Figure 4.10 shows example of the task notifications collection.

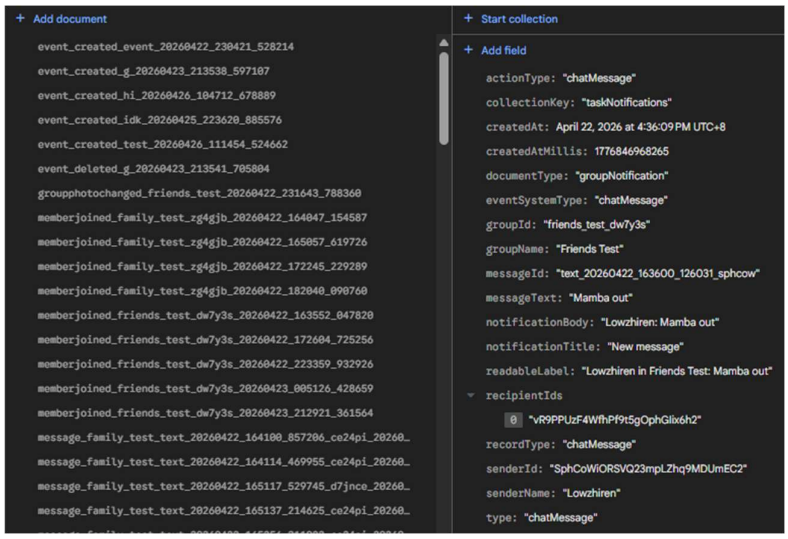


Figure 4.10 Example of Task Notifications Collection

4.4.6 Shared Location Collection

The shared location collection is important for the shared location system. It registers location metadata.

Figure 4.11 shows example of the shared location collection.

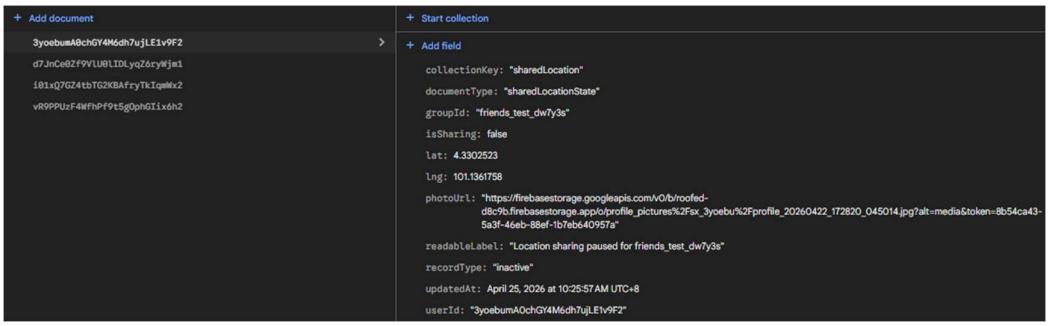


Figure 4.11 Example of Shared Location Collection

4.4.7 Family Advice Logs Collection

The family advice log collection is significant for the chat group system. It stores the data related to AI inquiries.

Figure 4.12 shows example of the family advice log collection.

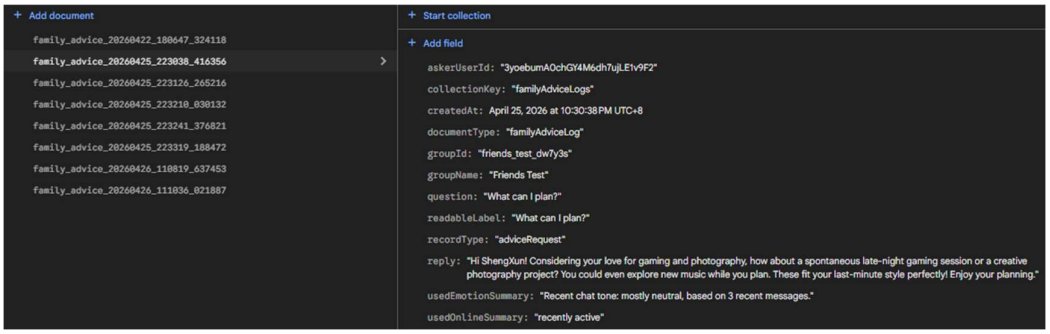


Figure 4.12 Example of Family Advice Log Collection

4.5 Backend Service Design

Table 4.5 shows the backend service design.

Table 4.5 Backend Service Design

Endpoint	Input	Processing	Output
/predict	Message text.	Normalizes shortforms, extracts TF-IDF and raw-text intensity features, runs Logistic Regression model.	emotion label, confidence, normalized text, score distribution.
/classify-image	imageUrl, gcsUri, or imageBase64.	Calls Google Vision for label detection, OCR, and face detection; applies rule-based category scoring.	category, confidence, reason, face count, detected labels, OCR preview.
/family-advice	Question, asker profile, target profile, recent emotion summary, online summary, relationship context.	Sanitizes profile context and calls Gemini with family-advice restrictions.	Short family-focused advice reply and model metadata.
Google Map API	Location name or address.	Converts textual location into latitude and longitude.	Coordinates and formatted address.

4.5.1 Emotion Classification API

The emotion prediction using sentiment text analysis is one of the main objectives in Roofed. When a user sends a text message in the chatroom, the system uses /predict function to the API. The backend then analyses the text and return the predicted emotion labels.

It is developed in a Python NLP model, based on scikit-learn pipeline. It uses TF-IDF text features, combine with a Logistic Regression and a message-intensity feature, so that it can better identify and classify emotions. This model is very lightweight and responsive compare to more complicated model like transformer. There are 5 total emotion labels: angry, annoyed, happy, neutral and sad.

The model's database contains a total of 10398 unique cases in csv format, specially crafted with Malaysian and Internet Slang in mind. Therefore, realistic communication can be better identified. Out of the training cases, 2080 are angry, 2140 are annoyed, 2060 are happy, 2058 are neutral and 2060 are sad examples. The training procedure

uses 8318 cases for training and 2080 cases for validation. In short, the vocabulary size of the model is 12006. At late development, the model was also added with unique instruction to elongated words, repeated words or punctuation marks. By doing so, the model can differentiate the intensity of message to better identify emotions.

The text pre-processing stage emphasis on handling short forms, Malaysian slang and emojis. For example, emojis such as 😜 will be translated to happy readable token before getting into the NLP model. Therefore, the model can identify emojis correctly instead of ignoring them like the previous low accuracy rate models.

Table 4.6 and 4.7 shows the summary of emotion classifier model.

Table 4.6 Summary of Emotion Classifier Model 1

Item	**Value**
Model type	TF-IDF text features + six handcrafted raw-text features + Logistic Regression
Emotion labels	angry, annoyed, happy, neutral, sad
Total labelled examples	10,398
Training split rows	8,318
Validation split rows	2,080
TF-IDF vocabulary size	12,000

Table 4.7 Summary of Emotion Classifier Model 2

Emotion Label	**Number of Examples**
angry	2,080
annoyed	2,140
happy	2,060
neutral	2,058
sad	2,060

After classifying, the output from the API is stored in Firestore. Then, it allows Home Page and Emotion Page to calculate and display the emotion summaries. But it does not directly use the raw output, instead, it will go through a decay function to better represent real-life situation, where emotion fades out after some time. The equation:

$$w_i = max(0, 1 - \frac{age_i}{7})$$

$$w_i = weight\ of\ message\ i$$
$$age_i = age\ of\ the\ message\ in\ days$$
$$7 = maximum\ recent - message\ window\ in\ days$$

After the weight is calculated, each message emotion score is multiplied by its decay weight. The weighted score for each emotion is then added together:

$$S_e = \Sigma(w_i \times score_{i,e})$$

$$S_e = total\ weighted\ score\ for\ emotion\ e$$
$$w_i = decay\ weight\ of\ message\ i$$
$$score_{i,e} = prediction\ score\ of\ emotion\ e\ for\ message\ i$$

Finally, the emotion percentage is calculated by normalising the weighted score:

$$P_e = \frac{S_e}{\Sigma S_k}$$

$$P_e = final\ displayed\ percentage\ for\ emotion\ e$$
$$S_e = total\ weighted\ score\ for\ emotion\ e$$
$$\Sigma S_k = total\ weighted\ score\ of\ all\ emotion\ classes$$

In conclusion, the decay functions reflect realistic situation by focusing on recent communication patterns, instead of relying on certain messages. By using it, the emotion will change over time, rather than having a permanent label.

4.5.2 Image Classification API

The image classification API is used to differentiate categories in the shared gallery function. When an image is uploaded in chatroom, Roofed calls the /classify-image endpoint to identify the image. It analyses the image via URL input, by detecting and identifying objects or text using OCR. Then, it is processed using rule-based logic to categorize the image into either people, bills or other.

Similar to the emotion classifier, the API returns the predicted category, its confidence value, reason, detected labels, face count and a short OCR text preview to Firestore. Finally, the Gallery Page can display the images by category. The reason only three reasons are chosen is to prevent overcomplicating this feature. Moreover, these categorizations are sufficient for daily use.

4.5.3 Family Advice API

The family advice API is integrated in the Emotion Page. Unlike general chatbot, the user needs to select a group and a member, so that the question can be focus to one single individual. Related information such as profile summary and recent emotion summary will be collected by the API, which help the AI to get important context, so that it can reply relevantly and correctly.

The AI model is also designed for advice usage only. If the user asks out of range questions like mathematical or scientific question, it will not answer those inquiries. Instead, it will prompt user to enter relevant question. A token limit is also set, so that the answer will be compacted and brief. Firestore will record the reply and users are able to review to old inquiries.

4.5.4 Google Map API

The Google Map API is a Google Cloud API that allows an app development to have geocoding features, such as showing map and pinging location. The location metadata is stored in Firestore for relevant feature use. In this case, the project uses it in its location sharing feature and the outing event creation.

4.6 UI Interaction Design and Page Flow Diagrams

This section will present the flow of the pages in Roofed.

Table 4.8 explains the page flow.

Table 4.8 Page Flow

Page / Screen	Purpose	Page Flow
Splash Page	Initial loading and routing screen.	Open app -> Show logo animation -> Initialize Firebase -> Check user session -> Check profile completion -> Route to Auth Entry / Profile Setup / App Shell.
Auth Entry Page	Entry page for unauthenticated users.	Auth Entry -> Choose Login / Sign Up / Guest -> If Firebase ready, continue -> If Firebase not ready, show warning.
Login Page	Registered user login.	Enter email or username -> Enter password -> Validate -> Resolve username to email if needed -> Firebase login -> Check profile -> App Shell or Profile Setup.
Signup Page	New account creation.	Enter username, email, password -> Validate username -> Check username uniqueness -> Create Firebase Auth account -> Create user document -> Profile Setup.
Profile Setup Page	Collects user profile and personalization data.	Load existing profile if any -> Enter basic info -> Select profile picture -> Select interests/preferences -> Pick routine details -> Save profile -> Mark profileCompleted true -> App Shell.
App Shell	Main navigation container.	Load retained pages -> Start notification listener -> Update lastActiveAt periodically -> User switches between Home, Emotion, Chat, Gallery, Calendar, Event, Location.
Home Page	Visual family dashboard and relationship overview.	Select group -> Load members and recent messages -> Compute emotion summary -> Display emotion circles and relationship nodes -> Tap member -> Show selected member card.

Page / Screen	Purpose	Page Flow
Emotion Page	Detailed selected-member emotion view.	Select group -> Select member -> Load recent text messages -> Compute weighted emotion percentage -> Show dominant emotion and details -> Open AI adviser if needed.
Family Advice Page	AI family advice interface.	Select group/member -> Enter question -> Load asker and target profile context -> Compute recent emotion and online summary -> Send /family-advice request -> Show reply -> Store advice log.
Chat Page	Group list and group access page.	Show create/join buttons -> Create group or enter join code -> Update groups collection -> Display group cards -> Tap group -> Chat Room.
Chat Room Page	Realtime group conversation page.	Load group messages -> Send text/image -> Save message -> Call emotion or image classifier -> Create notification trigger -> Display realtime updates.
Group Info Page	Group management page.	Open from chat header -> View group details -> Change photo/name/nickname/mute status -> Write system message and notification -> Return to chat.
Gallery Page	Shared image gallery generated from chat images.	Select group/category/grid size -> Read image messages -> Group by date -> Display grid -> Tap image -> Preview or save.
Calendar Page	Weekly schedule overview.	Select group -> Select week or swipe week -> Read visible tasks -> Bucket tasks by day -> Display seven-day schedule.
Event Page	Upcoming event management page.	Select group -> Read future tasks -> Show summary and event list -> Add/Edit/Delete event -> Update Firestore and notification records.
Add/Edit Event Page	Event creation and editing form.	Enter title/category/date/time -> Add category-specific fields -> Select visible members -> Save -> Write task -> Create system message and notification.
Location Page	Live group location sharing page.	Select group -> Enable sharing -> Check permission -> Write current location -> Stream

Page / Screen	Purpose	Page Flow
		updates -> Show map markers -> Disable sharing when finished.
Settings Page	Account and profile access page.	Open from Home -> Show profile summary -> Edit profile or logout -> Handle profile-write warning if needed.

Figure 4.13 represents the page flow diagram. To prevent confusion, the navigation bar will only be noted once in the home page.

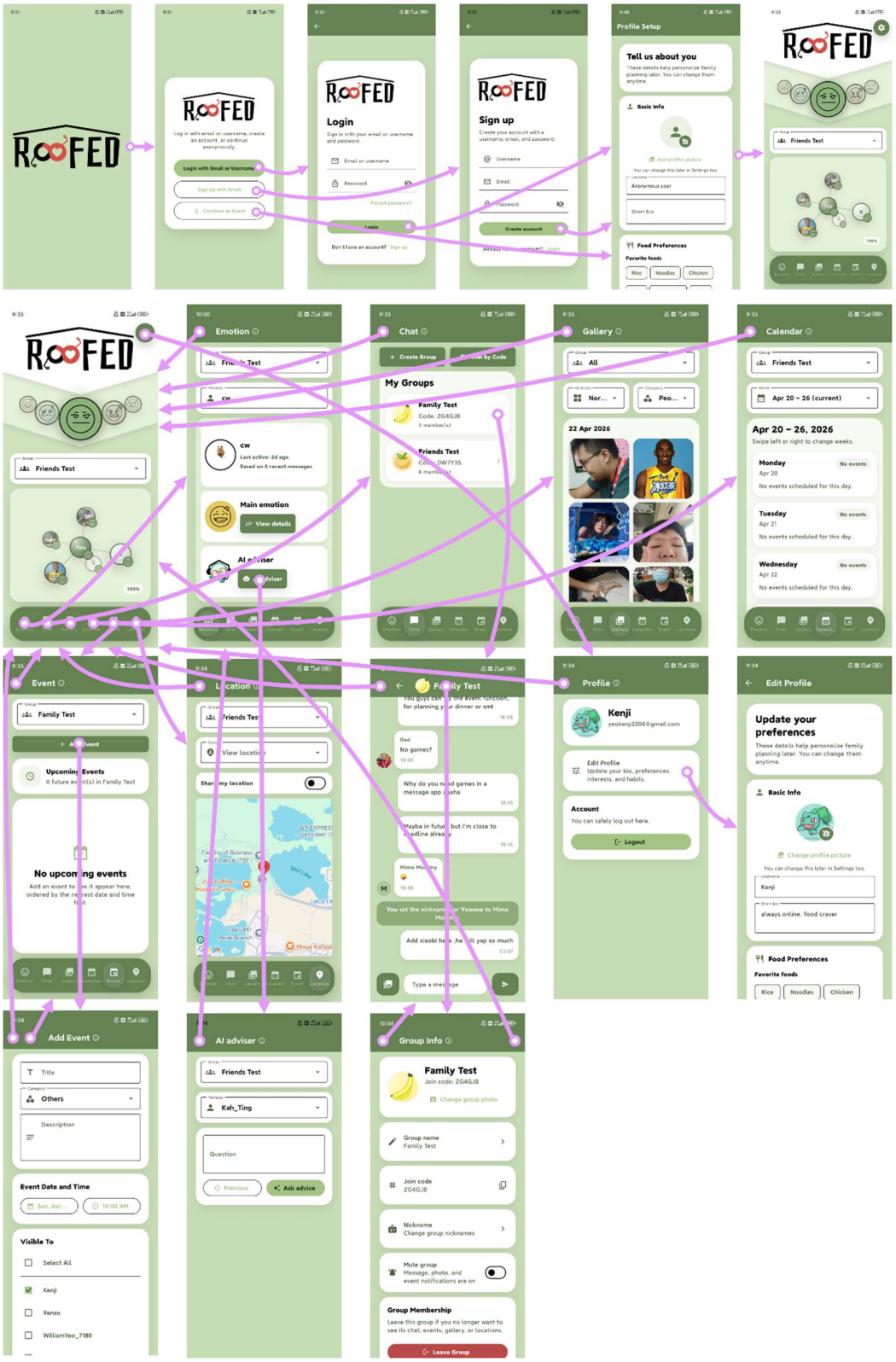


Figure 4.13 Page Flow Diagram

4.7 System Components Interaction Operations

Table 4.9 shows the operations interacted by system components.

Table 4.9 System Components Interaction Operations

Operation	Component Interaction
User authentication	SplashPage checks Firebase state -> Auth pages collect input -> AuthService calls Firebase Auth -> ProfileService checks users collection -> AppShell opens if completed.
Text chat and emotion detection	ChatRoomPage collects text -> EmotionDetectionService calls /predict -> message document stores text and emotion metadata -> HomePage and EmotionPage aggregate recent scores.
Image upload and gallery sorting	ChatRoomPage selects image -> OptimizedImageService uploads to Storage -> messages document stores image URL -> GalleryImageClassifierService calls /classify-image -> GalleryPage filters by category.
Event creation and reminder	AddTaskPage writes task -> system message written to messages -> taskNotifications record created -> Cloud Functions sends FCM -> CalendarPage reads task for weekly view.
Family advice generation	FamilyAdvicePage loads profiles and emotion summary -> FamilyAdviceService calls /family-advice -> reply shown in UI -> familyAdviceLogs stores history.
Live location sharing	LocationPage requests permission -> LocationSharingService writes sharedLocation -> live stream updates same record -> map markers refresh by group.
Group management	GroupInfoPage updates group name/photo/nicknames/mute -> groups collection updated -> system messages and notification records created when needed.

CHAPTER 5

System Implementation

This chapter demonstrates the implementation of Roofed. It describes the hardware, software, configuration and operation of the development.

5.1 Hardware Setup

The hardware involved in this project is computer and Android mobile devices. A computer is used for the develop of Roofed. Two Android mobile devices is used for testing and verify the behaviour of the application.

Table 5.1 lists the specifications of the development computer, while table 5.2 and 5.3 shows the specifications of two mobile phones.

Table 5.1 Specifications of Development Computer

Description	Specifications
Processor	AMD Ryzen 5 5600
Operating System	Windows 11
Graphic	AMD RX 6800 16GB DDR6
Memory	32GB DDR4 RAM
Storage	2TB NVME M.2

Table 5.2 Specifications of Mobile Phone 1

Description	Specifications
Model	Xiaomi 15 Pro
Processor	Qualcomm SM8750-AB Snapdragon 8 Elite
Operating System	Android 15, HyperOS 2.0
Graphic	Adreno 830
Memory	16GB RAM
Storage	512GB ROM

Table 5.3 Specifications of Mobile Phone 2

Description	Specifications
Model	Huawei Nova 5T
Processor	HiSilicon Kirin 980
Operating System	Android 10, EMUI 12.0.0
Graphic	Mali-G76
Memory	8GB RAM
Storage	128GB ROM

By using more than one Android device, it testing and validation process can be more accurate. For example, application behaviour may differ depending on Android version and screen size dimensions. Therefore, testing on two phones could identify issues that might not appear in an emulator or on a single device only.

5.2 Software Setup

Roofed was developed on Windows 11. The software setup consists of mobile app development tools like Flutter, backend development tools like Python, Firebase services, Google Cloud tools, coding assistance tools and more.

Table 5.4 shows the software setup in this development.

Table 5.4 Software Setup

Software / Tool	Purpose in Project
Flutter SDK	Used to build the Android mobile application with Dart.
Dart	Programming language used by the Flutter application.
Android Studio / Android SDK	Provides Android platform tools, emulator, SDK manager, command-line tools, and build support.
Visual Studio Code	Main code editor for Flutter, Dart, Python, and project file editing.

Software / Tool	Purpose in Project
Codex IDE extension	AI-assisted coding tool used inside VS Code for reading code, drafting fixes, refactoring, and implementation support.
Git	Version control and codebase management.
Firebase CLI / FlutterFire CLI	Used to connect the Flutter project with Firebase services and generate configuration files.
Node.js 20	Runtime used for Firebase Cloud Functions.
Python 3.13.7	Used for the emotion classifier and FastAPI backend.
FastAPI	Used to expose backend endpoints for emotion, image classification, and family advice.
Google Cloud SDK	Used to deploy the backend to Cloud Run and manage cloud project configuration.
Google Cloud Run	Hosts the FastAPI backend.
Google Cloud Vision	Supports image label, OCR, and face detection for gallery classification.
Vertex AI / Gemini	Supports the family advice feature.

Flutter and Android Studio are the essential tool in this project as Flutter requires an Android build environment and other mandatory files to be able to run and build the app. Unlike the previous FYP1 project, Visual Studio Code was used as the main editor due to its better interface and extension support that are required more during this development. It is important to note that Android Studio still remains crucial because it provides SDK management, Android command-line tools and good Android emulators for testing.

Moreover, OpenAI's Codex IDE extension was also used in VS Code as an AI assisted development tool. Vibe coding, which refers to guiding coding workflow where the developer describes a task, then request the AI assistant to inspect and generate relevant codes. Throughout this development, Codex IDE extension act as the main part in creating codes, combining features and cleaning up codes. However, most generated codes still required manual checking, therefore the developer role in developing an app is still important, especially at the testing and verification phase.

The backend software setup used in this project are Python language, while using libraries such as scikit-learn, pandas, joblib, Numpy and FastAPI. The NLP sentiment text analysis model was trained and saved as a joblib model bundle, then FastAPI loads the saved model and exposes it through JSON endpoints. FastAPI also did the same thing for the image classification and AI family advice system.

The Google Cloud setup included Cloud Run deployment for the FastAPI model. The cloud project used during development that connects the Flutter file with the Firebase Console, was roofed-d8c9b. This string act similar as an API key, so when inserted into the Flutter file, it can communicate and update with the console. The Cloud Run service name was emotion-api, in the asia-southeast1 region. The backend deployed by Cloud Run was used to provide /predict, /classify-image and /family-advice endpoints to Roofed.

5.3 Setting AND Configuration

5.3.1 Setting Up Flutter

Figure 5.1-5.4 represent the installation guides to Flutter.

1. Go to the link: https://docs.flutter.dev/get-started/install, and choose the platform and device.

2. Scroll Down and click the zip file **download button** to download **Flutter SDK**.

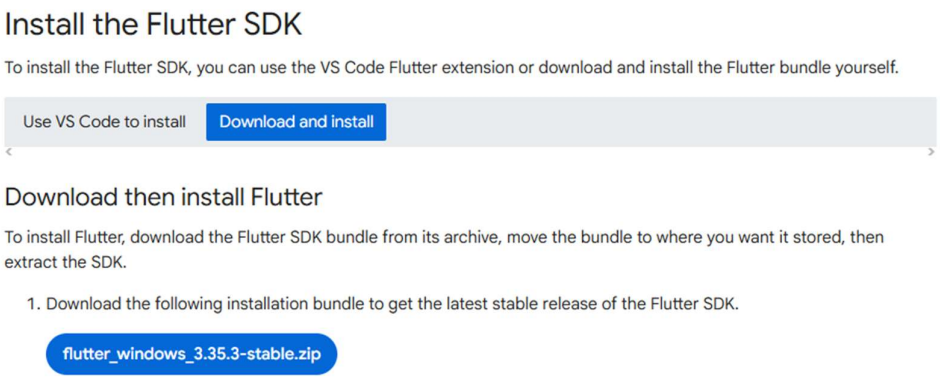


Figure 5.1 Installation Guide to Flutter 1

3. Unzip the file to a location.

4. Add it to the **environment variables path**, choose path then click edit. This allow Flutter to be access throughout the device instead of just in a limited file location.

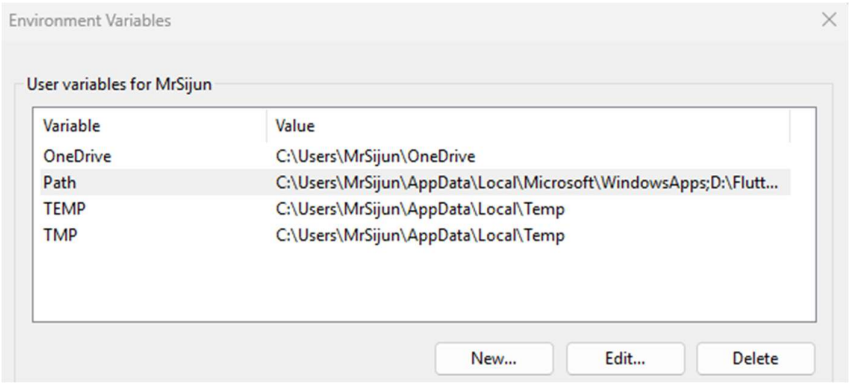


Figure 5.2 Installation Guide to Flutter 2

5. Create a new directory to the **Flutter bin file**.

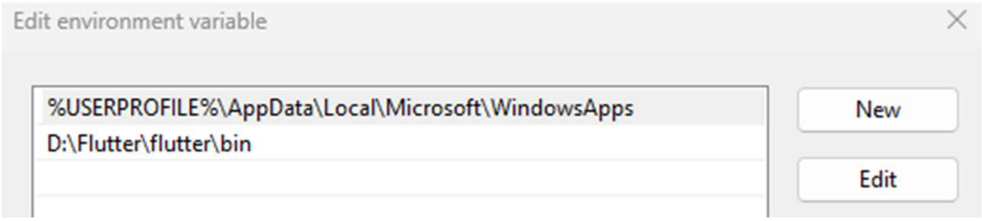


Figure 5.3 Installation Guide to Flutter 3

6. Open cmd and run, **flutter doctor**, to download require packages. Cmdline-tools are required to download in Android Studio.

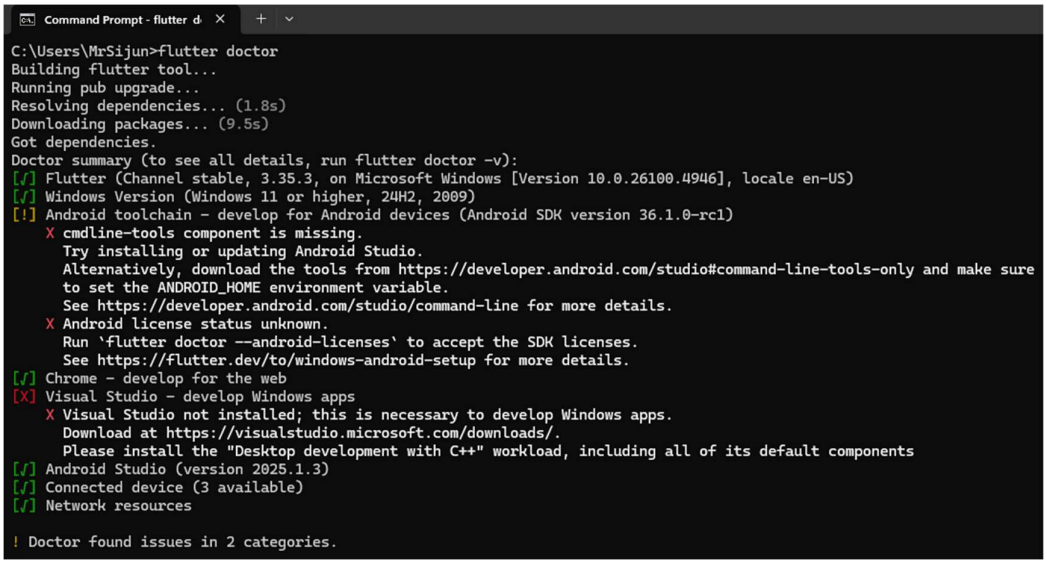


Figure 5.4 Installation Guide to Flutter 4

7. Open cmd and run, **flutter doctor --android-licenses**, accept all licenses.

5.3.2 Setting Up Android Studio

Figure 5.5-5.8 show the installation guides to Android Studio.

1. Go to the link: https://developer.android.com/studio, and click the **download button**.

2. To download relevant files for Flutter, go to settings and download **Android SDK Command-line Tools.**

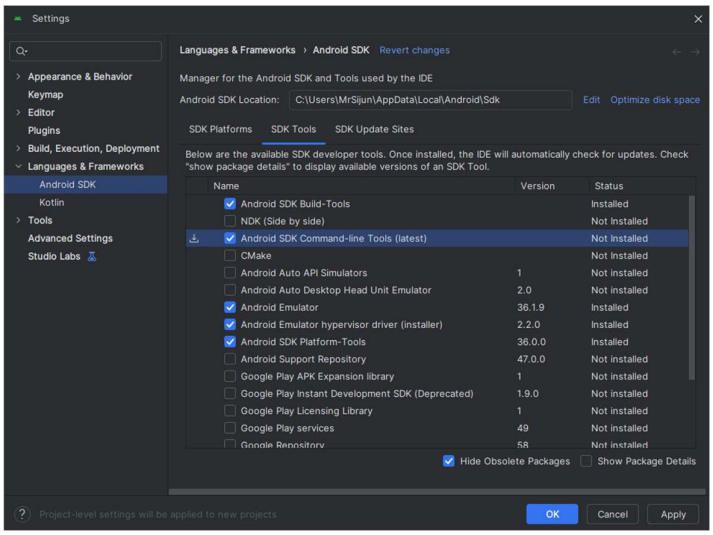


Figure 5.5 Installation Guide to Android Studio 1

3. To create Android Emulator. Go to main page, click **"more actions" menu**, then **Virtual Device Manager**.

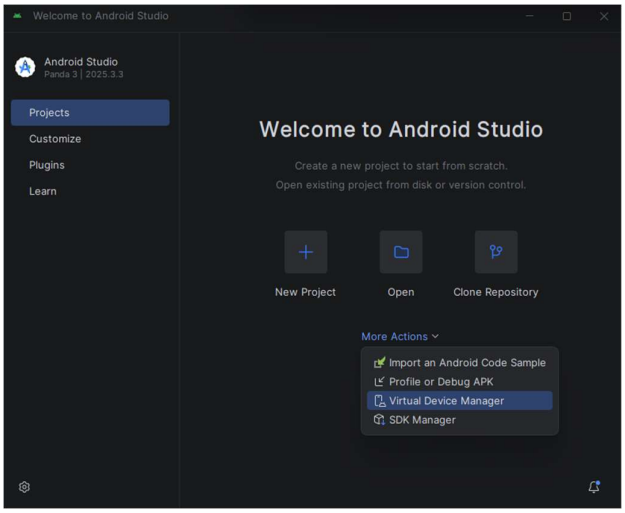


Figure 5.6 Installation Guide to Android Studio 2

4. In the manager, press **“+” button**, then choose device and press **“Next” button**.

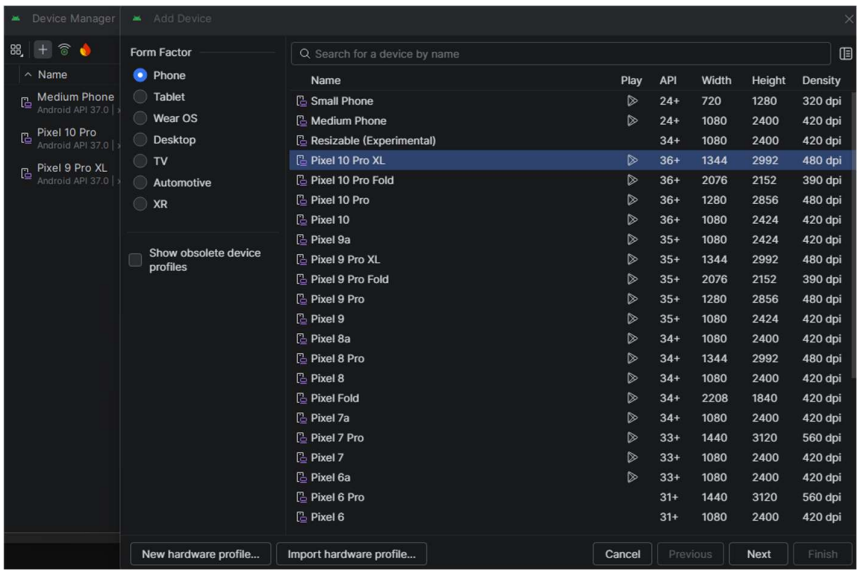


Figure 5.7 Install Guide to Android Studio 3

5. Lastly, press **“Finish” button** to create Android Emulator.

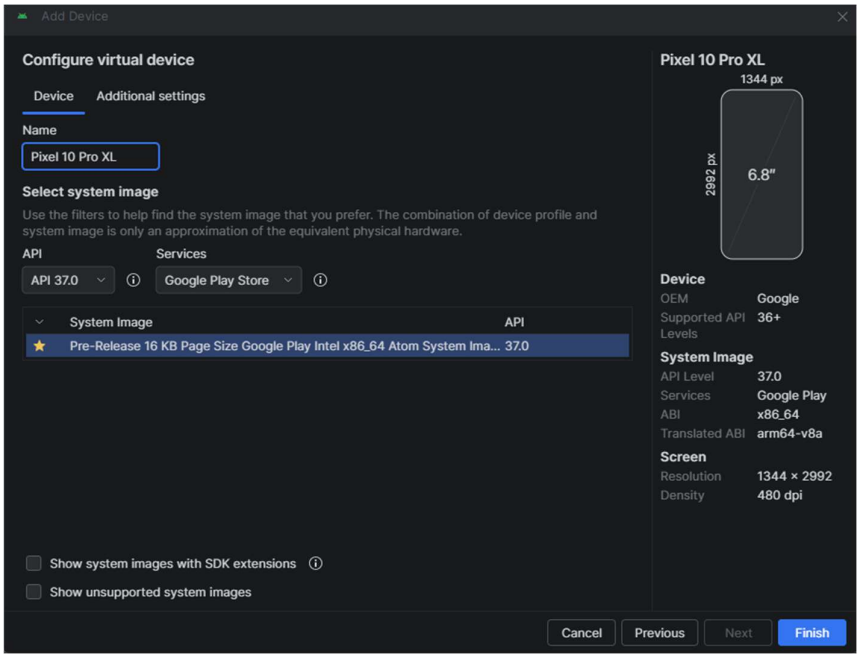


Figure 5.8 Install Guide to Android Studio 4

5.3.3 Setting Up Microsoft Visual Studio Code

Figure 5.9 and 5.10 represent the installation guides to Visual Studio Code.

1. Download VS Code from Microsoft Store or online.

Figure 5.9 Install Guide to VS Code 1

2. Similar to Flutter, allow it to **add to Path** when downloading.

3. In extension tab, install all relevant extensions required in this project.

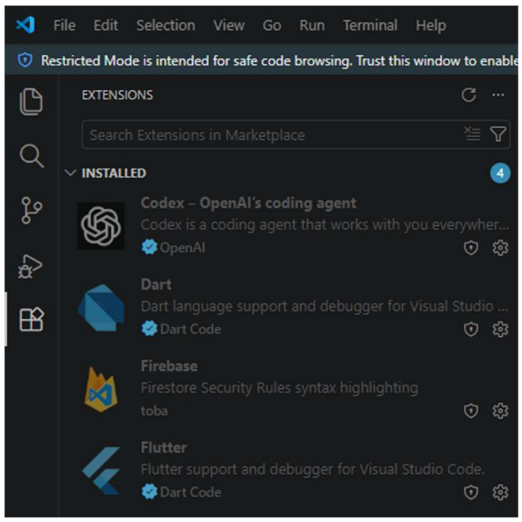


Figure 5.10 Install Guide to VS Code 2

5.3.4 Setting Up Python

1. Install python online or by Microsoft Store.

2. Similar to Flutter, allow it to **add to Path** when downloading.

5.3.5 Setting Up Firebase Console

Figure 5.11 shows the configuration guide to firebase console.

1. Create or Open a Firebase project for Roofed.

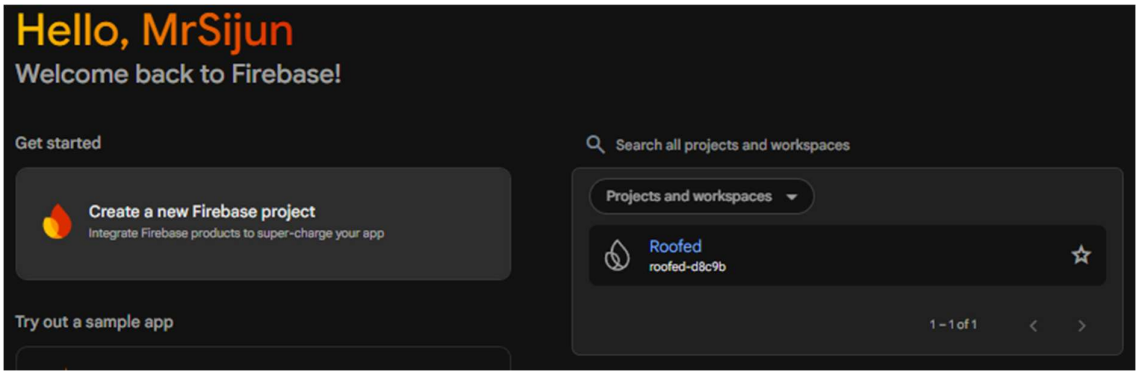


Figure 5.11 Configuration Guide to Firebase Console

2. Register the Android application in the console.

3. Install and run **FlutterFire CLI** to generate required libraries in the Flutter file.

4. Enable **Firebase Authentication, Firestore, Storage and Cloud Messaging** in the console.

5.3.6 Setting Up Google Cloud

Figure 5.12-5.14 represent the configuration guides to Google Cloud.

1. Choose the same project as the Firebase Console.

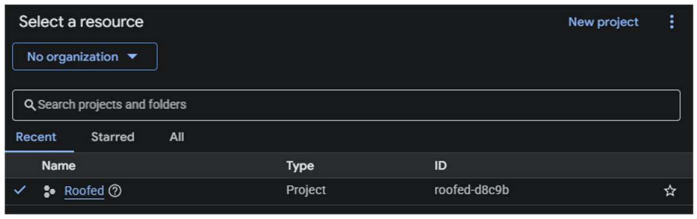


Figure 5.12 Configuration Guide to Google Cloud 1

2. Under APIs & Services tab, create a Google Map API key.

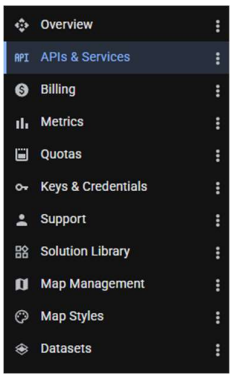


Figure 5.13 Configuration Guide to Google Cloud 2

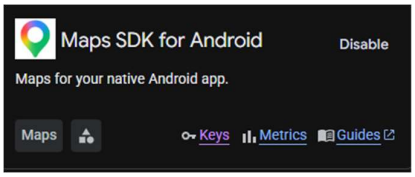


Figure 5.14 Configuration Guide to Google Cloud 3

3. Insert the API key inside the Flutter file.

5.3.7 Setting Up Backend Services

1. Prepare the training dataset in CSV format, with text sample and labels next to it

2. Prepare the shortform mapping file so informal expressions and abbreviations can be normalized.

3. Train the TF-IDF and Logistic Regression classifier using 1-gram to 3-gram TF-IDF phrases, no English stop-word removal, emoji/emoticon normalization, and six raw-text intensity features.

4. Save the trained model bundle as malaysian_emotion_model.joblib together with the replacement dictionary needed for consistent inference.

5. Create the FastAPI backend with routes for /predict, /classify-image, and /family-advice.

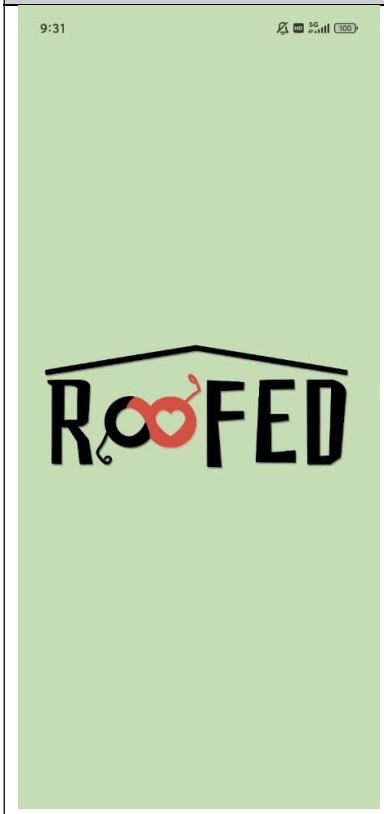
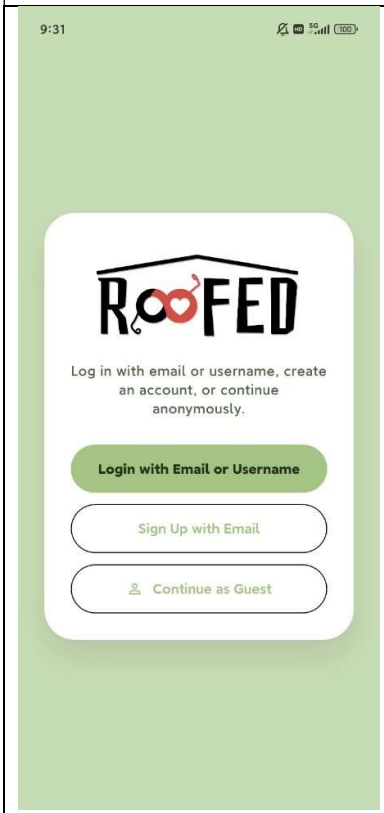
6. Install backend dependencies such as FastAPI, Uvicorn, pandas, scikit-learn, joblib, NumPy, google-cloud-vision, and google-genai.

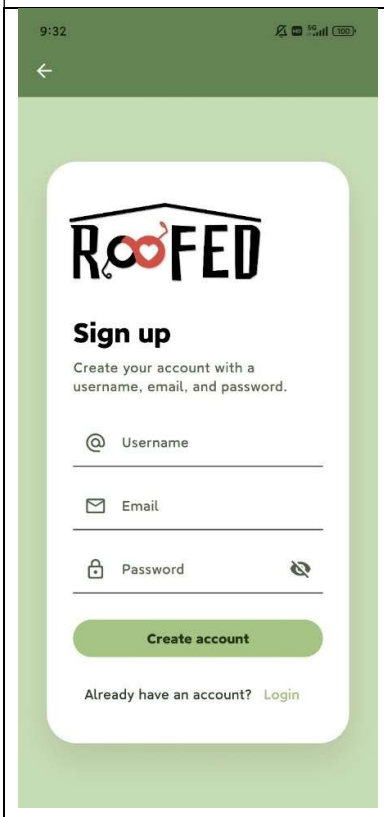
7. Configure Google Cloud project, enable required APIs, and authenticate Google Cloud CLI.

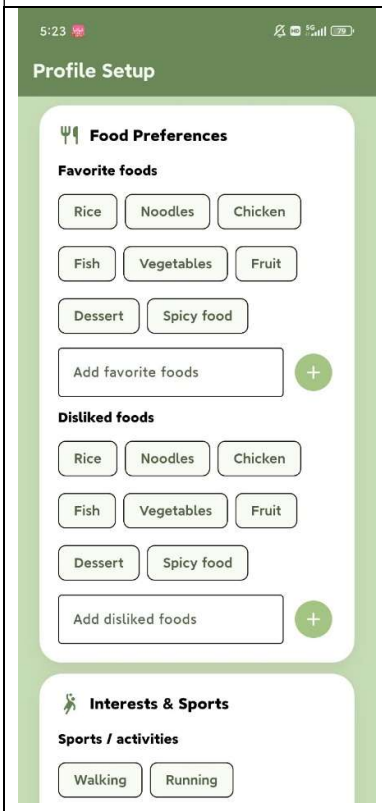
8. Deploy the backend folder to Google Cloud Run in the selected region.

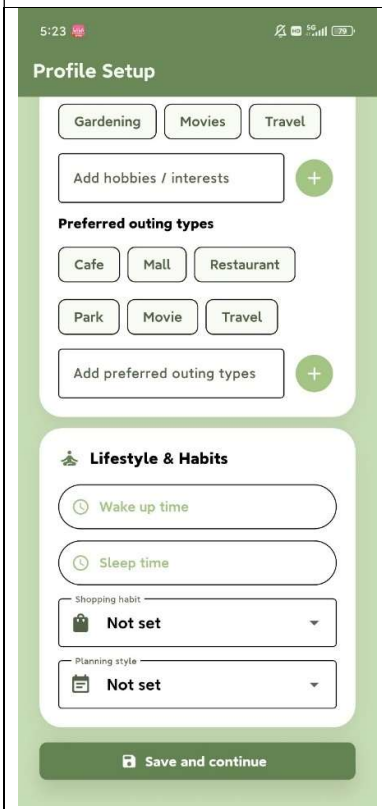
5.4 System Operation (with Screenshot)

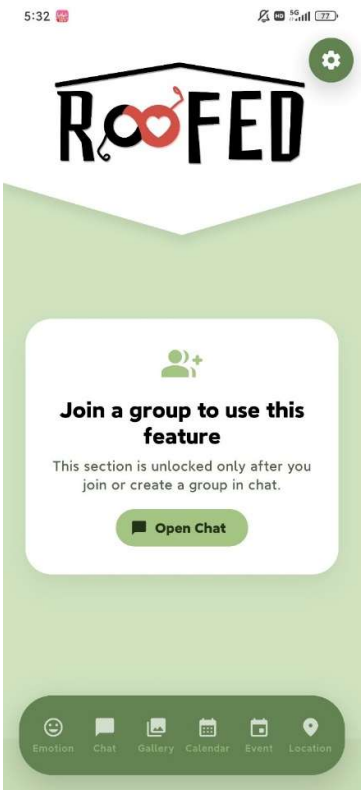
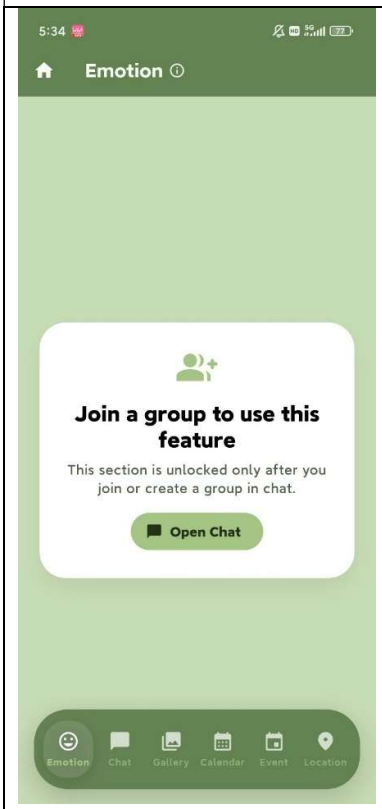
The figures below illustrate the interfaces of Roofed, screenshotted in Xiaomi 15 Pro, in 3200x1400 resolution. Figures 5.15 – 5.52 shows the screenshots of the app pages.

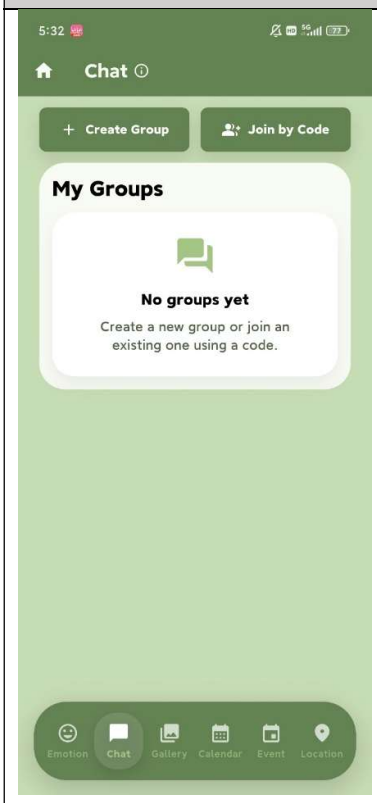
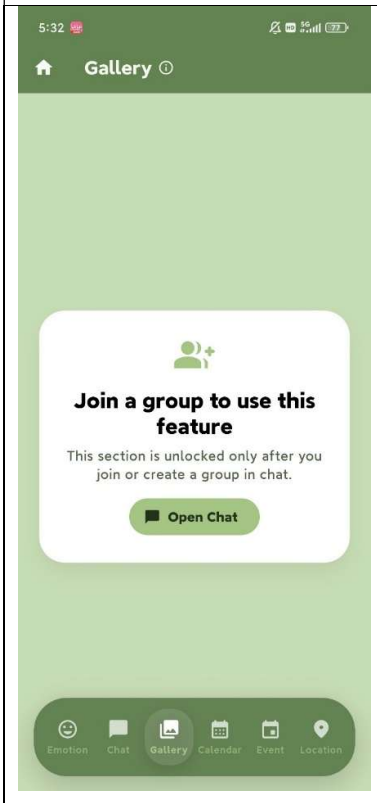
Screenshot	Explanation
	Figure 5.15 Screenshot of Splash Screen: - A 2-second-long splash screen containing the logo of Roofed.
	Figure 5.16 Screenshot of Authentication Page: - Contains three choices: login, sign up or guest session

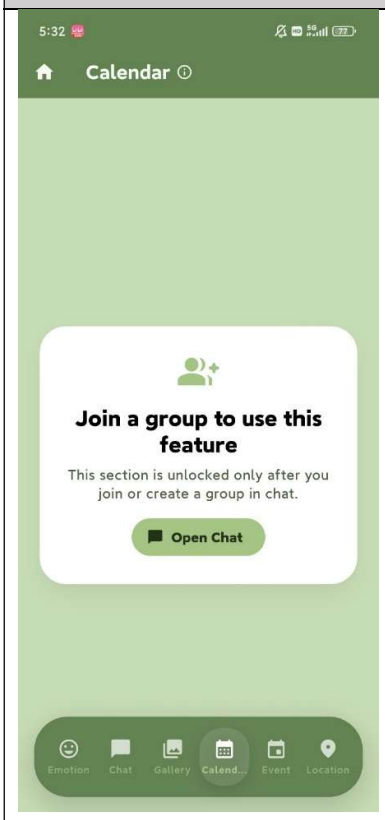
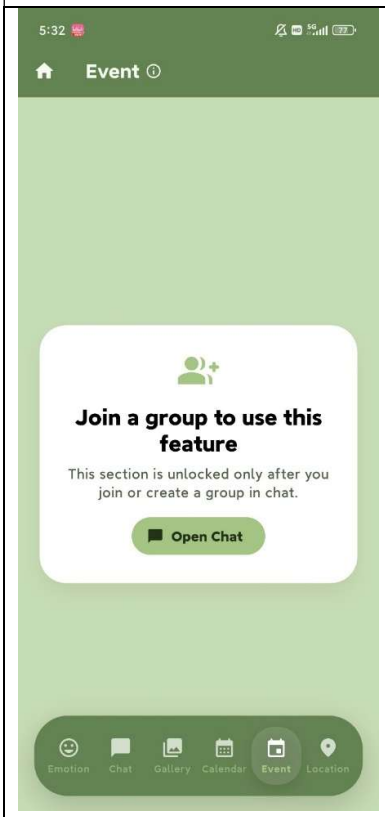
Screenshot	Explanation
	Figure 5.17 Screenshot of Login Page: - User can login with their username or Email. - If user forgot password, they can request for password renew, an Email will be sent for password renew.
	Figure 5.18 Screenshot of Sign up Page: - User creates account by entering a username, Email and password.

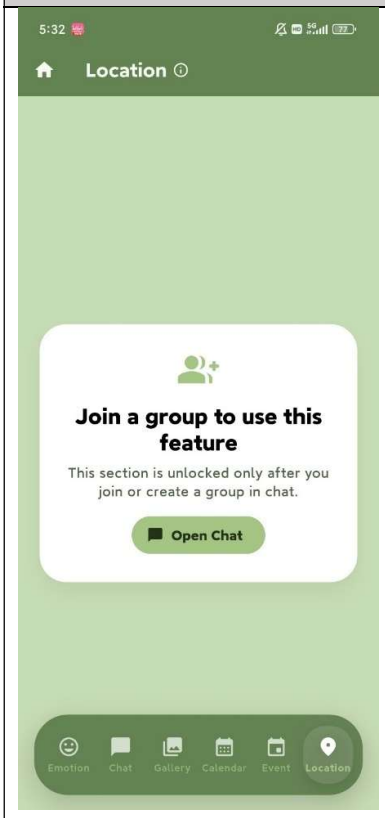
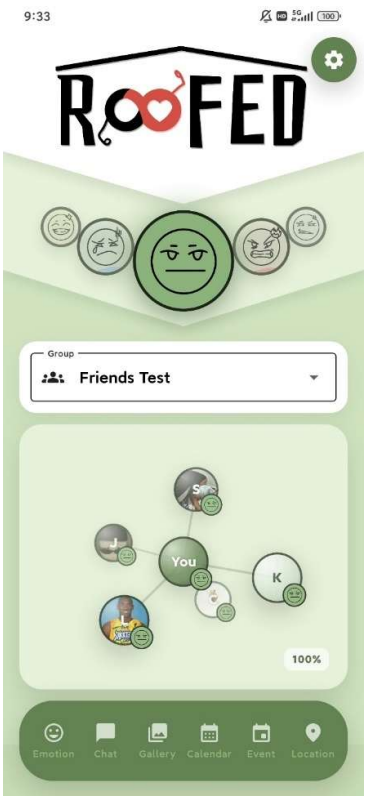
Screenshot	Explanation
	Figure 5.19 Screenshot of Profile Setup Page 1: - All profile setups except username are optional. - User can enter a short bio.
	Figure 5.20 Screenshot of Profile Setup Page 2: - User can choose their food preferences. - User can add their own choices for it.

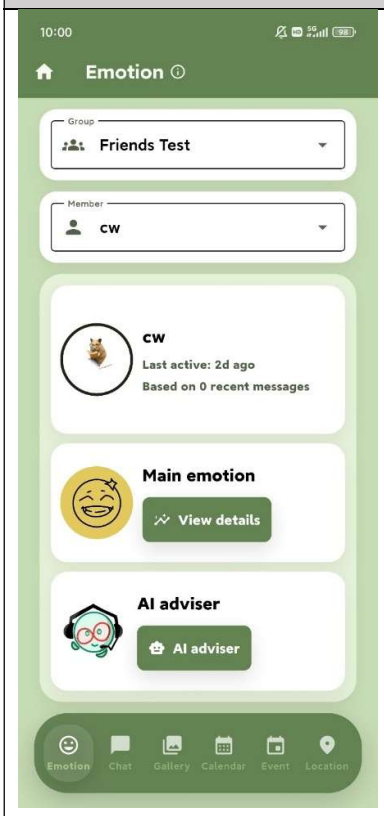
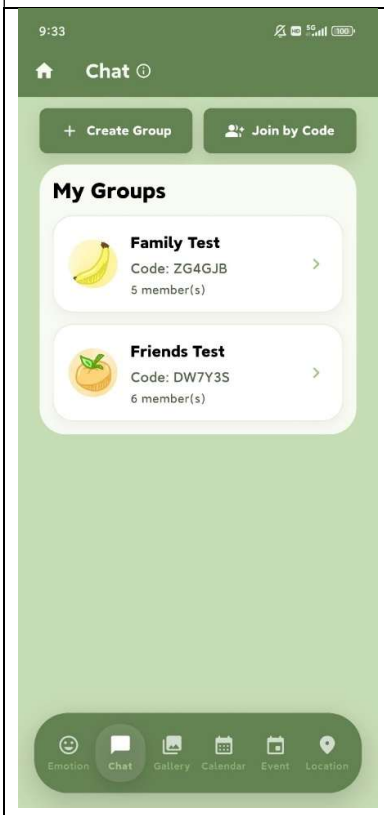
Screenshot	Explanation
	Figure 5.21 Screenshot of Profile Setup Page 3: - User can choose their interests and sports. - User can add their own choices for it.
	Figure 5.22 Screenshot of Profile Setup Page 4: - User can choose their lifestyle and habits. - User can add their own choices for it.

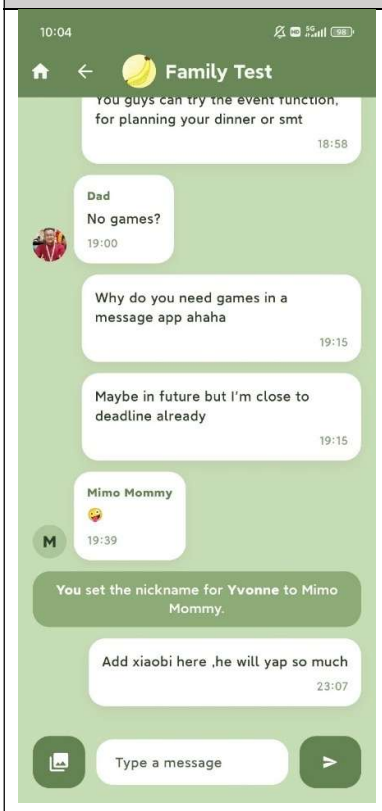
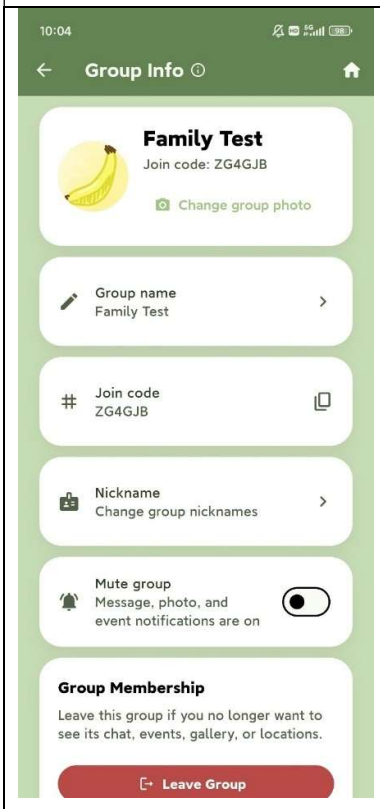
Screenshot	Explanation
	Figure 5.23 Screenshot of Home Page (No Group): - An instruction is shown to direct user to go to the Chat Page for unlocking further features. - A navigation menu is on the bottom. - The setting button is on the top right of the screen.
	Figure 5.24 Screenshot of Emotion Page (No Group): - An instruction is shown to direct user to go to the Chat Page for unlocking further features. - A navigation menu is on the bottom. - The title is clickable for instruction for this page.

Screenshot	Explanation
	Figure 5.25 Screenshot of Chat Page (No Group): - An instruction is shown to tell the user to join the group. - A navigation menu is on the bottom. - The title is clickable for instruction for this page.
	Figure 5.26 Screenshot of Gallery Page (No Group): - An instruction is shown to direct user to go to the Chat Page for unlocking further features. - A navigation menu is on the bottom. - The title is clickable for instruction for this page.

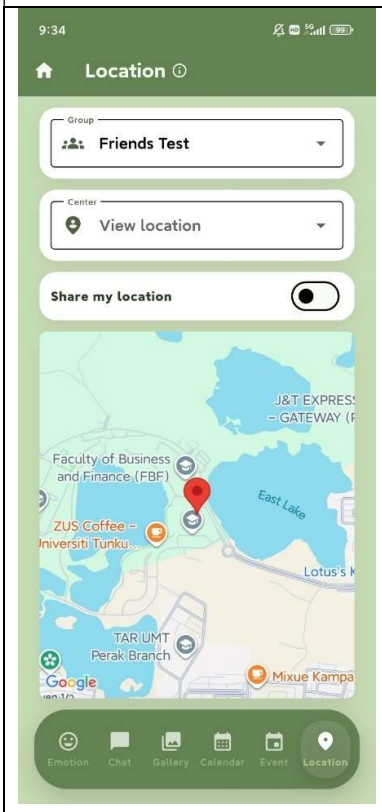
Screenshot	Explanation
	Figure 5.27 Screenshot of Calendar Page (No Group): - An instruction is shown to direct user to go to the Chat Page for unlocking further features. - A navigation menu is on the bottom. - The title is clickable for instruction for this page.
	Figure 5.28 Screenshot of Event Page (No Group): - An instruction is shown to direct user to go to the Chat Page for unlocking further features. - A navigation menu is on the bottom. - The title is clickable for instruction for this page.

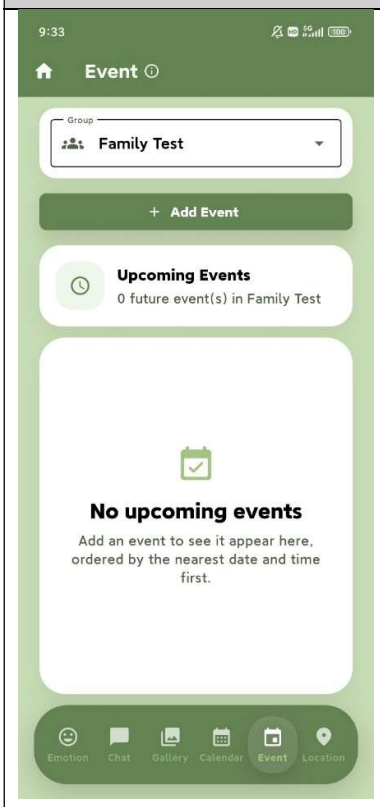
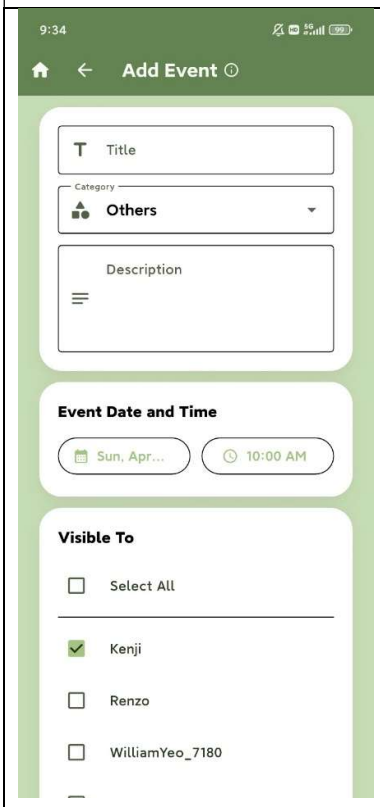
Screenshot	Explanation
	Figure 5.29 Screenshot of Location Page (No Group): - An instruction is shown to direct user to go to the Chat Page for unlocking further features. - A navigation menu is on the bottom. - The title is clickable for instruction for this page.
	Figure 5.30 Screenshot of Home Page: - A navigation menu is on the bottom. - In the middle section, it shows the user's main emotion. - In the bottom section, it shows the relationship diagram of selected group.

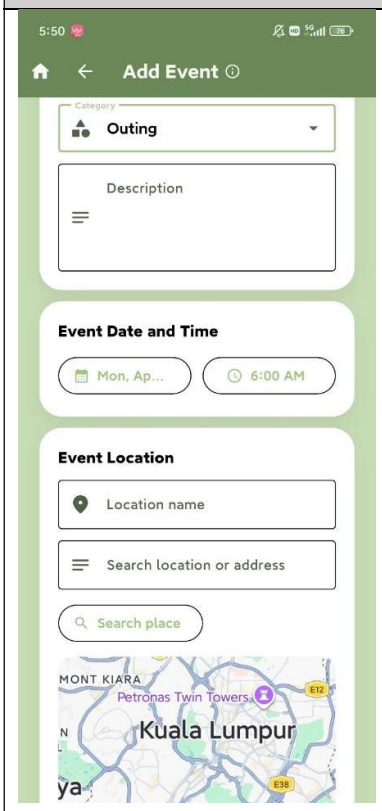
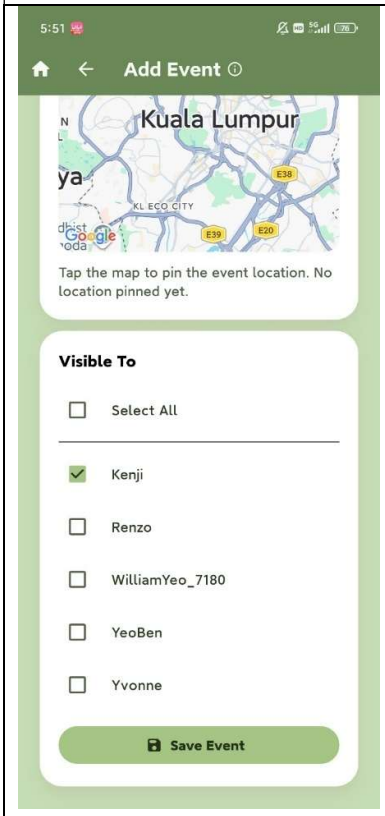
Screenshot	Explanation
	Figure 5.31 Screenshot of Emotion Page: - A navigation menu is on the bottom. - The title is clickable for instruction for this page. - User can view the main emotion and active status of selected member from selected group. - User can access to AI adviser Page.
	Figure 5.32 Screenshot of Chat Page: - A navigation menu is on the bottom. - The title is clickable for instruction for this page. - User can view and access to joined groups.

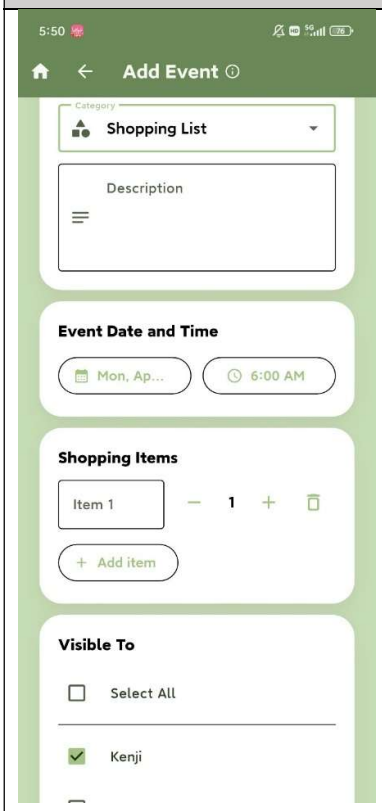
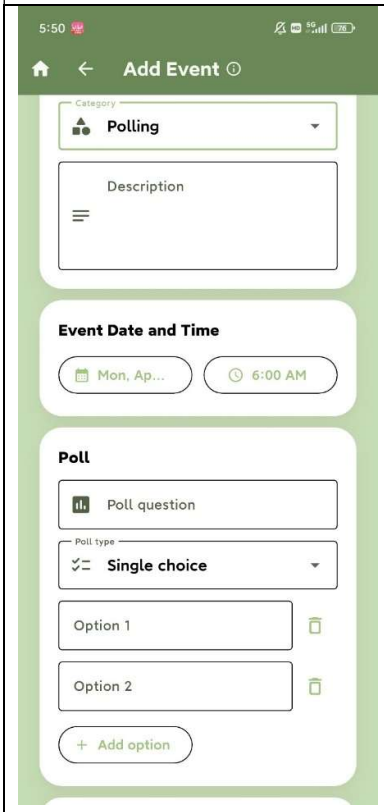
Screenshot	Explanation
	Figure 5.33 Screenshot of Chatroom Page: - User can send text or image messages. - User can view messages in chatroom space. - User can access to Group Info Page by clicking the title.
	Figure 5.34 Screenshot of Group Info Page: - The title is clickable for instruction for this page. - User can change group photo. - User can change group name. - User can copy the group code by clicking the code section. - User can change the nickname of all members in group. - User can mute notification of this group. - User can leave the group by clicking the “leave group” button.

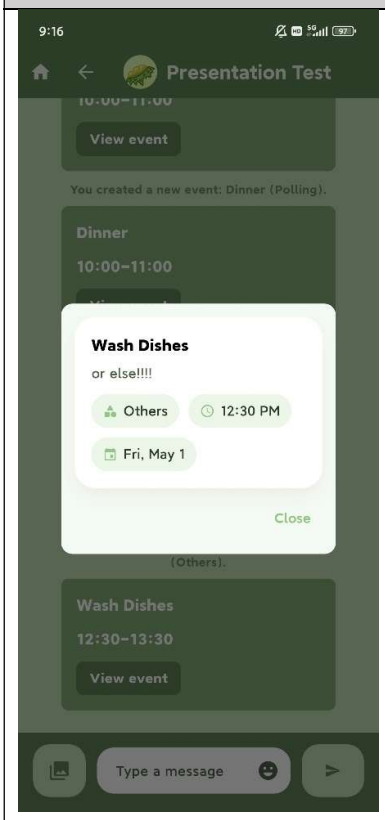
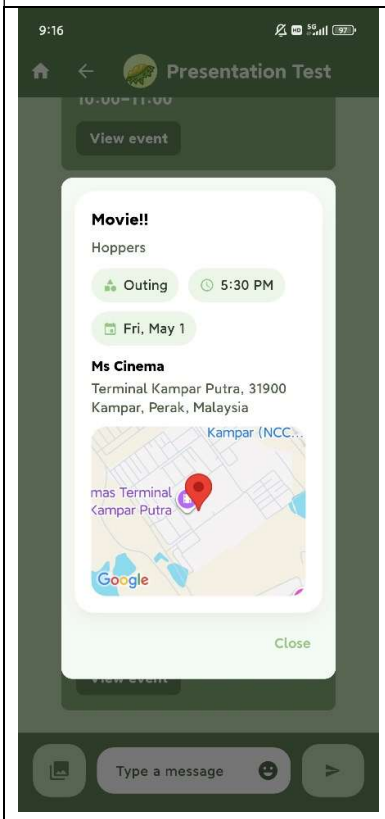
Screenshot	Explanation
	Figure 5.35 Screenshot of Calendar Page 1: - A navigation menu is on the bottom. - The title is clickable for instruction for this page. - User can view the chosen weekly calendar of selected group. - Events of the selected group will be shown on the calendar.
	Figure 5.36 Screenshot of Calendar Page 2: - A navigation menu is on the bottom. - The title is clickable for instruction for this page. - User can view the chosen weekly calendar of selected group. - Events of the selected group will be shown on the calendar.

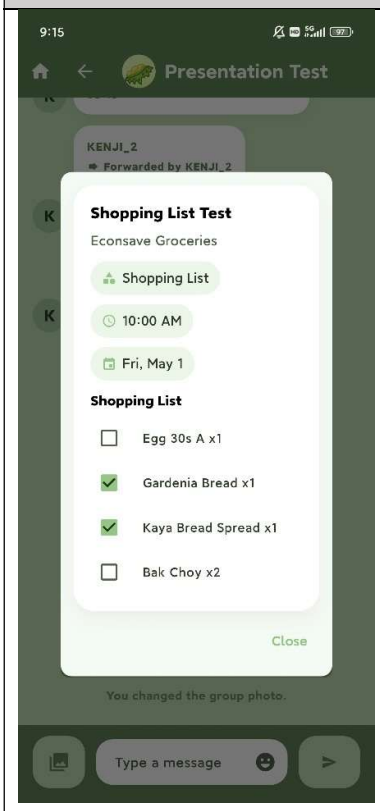
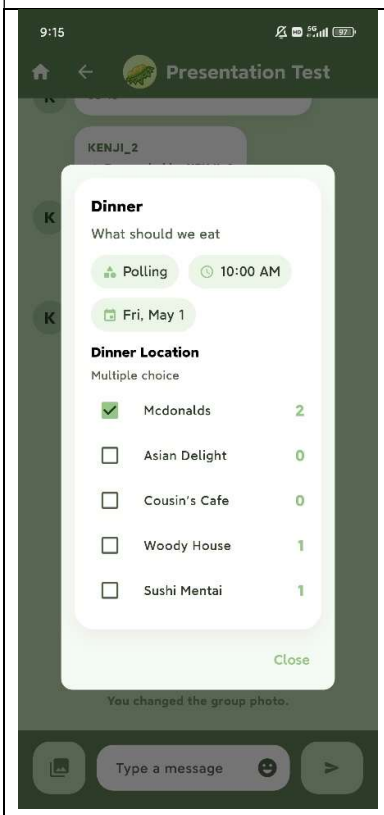
Screenshot	Explanation
	Figure 5.37 Screenshot of Gallery Page: - A navigation menu is on the bottom. - The title is clickable for instruction for this page. - User can view the photos sent in the selected group. - User can choose grid size: small (1x4), normal (1x2), big (1x1) - User can choose category: All, People, Bills, Other
	Figure 5.38 Screenshot of Location Page: - A navigation menu is on the bottom. - The title is clickable for instruction for this page. - User can share their location by the switch. - User can center their view of the map of the member in selected group by the drop-down menu. - User can zoom in and out by pinching the map. - User can view the map by swiping.

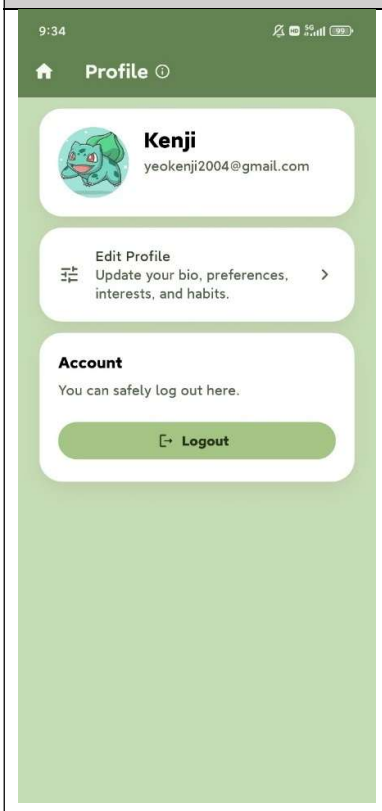
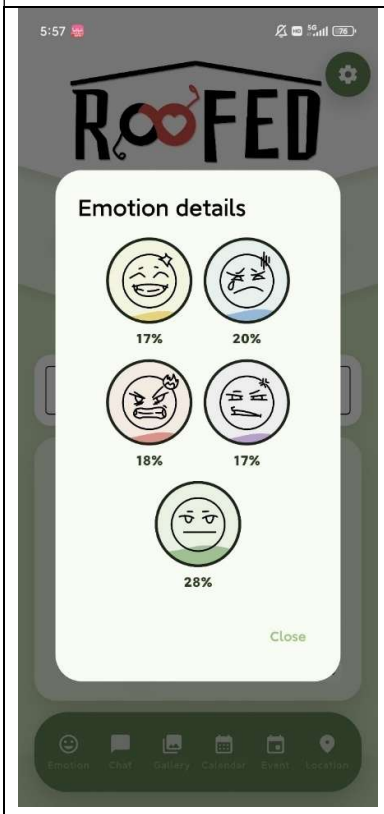
Screenshot	Explanation
	Figure 5.39 Screenshot of Event Page: - A navigation menu is on the bottom. - The title is clickable for instruction for this page. - User can add an event by clicking the button. - User can view upcoming events of selected group.
	Figure 5.40 Screenshot of Event Creation (Other) Page: - The title is clickable for instruction for this page. - User can enter the relevant information for events. - User can choose who the event will be visible to.

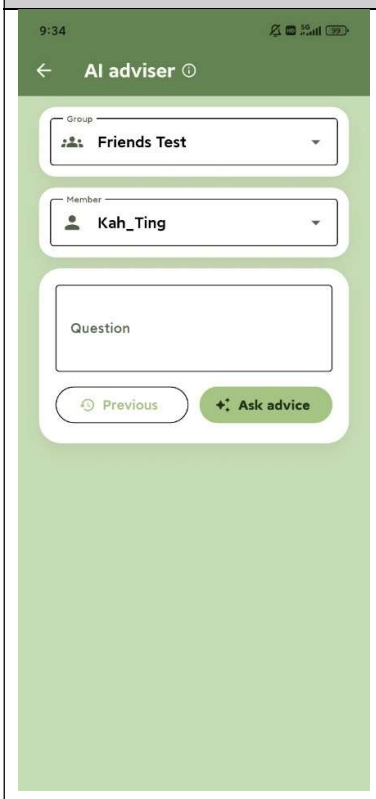
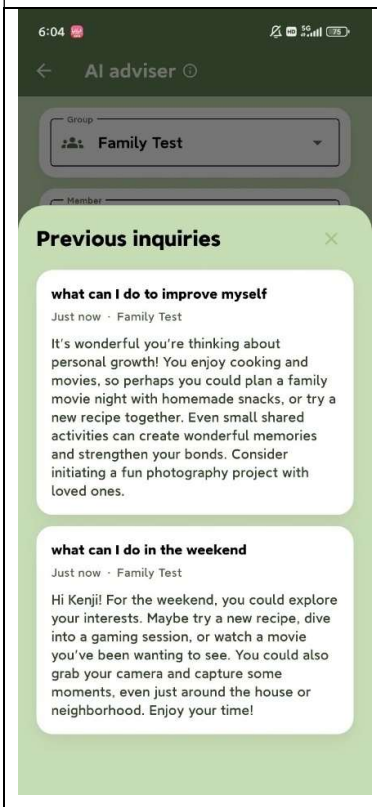
Screenshot	Explanation
	Figure 5.41 Screenshot of Event Creation (Outing) Page 1: - The title is clickable for instruction for this page. - User can enter the relevant information for events. - User can choose who the event will be visible to. - User can ping the location on the map.
	Figure 5.42 Screenshot of Event Creation (Outing) Page 2: - The title is clickable for instruction for this page. - User can enter the relevant information for events. - User can choose who the event will be visible to. - User can ping the location on the map.

Screenshot	Explanation
	Figure 5.43 Screenshot of Event Creation (Shopping List) Page: - The title is clickable for instruction for this page. - User can enter the relevant information for events. - User can choose who the event will be visible to. - User can write a list for shopping list. - User can add and remove item for the shopping list.
	Figure 5.44 Screenshot of Event Creation (Polling) Page: - The title is clickable for instruction for this page. - User can enter the relevant information for events. - User can choose who the event will be visible to. - User can add polling related functions, such as adding options for voting. - The poll can be either single choice or multiple choices.

Screenshot	Explanation
	Figure 5.45 Screenshot of Event Pop Up Window (Other): - User can view details of the event in the window.
	Figure 5.46 Screenshot of Event Pop Up Window (Outing): - User can view details of the event in the window.

Screenshot	Explanation
	Figure 5.47 Screenshot of Event Pop Up Window (Shopping List): - User can view details of the event in the window. - User can check off the shopping list.
	Figure 5.48 Screenshot of Event Pop Up Window (Polling): - User can view details of the event in the window. - User can vote for their choices. - User can view the number of votes for each choice.

Screenshot	Explanation
	Figure 5.49 Screenshot of Setting Page: - The title is clickable for instruction for this page. - User can edit their profile. - User can logout their account by clicking the log out button.
	Figure 5.50 Screenshot of Emotion Details pop up window: - Shown in Home Page and Emotion Page. - User can view the details of emotion.

Screenshot	Explanation
	Figure 5.51 Screenshot of AI Adviser Page: - The title is clickable for instruction for this page. - User can ask for advice for selected member in selected groups.
	Figure 5.52 Screenshot of History Inquiries Pop Up Window: - User can view the previous inquiries.

5.5 Implementation Issues and Challenges

Several implementation challenges were encountered during development. This is because Roofed combines many features that are normally separated in different apps in today's market. The main issues include UI problems, integration problems between Flutter, Firebase, Cloud Functions and backend services.

Table 5.5 represents some issues, with the full bug log available in the appendix.

Table 5.5 Some Implementation Issues and Challenges

Issue	**Affected Area**	**Problem**	**Implemented Fix / Report Explanation**
Notification delivery redesign	Notifications, Firebase, backend	Client-side notification handling was not reliable enough when app state changed or when users were not actively viewing a page.	Moved notification delivery toward FCM token storage and backend-triggered delivery using Cloud Functions. This centralised notification logic and improved reliability.
Calendar history retention	Event and Calendar	Past events could disappear from calendar history if old tasks were removed automatically.	Stopped automatic deletion of old task records. Event page can still filter future items, while Calendar page can retain history.
Emotion no-data state	Home and Emotion dashboard	New or inactive members could show leftover emotion values even without recent messages.	Changed no-data handling so members with no recent messages display zero or neutral fallback values instead of old data.
Shortform normalization	Emotion classifier backend	Informal Malaysian shortforms, slang, and abbreviations reduced emotion-classification quality.	Added reusable shortform replacement dictionary so training and prediction use consistent normalized text.

Issue	Affected Area	Problem	Implemented Fix / Report Explanation
Raw-text intensity preservation	Emotion classifier backend	Heavy normalization could weaken useful signals such as uppercase emphasis, repeated punctuation, and elongated words.	Added raw-text intensity features such as uppercase ratio, exclamation count, question mark count, repeated punctuation, repeated words, and elongated words.
Emotion API response parsing	Chat and emotion integration	Different backend response shapes could break label or score extraction in the app.	Improved response parsing to normalize label formats, score keys, confidence values, and percentage values.
Live location stability	Location and maps	Frequent updates could destabilize map focus and cleanup when the page lifecycle changed.	Limited aggressive camera movement and improved async cleanup when stopping location sharing or leaving the page.
Group chat query loading	Chat and Firestore	Message loading could fail due to Firestore query/index behaviour.	Adjusted query logic and sorted messages locally after retrieval where necessary.
Anonymous profile safety	Profile and authentication	Anonymous accounts could trigger profile write failures under backend permission constraints.	Added guarded initialization, readable warnings, and fallback labels to avoid crashes.
Large file complexity	Code maintainability	Some feature files became large because UI, state handling, and data operations were implemented in the same page file.	Identified future refactoring need. Shared services and planner widgets were used to reduce repeated logic where possible.
Public backend access during testing	Cloud Run backend	The backend was publicly reachable during development and testing.	Acceptable for prototype testing, but future production work should add Firebase token verification or another access-control layer.

5.6 Post-Testing Enhancements and Major Updates

After the initial implementation, Roofed was distributed to several testers for testing and acquiring feedbacks. This stage turned out to be crucial because several issues have been identified after the testing stage, which would otherwise be overlooked by the developer. Under this section, the improvements areas would be showed and explained, with the support of updated screenshots.

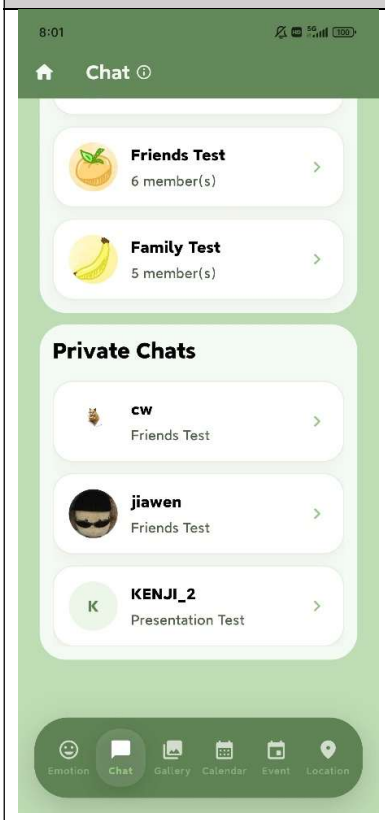
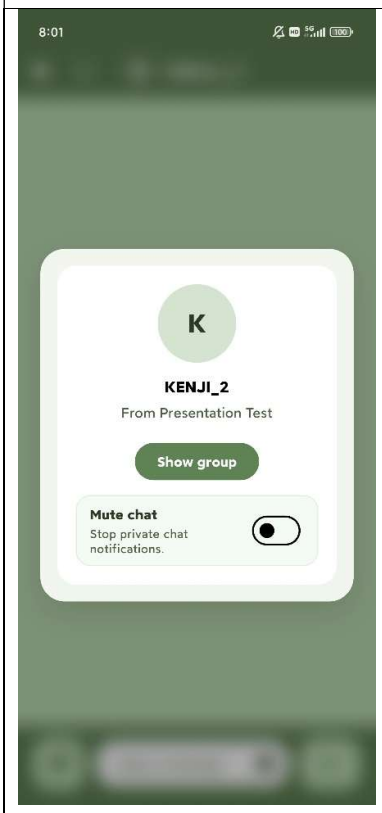
5.6.1 Chat System

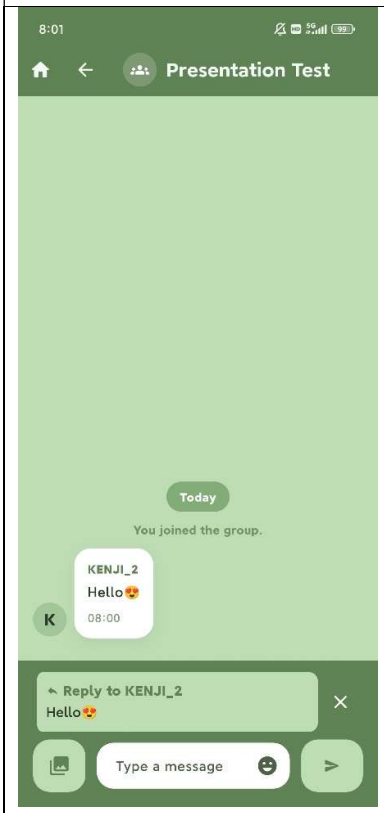
Table 5.6 demonstrates post-testing improvements for chat system.

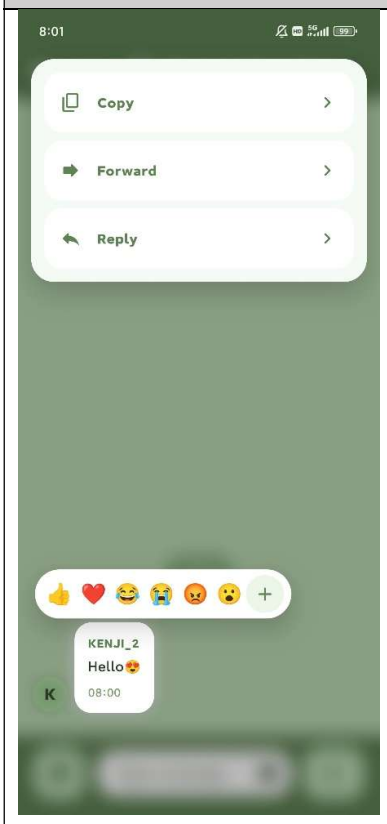
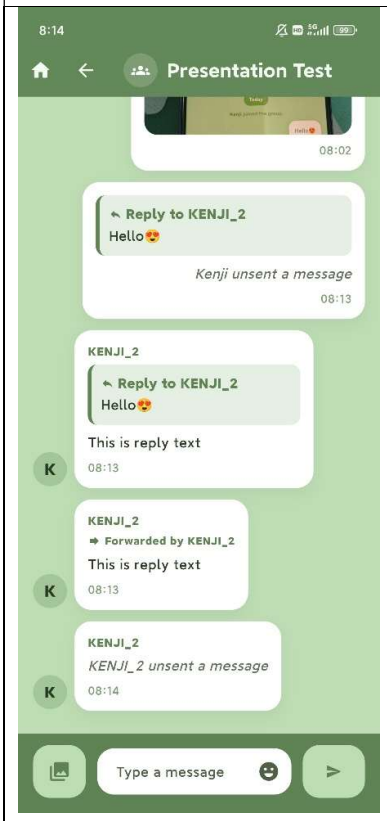
Table 5.6 Post-Testing Improvements for Chat System

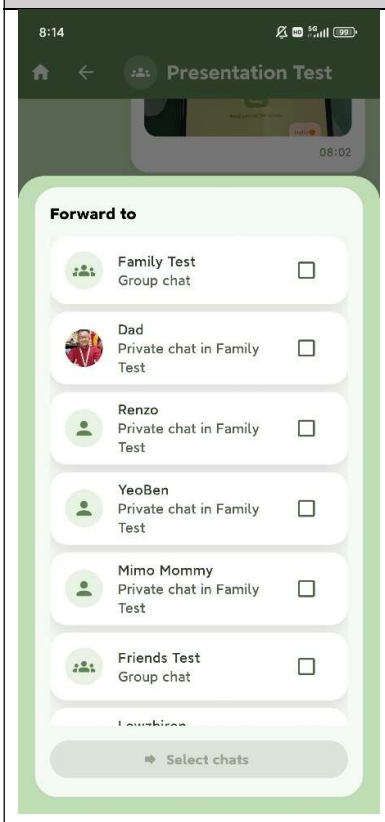
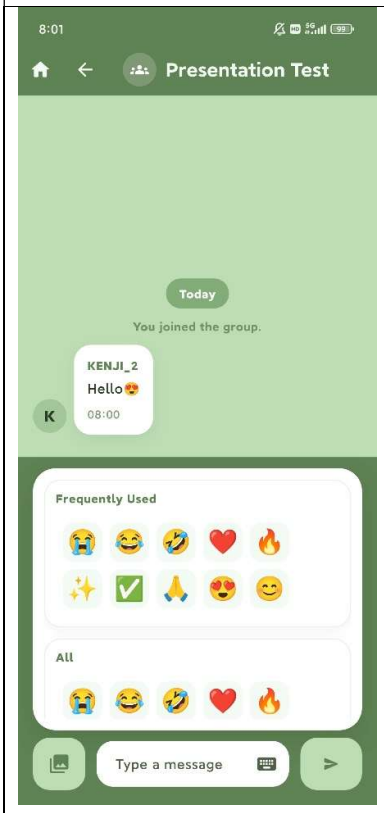
Issue / Limitation	Improvement Implemented	Impact
Chat flickering and unstable scroll	Stabilized Firestore stream and bottom-anchored behaviour	Smooth chat experience
Limited interaction features	Added reply, forward, unsend, emoji, reactions	Improved usability
No private messaging separation	Introduced dedicated private chat system	Better organization

Figures 5.53 – 5.60 shows the screenshots of the app pages after update related to chat system.

Screenshot	Improvements
	Figure 5.53 Screenshot of Chat Page (Updated) 1: - User can view and access to private chats. - All members from joined groups will be shown here. - Nickname from the groups given to the member will be shown instead of their username.
	Figure 5.54 Screenshot of Profile Pop Up Window (In Private Chat): - User can view info of member profile. - User can mute the private chat for that member. - User can access to the group chat conveniently by clicking the "Show group" button.

Screenshot	Improvements
	Figure 5.55 Screenshot of Profile Pop Up Window (In Group Chat): - User can view info of member profile. - User can access to the private chat conveniently by clicking the "Chat privately" button.
	Figure 5.56 Screenshot of Group Chatroom Page (Updated) 1: - User can reply to message by swiping right (for other members' messages) or swiping right (for users' own messages).

Screenshot	Improvements
	Figure 5.57 Screenshot of Message Menu: - User can access to this menu by holding any messages. - User can choose to either copy, forward or reply in this state. - User can react to the selected message with emojis.
	Figure 5.58 Screenshot of Group Chatroom Page (Updated) 2: - Examples of reply text, forward text and unsend text are shown.

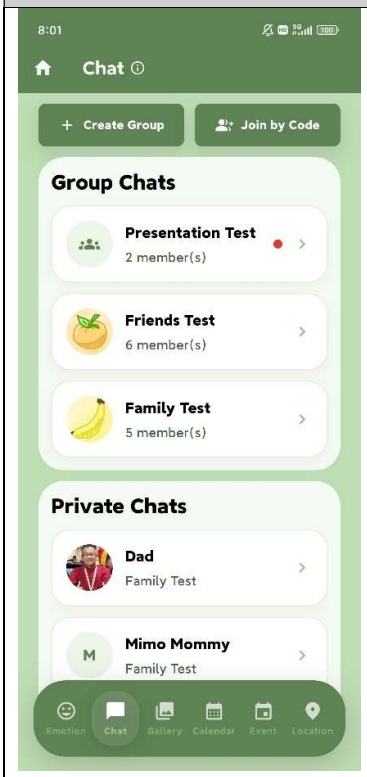
Screenshot	Improvements
	Figure 5.59 Screenshot of Forward Menu: - User can choose who to forward message (multiple choices available).
	Figure 5.60 Screenshot of Emoji Keyboard: - User can type emoji conveniently by the emoji keyboard. - It will show frequently used emojis by user. - When choose an emoji, it will be entered inside the input bar instead of directly sending out.

5.6.2 Notification System

Table 5.7 demonstrates post-testing improvements for notification system. Figure 5.61 shows the screenshot of the app pages after update related to notification system.

Table 5.7 Post-Testing Improvements for Notification System

Issue / Limitation	Improvement Implemented	Impact
Notifications failed when inactive	Migrated to FCM with Cloud Functions	Reliable push delivery
Generic notifications	Added reply and mention alerts	More meaningful alerts
No reminders	Added event reminders before 1 day and 1 hour	Better planning
New message notification in Chat Page	Unread message will show a red dot	Notifying User that there are new unread messages.

Screenshot	**Improvements**
	Figure 5.61 Screenshot of Chat Page (Updated) 2: - User can see chatroom that have unread messages via the red dot notification.

5.6.3 Emotion Classifier

Table 5.8 demonstrates post-testing improvements for emotion classifier.

Table 5.8 Post-Testing Improvements for Emotion Classifier

Issue / Limitation	Improvement Implemented	Impact
Poor slang handling	Added short form normalization	Better understanding
Lost emotion intensity	Added intensity features	Improved accuracy
API inconsistency	Improved parsing logic	Stable integration

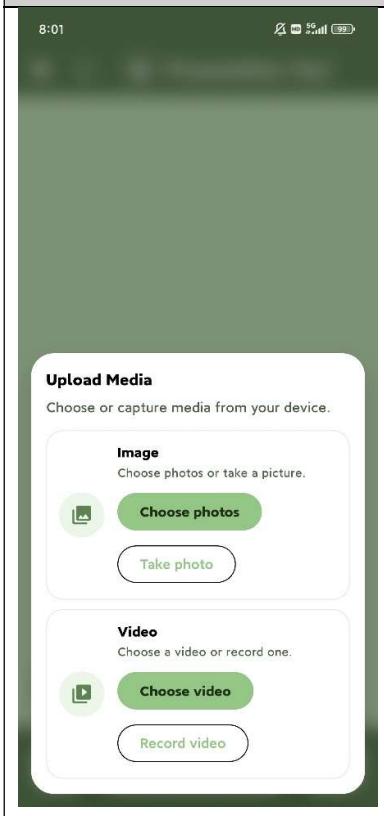
5.6.4 Media Handling

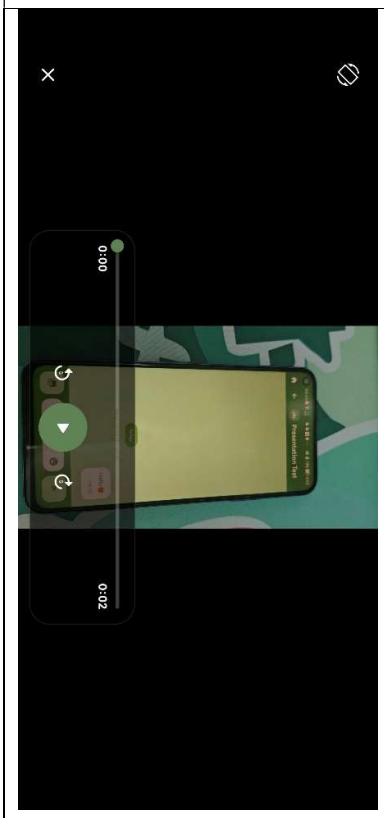
Table 5.9 demonstrates post-testing improvements for media handling.

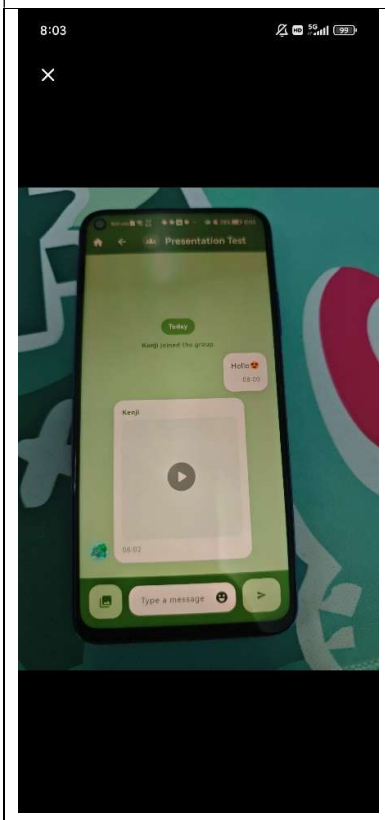
Table 5.9 Post-Testing Improvements for Media Handling

Issue / Limitation	Improvement Implemented	Impact
Preview issues	Improved layout and preview	Stable display
Limited interaction	Added fullscreen view for photo	Better UX
Slow loading	Optimized previews	Faster loading
No Video related feature	Added video related feature	Another type of media can be convenient in daily communication
Cannot take photo or record video	Allow for live camera input instead of only upload	Convenient and efficient

Figures 5.62 – 5.67 shows the screenshots of the app pages after update related to media handling.

Screenshot	Improvements
	Figure 5.62 Screenshot of Media Upload Menu: - User can choose to upload photo or video. - User can choose to take photo or record video by camera.
	Figure 5.63 Screenshot of Video Preview Pop Up Window: - User can download video. - User can play, reverse or skip video. - User can control video by timeline.

Screenshot	Improvements
	Figure 5.64 Screenshot of Video Fullscreen View (Vertical): - User can have better view of video. - User can exit this view by clicking X button. - User can rotate to horizontal view by clicking the upper right button.
	Figure 5.65 Screenshot of Video Fullscreen View (Horizontal): - User can have better view of video. - User can exit this view by clicking X button. - User can rotate to vertical view by clicking the upper right button.

Screenshot	Improvements
	Figure 5.66 Screenshot of Gallery Page (Updated): - User can view video. - Added category for video.
	Figure 5.67 Screenshot of Photo Fullscreen View: - User can have better view of video. - User can exit this view by clicking X button. - User can zoom by either pinching or double clicking the photo.

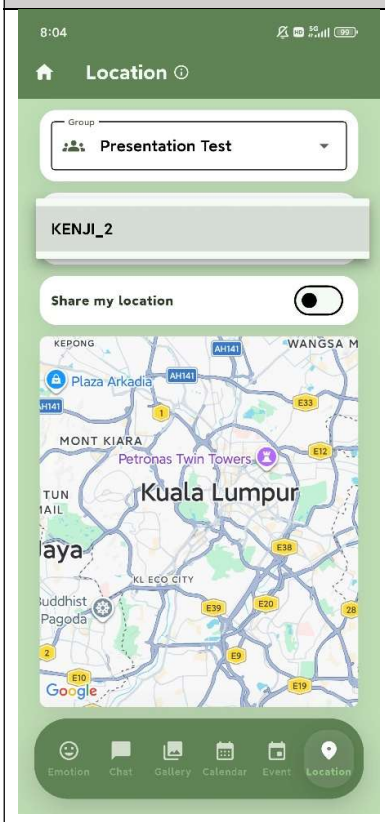
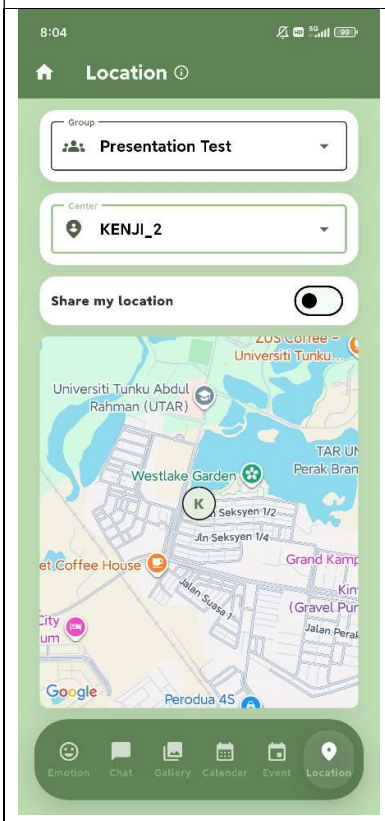
5.6.5 Location System

Table 5.10 demonstrates post-testing improvements for location system.

Table 5.10 Post-Testing Improvements for Location System

Issue / Limitation	Improvement Implemented	Impact
Map flickering	Optimized rendering	Stable map
Unknown member centred	Correctly display member nickname or username	Clear display
Unclear markers	Used profile-picture markers	Better visibility
Overlapping users	Improved prioritization	Clear display

Figures 5.68 and 5.69 shows the screenshots of the app pages after update related to location system.

Screenshot	Improvements
	Figure 5.68 Screenshot of Location Page (Updated) 1: - User can only see members that are currently sharing their location instead of showing all members. - The nickname or username of member are shown correctly in the drop-down menu.
	Figure 5.69 Screenshot of Location Page (Updated) 2: - Member profile picture will be shown in the map instead of the defaulted location marker.

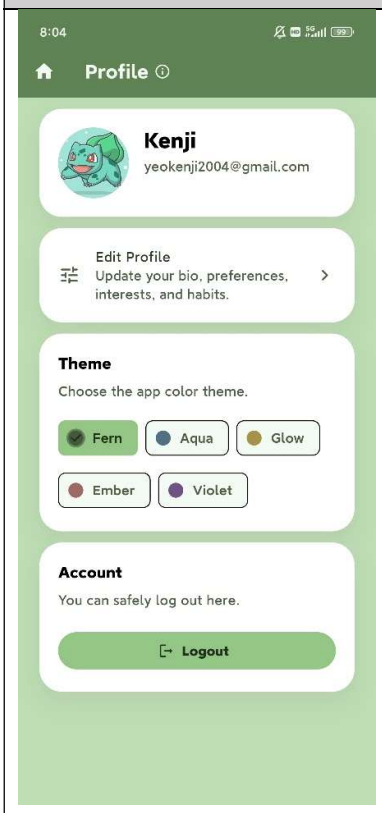
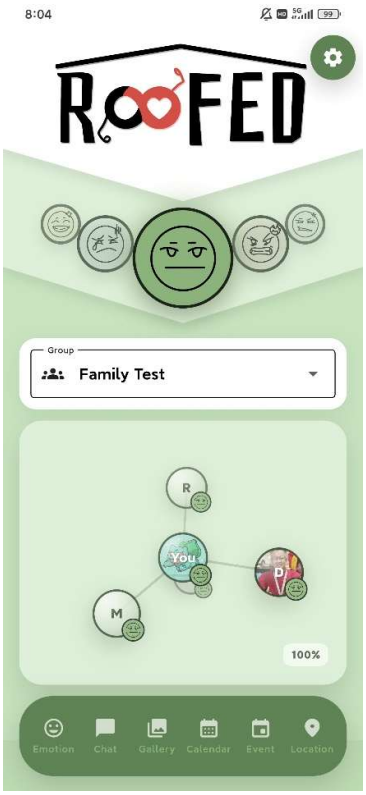
5.6.6 User Interface

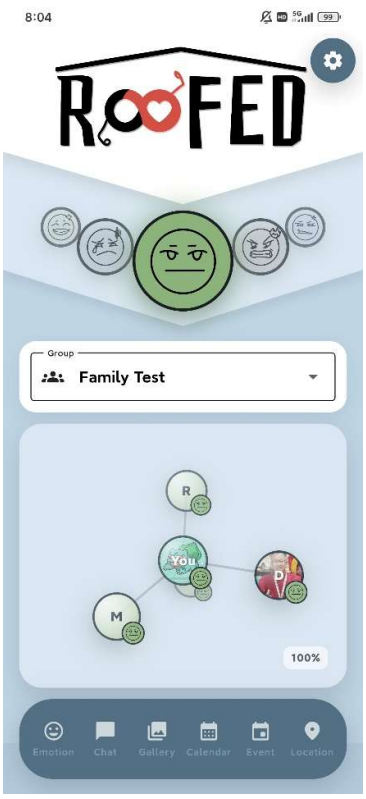
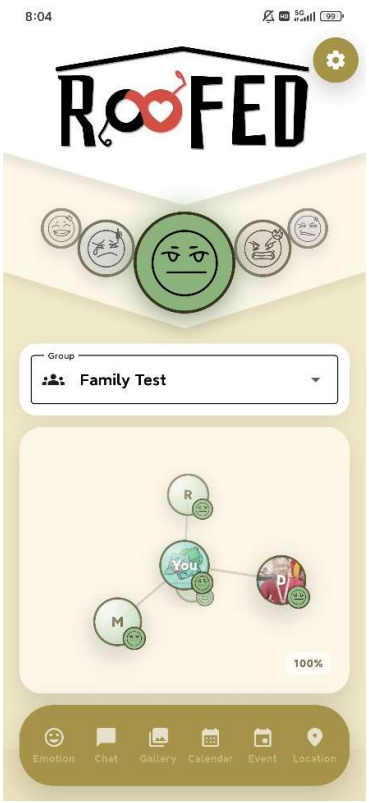
Table 5.11 demonstrates post-testing improvements for user interface.

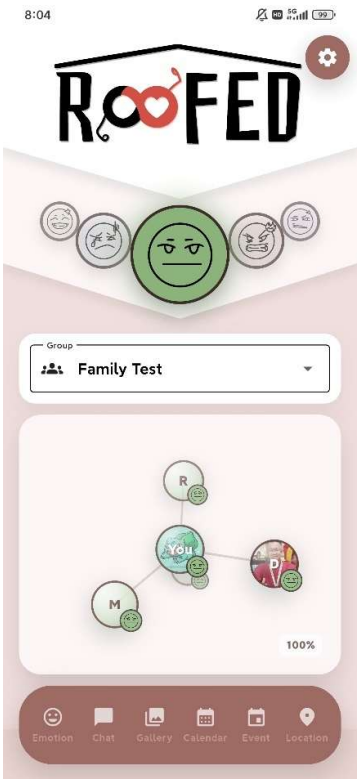
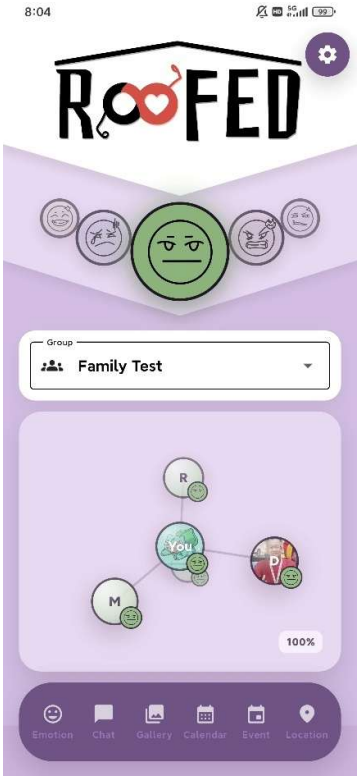
Table 5.11 Post-Testing Improvements for User Interface

Issue / Limitation	Improvement Implemented	Impact
Inconsistent UI	Standardized design	Better consistency
Redundant elements	Removed unnecessary UI	Cleaner interface
Overflow issues	Improved spacing	Responsive layout
Limited customization	Added Theme options	Provide customization

Figures 5.70 - 5.75 shows the screenshots of the app pages after update related to location system.

Screenshot	Improvements
	Figure 5.70 Screenshot of Setting Page (Updated): - User can choose one of the five themes for the app: Fern (defaulted), Aqua, Glow, Ember and Violet.
	Figure 5.71 Screenshot of Theme Examples (Fern):

Screenshot	Improvements
	Figure 5.72 Screenshot of Theme Examples (Aqua):
	Figure 5.73 Screenshot of Theme Examples (Glow):

Screenshot	Improvements
	Figure 5.74 Screenshot of Theme Examples (Ember):
	Figure 5.75 Screenshot of Theme Examples (Violet):

5.7 Concluding Remark

The implementation of Roofed demonstrates the project is a fully working mobile app rather than a prototype. The completed prototype successfully combined authentication, profile management, group communication, gallery support, event and calendar management, live location sharing, notification delivery, emotion classification and AI-assisted family advice within one coordinated application environment.

The implementation procedure also shows challenging parts, such as connecting text messages to the external emotion classification system, notification delivery system and more. Wiring UI to functions, to API, to database are complicated and long tasks. These issues were gradually resolved through iterative debugging, redesign and post-testing refinement.

In conclusion, the implementation stage provided a solid technical foundation for the later evaluation of the system. By the end of this stage, the major modules of Roofed had been successfully connected and were operating coherently and stably. The completed implementation therefore justified proceeding to Chapter 6, where the usability, stability, and practical value of the developed system were evaluated in detail.

CHAPTER 6

System Evaluation and Discussion

This chapter evaluates Roofed. The evaluation can be divided into functional testing, user testing, emotion classifier testing, project challenges and objective evaluation. The testing for Roofed was planned around two sections. Firstly, the application testing was carried out through small-group user testing. Then, the emotion classifier was evaluated by a set of test case, then performance metrics such as accuracies of it were documented.

6.1 System Testing and Performance Metrics

System testing and performance metrics are used to determine whether Roofed operates correctly and whether the app is suitable for family or close-group usage.

The main testing areas are listed in Table 6.1 below.

Table 6.1 Main Testing Areas

Testing Area	Metric / Evidence	Purpose of Evaluation
Functional correctness	Pass or fail observation for each major feature.	To check whether users can complete main tasks such as login, group creation, chat, event creation, and location sharing.
Usability	Google Form rating scale and written feedback.	To evaluate whether the interface, dropdowns, buttons, and navigation flow are understandable.
Stability	Crash, freeze, loading error, and bug reports by tester	To identify whether the prototype is stable enough for normal short-term testing.
Feature usefulness	User rating and comments for each feature.	To understand which features are useful for family or close-group communication.

Testing Area	Metric / Evidence	Purpose of Evaluation
Notification behavior	Tester feedback and direct observation.	To check whether chat, event, and reminder notifications are useful and sufficiently reliable.
Emotion classifier result	Accuracy, precision, recall, F1 score, confusion matrix, and ROC AUC.	To evaluate the five-class NLP classifier used by the emotion display module.

6.2 Testing Setup and Result

Two small testing groups were initiated to gather feedbacks. The details of the groups are illustrated in Table 6.2 below.

Table 6.2 List of Testing Group

Testing Group	Number of Members	Relationship Type	Testing Purpose
Friend group	6 members including the developer	Friends / close peer group	To test group chat, event planning, casual message emotion detection, image sharing, AI advice, and general usability among peer users.
Family group	4 members including the developer	Family group	To test whether the app concept is suitable for family communication, shared planning, location sharing, and emotional-awareness features.

The testing period lasted for five days. During this period, testers were instructed to install and try functions in the app. There were no complicated and forceful testing script for the testing groups. After the testing period ended, they are requested to complete a Google Form to report their user experience, rate features usability and describe issues they faced during their experience. The Google Form questions and response summaries can be referred in the appendix.

6.2.1 Quantitative Questionnaire User Feedback

A total of 8 Google Form responses were collected after the testing phase, where five of them were from the friend group and three came for family members. Among the respondents, most of them were between 18 to 24 years old, while one responder is below 18 and another is aged 45 or above. Therefor, this result was useful because it encompasses several age groups.

The overall feedback showed that the prototype was understandable and generally stable. All respondents did not face any issues for the installation and setup steps. No problems were reported for group creation or joining too, which indicates that the user entry flow and logic were sufficiently mature. Chat and event features received generally positive results, all related functions mostly work for all respondents. However, the emotion and AI-related features showed a more mixed outcome. The user flow shows no issue, but mainly the output and display from them were not convincing enough. Notification features were acceptable but not perfect, where on respondent noted that they are delayed notification. Table 6.3 lists the average ratings derived from the Google Form responses while Table 6.4 summarises the questionnaire results.

Table 6.3 Average Ratings Derived from The Google Form Responses

Rating Item	Average Score	Number of Ratings
Login / sign up / guest access clarity	4.50 / 5	8
Group join / code clarity	4.75 / 5	8
Chat and group page clarity	4.63 / 5	8
Emotion member clarity	4.00 / 5	8
Dominant emotion visibility	4.75 / 5	8
Emotion result reasonableness	3.75 / 5	8
Emotion page clarity	3.75 / 5	8
Event creation form clarity	4.50 / 5	8
Weekly calendar clarity	4.63 / 5	8
Event and calendar page clarity	4.38 / 5	8
AI reply relevance	3.88 / 5	8
AI reply clarity	4.12 / 5	8

Table 6.4 Summary of Selected Questionnaire Results

Feature / Observation	Most Common Result	Summary
Installation and app opening	Yes	8 (100.0%)
Account/profile setup problem	No problem	8 (100.0%)
Create or join group	Yes	8 (100.0%)
Chat messages sent/received properly	Yes	7 (87.5%)
Chat messages sometimes delayed	Sometimes delayed	1 (12.5%)
Image upload in chat	Yes	8 (100.0%)
Emotion page loaded properly	Yes	7 (87.5%)
Event creation successful	Yes	6 (75.0%)
Event appears correctly in Event page	Yes	7 (87.5%)
Event appears correctly in Calendar page	Yes	6 (75.0%)
Able to open AI family advice page	Yes	6 (75.0%)
Able to ask question and receive reply	Yes	5 (62.5%)
Received notifications	Yes	7 (87.5%)
Notification delayed	Yes	1 (12.5%)
Experienced bug/crash/freeze/wrong display	Yes	2 (25.0%)
Would continue using app if improved and completed	Maybe	5 (62.5%)

6.2.2 Qualitative Questionnaire User Feedback

Table 6.5 lists the main qualitative themes derived from tester comments.

Table 6.5 Main Qualitative Themes Derived from Tester Comments

Qualitative Theme	Summary
Most positively mentioned areas	Clean user interface; clear group chat; event planning; gallery browsing; location viewing; emotion detector concept.
Most requested improvements	AI adviser should provide more specific and complete replies; notification cleanup; clearer emotion explanation or purpose; more interaction in chat.
Requested additions from testers	Private messaging; replies; emoji; video or file upload; more background or theme customisation.
Location-specific observation	Location viewing was mentioned as a liked feature, while hidden member names in the location page were reported as a bug.
Important observation for later updates	Several user requests were later addressed in the major update, especially private chats, replies, emoji support, media improvements, and location-label improvements.

As mentioned before, for the emotion related features, several comments questioned the purpose of it and its accuracy at predicting correct emotion. For AI family advisor, some testers commented that the output should be more specific and complete rather than broad and vague suggestions. These reviews shows that the modules were technically functional but not yet sufficient in practical usefulness.

6.2.3 Bug Report from User Feedback

Although bug related reported are limited, but they still remained useful in helping the developer identify underlying issues. Only two main problems were reported from the testing group.

Table 6.6 shows the bugs reported in the Questionnaire.

Table 6.6 Bugs Report Derived from Tester Comments

No	Bugs
1	Chatroom position anchoring issues, where sometimes it will not redirect the user to the correct place (anchor user to older message instead of newest message)
2	Location centering member drop down menu not showing actual group members nicknames or username.

These comments and advices gathered from the questionnaire were ultimately adopt for the major updates mentioned in the later part of Chapter 5.

6.2.4 Emotion Classifier Testing Setup and Result

A test file containing 20 test cases with label are used to test the model. Table 6.7 shows the Test Cases and its result. With this result, multiple performance metric will be done.

Table 6.7 List of Test Cases and Result

No.	Text	Actual	Predicted	Correct	Confidence
1	hi guys remember arrive early for tmr debate	neutral	neutral	Yes	0.7009
2	thanks everyone for your contributions, glad we got this done and submitted	happy	happy	Yes	0.8053
3	waaaaaat	angry	happy	No	0.2773
4	grrrr ur so bad	angry	angry	Yes	0.3751
5	r we deadass	annoyed	annoyed	Yes	0.3631
6	OK DUDE	angry	angry	Yes	0.9204
7	maaan that was unfortunate...	sad	sad	Yes	0.6061
8	wait ya	neutral	neutral	Yes	0.6986
9	please let us know how can we help you	neutral	neutral	Yes	0.5021
10	walaoo the line is so long	annoyed	annoyed	Yes	0.4704
11	pukimak u jibaikia	angry	angry	Yes	0.5717
12	so r we meeting next week	neutral	neutral	Yes	0.4736
13	thanksss	happy	happy	Yes	0.4829
14	THank You sooo much	happy	happy	Yes	0.5814
15	pls check ur part PROPERLY!	angry	angry	Yes	0.7487
16	oii reply leh	annoyed	annoyed	Yes	0.4182
17	i alr wait for 30min, still haven't done meh	annoyed	annoyed	Yes	0.5149
18	i got rejected...	sad	sad	Yes	0.7407
19	deal, thanks for the offer	happy	happy	Yes	0.9115
20	when is the time for the test	neutral	neutral	Yes	0.7577

6.2.4.1 Accuracy

The accuracy is a measure of how accurate the model is, its function:

$$Accuracy = \frac{Correct\ Predictions}{Total\ Predictions}$$

In this case, there are 19 correct predictions out of 20 total predictions:

$$Accuracy = \frac{19}{20} = 0.95$$

6.2.4.2 Confusion Matrix

The confusion matrix is a tabular visualization of correlation between the predictions of the model and the actual class labels of the data points.

Table 6.8 represents the confusion matrix.

Table 6.8 Confusion Matrix

	Actually angry	**Actually annoyed**	**Actually happy**	**Actually neutral**	**Actually sad**
Predicted angry	4	0	0	0	0
Predicted annoyed	0	4	0	0	0
Predicted happy	1	0	4	0	0
Predicted neutral	0	0	0	5	0
Predicted sad	0	0	0	0	2

Then, the result can be further calculated in the following rates:

$$True\ Positive\ Rate\ (TPR) = \frac{True\ Positive}{Actual\ Positive}$$

$$True\ Negative\ Rate\ (TNR) = \frac{True\ Negative}{Actual\ Negative}$$

$$False\ Positive\ Rate\ (FPR) = \frac{False\ Positive}{Actual\ Negative}$$

$$False\ Negative\ Rate\ (FNR) = \frac{False\ Negative}{Actual\ Positive}$$

Table 6.9 and 6.10 show the calculations of TPR, TNR, FPR and FNR values.

Table 6.9 Calculations of TPR, TNR, FPR, FNR values 1

Label	True Positive	False Positive	True Negative	False Negative	Actual Positive	Actual Negative
angry	4	0	15	1	5	15
annoyed	4	0	16	0	4	16
happy	4	1	15	0	4	16
neutral	5	0	15	0	5	15
sad	2	0	18	0	2	18

Table 6.10 Calculations of TPR, TNR, FPR, FNR values 2

Label	TPR	TNR	FPR	FNR
angry	0.8000	1.0000	0.0000	0.2000
annoyed	1.0000	1.0000	0.0000	0.0000
happy	1.0000	0.9375	0.0667	0.0000
neutral	1.0000	1.0000	0.0000	0.0000
sad	1.0000	1.0000	0.0000	0.0000
Macro average	0.9600	0.9875	0.0125	0.0400
Weighted average	0.9500	0.9875	0.0125	0.0500

6.2.4.3 Precision and Recall

Precision represents how many times the positive prediction is actually positive. The closer the value is towards 1, it shows that the model is able to classify well between correct and incorrect label. The equation is as below:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Recall is the ratio of true positive to all the positives in ground truth. The closer the value is towards 1, it shows that the model is able to classify well between correctly and incorrectly labelling of dataset. The equation is as below:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Table 6.11 represents the calculations of precision and recall values.

Table 6.11 Calculations of Precision and Recall values

Label	Precision	Recall
angry	1.0000	0.8000
annoyed	1.0000	1.0000
happy	0.8000	1.0000
neutral	1.0000	1.0000
sad	1.0000	1.0000
Macro average	0.9600	0.9600
Weighted average	0.9600	0.9500

6.2.4.4 F_1 Score

The score is the harmonic mean of precision and recall. The equation is shown below:

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

Table 6.12 represents the calculations of F_1 Scores.

Table 6.12 Calculations of F_1 Scores

Label	F1 Score
angry	0.8889
annoyed	1.0000
happy	0.8889
neutral	1.0000
sad	1.0000
Macro average	0.9556
Weighted average	0.9500

6.2.4.5 Receiver Operating Characteristic (ROC)

A ROC visualizes the trade-off between the true positive rate and the false positive rate.

Figure 6.1 illustrates the ROC curve.

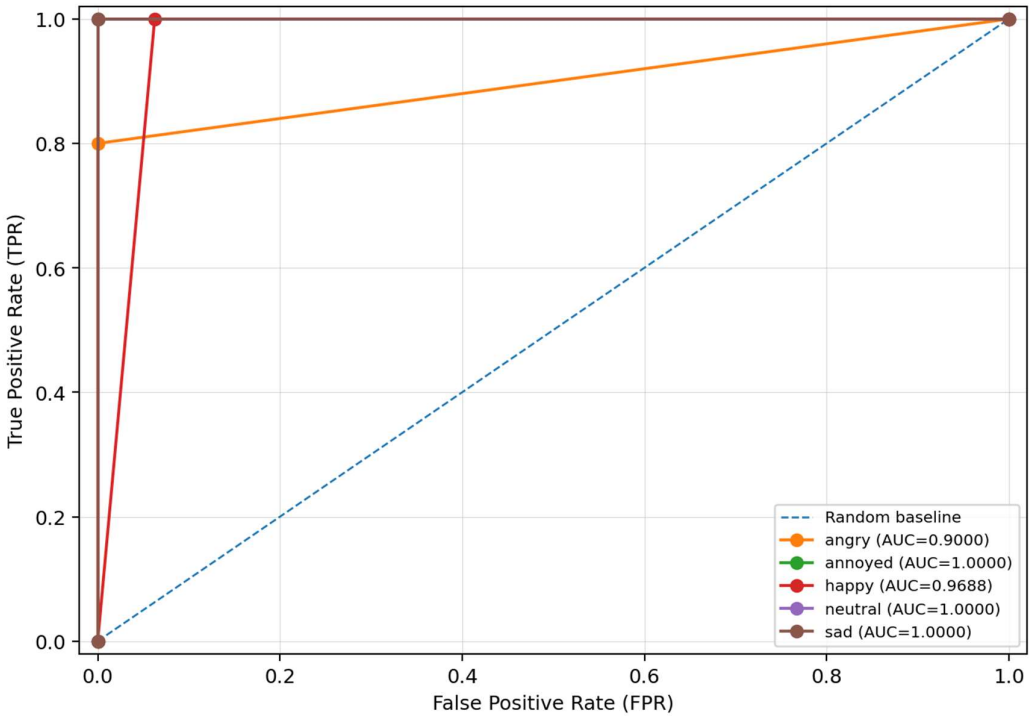


Figure 6.1 ROC Curve

6.2.4.6 Summary of Emotion Classifier Test Result

Table 6.13 summarise the emotion classifier test result.

Table 6.13 Summary of Emotion Classifier Test Result

Metric	**Result**
Number of test messages	20
Correct predictions	19 / 20
Accuracy	0.9500
Macro precision	0.9600
Macro recall	0.9600
Macro F1 score	0.9556
Weighted precision	0.9600
Weighted recall	0.9500
Weighted F1 score	0.9500
Macro ROC AUC	0.9947

6.3 Project Challenges

Several project challenges were encountered during the implementation and evaluation of Roofed. They are listed in Table 6.14.

Table 6.14 Project Challenges

No	Project Challenges	Explanation
1	Platform limitation	Due to IOS related development required macOS and Xcode for formal deployment, the project remained limited to Android. There are some ways to development IOS project in Windows, such as running a virtual device, but this method is not formal and stable.
2	Complexity of integrating many services	Roofed relied on Firebase Authentication, Cloud Firestore, Firebase Storage, Cloud Messaging, Cloud Functions, Google Maps or Geocoding and a Cloud Run backend for emotion classification, image classification and AI advice. During development, it was hard to identify the failure came from which part, thus debugging took a lot of time
3	Project cost to service access	Firebase Blaze Plan and Google Cloud subscriptions were required for image storage and location API access respectively. For better efficiency, ChatGPT plus subscription were also adopted, which also allows access to more token limit for Codex. Luckily, since these services are very “forgiving”, where some of them provides free usage within a limit, the cost pressure was acceptable for the overall development.
4	Late development refinements after testing	Although the working version provided to the testers were sufficient enough. Due to wanting to maintain high quality and perfecting Roofed, many features and bug fixes were developed in the later timeline. Therefore, there was time crunching concerns, but fortunately it was done in time.
5	Long term maintainability	Due to maintaining cost worries, without removing features supports, time and money have to be constantly spent to maintain the app even after this project.

6.4 Objectives Evaluation

There are two main objectives that can be evaluated for the final implementation, which are to develop a user-friendly family-oriented mobile application and to integrate NLP sentiment text analysis into the messaging system. They can be evaluated by the testing results.

Table 6.15 shows the evaluation of project objectives with evidence.

Table 6.15 Project Objectives Evaluation with Evidence

Objective	Evidence	Evaluation
Develop a family-oriented mobile application with core coordination features	Working prototype with chat, gallery, event, calendar, location, notifications, and profile modules.	Achieved
Integrate NLP sentiment text analysis into the messaging system	Deployed emotion API, stored emotion metadata, Home and Emotion summaries, and 0.9500 classifier accuracy on the test set.	Achieved

6.5 Concluding Remark

In the end, the evaluation results showed that Roofed was successful as a family-oriented mobile application with NLP sentiment text analysis features integrated into its messaging system. The system did not operate as a single isolated feature, but combined group communication, event planner functions, gallery sharing, emotional-awareness support, AI advice, notifications, and live location sharing within one coordinated application.

The feedback by testers indicated that the app was generally understandable, stable, and positively received. Functional modules such as login, group joining, chat, image upload, event handling, and general page navigation performed well according to the questionnaire results. The emotion-classifier test also showed strong performance, which supported the inclusion of the emotion module as a meaningful system component rather than a purely decorative addition.

At the same time, the evaluation also revealed areas for future improvement, such as enhancing the predictability of emotions, more useful features and refinement of interaction details such as location clarity and chat stability. Overall, the evaluation supported the main aims of the project had been achieved, while still leaving practical sectors for continued development.

CHAPTER 7

Conclusion and Recommendation

This chapter will conclude this report, giving closure to the development. It will also provide meaningful recommendations for future peers or continue development.

7.1 Conclusion

This project presented the design and development of Roofed, a family-oriented mobile app that combined communication, coordination, emotional-awareness support and AI-assisted interaction within one integrated Android app. The motivation of this project started from the increasing fragmentation of everyday digital communication, where families often rely on multiple separated applications for messaging, media sharing, planning, reminders, and location updates. To solve this issue, the project aimed to produce a single family-centred platform that could centralise important daily-life functions while also introducing an emotion-aware layer to support more thoughtful interpretation of text communication.

Furthermore, the final implementation showed that the proposed concept was feasible. The completed project prototype included authentication and profile setup, group creation and joining, real time chat, image sharing, gallery browsing, event creation, weekly calendar viewing, live location sharing, push notifications, emotion display and AI family advice. The system also demonstrated that Flutter, Firebase services, Google APIs, Cloud Functions, Cloud Run and a lightweight machine-learning classifier could be integrated effectively in an app.

Moreover, the evaluation results also supported the value of the project. Questionnaire feedback indicated that the prototype was generally easy to use, stable and useful for family or close-group communication. The emotion-classifier evaluation also produced strong performance within the prepared test set, showing that the selected TF-IDF plus Logistic Regression approach was sufficiently effective for this application context. In addition, tester

comments provided practical evidence that the post-testing refinement phase improved the project in meaningful ways.

Although the project achieved its main goals, it also identified limitations such as the emotion classifier still require extra training to increase its prediction accuracy. Nevertheless, these limitations did not undermine the value of this project. Instead, they help paint meaningful directions for future development.

In summary, Roofed achieved its central aim of producing a family-centred mobile application that integrated everyday coordination features with NLP sentiment text analysis function. The project therefore contributed not only a functioning system, but also a documented example of how mobile frontend development, Firebase cloud services, cloud-hosted backend APIs, and lightweight machine-learning components could be combined in a coherent final year project.

7.2 Recommendation

Before concluding the report, there are some recommendations that were derived, with the aim to assist future development or providing meaningful aid to future peers.

Table 7.1 shows the list of recommendations.

Table 7.1 List of Recommendations

No	Recommendation	Explanation
1	Refine the AI family advisor module	Future work should improve the specificity, completeness, and practical usefulness of the AI family-advice responses. This can be done through stronger prompt design, better context selection, safer but richer profile summarization, and additional response-quality checks before replies are shown to the user.
2	Improve emotion classifier prediction accuracy and usability	Future development should not focus only on model accuracy, but also on how ordinary users understand the emotion feature. Clearer explanation text, more transparent summary logic, and richer visual guidance would help users interpret the displayed emotion percentages more confidently.
3	Strengthen production readiness and security	If the project is continued beyond the current prototype, backend access control should be tightened and platform configuration should be cleaned further. For example, recording development logs for better reference in the future. Additional protection for cloud endpoints, safer handling of API keys, and more structured deployment settings would be important before real-world release.
4	Improve maintainability and automated testing	As the project grows, code maintainability becomes increasingly important. Large feature files should be broken into smaller components and shared services where appropriate. In addition, more automated tests should be added so that future changes can be verified more efficiently.
5	Extend long-term user evaluation and platform support	Future versions of Roofed would benefit from longer real-world testing with more family groups and a broader age distribution. If development resources permit, cross-platform expansion and a more production-ready release path could also be explored so that the system becomes accessible to a wider range of users.

REFERENCES

[1] N. Chokshi, “Your Kids Think You’re Addicted to Your Phone,” The New York Times, May 29, 2019. Accessed: Apr. 23, 2026. [Online]. Available: https://www.nytimes.com/2019/05/29/technology/cell-phone-usage.html

[2] M. Dhingra and R. K. Mudgal, “Historical Evolution of Social Media: An Overview,” in Int. Conf. Adv. Eng. Sci. Manag. Technol. (ICAESMT), Dehradun, India, 2019, pp. 1–8, doi: 10.2139/ssrn.3395665.

[3] E. Haddi, X. Liu, and Y. Shi, “The role of text pre-processing in sentiment analysis,” Procedia Comput. Sci., vol. 17, pp. 26–32, 2013, doi: 10.1016/j.procs.2013.05.005.

[4] R. Davie, C. Panting, and T. Charlton, “Mobile phone ownership and usage among pre-adolescents,” Telemat. Informat., vol. 21, no. 4, pp. 359–373, Nov. 2004, doi: 10.1016/j.tele.2004.04.001.

[5] D. DeSilver, “The average selling price of a smartphone,” Pew Research Center. Accessed: Apr. 23, 2026. [Online]. Available: https://www.pewresearch.org/short-reads/2013/09/10/the-average-selling-price-of-a-smartphone/

[6] J. McSkimming, “The restoration of contemplation and creative solitude,” Int. J. Narrative Therapy Community Work, no. 3, 2016. Accessed: Apr. 23, 2026. [Online]. Available: https://search.informit.org/doi/abs/10.3316/informit.460586860757043

[7] N. M. Fadzil, M. Y. Abdullah, and M. A. M. Salleh, “The level of tolerance sanctioning children using gadgets by parents lead to nomophobia: Early age gadgets exposure,” Int. J. Arts Sci., vol. 9, no. 2, pp. 615–622, 2016.

[8] D. M. Bidkar, A. G. Parthi, D. Maruthavanan, B. Pothineni, and S. R. Sankiti, “Developing user-facing experiences in Android applications: A focus on push notifications and background operations,” SSRN Electron. J., 2025, doi: 10.2139/ssrn.5042159.

[9] S. Abel, T. Machin, and C. Brownlow, “Social media, rituals, and long-distance family relationship maintenance: A mixed-methods systematic review,” New Media Soc., vol. 23, no. 3, pp. 632–654, 2021, doi: 10.1177/1461444820958717.

[10] M. Merkaš, M. Bodrožić Selak, and A. Žulec Ivanković, "Problematic smartphone use and communication in families with adolescents," Adolescents, vol. 4, no. 1, pp. 107–119, 2024, doi: 10.3390/adolescents4010008.

[11] A. Visuri, A. Sarsenbayeva, J. Goncalves, S. Hosio, and V. Kostakos, "Understanding smartphone notifications' user interactions, usage and impact," Int. J. Human-Comput. Stud., vol. 129, pp. 72–85, Sep. 2019, doi: 10.1016/j.ijhcs.2019.03.001.

[12] M. Gomez-Hernandez, X. Ferré, C. Moral, and E. Villalba-Mora, "Design guidelines of mobile apps for older adults: Systematic review and thematic analysis," JMIR Mhealth Uhealth, vol. 11, 2023, Art. no. e43186, doi: 10.2196/43186.

[13] E. Windmill, Flutter in Action. Shelter Island, NY, USA: Manning Publications, 2020. [Online]. Available: https://books.google.com.my/books?id=EzgzEAAAQBAJ. [Accessed: Apr. 23, 2026].

[14] Flutter, "Install Flutter," [Online]. Available: https://docs.flutter.dev/install. [Accessed: Apr. 23, 2026].

[15] Statcounter Global Stats, "Mobile Operating System Market Share Worldwide," *Statcounter Global Stats*. Accessed: Apr. 23, 2026. [Online]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide

[16] Firebase, "Add Firebase to your Flutter app," [Online]. Available: https://firebase.google.com/docs/flutter/setup. [Accessed: Apr. 23, 2026].

[17] Firebase, "Cloud Firestore documentation," [Online]. Available: https://firebase.google.com/docs/firestore. [Accessed: Apr. 23, 2026].

[18] Firebase, "Get started with Cloud Storage on Flutter," [Online]. Available: https://firebase.google.com/docs/storage/flutter/start. [Accessed: Apr. 23, 2026].

[19] Firebase, "Get started with Firebase Authentication on Flutter," [Online]. Available: https://firebase.google.com/docs/auth/flutter/start. [Accessed: Apr. 23, 2026].

[20] G. Bracha, The Dart Programming Language. Boston, MA, USA: Addison-Wesley, 2015. [Online]. Available: https://books.google.com.my/books?id=UHAlCwAAQBAJ. [Accessed: Apr. 23, 2026].

[21] A. M. Hassan, "JAVA and DART programming languages: Conceptual comparison," Indonesian Journal of Electrical Engineering and Computer Science, vol. 17, no. 2, pp. 845–851, Feb. 2020, doi: 10.11591/ijeecs.v17.i2.pp845-849.

[22] GeeksforGeeks, "Dart tutorial," GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/dart-tutorial/. [Accessed: Apr. 23, 2026].

[23] FastAPI, "FastAPI," [Online]. Available: https://fastapi.tiangolo.com/. [Accessed: Apr. 23, 2026].

[24] W3Schools, "Introduction to Python," W3Schools. [Online]. Available: https://www.w3schools.com/python/python_intro.asp. [Accessed: Apr. 23, 2026].

[25] P. Nandwani and R. Verma, "A review on sentiment analysis and emotion detection from text," Soc. Netw. Anal. Min., vol. 11, Art. no. 81, 2021, doi: 10.1007/s13278-021-00776-6.

[26] K. L. Tan, C. P. Lee, and K. M. Lim, "A survey of sentiment analysis: Approaches, datasets, and future research," Appl. Sci., vol. 13, no. 7, Art. no. 4550, 2023, doi: 10.3390/app13074550.

[27] scikit-learn, "TfidfVectorizer," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. [Accessed: Apr. 23, 2026].

[28] scikit-learn, "LogisticRegression," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. [Accessed: Apr. 23, 2026].

[29] T. Singh and M. Kumari, "Role of text pre-processing in Twitter sentiment analysis," Procedia Comput. Sci., vol. 89, pp. 549–554, 2016, doi: 10.1016/j.procs.2016.06.095.

[30] D. Demszky et al., "GoEmotions: A dataset of fine-grained emotions," in Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, 2020, pp. 4040–4054, doi: 10.18653/v1/2020.acl-main.372.

[31] Microsoft, "Documentation for Visual Studio Code," Visual Studio Code. [Online]. Available: https://code.visualstudio.com/docs. [Accessed: Apr. 30, 2026].

[32] Microsoft, "Debug code with Visual Studio Code," [Online]. Available: https://code.visualstudio.com/docs/debugtest/debugging. [Accessed: Apr. 30, 2026].

[33] Google Developers, "Meet Android Studio," *Android Developers*, 2019. [Online]. Available: https://developer.android.com/studio/intro. [Accessed: May 4, 2025].

[34] OpenAI, "Introducing Codex," May 16, 2025. [Online]. Available: https://openai.com/index/introducing-codex/. [Accessed: Apr. 30, 2026].

[35] OpenAI, “Introducing ChatGPT,” Nov. 30, 2022. [Online]. Available: https://openai.com/index/chatgpt/. [Accessed: Apr. 30, 2026].

[36] S. El-Salhi, F. Farouq, R. Obeidallah, and M. Al-Hami, “On developing an integrated family mobile application,” International Journal of Advanced Computer Science and Applications, vol. 10, no. 12, 2019, doi: 10.14569/ijacsa.2019.0101276.

[37] Ye. M. Hrabovskyi and Yu. D. Brusiltseva, “The methodology of developing a mobile application design for creating a genealogical tree,” 2022. [Online]. Available: http://repository.hneu.edu.ua/handle/123456789/28383

[38] Y.-M. Chung and J.-H. Choe, “A co-family managerial photo share mobile app UI development,” Apr. 2014. [Online]. Available: https://koreascience.kr/article/JAKO201414753677859.pdf

[39] Yellow Systems, “Family chat app development [How we created Famlicious],” Yellow Systems, Dec. 14, 2021. [Online]. Available: https://yellow.systems/blog/how-we-created-famlicious-a-family-chat-app-for-ios-and-android. [Accessed: May 1, 2025].

[40] Mbicycle, “Family organizer app development,” Mbicycle, Dec. 26, 2022. [Online]. Available: https://mbicycle.com/portfolio/family-organizer-app-development/. [Accessed: May 1, 2025].

[41] M. McCormick, “Waterfall vs. Agile Methodology,” McCormick PCS, Sep. 2012. [Online]. Available: http://www.mccormickpcs.com/images/Waterfall_vs_Agile_Methodology.pdf. [Accessed: May 4, 2025].

[42] C. Tozzi, “Waterfall vs. Agile methodology: Differences and examples,” TechTarget. [Online]. Available: https://www.techtarget.com/searchsoftwarequality/tip/Waterfall-vs-Agile-methodology-Differences-and-examples. [Accessed: May 4, 2025].

APPENDICES

Table A.1 shows the meeting log with FYP Instructor.

Table A.1 Meeting Log with FYP Instructor

Date	Topic
W2: 12/2/2026	Due to Hybrid Class, I was tasked to further finish my work. Redesign the UI elements of the app.
W4: 26/2/2026	Starting to implement backend services into my app, connect Firebase services to the local app.
W6: 12/3/2026	Show the 80% complete app, advised to tune some pages' functions, and make the functions more unique and useful.
W8: 26/3/2026	Finished the app development. Preparing to distribute to testers to gain feedbacks. I was advised to slightly redesign the app UI, to make main function more in focus.
W10: 9/4/2026	After testing phase and getting advices, improved some parts of the app that was lacking, added few features, mainly to chat behaviors.
W12: 23/4/2026	Finalizing report while continuing to discuss with instructor about weak parts.

Table A.2 shows the bug logs before testing phase.

Table A.2 Bug Logs Before Testing Phase

No	Affected Area	Issue Title	Description	Cause	Implemented Fix
1	auth / UI	Auth entry screen showed a redundant Login title under the logo	The first auth screen showed an extra `Login` title under the logo even though the page already explained the available actions.	The auth entry layout still included a standalone heading that duplicated the screen purpose.	Removed the extra `Login` title so the screen starts with the logo and helper text.
2	branding / app shell / platform config	App name casing was inconsistent across visible titles	The app name appeared as `roofed` in visible app metadata instead of `Roofed`, causing inconsistent branding across platforms.	Some display-name settings still used the old lowercase title.	Updated visible app title metadata to use `Roofed`.
3	event / calendar / Firebase	Past events were being deleted and disappearing from Calendar history	Finished events vanished completely, which made the Calendar page useless for viewing past events.	The shared planner task stream was auto-expiring past events by deleting their task documents from Firestore.	Removed the automatic task deletion flow so ended events stay in Firestore for Calendar, while the Event tab still hides them by showing only upcoming events.
4	auth / navigation	Signup footer sent users back to auth choices instead of Login	Tapping `Already have an account?` on the signup page returned users to the auth choice screen instead of taking them straight to Login.	The signup footer only called `maybePop()`, which depended on the previous route rather than explicitly opening the login form.	Changed the signup footer action to navigate directly to `LoginPage` and keep the flow on the auth form screens.
5	notification / Firebase / backend	Notifications depended on the sender device instead of Firebase delivery	Group message, photo, and event notifications were not reliably delivered when the app was in the background or terminated because the notification flow still depended on client-side handling.	The project was using a local Firestore-listener notification approach instead of a proper Firebase Cloud Messaging pipeline with backend delivery.	Switched notification delivery to Firebase Cloud Messaging, stored per-user FCM tokens in Firestore, added background/foreground handling in Flutter, and deployed Cloud Functions for group updates and event reminders.
6	notification / build	Notification service initialization broke after plugin API change	The app could fail to compile in the notification service because the local notifications plugin no longer accepted the old initialization call shape.	`flutter_local_notifications` v21 requires `initialize()` to receive `settings:` as a named argument, but the service was still passing the settings positionally.	Updated the notification service to call `_localNotifications.initialize(settings: initializationSettings, ...)`.
7	home / emotion	Emotion bubbles showed leftover test values for members with no recent messages	A newly joined group member could still show non-zero emotion percentages on the Home page even when they had no recent chat activity.	The emotion dashboard was falling back to seeded test-style values instead of a zeroed state when no weighted message data was available.	Changed the no-data emotion path to return an all-zero emotion map so empty history starts from `0%` across every bubble.

No	Affected Area	Issue Title	Description	Cause	Implemented Fix
8	gallery / UI	Gallery filter cards could overflow on narrow screens	The Gallery filter area could overflow horizontally by a few pixels, especially around the group/category dropdowns on tighter phone widths.	The dropdown content and selected labels could outgrow the compact filter card space.	Tightened the dropdown field layout and kept labels ellipsis-safe inside the shared dropdown card styling so the filter row fits cleanly.
9	location / UI / build	Location page dropdown restyle introduced a cascade of syntax errors	The Location page could fail to parse and show a large cascade of analyzer errors after the dropdown UI update.	A malformed dropdown key line in the location filter block broke the widget tree structure and caused follow-on parse errors.	Corrected the dropdown key/widget syntax and restored the location filter section to a valid build tree.
10	notification / build	Notification service build failed after plugin API change	The app could fail to compile after the notification service update because the local notification display call no longer matched the plugin API.	`flutter_local_notifications` was using the newer named-parameter `show()` signature, while the service still called it with positional arguments.	Updated the notification display call to use named parameters (`id`, `title`, and `body`) so the service builds correctly with the installed plugin version.
11	location / maps / state	Live location sharing could crash after updates kept arriving	The app could become unstable or crash shortly after location sharing was turned on and live updates started flowing in.	The map camera and sharing cleanup flow were being triggered too aggressively during repeated Firestore and location updates, which could surface async map or teardown errors.	Limited automatic map focusing, wrapped camera animation safely, and hardened sharing cleanup during page exit so repeated live updates no longer destabilize the screen.
12	profile / Firebase / auth	Anonymous profile setup could crash while opening the Profile page	Opening the Profile section or trying to save profile-related changes with an anonymous test account could crash the app.	Profile initialization ran an unhandled async Firestore write, and anonymous accounts could hit permission-related backend errors during that step.	Added guarded profile initialization, anonymous-safe fallback labels, and user-facing error handling so anonymous permission failures no longer crash the app.
13	emotion / backend integration	Shortform Normalization Missing in Emotion Inference	Slang and abbreviations such as `txt`, `y r u`, `omg`, and Malaysian short forms could reduce emotion prediction quality on real chat text if they were not normalized consistently.	Emotion inference needed a reusable replacement dictionary so the classifier would interpret common short forms more reliably.	Added reusable shortform normalization support so known abbreviations and Malaysian chat variants can be expanded consistently before classification.
14	emotion	Uppercase and Punctuation Emotion Signals Were Being Lost	Messages with all caps, repeated punctuation, repeated words, or elongated spelling could be under-classified because strong intensity signals were not being preserved well enough.	Emotion detection relied too heavily on normalized text and risked weakening emphasis cues from the original message form.	Added raw-text intensity support so uppercase, punctuation, repeated-word, and elongated-word signals are preserved and used during classification.
15	backend / deployment	Cloud Run Deployment Failed Due to Missing Builder Permission	The backend deployment to Google Cloud Run could fail during source build and prevent the API from going live.	The default build service account was missing the IAM permission required to build and deploy the Cloud Run service.	Granted the required Cloud Run builder permission to the default service account and redeployed successfully.

No	Affected Area	Issue Title	Description	Cause	Implemented Fix
16	ai-advice / backend	Family Advice Replies Could Stop Mid-Sentence	The family advice backend could sometimes return a reply that ended before the final sentence was completed.	The AI response could terminate too early, leading to visibly incomplete advice text.	Increased the generation allowance and added completion-focused handling so replies are more likely to finish cleanly.
17	ai-advice / backend integration	Family advice replies could stop mid-answer	The AI adviser could return an incomplete reply, leaving the user with a cut-off sentence instead of a full answer.	The family advice backend could sometimes stop before finishing the final sentence, and the app previously had no recovery step for incomplete replies.	Added incomplete-reply detection in `FamilyAdviceService`, retried the request with a stricter follow-up instruction, and surfaced a clearer user-facing error if the retry still returned an incomplete answer.
18	chat / emotion / backend integration	Emotion API parsing could fail on response format differences	Emotion tagging on chat messages could be lost when the backend response used different label names, different score keys, or percentage-style confidence values.	API integration needed to tolerate multiple backend response shapes and label variants such as `joy`, `anger`, or nested score maps instead of a single fixed schema.	Expanded `EmotionDetectionService` to normalize known label variants, accept multiple score field names, convert percentage values to 0-1 format, and fall back to the highest score when no direct label field is available.
19	chat / UI / state	Create and join group dialogs could crash when closing	The Create Group and Join Group dialogs could throw Flutter assertion errors when the user canceled or closed them.	The text input controller lifecycle was not isolated to the dialog, so the widget tree could still reference it during teardown.	Moved dialog input handling into a dedicated `_ChatInputDialog` widget that owns and disposes its own `TextEditingController`.
20	chat / Firebase / Firestore	Group chat message loading could fail with a Firestore index error	Opening a chat room could show a "Could not load messages" Firestore `failed-precondition` error even when the group and messages existed.	The message query used `groupId` filtering together with Firestore-side ordering, which required a missing composite index.	Streamed messages by `groupId` only and sorted them locally by `createdAt` before rendering the chat list.
21	calendar / UI	Weekly calendar view could overflow and hide the full 7-day schedule	The calendar page could show bottom overflow and made the weekly schedule hard to read in a full 7-day view.	The earlier week layout was too cramped for all day rows and task content on smaller screens.	Changed the calendar to a full weekly vertical layout with one day per row, compact dropdown controls, and whole-week swipe navigation.
22	chat / gallery / UI	Chat image upload entry could appear blank and expose the wrong picker flow	The chat image-upload button could appear blank, and the picker flow was not limited to the intended gallery-only path.	The upload trigger styling did not render clearly in the current theme, and the menu was not constrained to a photo-only option.	Added an explicit photo-library icon with visible styling and changed the upload menu to a single `Photos` choice that leads into the gallery permission flow.
23	auth / Firebase / navigation	Email login and guest access were not creating real Firebase sessions	The auth flow could show email login and guest options without reliably creating a real Firebase-authenticated user session before entering the app.	The form flow was still placeholder-driven, and guest access previously navigated forward without Firebase anonymous sign-in.	Wired the email login form to Firebase email/password auth and changed guest access to use Firebase anonymous sign-in before home navigation.
24	navigation / UI / build	Bottom navigation theme update introduced a build error	The app could fail to compile after the menu/icon color update because the bottom navigation item widget could not resolve the	Shared color constants were scoped inside `AppBottomNavigationBar` but referenced from the sibling `_BottomNavItem` class.	Moved the shared navigation color constants to file scope so both widgets can access them.

No	Affected Area	Issue Title	Description	Cause	Implemented Fix
			shared icon color constant.		
25	test / startup	Widget smoke test still referenced a removed app entry widget	The test suite could fail because the default widget test still tried to build `MyApp`, which no longer exists in this project.	The starter counter test was left unchanged after the app entry widget was replaced with `RoofedApp`.	Replaced the outdated test with a simple `RoofedApp` splash smoke test.
26	auth / startup / notification	Firebase startup options were not initialized consistently	App startup could fail to prepare authentication and other Firebase-backed features correctly on supported platforms.	Firebase initialization was not consistently guarded and tied to the generated platform options.	Added guarded initialization using `Firebase.apps.isEmpty` together with `DefaultFirebaseOptions.currentPlatform` during splash startup and Firebase-dependent background setup.
27	auth	Username-based login and reset flow did not resolve Firebase email auth	Users entering a username instead of an email could not reliably log in or request password reset through Firebase email/password APIs.	Firebase email authentication requires an email, while the UI now accepts email-or-username input.	Added username-to-email lookup in `AuthService`, including fallback support for legacy username records before calling Firebase auth methods.
28	auth / UI	Auth entry screen could become cramped on smaller devices	The login/signup entry screen risked layout crowding and button text compression on compact phone screens.	Large card content and long button labels were competing for limited vertical and horizontal space.	Updated the auth entry layout to use `LayoutBuilder`, `SingleChildScrollView`, compact spacing rules, and ellipsis-safe button labels.
29	startup / navigation	Duplicate entry file kept an outdated startup route	The project had a second entry file that still pointed to an older startup flow, which could confuse developers and leave stale navigation behavior in the repo.	`main_replacement.dart` was left behind after the main entrypoint changed.	Kept `main.dart` as the real entrypoint and converted `main_replacement.dart` into a small shim instead of leaving duplicated startup logic.
30	auth / navigation	Unauthenticated users were not fully gated before home access	The earlier auth flow allowed login/signup access from the home area instead of enforcing the intended startup path first.	Authentication entry was exposed from the home experience instead of being handled as part of app startup.	Moved the auth decision into the splash/startup flow and removed the temporary homepage auth shortcut.
31	chat / uploads / profile / groups	Chat uploads were slow and did not feel immediate	Text and image messages could feel delayed, and image/profile/group photo uploads did not provide quick visual feedback.	Messages waited too long on backend confirmation, and image uploads were not optimized before sending.	Added optimistic text updates, sending/uploading state, local image preview, basic upload progress support, and shared image compression/resizing before upload.
32	chat / groups / navigation	Leaving a group returned users to the wrong screen	After leaving a group, the app could send the user back into the old chatroom flow instead of the main Chat page.	The leave-group success path navigated to the wrong destination.	Updated the leave-group flow to go directly back to the main Chat page.
33	notification / Firebase / groups	Group action push notifications could show	Group photo changes, group renames, nickname changes, joins, and leaves could send	The server-side notification builder could fall through to the wrong event-based notification branch.	Added explicit notification metadata for group actions and updated the Cloud Function to use the provided notification title/body for accurate FCM content.

No	Affected Area	Issue Title	Description	Cause	Implemented Fix
		deleted-event text	an FCM message saying an event was deleted.		
34	chat / groups / notification	Group activity notices were incomplete inside chat	Important actions such as nickname changes, photo changes, joins, leaves, and group renames were not consistently shown as chatroom notices.	Some group actions updated backend records or push notifications without also writing a structured in-chat system message.	Added system chat notices for those group actions and updated the renderer to show the correct text, including `you` where appropriate.
35	home / emotion / presence	Active status could stay stale or inconsistent	Users could see `Last active unknown` or stale presence state even while the app was open, and self-view could show the wrong status.	Presence writes were not refreshed regularly in the foreground, and some screens relied on cached member data.	Added a 1-minute foreground heartbeat for `lastActiveAt`, refreshed dependent views, and forced self-view to display `active now`.
36	state / home / gallery / emotion / calendar / location	Dropdown selections were lost after relaunch	Chosen dropdown values such as selected group, week, and grid size reset after terminating and reopening the app.	Dropdown state was only kept in memory.	Added persisted dropdown selection support using `shared_preferences` for the main selectors.
37	UI / chat / emotion / event / gallery	Responsive layout broke on some phone sizes	Some screens could show overflow, awkward spacing, truncated action labels, or misaligned cards on smaller or larger phones.	Several layouts used fixed sizing and alignment that did not adapt well across screen sizes.	Adjusted responsive sizing and alignment for Chat, Emotion, Event, Gallery, and related bottom-spacing behavior while keeping the same app flow.

Table A.3 shows the bug logs after testing phase.

Table A.3 Bug Logs After Testing Phase

No	Affected Area	Issue Title	Description	Cause	Implemented Fix
1	chat / scroll / state	Chat room could jump between newest and oldest messages during open/reply	Entering a chat room or entering reply mode could flicker the view between bottom and top, making the room feel unstable.	The message list and related streams were rebuilding in ways that disturbed bottom anchoring and reply-preview layout changes.	Stabilized the chat streams, tightened bottom anchoring, and moved reply preview handling fully back into the composer layout.
2	chat / reply / UI state	Reply preview close button could stop working and the reply bar could shift the room	The reply bar close button could become unreliable, and opening reply mode could disturb the current room position.	The reply preview was not fully integrated into the composer layout and was interacting badly with rebuild/scroll behavior.	Reworked reply preview placement, restored the close interaction, and prevented reply mode from yanking the room toward older messages.
3	chat / hold-state UI	Reaction bar and hold-state menu could drift too far or cover the held message	Long-press reactions and the action sheet could appear too far from the selected message or partially block the held content.	Reaction/action positioning depended too much on surrounding layout instead of a fixed anchor to the selected message.	Standardized reaction spacing from the selected message, aligned it to the correct side, and let the action menu move above the held message when needed.
4	chat / video / media preview	Video previews could overflow, lose thumbnails, or place fullscreen	Vertical video previews could overflow, held video previews could regress to loading placeholders, and	Video layouts, thumbnail controller reuse, and fullscreen control placement needed tighter	Added safer preview constraints, shared thumbnail warming/reuse, and refined fullscreen control layouts.

No	Affected Area	Issue Title	Description	Cause	Implemented Fix
		controls over content	fullscreen controls could sit on top of the video content awkwardly.	constraints and better reuse.	
5	media / image preview	Image preview interaction was cluttered and mixed normal popup and fullscreen behaviors	The regular image popup was carrying zoom gestures directly, which made it busier than necessary and less consistent with the video flow.	Zoom and pinch handling lived in the normal popup instead of a dedicated fullscreen viewer.	Moved zoom interactions into a fullscreen image viewer with a close button and kept the normal popup cleaner.
6	profile / group info / UI consistency	Profile and group photo change popups used the wrong outer shell style	The photo-change popups still showed a larger outer green background that felt inconsistent with the newer media popup design.	Those flows still used an older sheet presentation style.	Restyled both to the same blur-backed, box-in-box popup approach used elsewhere.
7	location / maps / live updates	Location sharing could fall back to the old red pin, overlap badly, and flicker while sharing	Shared user locations could still show the old pin, be too large, overlap badly, and flicker when sharing updates arrived.	Marker rendering, fallback logic, z-ordering, and update handling were not yet fully aligned to the avatar-marker approach.	Switched to profile/initial avatar markers with a black border, reduced marker size, layered the selected user on top, filtered the center list to active sharers, and stabilized live marker updates.
8	chat page / scrolling / layout	Group and private chat sections could clip content or hide newly active chats	Chat-page sections could clip cards, sit too close to bottom navigation, and hide newly active chats when the user had scrolled deeper into the section.	Section height, section scrolling, and reorder handling were not tuned for the final 3-visible-item layout.	Tuned section sizing, page bottom spacing, and section scroll reset behavior so newly active chats become visible immediately.
9	chat / forwarding / notifications	Forward labels and forwarded-notification behavior did not match chat context	Forwarded messages could show the wrong `by/from` label, and forwarded items were not fully aligned with normal notification behavior.	Forward behavior was previously simplified and did not fully encode destination-context rules.	Added destination-aware forward labels plus notification handling that treats forwarded text/images like normal messages while keeping videos distinct.
10	notifications / Firebase / Cloud Functions	Reply, mention, and private-chat notification wording was inconsistent	Notifications did not fully differentiate reply-to-you, mention, private-chat username usage, or one-day event reminders.	The existing chat notification pipeline and Cloud Function fallback text covered only the simpler base cases.	Standardized notification metadata and server-side wording for reply-to-you, mention alerts, private username-based notifications, and both one-hour and one-day event reminders.

Questionnaire Questions (10 Sections):

Table A.4 – A.13 shows the Google Form questions for 10 sections.

Table A.4 Section 1 - Tester Information

No.	Question	Question Type	Options / Scale	Required
1	Tester Name	Short answer	Free-text answer	Yes
2	Age Range	Multiple choice	Below 18; 18-24; 25-34; 35-44; 45 and above	Yes
3	Relationship To Tester Group	Multiple choice	Family Members; Friends	Yes

Table A.5 Section 2 - App Setup and First Impression

No.	Question	Question Type	Options / Scale	Required
1	Were you able to install and open the app successfully?	Multiple choice	Yes; No; Yes, but with some issue	Yes
2	What issue(s) did you face during installation? (Leave blank if none)	Paragraph	Free-text answer	No
3	Was the login/sign-up/guest access process easy to understand?	Linear scale	1 = Very Confusing; 5 = Very Easy	Yes
4	Did you face any problem during account setup or profile setup?	Multiple choice	No problem; Minor Problem; Serious Problem	Yes
5	Please describe any setup/login/profile problem you faced. (Leave blank if none)	Paragraph	Free-text answer	No
6	What was your first impression of the app design? (UI, Layout, Readability)	Paragraph	Free-text answer	Yes

Table A.6 Section 3 - Chat and Group Features

No.	Question	Question Type	Options / Scale	Required
1	Were you able to create or join a group successfully?	Multiple choice	Yes; No; Did not test	Yes
2	Was the group code / join group process clear?	Linear scale	1 = Very Confusing; 5 = Very Clear	Yes
3	Were chat messages sent and received properly?	Multiple choice	Yes; Sometimes delayed; No; Did not test	Yes
4	Did image upload in chat work properly?	Multiple choice	Yes; Sometimes failed; No; Did not test	Yes
5	Did the group(s) make sense? (Group info page and chat room page)	Linear scale	1 = Very Confusing; 5 = Very Clear	Yes
6	What are your suggestion to this feature?	Paragraph	Free-text answer	No

Table A.7 Section 4 - Emotion Display Features

No.	Question	Question Type	Options / Scale	Required
1	Did the emotion display page load properly?	Multiple choice	Yes; No; Did not test	Yes
2	Was it clear which member's emotion was being displayed?	Linear scale	1 = Very Confusing; 5 = Very Clear	Yes
3	Was the dominant emotion easy to notice?	Linear scale	1 = Not Noticeable; 5 = Very Noticeable	Yes
4	Did the emotion result feel reasonable based on the message?	Linear scale	1 = Very Inaccurate; 5 = Very Accurate	Yes
5	Did the page(s) make sense?	Linear scale	1 = Very Confusing; 5 = Very Clear	Yes
6	What are your suggestion to this feature?	Paragraph	Free-text answer	No

Table A.8 Section 5 - Event and Calendar Features

No.	Question	Question Type	Options / Scale	Required
1	Were you able to create an event successfully?	Multiple choice	Yes; No; Did not test	Yes
2	Was the event creation form easy to understand?	Linear scale	1 = Very Confusing; 5 = Very Clear	Yes
3	Did the event appear correctly in the Event page?	Multiple choice	Yes; No; Did not test	Yes
4	Was weekly calendar navigation clear?	Linear scale	1 = Very Confusing; 5 = Very Clear	Yes
5	Did the event appear correctly in the Calendar page?	Multiple choice	Yes; No; Did not test	Yes
6	Did the page(s) make sense? (Event page and Calendar page)	Linear scale	1 = Very Confusing; 5 = Very Clear	Yes
7	What are your suggestion to this feature?	Paragraph	Free-text answer	No

Table A.9 Section 6 - AI Family Advice Features

No.	Question	Question Type	Options / Scale	Required
1	Were you able to find and open the AI family advice page?	Multiple choice	Yes; No; Did not test	Yes
2	Were you able to ask a question and receive a reply?	Multiple choice	Yes; No; Sometimes delayed; Did not test	Yes
3	Was the AI reply relevant to the question?	Linear scale	1 = Not Relevant; 5 = Very Relevant	Yes
4	Was the AI reply easy to understand?	Linear scale	1 = Very Confusing; 5 = Very Clear	Yes
5	Did the previous inquiries/history function work?	Multiple choice	Yes; No; Did not test	Yes
6	Did the page(s) make sense?	Linear scale	1 = Very Confusing; 5 = Very Clear	Yes
7	What are your suggestion to this feature?	Paragraph	Free-text answer	No

Table A.10 Section 7 - Notifications Features

No.	Question	Question Type	Options / Scale	Required
1	Did you receive notifications from the app?	Multiple choice	Yes; No	Yes
2	Is the notifications delayed?	Multiple choice	Yes; No	Yes
3	Which notifications did you receive?	Checkboxes	Chat Messages; Event Creation; Event Reminder; Group Updates (New member join, nickname change...); Not Sure	Yes
4	Were the notifications useful?	Linear scale	1 = Not Useful; 5 = Very Relevant	Yes
5	What are your suggestion to this feature?	Paragraph	Free-text answer	No

Table A.11 Section 8 - Overall Usability

No.	Question	Question Type	Options / Scale	Required
1	Overall, how easy was the app to use?	Linear scale	1 = Very Difficult; 5 = Very Easy	Yes
2	How stable did the app feel?	Linear scale	1 = Very Unstable; 5 = Very Stable	Yes
3	How would you rate the app design?	Linear scale	1 = Very Poor; 5 = Very Good	Yes
4	How useful do you think this app is for family or close group communication?	Linear scale	1 = Not Useful; 5 = Very Useful	Yes
5	Which feature(s) did you like the most? Explain Briefly	Paragraph	Free-text answer	Yes
6	Which feature(s) needs the most improvement? Explain Briefly	Paragraph	Free-text answer	Yes
7	Would you continue using the app if it was improved and completed?	Multiple choice	Yes; Maybe; No	Yes

Table A.12 Section 9 - Bug Report

No.	Question	Question Type	Options / Scale	Required
1	Did you experience any bugs, crashes, freezing, or wrong display?	Multiple choice	Yes; No	Yes
2	What bug(s) did you find? Explain Briefly (Where, What and How)	Paragraph	Free-text answer	No
3	Can the bug happen again if you repeat the same steps?	Multiple choice	Yes; No; Not Sure	Yes

Table A.13 Section 10 - Final Feedback

No.	Question	Question Type	Options / Scale	Required
1	What should be added to the app in the future? (Leave blank if none)	Paragraph	Free-text answer	No
2	What should be removed or simplified? (Leave blank if none)	Paragraph	Free-text answer	No
3	Any final comment for the developer? (Leave blank if none)	Paragraph	Free-text answer	No

Questionnaire Response (8 Respondents):

Table A.14 shows the Google Form responses from 8 respondents respectively.

Table A.14 Questionnaire Response from Respondents

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R1	45 and above	Family Members	Timestamp	4/26/2026 10:10:39
R1	45 and above	Family Members	Were you able to install and open the app successfully?	Yes
R1	45 and above	Family Members	Was the login/sign-up/guest access process easy to understand?	5
R1	45 and above	Family Members	Did you face any problem during account setup or profile setup?	No problem
R1	45 and above	Family Members	What was your first impression of the app design?	Normal, easy to use
R1	45 and above	Family Members	Were you able to create or join a group successfully?	Yes
R1	45 and above	Family Members	Was the group code / join group process clear?	4
R1	45 and above	Family Members	Were chat messages sent and received properly?	Yes
R1	45 and above	Family Members	Did image upload in chat work properly	Yes
R1	45 and above	Family Members	Chat and group pages made sense	4
R1	45 and above	Family Members	Suggestions for chat/group feature	Maybe can have conference call or video call
R1	45 and above	Family Members	Did the emotion display page load properly?	Did not test
R1	45 and above	Family Members	Was it clear which member's emotion was being displayed?	3
R1	45 and above	Family Members	Was the dominant emotion easy to notice?	4
R1	45 and above	Family Members	Did the emotion result feel reasonable based on the message?	4
R1	45 and above	Family Members	Emotion page made sense	4
R1	45 and above	Family Members	Suggestions for emotion feature	No comment
R1	45 and above	Family Members	Were you able to create an event successfully?	Did not test
R1	45 and above	Family Members	Was the event creation form easy to understand?	4
R1	45 and above	Family Members	Did the event appear correctly in the Event page?	Yes
R1	45 and above	Family Members	Was weekly calendar navigation clear?	4
R1	45 and above	Family Members	Did the event appear correctly in the Calendar page?	Did not test
R1	45 and above	Family Members	Event and calendar pages made sense	4
R1	45 and above	Family Members	Suggestions for event/calendar feature	No comment
R1	45 and above	Family Members	Were you able to find and open the AI family advice page?	Did not test
R1	45 and above	Family Members	Were you able to ask a question and receive a reply?	Did not test

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R1	45 and above	Family Members	Was the AI reply relevant to the question?	4
R1	45 and above	Family Members	Was the AI reply easy to understand?	4
R1	45 and above	Family Members	Did the previous inquiries/history function work?	Did not test
R1	45 and above	Family Members	AI advice page made sense	4
R1	45 and above	Family Members	Suggestions for AI advice feature	No comment
R1	45 and above	Family Members	Did you receive notifications from the app?	Yes
R1	45 and above	Family Members	Is the notifications delayed?	Yes
R1	45 and above	Family Members	Which notifications did you receive?	Chat Messages, Event Creation, Event Reminder, Group Updates (New member join, nickname change....)
R1	45 and above	Family Members	Were the notifications useful?	4
R1	45 and above	Family Members	Notification-related pages made sense	4.0
R1	45 and above	Family Members	Suggestions for notification feature	No comment
R1	45 and above	Family Members	Overall, how easy was the app to use?	4
R1	45 and above	Family Members	How stable did the app feel?	4
R1	45 and above	Family Members	How would you rate the app design?	4
R1	45 and above	Family Members	How useful do you think this app is for family or close group communication?	4
R1	45 and above	Family Members	Which feature(s) did you like the most?	No comment
R1	45 and above	Family Members	Which feature(s) needs the most improvement?	No comment
R1	45 and above	Family Members	Would you continue using the app if it was improved and completed?	Maybe
R1	45 and above	Family Members	Did you experience any bugs, crashes, freezing, or wrong display?	No
R2	18-24	Family Members	Timestamp	4/26/2026 10:18:02
R2	18-24	Family Members	Were you able to install and open the app successfully?	Yes
R2	18-24	Family Members	Was the login/sign-up/guest access process easy to understand?	5
R2	18-24	Family Members	Did you face any problem during account setup or profile setup?	No problem
R2	18-24	Family Members	What was your first impression of the app design?	Clean UI design
R2	18-24	Family Members	Were you able to create or join a group successfully?	Yes
R2	18-24	Family Members	Was the group code / join group process clear?	4
R2	18-24	Family Members	Were chat messages sent and received properly?	Yes
R2	18-24	Family Members	Did image upload in chat work properly	Yes
R2	18-24	Family Members	Chat and group pages made sense	5
R2	18-24	Family Members	Suggestions for chat/group feature	-
R2	18-24	Family Members	Did the emotion display page load properly?	Yes

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R2	18-24	Family Members	Was it clear which member's emotion was being displayed?	5
R2	18-24	Family Members	Was the dominant emotion easy to notice?	4
R2	18-24	Family Members	Did the emotion result feel reasonable based on the message?	2
R2	18-24	Family Members	Emotion page made sense	2
R2	18-24	Family Members	Suggestions for emotion feature	There should be a description on what this feature is
R2	18-24	Family Members	Were you able to create an event successfully?	Did not test
R2	18-24	Family Members	Was the event creation form easy to understand?	3
R2	18-24	Family Members	Did the event appear correctly in the Event page?	Did not test
R2	18-24	Family Members	Was weekly calendar navigation clear?	3
R2	18-24	Family Members	Did the event appear correctly in the Calendar page?	Did not test
R2	18-24	Family Members	Event and calendar pages made sense	3
R2	18-24	Family Members	Suggestions for event/calendar feature	-
R2	18-24	Family Members	Were you able to find and open the AI family advice page?	Did not test
R2	18-24	Family Members	Were you able to ask a question and receive a reply?	Did not test
R2	18-24	Family Members	Was the AI reply relevant to the question?	3
R2	18-24	Family Members	Was the AI reply easy to understand?	3
R2	18-24	Family Members	Did the previous inquiries/history function work?	Did not test
R2	18-24	Family Members	AI advice page made sense	3
R2	18-24	Family Members	Suggestions for AI advice feature	-
R2	18-24	Family Members	Did you receive notifications from the app?	Yes
R2	18-24	Family Members	Is the notifications delayed?	No
R2	18-24	Family Members	Which notifications did you receive?	Chat Messages
R2	18-24	Family Members	Were the notifications useful?	5
R2	18-24	Family Members	Notification-related pages made sense	2.0
R2	18-24	Family Members	Suggestions for notification feature	Notification of already seen messages is still there, this makes it accumulate a lot of notification
R2	18-24	Family Members	Overall, how easy was the app to use?	4
R2	18-24	Family Members	How stable did the app feel?	4
R2	18-24	Family Members	How would you rate the app design?	5
R2	18-24	Family Members	How useful do you think this app is for family or close group communication?	3
R2	18-24	Family Members	Which feature(s) did you like the most?	Emotion detector was cool
R2	18-24	Family Members	Which feature(s) needs the most improvement?	Notification
R2	18-24	Family Members	Would you continue using the app if it was improved and completed?	Maybe
R2	18-24	Family Members	Did you experience any bugs, crashes, freezing, or wrong display?	Yes

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R2	18-24	Family Members	What bug(s) did you find?	When typing a message the chatroom ui sometimes jump up and down
R2	18-24	Family Members	Can the bug happen again if you repeat the same steps?	Yes
R2	18-24	Family Members	What should be added to the app in the future?	Direct messaging instead of needing groups
R3	Below 18	Family Members	Timestamp	4/26/2026 11:23:17
R3	Below 18	Family Members	Were you able to install and open the app successfully?	Yes
R3	Below 18	Family Members	Was the login/sign-up/guest access process easy to understand?	4
R3	Below 18	Family Members	Did you face any problem during account setup or profile setup?	No problem
R3	Below 18	Family Members	What was your first impression of the app design?	UI
R3	Below 18	Family Members	Were you able to create or join a group successfully?	Yes
R3	Below 18	Family Members	Was the group code / join group process clear?	5
R3	Below 18	Family Members	Were chat messages sent and received properly?	Yes
R3	Below 18	Family Members	Did image upload in chat work properly	Yes
R3	Below 18	Family Members	Chat and group pages made sense	4
R3	Below 18	Family Members	Suggestions for chat/group feature	Could add GIF or emojis and also allow us to reply to a message.
R3	Below 18	Family Members	Did the emotion display page load properly?	Yes
R3	Below 18	Family Members	Was it clear which member's emotion was being displayed?	5
R3	Below 18	Family Members	Was the dominant emotion easy to notice?	5
R3	Below 18	Family Members	Did the emotion result feel reasonable based on the message?	4
R3	Below 18	Family Members	Emotion page made sense	4
R3	Below 18	Family Members	Suggestions for emotion feature	Tell me the emotions instead of only emojis
R3	Below 18	Family Members	Were you able to create an event successfully?	Yes
R3	Below 18	Family Members	Was the event creation form easy to understand?	5
R3	Below 18	Family Members	Did the event appear correctly in the Event page?	Yes
R3	Below 18	Family Members	Was weekly calendar navigation clear?	5
R3	Below 18	Family Members	Did the event appear correctly in the Calendar page?	Yes
R3	Below 18	Family Members	Event and calendar pages made sense	5
R3	Below 18	Family Members	Suggestions for event/calendar feature	Make a reminder for the event
R3	Below 18	Family Members	Were you able to find and open the AI family advice page?	Yes
R3	Below 18	Family Members	Were you able to ask a question and receive a reply?	Yes
R3	Below 18	Family Members	Was the AI reply relevant to the question?	5
R3	Below 18	Family Members	Was the AI reply easy to understand?	5
R3	Below 18	Family Members	Did the previous inquiries/history function work?	Yes
R3	Below 18	Family Members	AI advice page made sense	5

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R3	Below 18	Family Members	Suggestions for AI advice feature	Nonea
R3	Below 18	Family Members	Did you receive notifications from the app?	Yes
R3	Below 18	Family Members	Is the notifications delayed?	No
R3	Below 18	Family Members	Which notifications did you receive?	Not Sure
R3	Below 18	Family Members	Were the notifications useful?	4
R3	Below 18	Family Members	Notification-related pages made sense	4.0
R3	Below 18	Family Members	Overall, how easy was the app to use?	4
R3	Below 18	Family Members	How stable did the app feel?	5
R3	Below 18	Family Members	How would you rate the app design?	5
R3	Below 18	Family Members	How useful do you think this app is for family or close group communication?	4
R3	Below 18	Family Members	Which feature(s) did you like the most?	I like the AI feature, it was fast I replying, explained well how to solve my emotional problems.
R3	Below 18	Family Members	Which feature(s) needs the most improvement?	Chat feature, could improve in adding emojis, allow tagging other members.
R3	Below 18	Family Members	Would you continue using the app if it was improved and completed?	Maybe
R3	Below 18	Family Members	Did you experience any bugs, crashes, freezing, or wrong display?	Yes
R3	Below 18	Family Members	What bug(s) did you find?	Location feature, members name are hidden.
R3	Below 18	Family Members	Can the bug happen again if you repeat the same steps?	Yes
R3	Below 18	Family Members	Any final comment for the developer?	Good app.
R4	18-24	Friends	Timestamp	4/26/2026 14:35:13
R4	18-24	Friends	Were you able to install and open the app successfully?	Yes
R4	18-24	Friends	Was the login/sign-up/guest access process easy to understand?	5
R4	18-24	Friends	Did you face any problem during account setup or profile setup?	No problem
R4	18-24	Friends	What was your first impression of the app design?	UI design is good, easy to understand
R4	18-24	Friends	Were you able to create or join a group successfully?	Yes
R4	18-24	Friends	Was the group code / join group process clear?	5
R4	18-24	Friends	Were chat messages sent and received properly?	Somtimes delayed
R4	18-24	Friends	Did image upload in chat work properly	Yes
R4	18-24	Friends	Chat and group pages made sense	5
R4	18-24	Friends	Suggestions for chat/group feature	It seems like cannot delete or replied sent messages directly
R4	18-24	Friends	Did the emotion display page load properly?	Yes
R4	18-24	Friends	Was it clear which member's emotion was being displayed?	3
R4	18-24	Friends	Was the dominant emotion easy to notice?	5
R4	18-24	Friends	Did the emotion result feel reasonable based on the message?	4
R4	18-24	Friends	Emotion page made sense	3

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R4	18-24	Friends	Suggestions for emotion feature	Not really sure the purpose for this feature, I didn't notice too much while using the app
R4	18-24	Friends	Were you able to create an event successfully?	Yes
R4	18-24	Friends	Was the event creation form easy to understand?	5
R4	18-24	Friends	Did the event appear correctly in the Event page?	Yes
R4	18-24	Friends	Was weekly calendar navigation clear?	5
R4	18-24	Friends	Did the event appear correctly in the Calendar page?	Yes
R4	18-24	Friends	Event and calendar pages made sense	5
R4	18-24	Friends	Suggestions for event/calendar feature	I think this feature is good enough
R4	18-24	Friends	Were you able to find and open the AI family advice page?	Yes
R4	18-24	Friends	Were you able to ask a question and receive a reply?	Yes
R4	18-24	Friends	Was the AI reply relevant to the question?	5
R4	18-24	Friends	Was the AI reply easy to understand?	5
R4	18-24	Friends	Did the previous inquiries/history function work?	Did not test
R4	18-24	Friends	AI advice page made sense	5
R4	18-24	Friends	Suggestions for AI advice feature	Only have limited topic can be discussed, maybe can provide more
R4	18-24	Friends	Did you receive notifications from the app?	No
R4	18-24	Friends	Is the notifications delayed?	No
R4	18-24	Friends	Which notifications did you receive?	Not Sure
R4	18-24	Friends	Were the notifications useful?	3
R4	18-24	Friends	Notification-related pages made sense	3.0
R4	18-24	Friends	Suggestions for notification feature	Not sure, I have never received any notifications
R4	18-24	Friends	Overall, how easy was the app to use?	5
R4	18-24	Friends	How stable did the app feel?	5
R4	18-24	Friends	How would you rate the app design?	5
R4	18-24	Friends	How useful do you think this app is for family or close group communication?	5
R4	18-24	Friends	Which feature(s) did you like the most?	event, like everyone will have a clear figure for the upcoming event that someone planned or it can inform other people effectively.
R4	18-24	Friends	Which feature(s) needs the most improvement?	The problem for messages I mention before, not much interaction between messages in the group chat.
R4	18-24	Friends	Would you continue using the app if it was improved and completed?	Yes
R4	18-24	Friends	Did you experience any bugs, crashes, freezing, or wrong display?	No
R4	18-24	Friends	Any final comment for the developer?	Overall is a good app actually, quite fun
R5	18-24	Friends	Timestamp	4/27/2026 9:12:53
R5	18-24	Friends	Were you able to install and open the app successfully?	Yes
R5	18-24	Friends	Was the login/sign-up/guest access process easy to understand?	2

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R5	18-24	Friends	Did you face any problem during account setup or profile setup?	No problem
R5	18-24	Friends	What was your first impression of the app design?	readability
R5	18-24	Friends	Were you able to create or join a group successfully?	Yes
R5	18-24	Friends	Was the group code / join group process clear?	5
R5	18-24	Friends	Were chat messages sent and received properly?	Yes
R5	18-24	Friends	Did image upload in chat work properly	Yes
R5	18-24	Friends	Chat and group pages made sense	5
R5	18-24	Friends	Suggestions for chat/group feature	the profile pic of other cannot enlarge to see
R5	18-24	Friends	Did the emotion display page load properly?	Yes
R5	18-24	Friends	Was it clear which member's emotion was being displayed?	3
R5	18-24	Friends	Was the dominant emotion easy to notice?	5
R5	18-24	Friends	Did the emotion result feel reasonable based on the message?	4
R5	18-24	Friends	Emotion page made sense	4
R5	18-24	Friends	Suggestions for emotion feature	let people can choice the emotion
R5	18-24	Friends	Were you able to create an event successfully?	Yes
R5	18-24	Friends	Was the event creation form easy to understand?	4
R5	18-24	Friends	Did the event appear correctly in the Event page?	Yes
R5	18-24	Friends	Was weekly calendar navigation clear?	5
R5	18-24	Friends	Did the event appear correctly in the Calendar page?	Yes
R5	18-24	Friends	Event and calendar pages made sense	3
R5	18-24	Friends	Suggestions for event/calendar feature	it should combine both page to one
R5	18-24	Friends	Were you able to find and open the AI family advice page?	Yes
R5	18-24	Friends	Were you able to ask a question and receive a reply?	Yes
R5	18-24	Friends	Was the AI reply relevant to the question?	2
R5	18-24	Friends	Was the AI reply easy to understand?	3
R5	18-24	Friends	Did the previous inquiries/history function work?	Did not test
R5	18-24	Friends	AI advice page made sense	2
R5	18-24	Friends	Suggestions for AI advice feature	the ans of ai is very limit
R5	18-24	Friends	Did you receive notifications from the app?	Yes
R5	18-24	Friends	Is the notifications delayed?	No
R5	18-24	Friends	Which notifications did you receive?	Chat Messages
R5	18-24	Friends	Were the notifications useful?	5
R5	18-24	Friends	Suggestions for notification feature	good
R5	18-24	Friends	Overall, how easy was the app to use?	4

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R5	18-24	Friends	How stable did the app feel?	5
R5	18-24	Friends	How would you rate the app design?	4
R5	18-24	Friends	How useful do you think this app is for family or close group communication?	4
R5	18-24	Friends	Which feature(s) did you like the most?	emoji
R5	18-24	Friends	Which feature(s) needs the most improvement?	event page
R5	18-24	Friends	Would you continue using the app if it was improved and completed?	Maybe
R5	18-24	Friends	Did you experience any bugs, crashes, freezing, or wrong display?	No
R5	18-24	Friends	Any final comment for the developer?	
R6	18-24	Friends	Timestamp	4/27/2026 11:34:23
R6	18-24	Friends	Were you able to install and open the app successfully?	Yes
R6	18-24	Friends	Was the login/sign-up/guest access process easy to understand?	5
R6	18-24	Friends	Did you face any problem during account setup or profile setup?	No problem
R6	18-24	Friends	What was your first impression of the app design?	Layout
R6	18-24	Friends	Were you able to create or join a group successfully?	Yes
R6	18-24	Friends	Was the group code / join group process clear?	5
R6	18-24	Friends	Were chat messages sent and received properly?	Yes
R6	18-24	Friends	Did image upload in chat work properly	Yes
R6	18-24	Friends	Chat and group pages made sense	4
R6	18-24	Friends	Suggestions for chat/group feature	Hope can upload file and video to the chat
R6	18-24	Friends	Did the emotion display page load properly?	Yes
R6	18-24	Friends	Was it clear which member's emotion was being displayed?	3
R6	18-24	Friends	Was the dominant emotion easy to notice?	5
R6	18-24	Friends	Did the emotion result feel reasonable based on the message?	2
R6	18-24	Friends	Emotion page made sense	3
R6	18-24	Friends	Suggestions for emotion feature	I think this feature is not necessary
R6	18-24	Friends	Were you able to create an event successfully?	Yes
R6	18-24	Friends	Was the event creation form easy to understand?	5
R6	18-24	Friends	Did the event appear correctly in the Event page?	Yes
R6	18-24	Friends	Was weekly calendar navigation clear?	5
R6	18-24	Friends	Did the event appear correctly in the Calendar page?	Yes
R6	18-24	Friends	Event and calendar pages made sense	5
R6	18-24	Friends	Suggestions for event/calendar feature	no

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R6	18-24	Friends	Were you able to find and open the AI family advice page?	Yes
R6	18-24	Friends	Were you able to ask a question and receive a reply?	Sometimes delayed
R6	18-24	Friends	Was the AI reply relevant to the question?	4
R6	18-24	Friends	Was the AI reply easy to understand?	4
R6	18-24	Friends	Did the previous inquiries/history function work?	Yes
R6	18-24	Friends	AI advice page made sense	5
R6	18-24	Friends	Suggestions for AI advice feature	no
R6	18-24	Friends	Did you receive notifications from the app?	Yes
R6	18-24	Friends	Is the notifications delayed?	No
R6	18-24	Friends	Which notifications did you receive?	Chat Messages, Event Creation, Group Updates (New member join, nickname change....)
R6	18-24	Friends	Were the notifications useful?	5
R6	18-24	Friends	Suggestions for notification feature	no
R6	18-24	Friends	Overall, how easy was the app to use?	5
R6	18-24	Friends	How stable did the app feel?	4
R6	18-24	Friends	How would you rate the app design?	4
R6	18-24	Friends	How useful do you think this app is for family or close group communication?	5
R6	18-24	Friends	Which feature(s) did you like the most?	Group chat, it is very clear
R6	18-24	Friends	Which feature(s) needs the most improvement?	Emotional display, I think it is not necessary
R6	18-24	Friends	Would you continue using the app if it was improved and completed?	Yes
R6	18-24	Friends	Did you experience any bugs, crashes, freezing, or wrong display?	No
R6	18-24	Friends	Can the bug happen again if you repeat the same steps?	No
R7	18-24	Friends	Timestamp	4/27/2026 14:29:01
R7	18-24	Friends	Were you able to install and open the app successfully?	Yes
R7	18-24	Friends	Was the login/sign-up/guest access process easy to understand?	5
R7	18-24	Friends	Did you face any problem during account setup or profile setup?	No problem
R7	18-24	Friends	What was your first impression of the app design?	layout
R7	18-24	Friends	Were you able to create or join a group successfully?	Yes
R7	18-24	Friends	Was the group code / join group process clear?	5
R7	18-24	Friends	Were chat messages sent and received properly?	Yes
R7	18-24	Friends	Did image upload in chat work properly	Yes
R7	18-24	Friends	Chat and group pages made sense	5
R7	18-24	Friends	Suggestions for chat/group feature	Excellent
R7	18-24	Friends	Did the emotion display page load properly?	Yes

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R7	18-24	Friends	Was it clear which member's emotion was being displayed?	5
R7	18-24	Friends	Was the dominant emotion easy to notice?	5
R7	18-24	Friends	Did the emotion result feel reasonable based on the message?	5
R7	18-24	Friends	Emotion page made sense	5
R7	18-24	Friends	Suggestions for emotion feature	Excellent
R7	18-24	Friends	Were you able to create an event successfully?	Yes
R7	18-24	Friends	Was the event creation form easy to understand?	5
R7	18-24	Friends	Did the event appear correctly in the Event page?	Yes
R7	18-24	Friends	Was weekly calendar navigation clear?	5
R7	18-24	Friends	Did the event appear correctly in the Calendar page?	Yes
R7	18-24	Friends	Event and calendar pages made sense	5
R7	18-24	Friends	Suggestions for event/calendar feature	Excellent
R7	18-24	Friends	Were you able to find and open the AI family advice page?	Yes
R7	18-24	Friends	Were you able to ask a question and receive a reply?	Yes
R7	18-24	Friends	Was the AI reply relevant to the question?	5
R7	18-24	Friends	Was the AI reply easy to understand?	5
R7	18-24	Friends	Did the previous inquiries/history function work?	Yes
R7	18-24	Friends	AI advice page made sense	5
R7	18-24	Friends	Suggestions for AI advice feature	Excellent
R7	18-24	Friends	Did you receive notifications from the app?	Yes
R7	18-24	Friends	Is the notifications delayed?	No
R7	18-24	Friends	Which notifications did you receive?	Chat Messages
R7	18-24	Friends	Were the notifications useful?	5
R7	18-24	Friends	Suggestions for notification feature	Excellent
R7	18-24	Friends	Overall, how easy was the app to use?	5
R7	18-24	Friends	How stable did the app feel?	5
R7	18-24	Friends	How would you rate the app design?	5
R7	18-24	Friends	How useful do you think this app is for family or close group communication?	5
R7	18-24	Friends	Which feature(s) did you like the most?	View location, can see friends' locations at any time.
R7	18-24	Friends	Which feature(s) needs the most improvement?	Emotional details, maybe can draw own expressions
R7	18-24	Friends	Would you continue using the app if it was improved and completed?	Yes
R7	18-24	Friends	Did you experience any bugs, crashes, freezing, or wrong display?	No
R7	18-24	Friends	Can the bug happen again if you repeat the same steps?	No
R8	18-24	Friends	Timestamp	4/28/2026 15:01:54

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R8	18-24	Friends	Were you able to install and open the app successfully?	Yes
R8	18-24	Friends	Was the login/sign-up/guest access process easy to understand?	5
R8	18-24	Friends	Did you face any problem during account setup or profile setup?	No problem
R8	18-24	Friends	What was your first impression of the app design?	A bit monotonous, can add some background changing function.
R8	18-24	Friends	Were you able to create or join a group successfully?	Yes
R8	18-24	Friends	Was the group code / join group process clear?	5
R8	18-24	Friends	Were chat messages sent and received properly?	Yes
R8	18-24	Friends	Did image upload in chat work properly	Yes
R8	18-24	Friends	Chat and group pages made sense	5
R8	18-24	Friends	Suggestions for chat/group feature	Can add private message function.
R8	18-24	Friends	Did the emotion display page load properly?	Yes
R8	18-24	Friends	Was it clear which member's emotion was being displayed?	5
R8	18-24	Friends	Was the dominant emotion easy to notice?	5
R8	18-24	Friends	Did the emotion result feel reasonable based on the message?	5
R8	18-24	Friends	Emotion page made sense	5
R8	18-24	Friends	Suggestions for emotion feature	Good idea, very special. Can add more emotions to let it more accurate for every situations.
R8	18-24	Friends	Were you able to create an event successfully?	Yes
R8	18-24	Friends	Was the event creation form easy to understand?	5
R8	18-24	Friends	Did the event appear correctly in the Event page?	Yes
R8	18-24	Friends	Was weekly calendar navigation clear?	5
R8	18-24	Friends	Did the event appear correctly in the Calendar page?	Yes
R8	18-24	Friends	Event and calendar pages made sense	5
R8	18-24	Friends	Suggestions for event/calendar feature	No, I think this part is good nothing need to change.
R8	18-24	Friends	Were you able to find and open the AI family advice page?	Yes
R8	18-24	Friends	Were you able to ask a question and receive a reply?	Yes
R8	18-24	Friends	Was the AI reply relevant to the question?	3
R8	18-24	Friends	Was the AI reply easy to understand?	4
R8	18-24	Friends	Did the previous inquiries/history function work?	Yes
R8	18-24	Friends	AI advice page made sense	5
R8	18-24	Friends	Suggestions for AI advice feature	Still can improve because the answer is not very specific, it just give the general answer, I expect that it can give me a complete plan not only what I can do then I need to do the whole myself.
R8	18-24	Friends	Did you receive notifications from the app?	Yes
R8	18-24	Friends	Is the notifications delayed?	No

Respondent ID	Age Range	Relationship Group	Question / Field	Response
R8	18-24	Friends	Which notifications did you receive?	Chat Messages
R8	18-24	Friends	Were the notifications useful?	5
R8	18-24	Friends	Suggestions for notification feature	No suggestion.
R8	18-24	Friends	Overall, how easy was the app to use?	5
R8	18-24	Friends	How stable did the app feel?	5
R8	18-24	Friends	How would you rate the app design?	4
R8	18-24	Friends	How useful do you think this app is for family or close group communication?	4
R8	18-24	Friends	Which feature(s) did you like the most?	Gallery. Because I can search all the photos easily. This very useful I like it most.
R8	18-24	Friends	Which feature(s) needs the most improvement?	AI adviser. I hope it can have a more specific answer or plan that I can use it directly. Not just give me an idea then I still need to search everything myself.
R8	18-24	Friends	Would you continue using the app if it was improved and completed?	Maybe
R8	18-24	Friends	Did you experience any bugs, crashes, freezing, or wrong display?	No
R8	18-24	Friends	What should be added to the app in the future?	A friend list and private message function.
R8	18-24	Friends	Any final comment for the developer?	Very good but still can improve such as the background can let users design themselves and the AI adviser need to be improved most.

POSTER

Bachelor of Information Technology (Honours) Computer Engineering

Faculty of Information and Communication Technology (Kampar Campus), UTAR