VOICE-ACTIVATED DIARY WITH TEXT CONVERSION : TRANSFORMING SPOKEN WORDS INTO DIGITAL MEMORY

BY

WILSON FOOK WEI SHENG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

FEBRUARY 2026

ACKNOWLEDGEMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr Tan Joi Shan, for their continuous guidance, support, and encouragement throughout the development of this Final Year Project 2. Their valuable advice, constructive feedback, and patience have greatly contributed to the completion of this work.

I would also like to extend my appreciation to the lecturers and staff of Universiti Tunku Abdul Rahman (UTAR), who have provided me with the necessary knowledge, resources, and facilities that made this project possible.

Special thanks go to my family for their endless support, love, and understanding during this period of study. Their encouragement has been a source of motivation and strength throughout my academic journey.

Lastly, I would like to thank my friends and course mates for their cooperation, discussions, and moral support. Their companionship has made this journey more meaningful and enjoyable.

COPYRIGHT STATEMENT

This Final Year Project proposal is submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Honours) at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project proposal represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project proposal 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.

ABSTRACT

Many people find it challenging to keep writing in a diary because traditional handwritten diaries are time-consuming and difficult to store. Therefore, this project was created to provide users with a faster, easier, and more modern way to journal by using their voice. Voice Diary is built by using Visual Studio Code with Flutter SDK and Dart language. The key features include voice-to-text conversion and AssemblyAI use as a speech recognition tool to convert voice messages into text messages that support multiple languages. Mood detection features are included in this mobile app using the DistillBERT Base Uncased GoEmotions Student sentiment analysis library. Other modules like tagging, search, calendar integration, multimedia embedding, emotional detection, location and weather detection, theme colour selection, admin dashboard, pin protection, and community diary sharing are implemented to enhance users' experience. Agile methodology by Scrum framework is implemented in this project to guide the iterative development of the application. Voice Diary allows users to record their diaries using voice and then automatically converted into text messages. Users can track their emotional trends over time using mood ratings and graphs. Voice Diary turns the conventional journaling approach into an advanced yet practical and emotionally reflective activity. This is because Voice Diary makes use of voice-based technologies together with some sophisticated capabilities including mood detection and trend analysis to make journal writing more convenient and reflective. According to the outcomes of user testing, this suggested solution was effective in ensuring high usability and user satisfaction.

Area of Study: Mobile Application Development, Human-Computer Interaction (HCI)

Keywords: Mobile Application, Firebase, Flutter, Speech-to-Text, Emotion Detection, Community Sharing System

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
APK	Android Package Kit
CRUD	Create, Read, Update, Delete
DBMS	Database Management System
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
OS	Operating System
UTP	Univesiti Teknologi Petronas
XML	Extensible Markup Language
SQL	Structured Query Language
UI	User Interface
PHP	Hypertext Preprocessor
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
DBMS	Database Management System
IDE	Integrated Development Environment
SDK	Software Development Kit
ASR	Automatic Speech Recognition
UAT	User Acceptance Testing

CHAPTER 1 PROJECT BACKGROUND

1.1 Overview

A Diary is a record of daily events that may affect the emotion of the writer, or a personal record of events that are stored by people for their own purposes. A personal diary acts as a reflection tool that records the personal experience, thoughts, memory preservation, feelings, and tracking of their progress of work. Traditionally, diaries are displayed in handwritten formats that are not easy to store, they need to be store at a dry and safe place to prevent damage from moisture, fire, or wear and tear. However, handwritten diaries can be difficult to store and organize because of the number of entries that grow.

People use diaries for various reasons, such as expressing thoughts that they may not share with others, planning for the future, and acting as a private space for self-reflection. This allows people to write down their feelings and ideas without fear of judgment from others. It provides a tree of holes that allows people to write down the secret at the bottom of their hearts. Besides emotional expression, people also use diaries for planning and organization. People often write down their plans, outlines of important tasks, and draft ideas of exercise because this will help them maintain productivity during their free time. The function of a diary is not limited to these, some individuals may use it for specialized diaries, including gratitude journals, dream journals, and travel journals. Each of them serves a unique and unmatched function in personal development.

With advancements in technology, the traditional handwritten diary is being replaced by the digital diary, which provides security, and accessibility. So that a mobile application is developed to allow people to create their own new generation of diary to let the diary be stored in a digital way for longer periods of time. Digital diaries allow people to store unlimited diaries, easily search past diaries, and integrate photos and voice recordings. Additionally, a digital diary can provide enhanced privacy features like password protection of their personal diary to make sure that personal reflection is more secure. These changes will cause the demand for records of personal memories in more convenient and efficient ways.

1.2 Problem Statement

Nowadays, individuals find that it is difficult to maintain the habit of writing a diary day by day. With the busy schedules and the increasing reliance on technology, more people prefer to

record diaries in a digital version, as it provides many advantages compared to handwritten diaries. This is due to the difficulty of storing and preserving physical diaries, the lack of motivation of existing diary solutions, and the inconvenience of a handwritten diary.

i. Difficulty of storing and preserving physical diaries

Physical diaries need a physical place in a dry and safe condition to prevent damage from fire, moisture, or aging. The pages will become unreadable or even get lost due to the pages deteriorating or the ink fading, resulting in the loss of some personal memories and reflections. Additionally, handwritten diaries need physical space for storing, and over time the large quantity of diaries makes it hard for individuals to keep and organize them wisely. Due to the lack of a digital search function, if users need to search for a specific entry, they must search through pages physically to retrieve previous entries. The process becomes time-consuming and inefficient. Many people feel sadness, depression and loss when they are unable to find past entries or when they lose precious memories due to unforeseen circumstances.

ii. Lack of motivation of existing diary solutions

Traditional diaries serve as tools for self-expression, but they do not have any elements that are interactive to keep users from engaging with them. Most people start journaling with passion but eventually lose their interest or forget to write due to lack of reminders and feedback from their entries. Writing in a diary can feel like a repetitive daily task, and this is one of the major reasons why people stop journaling. People will find it tedious and boring to write about their expected and predictable daily routine because it is unexciting. This leads to writing becoming a task that needs to be completed rather than an enjoyable activity. Writing a diary is an activity that will relieve stress, but users may feel isolated because they do not receive feedback and support in their journaling. Furthermore, journaling needs users to put in their discipline and persistence, which many users find hard to maintain.

iii. Inconvenience of handwritten diary

Handwriting a diary entry each day requires much time and effort, making it difficult for many to maintain a consistent journaling habit. Many people struggle to find time to sit down and write in a diary, especially when balancing work, studies, and personal tasks. Before writing a diary, people need to organize their thoughts, which will discourage individuals from

journaling regularly. Handwritten diaries require people to turn pages to find past entries. If someone wanted to recap their specific memory or track progress over time, it would become inefficient. Handwritten diaries are difficult for users to store photos, videos or other multimedia content. In this digital era, people capture significant moments by using smartphones and cameras, but traditional diaries do not provide any way to integrate these elements. While some may print out the photo to attach to the diary, this method will be inconvenient and bulky. Lack of multimedia support causes users to not be able to combine their visual memories and voice recordings which could enhance their reflections.

1.3 Project Objective

The primary objective of this project is to develop a mobile application called “Voice Diary.” This application allows users to record and store their daily entries by using voice recording and then converting them into text messages. It aims to solve common problems faced with traditional handwritten diaries by enhancing convenience, interactivity, and accessibility through modern technologies.

i. To solve storage issues and improve the organization of diary entries

By using a mobile application, a diary can be stored in digital storage to prevent the risk of damage from fire, moisture, or aging. The diary entries will be stored securely in the devices of users without the need for physical space. A searching function and tagging system will be implemented into the application for easy categorization and navigation of past entries. Users can search previous entries by using the key in the keywords, tags, and date.

ii. To increase engagement through emotion detection and mood rating

The implementation of automatic emotion detection through the content of a user's diary will provide interaction with the user. The emotional pattern will be represented in statistical analysis by using a graphical chart. Besides automatic emotion detection, users can use manual mood ratings that are contained in the diary to make the emotional detection system be more accurate. Users can track their emotional pattern over time through the combination of manual and automatic mood rating features. This function will attract user interest and cultivate the habits of journaling because it makes the journal-writing process more interesting and enjoyable.

iii. To simplify the diary process using voice-to-text technology

Users just need to press a recording button, then speak to the device, and then the voice will automatically convert into a text message to save in the diary entries. By using voice-to-text technology, users can save time typing words to record their daily routine without having to use it like other digital diary apps. After the text conversion, the user can review and edit the transcribed text to ensure accuracy and personalization.

1.4 Project Scope

The scope of the "Voice Diary" application includes the design, development, and deployment of a mobile app to record the diary entries of users in the form of voice recordings. The main function of this app is the conversion of a voice recording to a text message, which can enhance user experience in journaling. Other features include detecting emotion from the journal's entries, multimedia embedding, tags that separate the diary, predefined templates, statistical analysis, and a search function for past entries.

i. Voice-to-text conversion module

Users can press a button to start the voice recording, and then the recording will automatically convert into a text message to convenience the user by replacing the typing process. The converted text will be stored in the database of the app, and the user will be allowed to review and edit it. This will speed up the journaling process because voice-to-text conversion is fast and simple. It also support multi-language voice-to-text convert and have a translation to translate others language like Malay, Chinese, Japanese to English for sentiment analysis.

ii. Emotion detection module

The app can analyse a user's diary content to detect the emotion of user at that specific time. Besides the automatic detection, the user is also allowed to rate their mood manually by rating the emotion percentage. Users are allowed to track and visualize their emotions insightfully.

iii. Multimedia embedding module

Adding photos, videos, and files can help enrich users' diary entries and increase their impression. Attaching a photo and video from the user's gallery can act as visual memories that can enhance the emotional expression and immerse the user.

iv. Tagging system module

Users can assign tags to each diary entry to ease the searching step of previous entries. Users can choose the predefined tags or manually add tags like vacation, work, family, and so on. This is helpful in the searching process of users to find their previous diary entries after a long time and improve overall diary organization.

v. Search function module

The searching function is important to allow users to find their past entries by using keywords or tags. At the top of the search pages, there can be a search bar to allow users to search previous entries, and then it will show the results. Users can choose to search by keywords, tags, mood, date, or photos that are attached to the diary. This can ensure that users can find past entries without scrolling through the entire journal.

vi. Searchable calendar interface

Users can know the day that they have done their diaries by just viewing the simple calendar pages. The calendar interface will have a blue dot below the dates with saved diary entries to let the user identify them easily. Users click on the date and then directly shown the previous diary entries.

vii. Location & weather detection module

The apps will capture the current location and weather data at the time of making a diary entry. The location of the user is detected by the device's GPS, and the coordinates will connect with the WeatherAPI.com to fetch the weather conditions such as temperature, humidity, and sky status. This contextual information will be stored together with the diary entry to provide a reflection of the user's thoughts on weather conditions.

viii. Theme colour selection module

This can let user personalize the overall look of their diary application. Users can select six different theme colors that match their preferences or moods. Its target is to enable users to choose from a pool of pre-decided theme colours, thus increasing eye comfort and user satisfaction without compromising basic functionality.These customization features can enhance user experience make the diary entry feel more personal and engaging. This can improve the overall user satisfaction and encourage app usage.

ix. Mood analysis dashboard module

Mood Analysis dashboard helps users monitor trends in emotion, reflect upon patterns, and enhance their understanding of their mental and emotional well-being. This provides users with a sense of their emotional well-being over long-term analysis of their diary material. This receives both textual sentiment analysis (e.g., positive, neutral, or negative mood identification) and user-rated mood inputs. The output is displayed on a specialized dashboard that has statistical summaries, charts, and mood distribution findings.

x. Login and signup module

Signup and login module aims at delivering safe user authentication and account management for the diary program. The main task is to make sure that every user has a customized and secured account that gives access to private diary content, settings, and user-specific features like mood history and multimedia storage. It provides new users with the chance for registering for an account, and existing users can securely authenticate by means of their registered credentials. It also offers account recovery and makes sure that it integrates smoothly into Firebase Authentication services for safety and scalability.

xi. Community diary sharing module

Community Diary Sharing Module is built to change the diary app from a personal journaling app into a social interaction platform. It aims at allowing users to share some of their chosen diary content with the broader community, allowing for interaction, linking, and shared storytelling while maintaining user control over privacy. It offers a special community section where shared content is viewed within a feed-style interface based on access by all users. The coverage covers community interaction features like user likes and

comments and user profiles, while maintaining that privacy and moderation controls are in place.

xii. Admin Dashboard Module

Admin Dashboard Module is a feature which enables the system administrator to track all activities taking place within the system. It acts as a source of data for the administrator. With this module, the administrator will be able to view user information, take care of post made by users in the community section, and analyze statistics generated through diaries written by the users. From the dashboard, the admin will be able to view all users who have registered on the application including details concerning the status of their accounts, whether active or not.

xiii. Mood Trend Notification Module

Mood Trend Notification Module is designed to track users' moods over time and deliver proactive assistance through the notification process. In order for the module to work properly, it keeps analysing the mood results that have been derived based on the information provided by the users in their diaries. If the system finds out that there is a constant negative mood trend involving the recurring appearance of emotions like sadness, anger, and fear for a predetermined number of entries, the pop-up notification will appear.

xiv. Password PIN Protection Module

Password PIN Protection Module has been created to enhance security and confidentiality of the diaries written by the users of the software program. Because there may be some confidential details in the diary entry that belong to the user, the Password PIN Protection Module ensures that only authorized people can gain access to the diary entry. As soon as the PIN protection option is activated, you will be required to enter your PIN.

1.5 Proposed Approach / Study

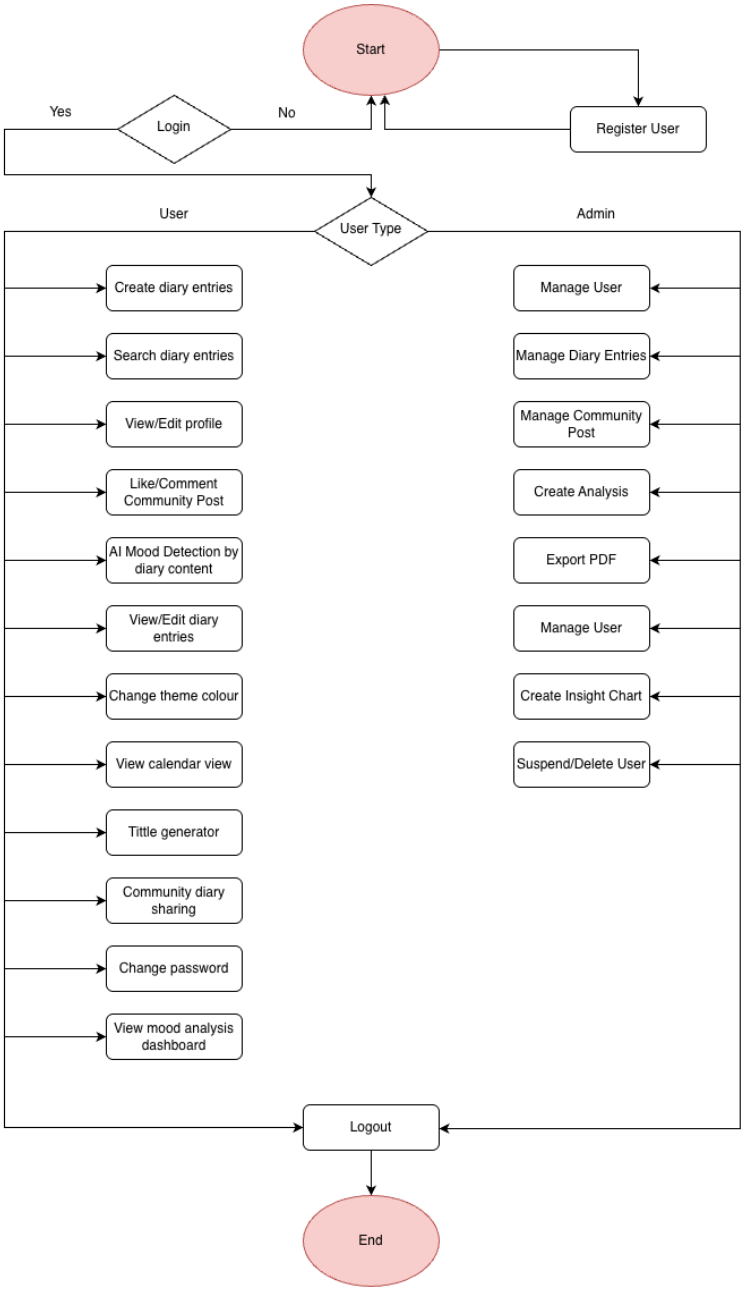


Figure 1.5.1 Flowchart of Voice Diary Application

Figure 1.5.1 is the flowchart that show how is the Voice Diary Application works. It is showing the interaction between user and admin role, the system starts with the authentication then separated into two different branches based on user types. The flow begins with the starting point where user expected to sign into the system. If user does not have any account, user will guide to the process of registration in the system. After done of registration, user will be possible to log into the system, and system will identify whether the user role is a user or admin.

If it is a User, he or she is allowed to use the various functions in the diaries. Some of them include creation of diary entries by using text/voice input. Additionally, the system helps users to search and view various previously made diary entries. Furthermore, the user can modify any of the diary entries that he or she wishes to. For customizing purpose, the user is allowed to see and modify his or her profile details. Moreover, the system has a function of allowing the user to choose his or her application color themes. Also, the system features social media integration that allows users to publish certain diary entries on community section. Other users can also comment and give likes to the diary entries that have been posted. In addition, users can also look at the correlation and trend of their moods by accessing the mood analysis dashboard.

Admin will be led into the administrative features of the system. The admin is the one in charge of user management, which includes monitoring of the user profile and doing operations like suspending and deleting of profiles. Furthermore, the admin can manage the diaries and posts made by other users to ensure that proper and suitable contents are provided in the system. There is the feature of analytics in the system for the benefit of the administrator, where the admin is allowed to make mood analysis reports and insights charts.

1.6 Impact, significance, and contribution

The development of the Voice Diary application for users can provide a meaningful impact on individuals who wish to record their lives consistently. It offers a new way of personal reflection by using voice activation and emotion tracking to replace traditional handwritten journals. Users just need to press a button to record their moods and events of the moment. The voice-to-text conversion features will reduce the time and effort needed for journaling and make it faster and more accessible for people who have busy lifestyles. It reduces the users' stress and suffering in the journal process. The voice recognition relies on the AssemblyAI API, an cloud based speech recognition service that converts spoken words into text. As an online API, users must have internet service to have a seamless diary experience. This will provide fast, scalable, and accurate voice-to-text conversion experience for users.

The emotional detection and mood tracking features help users to identify and record their emotions day by day. This encourages users to reflect on their feelings to contribute to good mental health and self-awareness. Users need to rate their mood after each diary process or the

automatic analysis of emotion based on the content of their entry. The collected mood data will be displayed in a visual graph to help users recognize their emotional patterns. Users are not just journaling in this app, they also receive real value from emotional insight to let them stay consistent in journaling.

Digital storage of diary entries can help users to prevent missing and lost entries due to physical damage. Digital storage will enhance the security, durability, and longevity of personal diary entries. Users can choose to back up their entries to cloud storage. If a mobile device is lost, damaged, or replaced, users can have a seamless experience with multiple devices, such as phones or tablets. This will ensure no happening of losing memories that cause emotional distress to the user.

1.7 Summary

This chapter discusses the scope, objective, and contributions of the Voice Diary project. The apps are used to provide a way to use voice to record their diary entries. The modules proposed for this project are voice-to-text conversion, emotion detection, multimedia embedding, a tagging system, a search function, a searchable calendar interface, location and weather detection, theme colour selection, mood analysis dashboard, login and signup, community diary sharing, mood trend notification, and password PIN protection. The definition and significance of a diary are introduced in the overview, highlighting its help in self-reflection, emotional expression, and personal organization. However, there are three key problems in traditional handwritten diaries, which make it difficult for individuals to maintain consistent journaling habits. Thus, creating a voice-activated diary with text conversion is needed to provide a more convenient and effective journaling experience.

CHAPTER 2 LITERATURE REVIEW

2.1 Overview

This chapter will discuss and review existing applications and projects that are relevant to the diary. The few examples of existing applications are used as a guideline while developing a new voice diary mobile application. Besides, a review of the existing project is to understand the project objective and the process of developing the project. The existing applications are reviewed, and their strengths and limitations are compared to provide a summary.

2.2 Similar project

2.2.1 Mobile Application for Personal Journal/Diary (Nazrin 2020)

Mobile Application for Personal Journal/Diary is a project that was developed by a student at Universiti Teknologi PETRONAS (UTP) in the year 2013. The objective of this project is to study the expressive writing used in diary writing, develop a mobile application that resembles a physical diary, and test and evaluate users' behavior based on their experience in using the diary application. It is used to record the individual's series of daily life experiences or thoughts [1].

Interactive and incremental development are used in this project as its model framework, which is useful for projects that require continuous improvement and regular feedback. At the initial planning stage, the author conducted a survey to get the data from students at UTP by using e-mail. Milestones are used to monitor the progress and accomplish the project. Gantt charts are generated for tracking the progress of the project.

Furthermore, the authors find that modern mobile application architectures are the most suitable architectures to implement in this project. This architecture consists of three different layers, which are client-side, server-side, and database. Client-side is the interface and application login on the device, which framework was developed by XML and programmed using the JAVA language. Server-side acts as bridge that connects the database and the client using cloud servers. Databases are the data stored by using SQL language.

To develop an application, the author has used the Android Software Development Kit (SDK) to build the diary application. Adobe Photoshop is used for editing and developing the basic

framework for the application. Parse acts as a cloud database service for the application to store the relevant data.

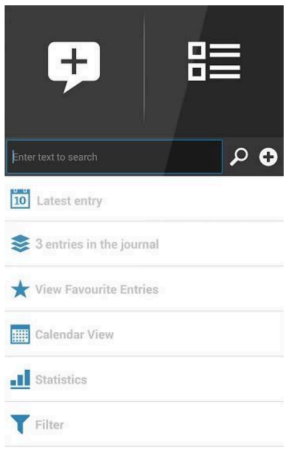


Figure 2.2.1.1 Final Prototype of Diary Application [1]

This is the final prototype of the diary application. The top right icon is for the user to write a new diary post, and the top right side will be the list of views to allow the user to find their previous entries, which are arranged in order. It uses icon to represent, this will make new user be complicated with it. Next will be the search bar, which allows users to find their specific diary with the searching navigation through the tag attached to the diaries. For the latest entry, it is for the user to check back on their last diary entry. Calendar view is an option that let users view their previous' posts by choosing the date they published the diary. Lastly, the statistic function will let users know the number of words that they had written in the last month and the posts per week by a graphical statistics graph.

There were two important features in the new post section that helped users to manage their content. Users are allowed to use the tags function to label their entry to make it easy to be found by using the search bar, which will simplify the searching process for users to find their entry. Users can use the categories feature to separate their diary into different categories such as finance, happy, dating, and so on. There is an indicator about the number of words written by the user in the diary, this is used to remind the user of cautions about the number of words they have written. For the latest post section, users are allowed to upload their multimedia content, such as photos and audio, to make their diary more attractive. There are also other features to make the diary more interesting: mood indicator, weather, and location.

In conclusion, this is a complete diary application that allows user to write their diary in a digital way. It provides two searching features for users to find back the previous entries one is the calendar view by the specific date, and the other is the search bar by searching the tag of the diary. However, the user interface (UI) of this diary application can have some improvement. The UI of the application is not user friendly, this will reduce overall user experience. An intuitive and visually appealing design is essential to make users more comfortable when using the application. Lastly, data security is an important part of software development, the author should improve the security of the diary application because a diary is an individual's personal privacy. To solve this problem, the application should implement strong encryption protocols and two-factor authentication to ensure that the user data are protected.

2.2.2 Web Application "Personal Diary" (Shorokhova 2023)

Web Application "Personal Diary" is a project done by a student named Yevheniia Shorokhova from the University of Finance and Administration. This project shows that digital diaries are more user-friendly compared to traditional handwritten diaries, digital diaries provide a convenient way for users to keep a diary. Traditional diaries require pen and paper, but digital diaries just need a digital device, which is more convenient, and accessible anytime. Text editing capability is the most important feature provided by digital diaries, this keeps the notes neat and organized. The personal diary website is used to solve these problems and provide convenience, enhance organization, and user customization. Digital diaries can be accessed anywhere and anytime on devices that are connected to the internet, allowing for more structured entries to improve user experience. The personal diary website provides customization for users to select, which can enhance both usability and satisfaction [2].

This project uses PHP to write the website. The most important file, function.php**,** is the place that stores all of the core functions, including database operations, user authentication, input handling, session management, page redirection, and database table creation. PHP is used to handle user authentication processes, such as registration, logging in, and managing sessions. It can hash passwords before storing them in the database and verify users during login attempts. Besides, PHP is also used for managing user sessions; the session variable that holds user information will be stored to enhance user experience. PHP can handle and register errors that occur while executing the script. This is useful for troubleshooting problems that occur within the website so that the website runs smoothly.

Criterea	Languages						
	C#	JavaScript	PHP	Python	HTML	CSS	SQL
Purpose	Web applications and services, desktop applications development	Web page development	General-purpose scripting language for web development	Machine learning,data analytics	Creation of web pages and applications	Description of the presentation of documents written in HTML	Store, manage and manioulate data within a databse
Platform	Microsoft Visual Studio,MonoDevelop(Linux), Windows, MacOS	Microsoft Visual Studio, Linux, Windows, MacOS	CakePHP, Laravel, Symfony, Windows, Unix, Linux, MacOS	Linux, MacOS, Windows	Microsoft Visual Studio, Notepad++, Atom, TextEdit(MacOs)	Microsoft Visual Studio, Komodo IDE, Atom	Microsoft Azure/ SQL server, Oracle Database, MySQL
Used for	Flexible memory management	Interactive wep pages	Websites, web/ mobile apps, blogging and more	Conduction of data analysis, writing of python scripts, automating tasks	Building web page structures	Web page design	Database communication, structuring and storing the data
Advantages	Object-oriented, effective memory management, standatd library, fast compilation	Less server interaction, rich interfaces, extreme interactivity	Open-source and free to use, relatively easy to learn, efficient in performance, easy connection to database	Easy to write, high variety of supported libraries, highly efficient	Supported by every browser, free, easy to use	Can also be used in HTML files, consistant design, better device compability	Ensured data integrity, no data duplicates, easily used between different systems
Disadvantages	Low execution speed, does not come with independent compiler, does not allow multiple inheritance	Lack of debugging facility, single inheritance, lack of classes (cannot be concidered fully obkec-oriented)	Not as secure due to being open-source, not suitable for corporate big projects, not as versatile outside of web development	Database limitations, slow runtime speed, high memory consumption	For more modern features needs to be used along with JavaScript or ASP, weak security, the code can get overcomplicated easily	weak security, cross browser issues	Security risks, some versions are priced.

Figure 2.2.2.1 The Language Applies on Web Application “Personal Diary”

HTML, Bootstrap, and CSS are utilized together to create an attractive, responsive, and user-friendly interface. HTML acts as the backbone of website applications because it defines the structure and layout of the pages. In this project, HTML is used to present the diary, navigation element, and user interface component, which ensures the content is accessible. While CSS controls the presentation of elements to make sure that the website is visually engaging and aligned with the project. CSS also enables the implementation of responsive design, which makes the website fit with different sizes of devices, which provides a good user experience across different platforms. Bootstrap has many different built-in components that save time and effort in design. This allows the author to easily create a unique design and a responsive website layout. The combination of HTML, CSS, and Bootstrap will create a polished and interactive website application that meets user needs effectively.

MySQL and XAMPP play an important role in database management and server environment setup. MySQL is the database management system (DBMS) for this diary website. It is used to create and manage user data and diary entries such as usernames, passwords, email addresses, and diary content with timestamps. MySQL can implement create, read, update, and delete (CRUD) operations through SQL queries. When a new diary is written, the website will execute an INSERT statement to store the journal in the database. MySQL also provides mechanisms for ensuring data integrity and security, such as data validation and user authentication to protect user privacy. XAMPP is used to set up a local server on the author’s machine to allow the testing of PHP script and MySQL database interaction. This is essential for debugging to ensure all functionalities are working before the website is published online. MySQL and XAMPP create a good environment for developing the personal diary website.

Database handling is used for creating, managing, and interacting with the MySQL database. The most important are the users and post tables. The users table contains fields like user ID, username, email, and hashed password. The posts table includes fields such as post ID, title, content, date created, photo, and so on. Each post is linked to a specific user, and it ensures that only the user can access through the journal only to protect user privacy. This website also provides an administrative interface for managing users and posts. Users are allowed to change their name, email, password or delete their diaries that are store in the database.

2.3 Existing Application

2.3.1 Day One

Day One is a well-known journaling application that is suitable for users to write down their daily routine [3]. This application is available on macOS, Windows, Android, and iOS, it provides an alternative way to write down their journal in a digital form compared to handwritten diary. This application is known for its intuitive design, simplicity, and powerful features, making it a perfect application for digital journaling. Day One has the maps feature that records the place that the user is attached to in the journal entries.

For the new user, he/she can click on a plus icon on the bottom right side to start writing a diary instantly as shown in Figure 2.3.1.1. Multiple design templates are also provided for the users to simplify their process in completing their daily journal as shown in Figure 2.3.1.2. These templates are selected by users to make it easier to organize and structure their writing. The template includes various layouts to meet different users' needs, such as daily gratitude, to-do lists, mood trackers, goal plans, and travel recaps, which provide for different purposes. This acts as a guideline to help people who do not write a diary regularly or who do not know how to begin. Daily reminders and notifications are also important features to prompt users to write their journal at a specified time every day. This will help users to stay engaged with their journaling habit and develop user patience.

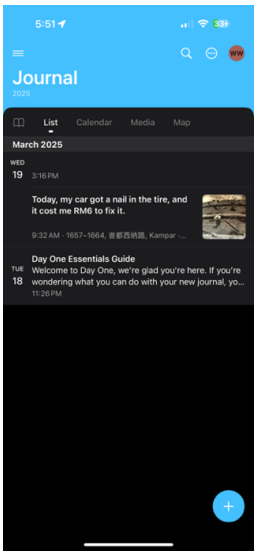


Figure 2.3.1.1 Main Page

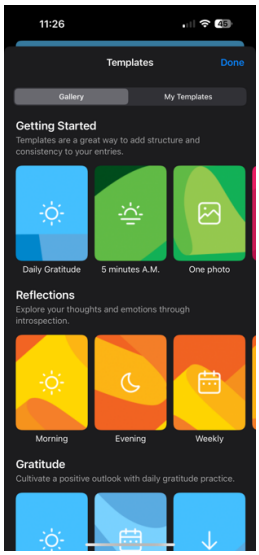


Figure 2.3.1.2 Template Sample

Day One is a cross-platform digital diary application so that it supports seamless synchronization across multiple devices by using Dropbox or iCloud as shown in Figure 2.3.1.3. This ensures users is able to access their journal entries from any device that is connected to the internet. This feature provides flexibility for users to record and view their journal entries anytime and anywhere, which keeps everything up to date. By using these cloud services, users can back up their journal entries online, which provides convenience for multi-device users. The standout feature for digital diaries is the multimedia integration. Day One also provides this feature, which allows users to attach photos, videos, audio, and files to journal entries as shown in Figure 2.3.1.4. Users are allowed to insert photos that are related to the content in their diaries from their gallery with just a click on the screen, and they also can record the audio and then attach it. This makes the journaling experience more interactive and engaging.

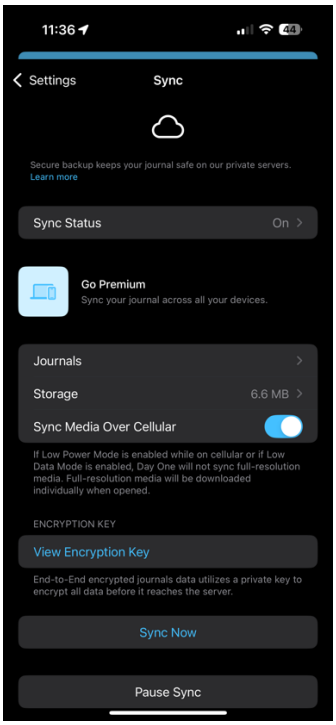


Figure 2.3.1.3 Cloud Synchronize

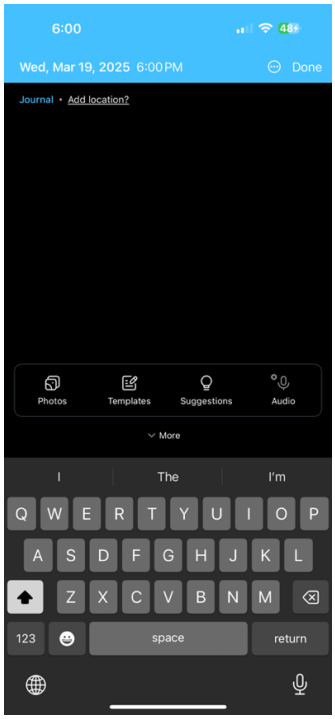


Figure 2.3.1.4 Diary Writing

Day One put users' privacy in first place, so it offered end-to-end encryption to secure users' personal journal entries. This feature ensures that only the user can access their own diary, no one else has the permission to view or edit the content in the diaries. Day One also provides the password feature when users want to open the application as shown in Figure 2.3.1.5. This will protect no one from accessing the application even though he gets the users' device. At the top right corner of the Day One, have a search feature that allows users to search their

journal entries by tag, place, date, checklist, and so on as shown in Figure 2.3.1.6. This helps the user to find the relevant entries with a few simple steps only. Users can use the tagging system to separate those entries into different categories.

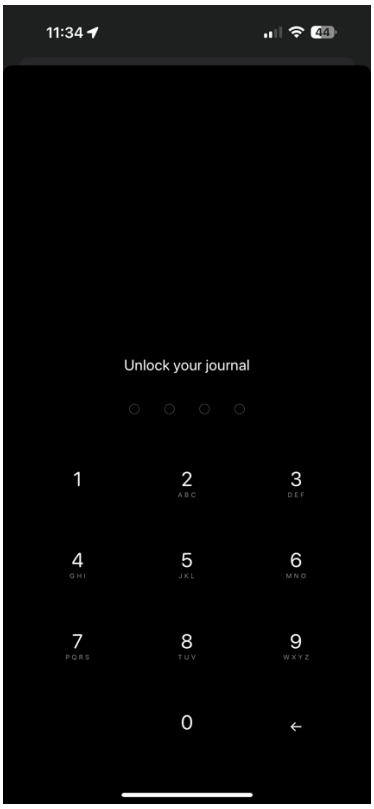


Figure 2.3.1.5 Password Secure

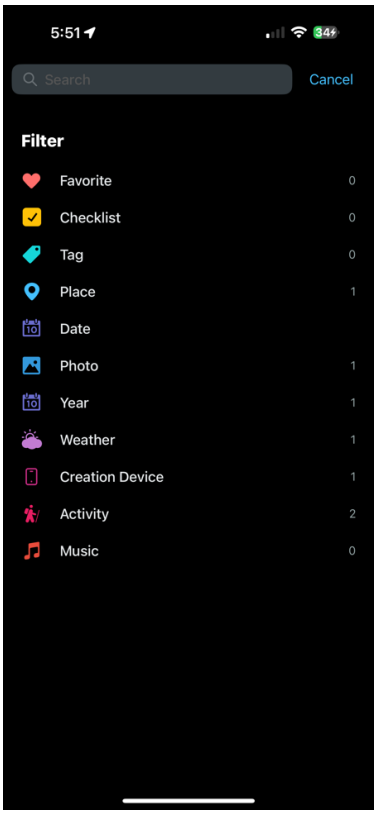


Figure 2.3.1.6 Search Function

Strength

- Users can have seamless syncing across multiple devices.
- Offers customized template for various types of entries for a personalized experience.
- User can export journal entries in different format for backup or sharing purpose.
- User is allowing to add photo, video, audio into journals entries.
- User can tag their journals entries into different categories.

Weakness

- Lack of social interaction features for sharing at the social media platform

- Do not provide time-related reflection feature that help user revisit their memories or track personal progress over time.
- Many advance features require paid subscription to unlock.
- Free users cannot use the cloud syncing and backup.

2.3.2 Journey

Journey is a personal diary application that is designed for users who want to track their thoughts, emotions, and memories over time [4]. Journey is available on multiple platforms, including Android, iOS, Linux, and Windows. Journey provides many interesting features that other digital diary applications do not, such as Journey Odyssey AI, coach programs, and statistical analysis.

At the main page of the Journey, the user is able to press the plus options at the bottom right to quickly write their journal entry for the day as shown in Figure 2.3.2.1. The surprising thing is it provides three different types of text size, ordered lists, unordered lists, task lists, and tables for users to complete their diaries as shown in Figure 2.3.2.2. Besides, Journey also provides various designs of templates to meet users' personalities. Users can do some customization on their journals to meet their style that enhances engagement. These customizations help users create more creative journals that other diary applications do not provide. Journey allows users to integrate photos, videos, audio, and files into the journal's entries. This is suitable for users who prefer to record through photo or video because this makes the diary more interesting and attractive when they recap it many years later.

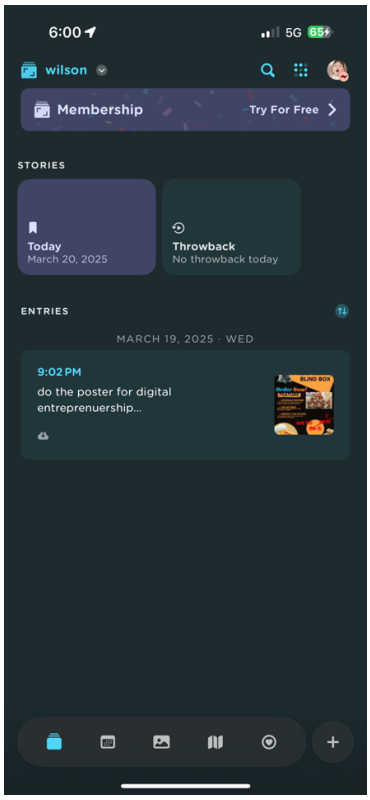


Figure 2.3.2.1 Main Page

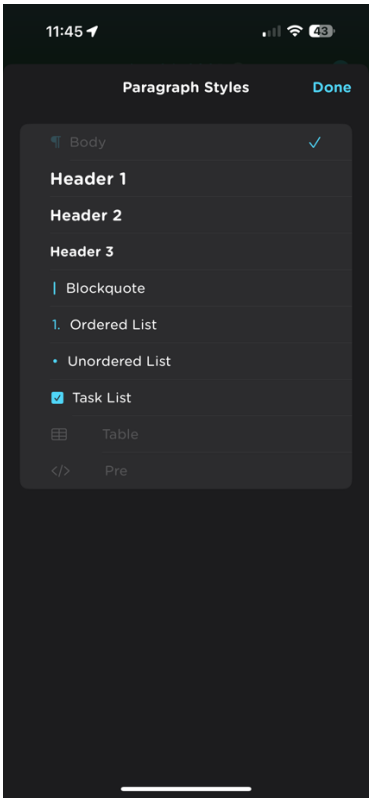


Figure 2.3.2.2 Selection of Font Size

Users are allowed to set password and Face ID authentication to protect the content in the application and prevent any leakage of the user's privacy to others as shown in Figure 2.3.2.3. All the journal's entries are protected by end-to-end encryption when synced to the cloud to prevent the happening of unauthorized access. Journey also provides a search bar for users to find their previous entries more easily. Users can search their journals by tags, starred, activity, sentiment, or keywords. Beside the search bar, Journey provides a gallery and calendar to help users quickly navigate to the previous entries as shown in Figure 2.3.2.4. These searching functions are the main difference that separates digital diaries and traditional diaries. Journey has included a reminder feature to encourage users to write journals regularly. Users can set the reminder to daily or weekly or customize it to help users track their progress. This can cultivate the habit of journaling among the users.

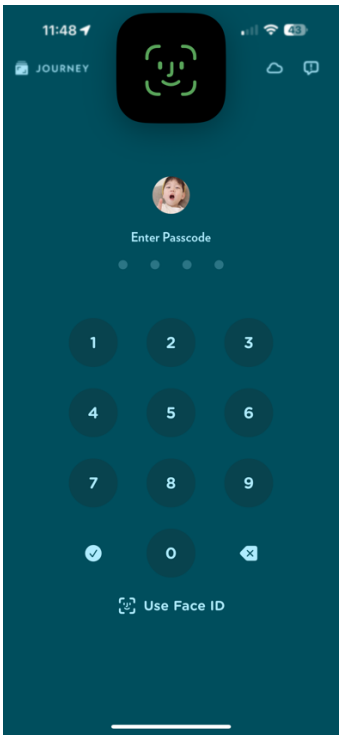


Figure 2.3.2.3 Face ID Unlock

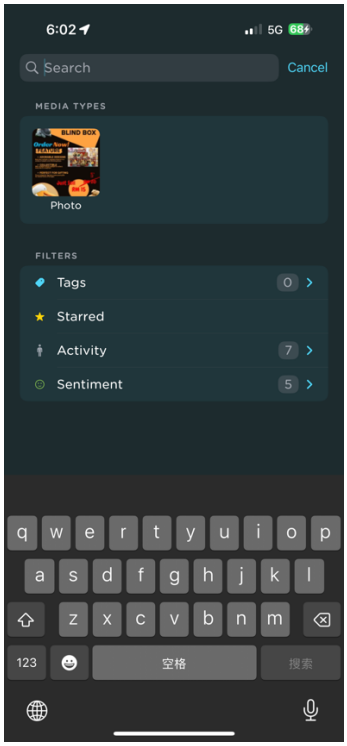


Figure 2.3.2.4 Search Function

Journey uses artificial intelligence (AI) technology to enhance the user's journaling experience by introducing Journey Odyssey AI as assistance for the user as shown in Figure 2.3.2.5. Journey Odyssey AI will analyse users' journal entries to provide personalized insights about their emotions and behaviour. This AI also tracks users' moods through the specific words, phrases, or sentiments that users select to do mood analysis. The analysis will be represented in a visual mood graph to show how the user's mood changes over time, letting the user track their mood easily as shown in Figure 2.3.2.6. By applying AI technology to the Journey application, Journey can have a better understanding of their users' emotions and behaviours. AI makes the journaling process meaningful, providing suggestions and feedback for users after they write down their diary. Journeys provides a statistical analysis for users to track their frequency of journaling and mood tracking. These data are represented in a graphical way, which creates a more insightful experience for the user.

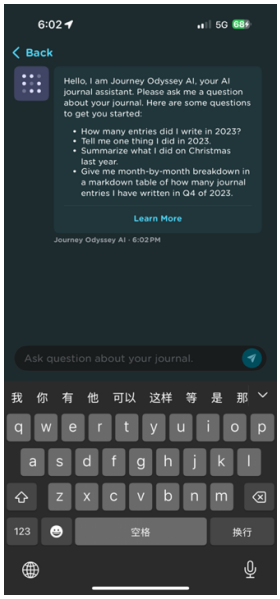


Figure 2.3.2.5 Odyssey AI

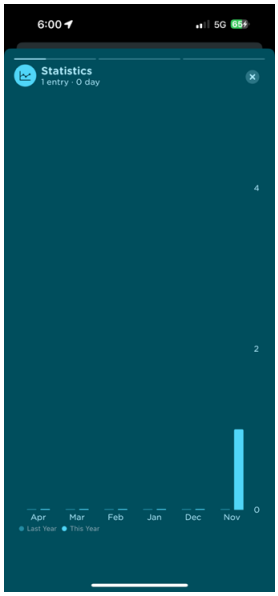


Figure 2.3.2.6 Statistics Analysis

The coach program provided by Journey is a feature used to enhance users' journaling experience by providing personalized guidance. At the coach program, Journey has provided much personalized advice for users based on their journaling habits. It offers specific suggestions to help users have a good journaling experience. Journals have cooperated with many featured coaches to provide self-care and mindfulness programs for users to participate in to customize their unique diaries.

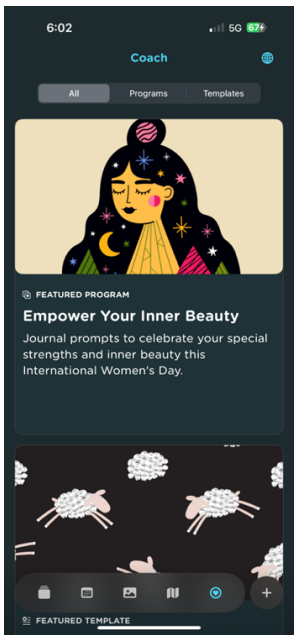


Figure 2.3.2.7 Coach Page

Strength

- Offers mood tracking feature that allows user to monitor their emotional state.
- Provide various type of font and changeable text size for customization.
- Provide personalized prompt and suggestions to encourage daily journaling.
- Include Journal Odyssey AI for journal assistant.
- Provide statistics analysis and throwback feature.

Weaknesses

- Limitation in social sharing features for sharing and collaborating.
- Many advance features require membership subscription to unlock.
- Journal Odyssey AI require Journal cloud service to unlock.
- Only provide subscription plan in one year term.

2.3.3 Grid Diary

Grid Diary is different from other diary applications; it offers a structured approach to self-reflection, tracks personal goals, records users' emotions, and helps users organize their thoughts. Grid Diary uses a grid-based system that provides a framework for users to fill in their entries into different categories such as plans, personal development, health, and gratitude [5]. This diary application is suitable for users who are not familiar with the open-ended diary nature. This application was first released in the year 2013 and now has over a million users.

Grid Diary has provided a unique grid-based system that divides daily, weekly, monthly, and yearly journal entries into different categories as shown in Figure 2.3.3.1. For example, 'What did I learn today?', 'To-dos for next week,' and 'Highlights from the past month' are given guidelines for users to reflect on different aspects and record their emotions and thoughts into it. The grid provides users a way to think about their daily routine in different categories, leading users to focus on certain aspects of their lives. The grid is customizable, where users can select their own categories for their diary journaling. Customization allows users to select a grid based on their needs and preferences to accommodate the different users' diary purposes. Users may require some time to learn how to use the grid-based system diary because this is different from other diary applications that occur on the market.

Figure 2.3.3.1 Diary Writing

Furthermore, on the main page of Grid Diary, have a quotation from famous people, which will encourage users to write in their diaries regularly as shown in Figure 2.3.3.2. The quotations act as motivation prompts for users to read or even copy into their journal entries to motivate them. Prompt lists are provided for users to choose to add them into their personal diaries to encourage them. The funny function of this application is the click of every button will have a sound effect to attract the user's interest. This diary application also includes cloud sync and backup functions to let users sync their journal entries across multiple devices by using online cloud services as shown in Figure 2.3.3.3. This makes sure that users can access all the journals with different devices with a seamless experience. The ability to access journal entries easily and securely will enhance user trust in Grid Diary.

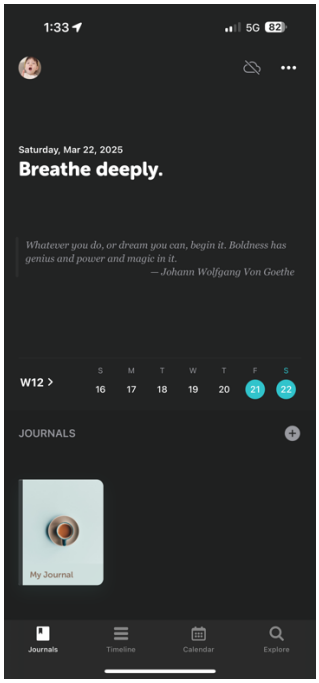


Figure 2.3.3.2 Main Page

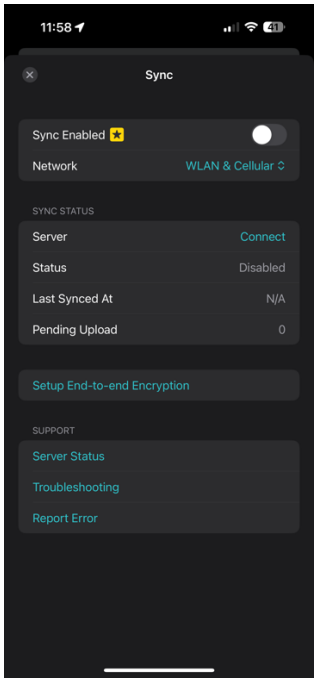


Figure 2.3.3.3 Cloud Synchronize

Users can click to the explore pages to search their previous journal entries by search keywords or photos included in the diary as shown in Figure 2.3.3.5. Statistical analysis is provided for users to check the total journals written in a month, total characters for all the journals, and the current streak that the day's user continues journaling as shown in Figure 2.3.3.6. It uses a line graph to show the use of characters each day. There is a list of titles that users have used in the Grid Diary to simplify the finding step for users.

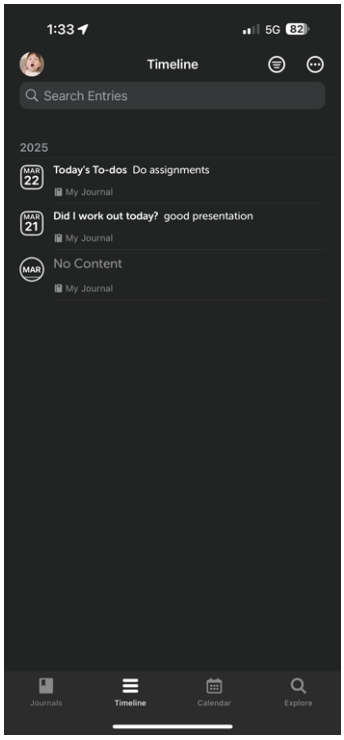


Figure 2.3.3.4 Search Function

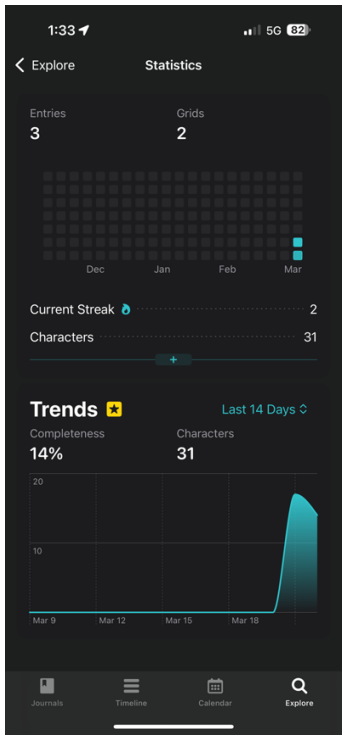


Figure 2.3.3.5 Statistics Analysis

Strength

- Customized grid and personalized prompts meet users' specific need.
- Quotes provide extra motivation and help users feels inspired to engage their journaling practice.
- Grid system helps users focus on specific aspect of their lives.
- Sound effect after clicking a button.

Weakness

- The grid system may cause inconvenience for users who prefer freeform journaling.
- Users cannot attach voice recording into the journal.
- Basic password functions are limited to premium users only.
- Lack of unique template for user to choose.
- No throwback feature for users to revisit their memories.

2.4 Summary

In this chapter, the mobile application for personal journal/diary and the web application "Personal Diary" project have been reviewed. The mobile application for a personal journal/diary used interactive and incremental development to ensure regular feedback and improvements throughout the process. This application is developed based on the Android operating system by using the Java programming language and implementing online cloud service by Parse.com for database management. Key features for this diary application are diary entry management, a categories system, multimedia support, data analysis, and calendar view. The limitations are that the UI design is not user-friendly, and the security of the application can be enhanced by implementing strong encryption and two-factor authentication to protect user privacy. Next, the web application "Personal Diary" provides a user-friendly, accessible, and organized website for journaling. The website is built by using PHP, HTML, CSS, Bootstrap, MySQL, and XAMPP to provide users a place to write, store, and manage their diary entries. The key features of this web application are user authentication, diary management, responsive design, and database management. A few limitations of this website are the lack of interactive features to attract users and the fact that the website requires internet access, making it unsuitable for users who want to journal offline. Both projects are digital diaries that replace the traditional handwritten diary by offering multimedia support and a structured journaling experience.

Three existing mobile applications are reviewed which is Day One, Journey, and Grid Diary. The weaknesses and strengths are compared and summarized in Table 2.4.1.

Table 2.4.1 Analysis of strengths and weaknesses of existing applications

	Day One	**Journey**	**Grid Diary**	**Voice Diary**
Multimedia Integration	Photo, video, audio with text conversion, location and file	Photo, video, audio, location and file	Photo and video only	Photo, video, audio with text conversion, and location

Customized Template	Many different types of templates	Many different types of templates	Not provided	Not provided
Grid System	Not available	Not available	Fully customizable grid system	Not available
Personalized Prompts	Provided daily prompts	Not available	Prompts are provided to guide user	Not available
Audio to text conversion	Yes, support voice converts to text function	No voice-to-text feature	No voice-to-text feature	Yes, support voice converts to text function
Body type and text size feature	Provided but hard to found	Yes, can change easily while journaling	Provided but hard to found	Not provided
User Interface (UI)	Elegant, minimalist design, easy-to-use	User-friendly, clean, focus on simplicity	Grid-based, structured interface	Clean, modern, card-based layout
Artificial Intelligent (AI) technique	Not available	Yes, Journey Odyssey AI	Not available	Yes, DistilBERT Base Uncased GoEmotions Student and Gemini AI
Target Audience	Users who want a sleek	Users focused on emotional tracking and personal growth	Users who prefer structured journaling and guided prompts	User who is tech savvy and privacy concious

CHAPTER 3 METHODS/TECHNOLOGIES INVOLVED

3.1 Overview

In this chapter, the development process for the Voice Diary application is discussed. The functioning of the application is described through use-case diagrams, use-case descriptions, and activity diagrams.

3.2.1 Use-Cases Diagram

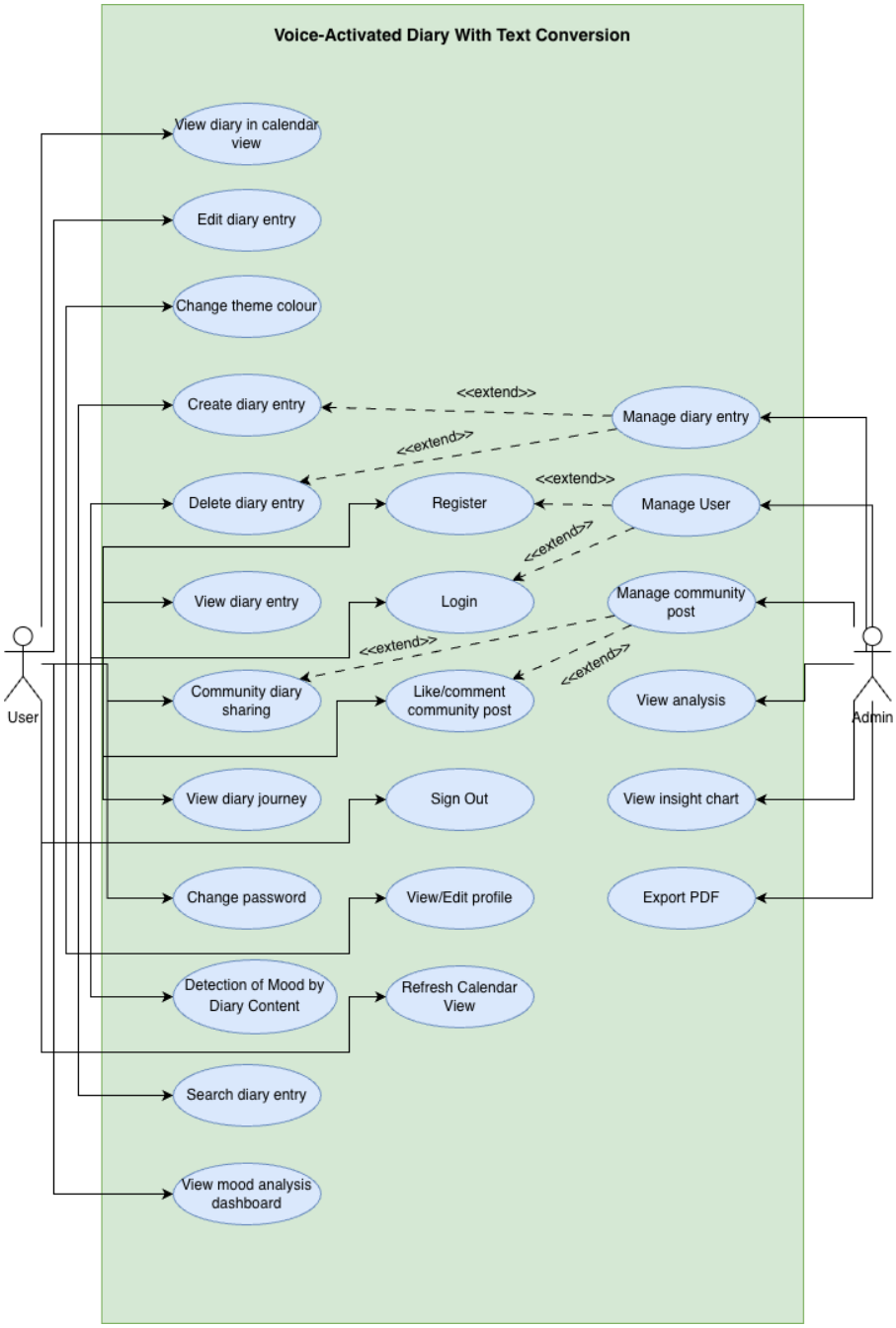


Figure 3.2.1.1 Use-case Diagram for Voice Diary Mobile Application

3.2.2 Use-case Description

Table 3.2.2.1 Login Use Case Description

Use Case ID	00001
Use Case Name	Login
Brief Description	Allows registered users to access their personal diary
Actor	Users
Trigger	Users' first-time access to the applications.
Precondition	The user must have already registered an account in the system.
Normal flow of event	1. Users enter their email and password. 2. Users click the login button. 3. The system validates the inputs and sends the credentials to the authentication server. 4. If the credentials are correct, the system will confirm the login. 5. The system will redirect the user to the diary home page.
Sub flows	-
Alternate flows	3a. If the combination of credentials does not match, the system will display an error message to users. 3b. If the system cannot connect to the authentication server because of a network problem, it will show an error message about the network error to users.

Table 3.2.2.2 Register Use Case Description

Use Case ID	00002
Use Case Name	Register
Brief Description	Allows a new user to create an account in the application
Actor	New users
Trigger	The user selects the "Sign up" button on the login screen.
Precondition	The user does not have any account that was registered with this same email.
Normal flow of event	1. Users fill in the email address, password, and confirm password.

	2. The system checks the inputs that match the format. 3. Users click the “Sign up” button to register. 4. The system sends the registration data to the authentication server. 5. If the registration is successful, the system will create a new account and redirect the user to the diary home page.
Sub flows	-
Alternative flows	2a. If the input is invalid, the system will prompt the error message and ask users to correct the input. 4a. If the email is already registered, the system will display “ The email address is already in use by another account”.

Table 3.2.2.3 Create Diary Entry Use Case Description

Use Case ID	00003
Use Case Name	Create Diary Entry
Brief Description	User creates a new diary entry using text and voice input.
Actor	Users
Trigger	Users click the “New Entry” or “+” button on the diary home page.
Precondition	The user is required to log in to system and be connected to the internet.
Normal flow of event	1. Users click the “New Entry” button to enter the new diary screen. 2. Users enter a title for the diary entry. 3. Users can rate their current feeling by selecting the emoji. 4. Users can enter their diary content through text or record voice by clicking the “microphone” button. 5. The system will auto convert the voice input into text, then show it in the textbox. Location and weather data will be auto-detected through the device’s GPS. 6. Users can select to add tags, images, videos, or audio files to it to enhance the diary entry.

	7. Users click the “SAVE” button. 8. The system will save the diary entry into the database, then confirm with a “Entry saved successfully!” message and update the calendar view.
Sub flows	5a. The user can click the “clear” button to not record the location and weather data into the diary entry. 6a. For adding images, users can select the source of a new photo from the camera or an existing photo from the gallery.
Alternative flows	5a. If the voice recognition fails due to the quality of the audio, the system will display “Transcription error” for users to try again. 6a. If camera or gallery access is denied, the system will ask for permission. 8a. If no content in the diary entry, the system will display “Please add a title or content” to remind users.

Table 3.2.2.4 Edit Diary Entry Use Case Description

Use Case ID	00004
Use Case Name	Edit Diary Entry
Brief Description	Allows users to modify existing diary entries by updating the content or media.
Actor	Users
Trigger	The user selected an existing diary entry and clicked the edit option.
Precondition	The user is required to log in to system, and the diary should already exist in the system.
Normal flow of event	1. User can select any existing diary entry from the home page, calendar view page or search page. 2. The system will retrieve the selected diary entry from Firebase. 3. The system will display the diary entry in edit mode.

	4. User can edit the date of entry, moods, content, location and weather, tags, and media content in the diary. 5. Users click the “SAVE” button. 6. The system will update the modified entry in the Firebase and display a “Entry updated successfully!” confirmation message.
Sub flows	4a. The user can click the “clear” button to not record the location and weather data into the diary entry. 4b. For adding images, users can select the source of a new photo from the camera or an existing photo from the gallery.
Alternative flows	6a. If no content in the diary entry, the system will display “Please add a title or content” to remind users.

Table 3.2.2.5 Delete Diary Entry Use Case Description

Use Case ID	00005
Use Case Name	Delete Diary Entry
Brief Description	Allow users to remove the existing diary entry from the system.
Actor	Users
Trigger	The users select a diary entry, then choose the delete options.
Precondition	The user is required to log in to system, and at least one diary entry in the system.
Normal flow of event	1. User can select any existing diary entry from the home page, calendar view page or search page. 2. Users click the “Delete” button. 3. The system prompts a confirmation dialogue to users for double confirmation. 4. Users confirm the deletions. 5. The system removes the diary entry from the Firebase, then displays a “Entry deleted successfully” message to users.
Sub flows	-

Alternative flows	4a. If users cancel the confirmation dialogues, the system will not delete the diary entry.

Table 3.2.2.6 Change Password Use Case Description

Use Case ID	00006
Use Case Name	Change Password
Brief Description	Allow user to change their existing account password while logged in.
Actor	Users
Trigger	Users click on the profile page and select the Change Password option.
Precondition	The user is required to log in to system and know their current password.
Normal flow of event	1. Users click the change password option on the profile pages. 2. The system displays the change password screen to let users enter the current password, new password, and confirm the new password. 3. Users enter their current password and new password. 4. Users click the “Change Password” button. 5. The system validates the inputs. 6. The system updates the password in the authentication database and displays the confirmation message “Password changed successfully!”.
Sub flows	-
Alternative flows	5a. If the current password is invalid, the system displays the “Current password is incorrect” message to users. 5b. If the new password does not meet the requirement, the system will display the “Password must be at least 6 characters long”. 5c. If the new password and the confirm password do not match, the system will display “New passwords do not match”.

	5d. If the new password is the same as the old password, the system will display an error message.

Table 3.2.2.7 Change Theme Colour Use Case Description

Use Case ID	00007
Use Case Name	Change Theme Colour
Brief Description	Allow the user to personalise the application by selecting a different theme colour.
Actor	Users
Trigger	Users click on the profile pages and select a theme.
Precondition	The user is required to log in to system.
Normal flow of event	1. Users click on the profile pages and there are 6 colours of theme selection. 2. Users select the theme color options. 3. The system applies the selected theme color across the application user interface. 4. The system displays the change theme message and redirects users to the home page of the application.
Sub flows	-
Alternative flows	-

Table 3.2.2.8 View Dairy in Calendar View Use Case Description

Use Case ID	00008
Use Case Name	View Diary in Calendar View
Brief Description	Allows users to view and navigate diary entries by select dates on the calendar interface.
Actor	Users
Trigger	Users select the calendar options on the diary home page.
Precondition	The user is required to log in to system.
Normal flow of event	1. Users click the “View Calendar” button on the home page. 2. The system retrieves all the diary entries from the Firebase.

	3. The system will have a blue dot below the a have diary entry. 4. Users click on the dates with blue dots. 5. The system will display the diary entry list for that date. 6. Users can select any entry to view, edit, or delete.
Sub flows	-
Alternative flows	4a. If users click on the dates with no entry, the system will show a "No entries for this date".

Table 3.2.2.9 View Diary Entry Use Case Description

Use Case ID	00009
Use Case Name	View Diary Entry
Brief Description	Allow users to open and read existing diary entries.
Actor	Users
Trigger	Users choose a diary entry from the calendar view and the search result.
Precondition	The user is required to log in to system and have at least one diary entry in their account.
Normal flow of event	1. Users click on the diary entry. 2. The system retrieves the selected diary entry from Firebase. 3. The system loads the associated metadata and displays the diary content along with all related details. 4. Users can read the diary entry.
Sub flows	-
Alternative flows	2a. If no diary entries exist, the system will display "No recent entries" to users.

Table 3.2.2.10 Refresh Calendar View Use Case Description

Use Case ID	00010
Use Case Name	Refresh Calendar View
Brief Description	Allow the user to update the calendar interface to the most recent diary entry.

Actor	Users
Trigger	Users perform the refresh action while viewing the calendar.
Precondition	The user is required to log in to system.
Normal flow of event	1. Users click the “Calendar” button on the bottom navigation bar to access the calendar view pages. 2. Users swipe down on the calendar view screens. 3. The system will request the latest diary entries from Firebase. 4. The system updates the most recent data to the calendar interface. 5. The user can view the latest diary data on the calendar interface.
Sub flows	-
Alternative flows	3a. If no diary entries exist, the system will display “No recent entries” to users.

Table 3.2.2.11 View Mood Analysis Dashboard Use Case Description

Use Case ID	00011
Use Case Name	View Mood Analysis Dashboard
Brief Description	Allow user to access their mood analysis dashboard
Actor	Users
Trigger	Users enter the profile pages and view the mood analysis dashboard.
Precondition	The user is required to log in to system.
Normal flow of event	1. Users click the profile pages and scroll down to the mood analysis dashboard. 2. The system will retrieve the most recent 100 entries from Firebase. 3. The system generates a mood distribution bar chart that is separated into different colours. 4. The system displays a dashboard with the mood insights and average mood. 5. Users review their mood patterns and reflections.

Sub flows	-
Alternative flows	2a. If no mood entries exist, the system will display “No mood data available”.

Table 3.2.2.12 View Diary Journey Use Case Description

Use Case ID	00012
Use Case Name	View Diary Journey
Brief Description	Allow the user to view their diary journey progress, including day streak and total number of entries.
Actor	Users
Trigger	Users select the option to view their diary journey from the profile page.
Precondition	The user is required to log in to system.
Normal flow of event	1. Users navigate to the profile screen and scroll down to the diary journey section. 2. The system retrieves the total number of diary entries associated with users. 3. The system will calculate the user’s current day streak. 4. The system will display all the details on the diary journey section.
Sub flows	-
Alternative flows	3a. If the streak is broken, the system will reset the streak count to zero.

Table 3.2.2.13 Community Diary Sharing Use Case Description

Use Case ID	00013
Use Case Name	Community Diary Sharing
Brief Description	Allow the user to share selected diary entries in a community space to let other users read and interact.
Actor	Users
Trigger	Users select the option to share a diary entry to the community space.

Precondition	The user is required to log in to system.
Normal flow of event	1. Users select the diary entry that they want to share to the community space. 2. The system will show a “Share to community” button option. 3. The system validates the diary contents against community rules. 4. The diary entries published in the community space then allow other users to view and interact with the shared diary entries.
Sub flows	-
Alternative flows	3a. If the diary content violates rules, the system will block the sharing process and display “ Your diary entry cannot be shared due to policy violation”.

Table 3.2.2.14 Detection of Mood by Diary Content Use Case Description

Use Case ID	00014
Use Case Name	Detection of Mood by Diary Content
Brief Description	Automatically analyse the diary content and detect the user’s mood.
Actor	Users
Trigger	When users save or edit a diary entry, the system will automatically detect mood.
Precondition	The user is required to log in to system.
Normal flow of event	1. Users write or edit a diary entry. 2. The system will capture the text content and send it to the sentiment analysis module. 3. The sentiment analysis module processes the content and determines the mood, then stores it along with the diary entry in the Firebase. 4. The system will display the detected mood in the diary content.
Sub flows	-

Alternative flows	

Table 3.2.2.15 Like/Comment Community Post Use Case Description

Use Case ID	00015
Use Case Name	Like/Comment Community Post
Brief Description	Allows users to interact with community-shared diary posts by liking or commenting on them.
Actor	Users
Trigger	User selects a community post from the community page.
Precondition	The user is required to log in to system and the selected post must be available in the community section.
Normal flow of event	1. Users opens the community page. 2. User selects a diary post shared by another user. 3. User chooses to like the post or write a comment. 4. User submits the action. 5. System records the like or comment in the database. 6. System updates the post interaction count and displays the latest comment.
Sub flows	3a. User may remove a previously added like from the post.
Alternative flows	4a. If the comment is empty, the system displays a validation message.

Table 3.2.2.16 Search Diary Entry Use Case Description

Use Case ID	00017
Use Case Name	Search Diary Entry
Brief Description	Allow the user to quickly locate the diary entries based on keywords, tags, or titles.
Actor	Users
Trigger	Users enter search queries in the search bar.
Precondition	The user is required to log in to system.

Normal flow of event	1. Users enter the search diary pages by clicking the “Search” option in the bottom navigation bar. 2. Users enter the keyword, tag, or title into the search bar. 3. The system will retrieve diary entries that match the query. 4. The system will show the search result in a list format for the user to select. 5. Users can select the result to open and view the details of diary entry.
Sub flows	-
Alternative flows	3a. If no entries match the search query, the system will display “No results found for …” message to the user.

Table 3.2.2.17 Filter Search by Mood Use Case Description

Use Case ID	00017
Use Case Name	View/Edit Profile
Brief Description	Allows users to view or update their personal information in application.
Actor	Users
Trigger	User selects the profile section from the application menu.
Precondition	The user is required to log in to system.
Normal flow of event	1. User opens the profile page. 2. System displays current profile information. 3. User selects the edit option. 4. User updates personal details such as username, profile image, or gender. 5. User clicks save. 6. System validates and updates the information in the database.
Sub flows	-
Alternative flows	6a. If required fields are empty, the system displays a validation message.

	6b. If have any fail update system will displays an error message.

Table 3.2.2.18 Sign Out Use Case Description

Use Case ID	00018
Use Case Name	Sign Out
Brief Description	Allow the user to securely end their current session in the application.
Actor	Users
Trigger	Users select the sign-out option in the account settings section.
Precondition	The user is required to log in to system.
Normal flow of event	1. Users navigate to profile pages and scroll to the account setting section. 2. Users click on the “Sign Out” button options. 3. The system terminates the user session. 4. Users will be redirected to the login screen
Sub flows	-
Alternative flows	-

Table 3.2.2.19 Manage User Use Case Description

Use Case ID	00019
Use Case Name	Manage User
Brief Description	Allows administrators to monitor and manage registered user accounts within the system.
Actor	Admin
Trigger	Admin selects the user management section from the admin dashboard.
Precondition	The administrator is required to log in to the admin panel.
Normal flow of event	1. Admin opens the user management page. 2. System displays the list of registered users. 3. Admin selects a specific user account. 4. Admin views user details and account status. 5. Admin may update account status or monitor user activity.

	6. System saves the updated changes.
Sub flows	-
Alternative flows	6a. If have any fail update system will displays an error message.

Table 3.2.2.20 Manage Diary Entry Use Case Description

Use Case ID	00020
Use Case Name	Manage Diary Entry
Brief Description	Allows administrators to monitor and manage diary entries stored in the system for moderation and analysis purposes.
Actor	Admin
Trigger	Admin selects the diary management section from the admin dashboard.
Precondition	The administrator is required to log in to the admin panel.
Normal flow of event	1. Admin opens the diary management page. 2. System displays all diary entries stored in the system. 3. Admin selects a diary entry. 4. Admin reviews the diary content and related mood analysis result. 5. Admin performs management actions if necessary. 6. System updates the record accordingly.
Sub flows	5a. Admin may search diary entries by user ,content, or mood category.
Alternative flows	6a. If the selected diary entry cannot be retrieved, the system displays an error message.

Table 3.2.2.21 Manage Community Post Use Case Description

Use Case ID	00021
Use Case Name	Manage Community Post
Brief Description	Allows administrators to review and ensure community-shared diary posts to maintain a safe and clean environment.
Actor	Admin

Trigger	Admin selects the community management section from the admin dashboard.
Precondition	The administrator is required to log in to the admin panel.
Normal flow of event	1. Admin opens the community post management page. 2. System displays all shared community posts, including posts reported by users. 3. Admin selects a specific post for review. 4. Admin determines whether the post against the rules of the community. 5. If the post is acceptable, admin allows the post to remain. 6. If the post is unacceptable, admin will takes down the post. 7. System updates the database accordingly.
Sub flows	3a. Admin may search or filter posts based on username, report status, date, or content type.
Alternative flows	7a. If the deletion process fails, the system displays an error message and ask admin to retry.

Table 3.2.2.22 View Analysis Use Case Description

Use Case ID	00022
Use Case Name	View Analysis
Brief Description	Allows administrators to view mood-related analytics and overall system usage statistics.
Actor	Admin
Trigger	Admin selects the analysis section from the admin dashboard.
Precondition	The administrator is required to log in to the admin panel and system data must be available.
Normal flow of event	1. Admin opens the analysis page. 2. System retrieves mood detection data and user activity statistics. 3. System displays analytical summaries such as mood distribution and engagement trends. 4. Admin reviews the generated analysis.

Sub flows	3a. Admin can filter analysis based on date range.
Alternative flows	2a. If no data is available, the system displays an empty analysis notification.

Table 3.2.2.23 View Insight Chart Use Case Description

Use Case ID	00023
Use Case Name	View Insight Chart
Brief Description	Allows administrators to visualize mood trends and user behaviour through charts and graphical reports.
Actor	Admin
Trigger	Admin selects the insight chart section from the admin dashboard.
Precondition	The administrator is required to log in to the admin panel and analysis data must exist in the system.
Normal flow of event	1. Admin opens the insight chart page. 2. System generates charts based on diary mood data and user activities. 3. Charts such as pie charts, bar charts, and trend graphs are displayed. 4. Admin reviews the visual insights for decision-making purposes.
Sub flows	2a. Admin can filter analysis based on date range.
Alternative flows	2b. If chart generation fails, the system displays an error message.

Table 3.2.2.24 Export PDF Use Case Description

Use Case ID	00024
Use Case Name	Export PDF
Brief Description	Allows administrators to export analytical reports and system insights into PDF format for documentation and further evaluation.
Actor	Admin
Trigger	Admin clicks the Export PDF button from the admin dashboard.
Precondition	The administrator is required to log in to the admin panel and report data must be available.

Normal flow of event	1. Admin selects the export option. 2. System prepares the selected report data. 3. System generates the report in PDF format. 4. System prompts the admin to download or save the PDF file.
Sub flows	-
Alternative flows	3a. If PDF generation fails, the system displays an export error message.

3.2.3 Activity Diagram

3.2.3.1 Create New Diary Entry Activity Diagram

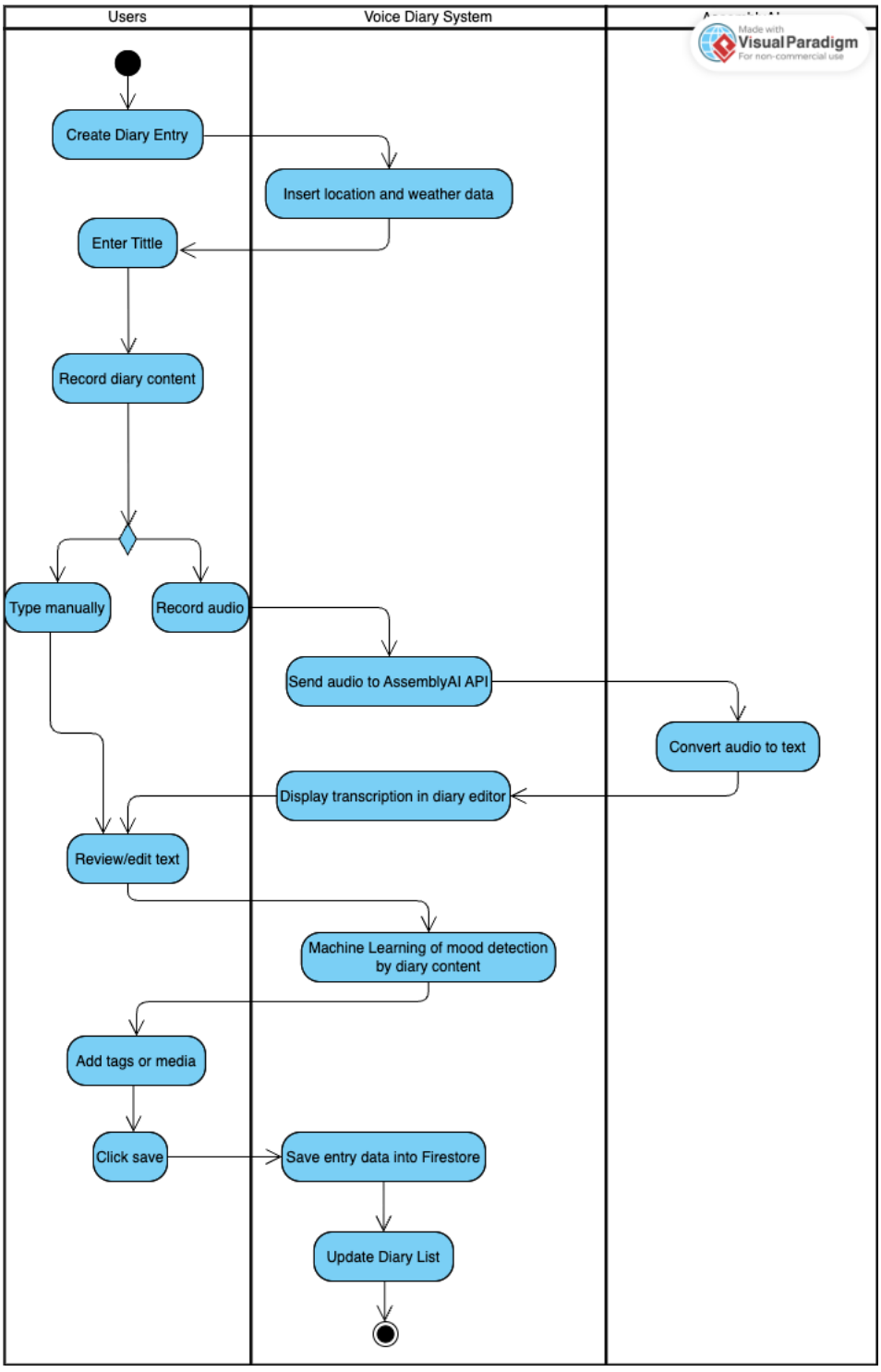


Figure 3.2.3.1 Activity Diagram for Diary Entry Creation

3.2.3.2 Login Activity Diagram

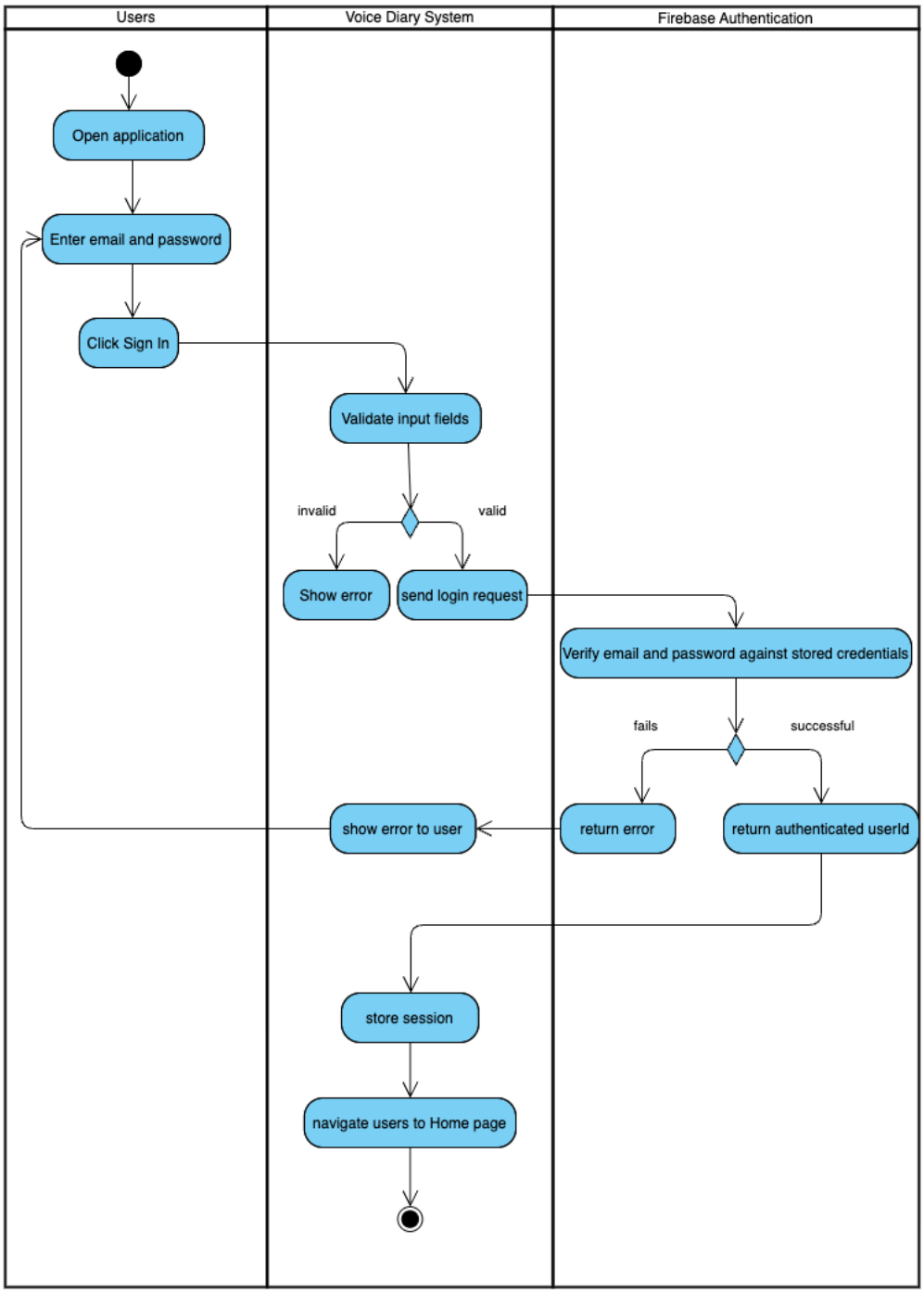


Figure 3.2.3.2 Activity Diagram for Login Application

3.2.3.3 Search diary Entries Activity Diagram

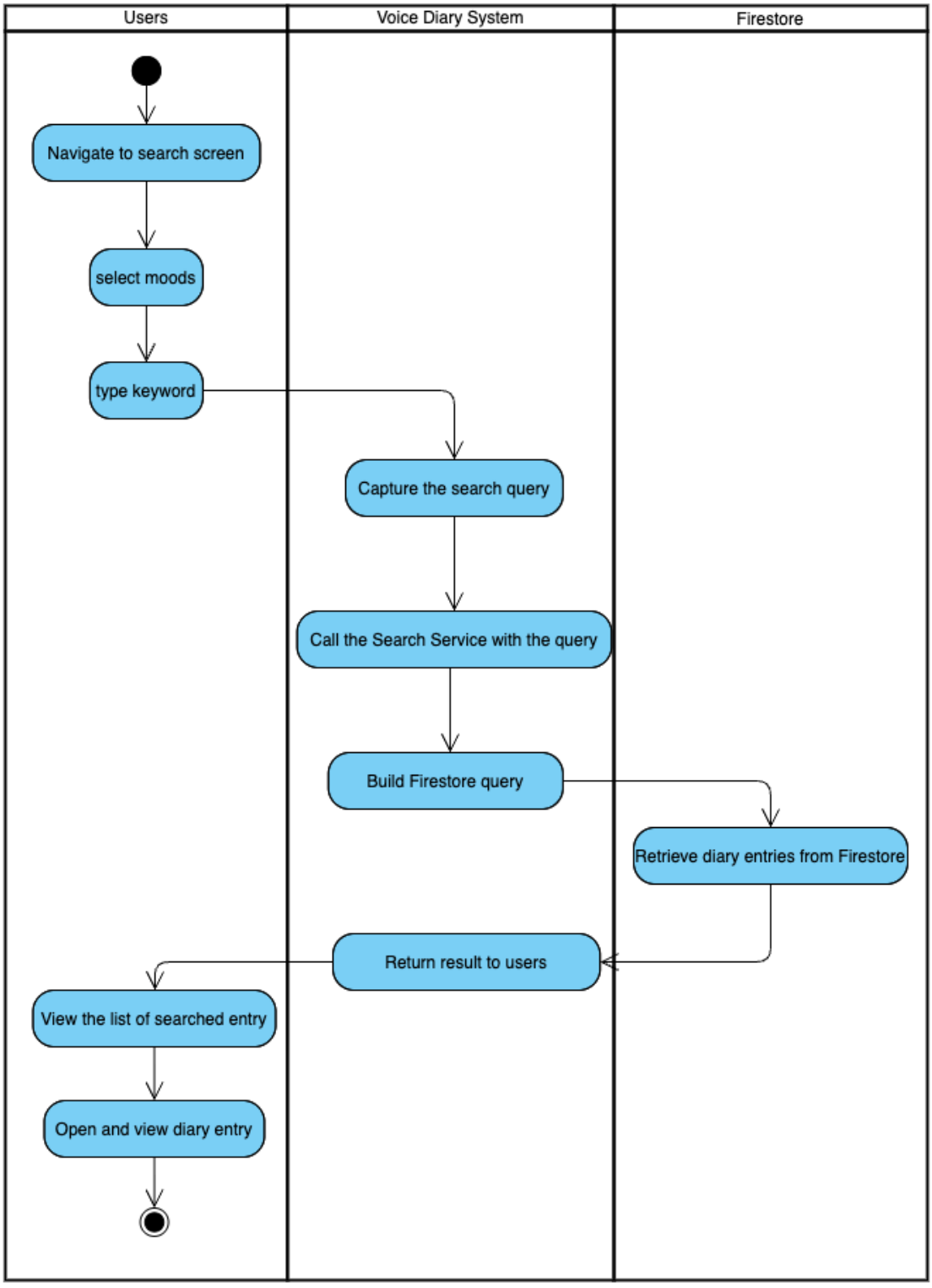


Figure 3.2.3.3 Activity Diagram for Searching Diary Entries

3.2.3.4 Change Theme Activity Diagram

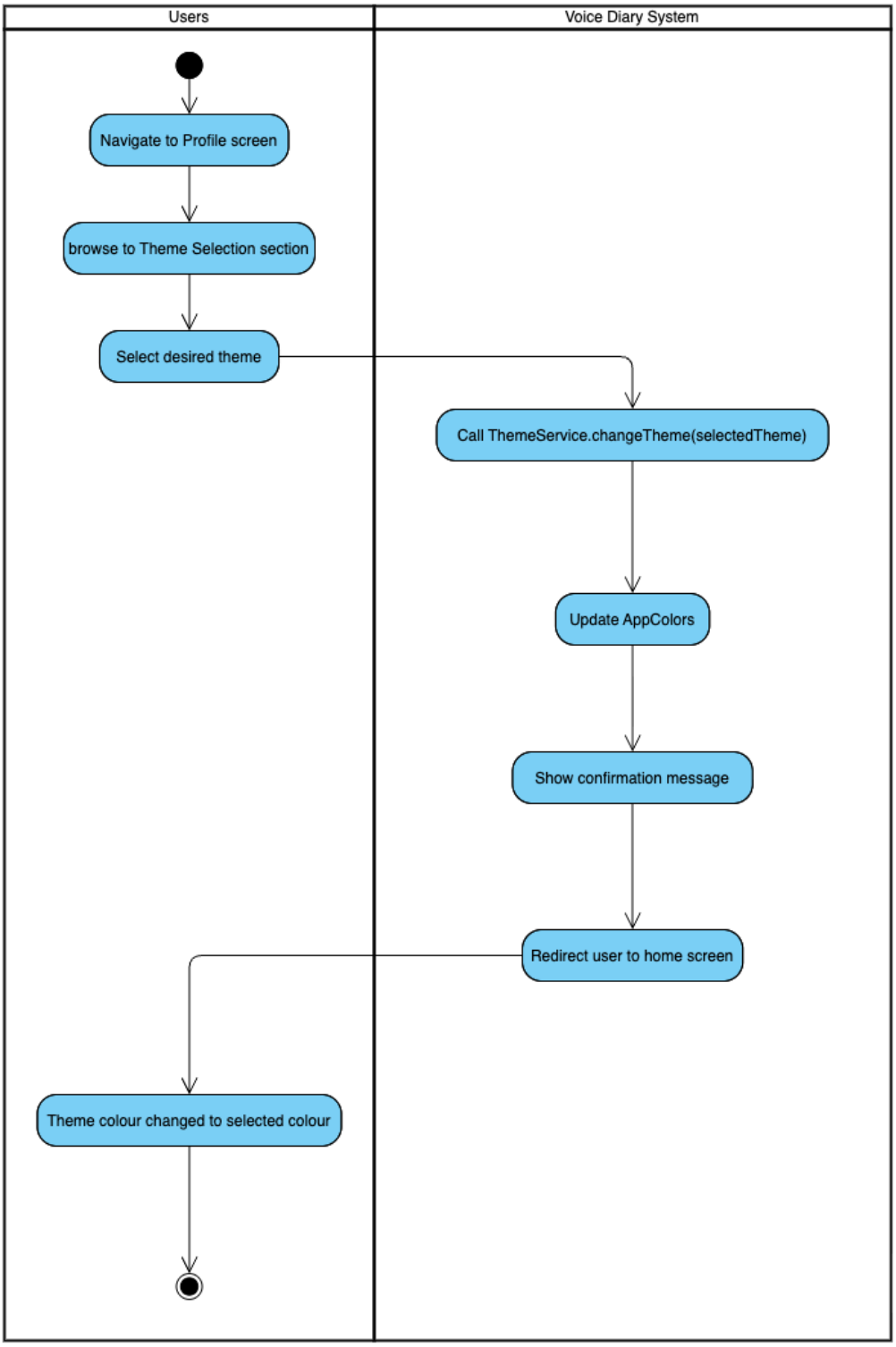


Figure 3.2.3.4 Activity Diagram for Theme Selection

3.2.3.5 Mood Trend Notification Activity Diagram

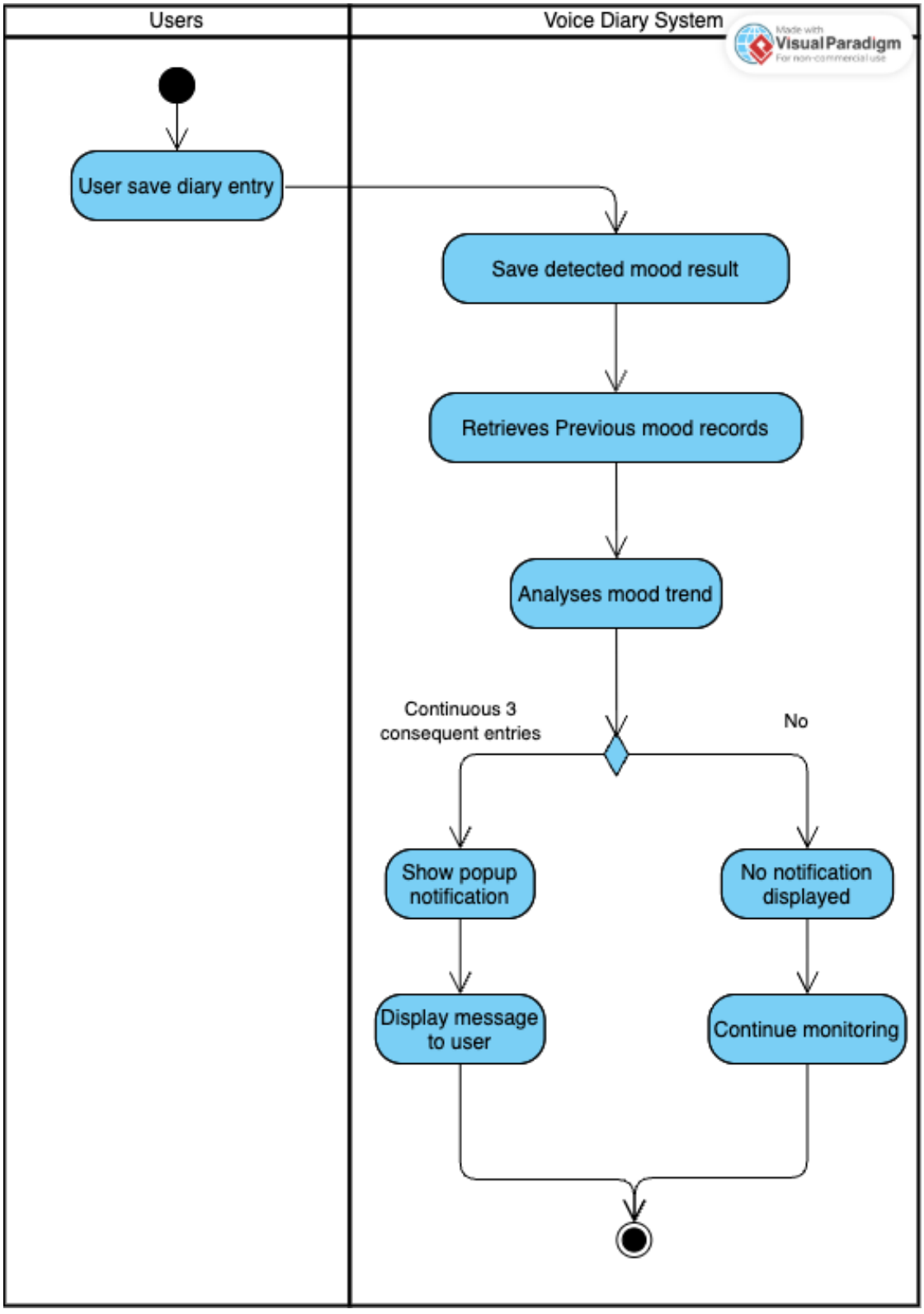


Figure 3.2.3.5 Activity Diagram for Mood Trend Notification

3.2.3.6 Community Sharing Activity Diagram

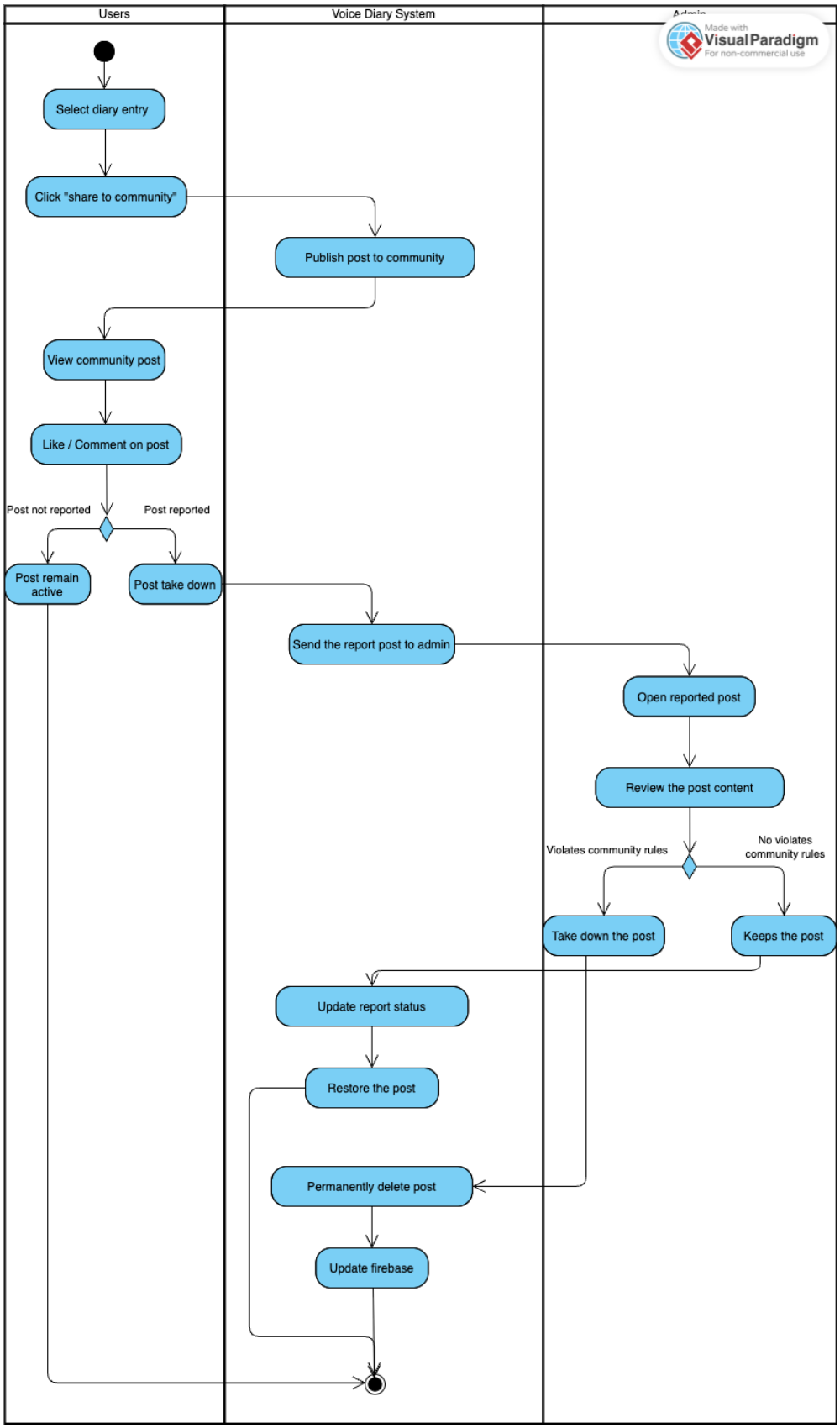


Figure 3.2.3.6 Activity Diagram for Community Sharing

3.2.3.7 Admin Dashboard Export PDF Activity Diagram

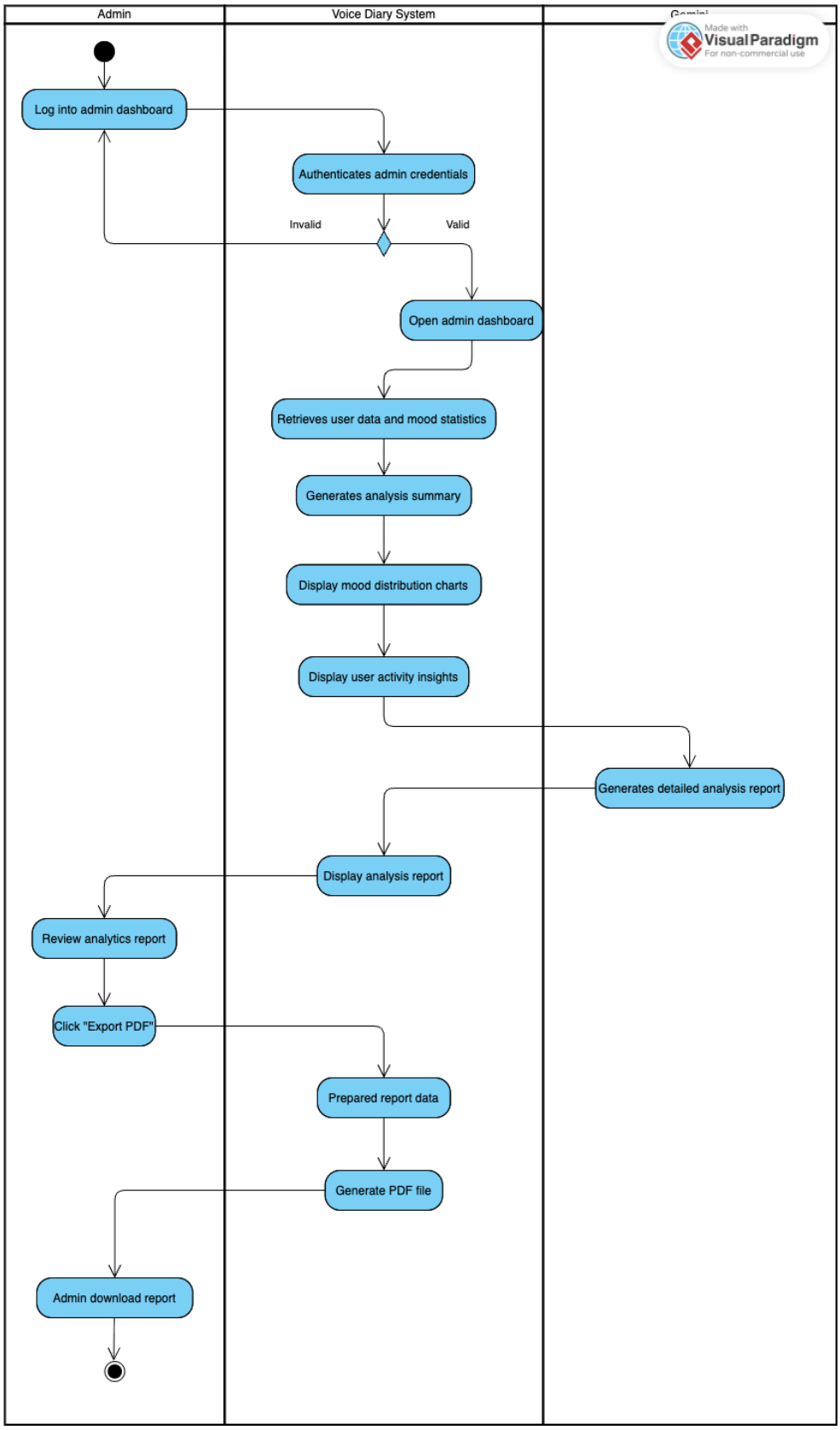


Figure 3.2.3.7 Activity Diagram for Admin Dashboard Export PDF

3.2.3.8 PIN Protection for Diary Entry Activity Diagram

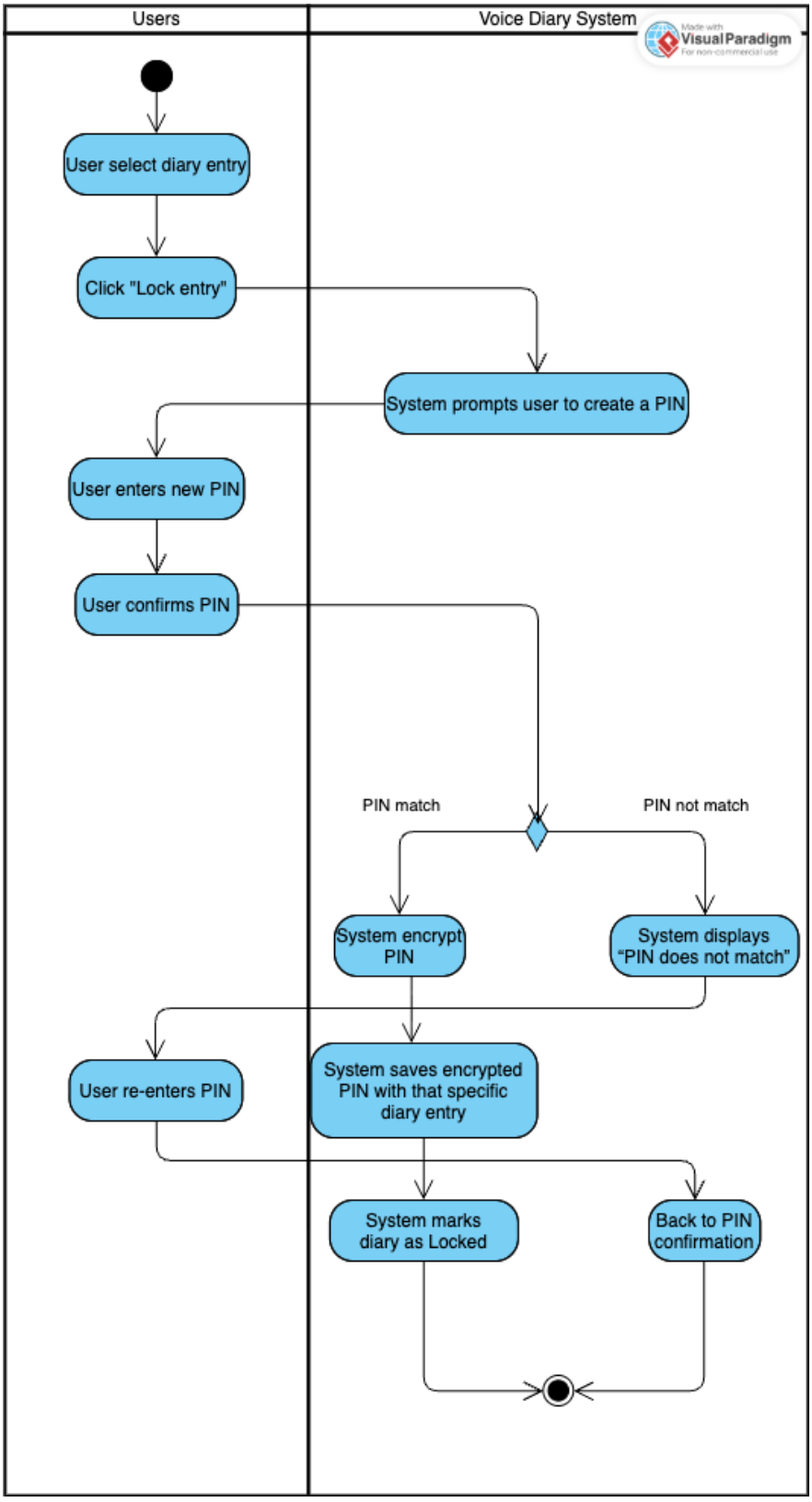


Figure 3.2.3.8 Activity Diagram for PIN Protection for Diary Entry

3.3 Program Development

3.3.1 Frontend Development

The frontend of the Voice Diary mobile application is developed using the Flutter SDK, which helps in achieving a responsive cross-platform design. In the architecture of the mobile application frontend, there is a presentation layer, a state management layer which is Provider with services like ChangeNotifier, and a data layer responsible for working with Firebase services. This architecture makes it more maintainable and scalable. In the initialization process of the application, Firebase is initialized, and global providers such as AuthService, ThemeService, and AdminService are initialized through MultiProvider inside main.dart file, and the MaterialApp is created with the dynamic theme. Figure 3.3.1.1 listed down all the dependencies that used in the development of application by flutter.

There are several modules in the frontend of the mobile application. The first one is authentication, which is responsible for user login or registering process, and the validation and loading processes are managed by AuthService. There is also a module, which manages the navigation screen and consists of bottom-navigation-based screens. There are Home screen, New Entry screen, Search screen, Community screen, and Profile screen.

Figure 3.3.1.2 home_screen.dart is used as the main dashboard which showcases the latest entries along with the navigational shortcuts. Figure 3.3.1.3 new_entry_screen.dart enables the users to create new diary entries where they can type text, transcribe voice from a number of different languages, upload multimedia content, add tags, as well as make use of GPS and weather APIs. Furthermore, the process uses an AI technique for determining the mood, utilizing a debouncing mechanism to enhance performance. The calendar_screen.dart and search_screen.dart enable the user to navigate their entries either through dates or specific search terms. Finally, the community screen allows the user to interact by posting, liking, and commenting, while the profile screen gives the user’s stats and mood analysis.

A dedicated admin section has been developed with an independent navigation system for managing users, diary entries, community postings, and accessing analytics dashboards. All reusable UI elements are placed in the lib/widgets folder for uniformity and avoiding redundancy. Provider manages state management within the app, with global states being available for all screens and local states responsible for transient UI interactions.

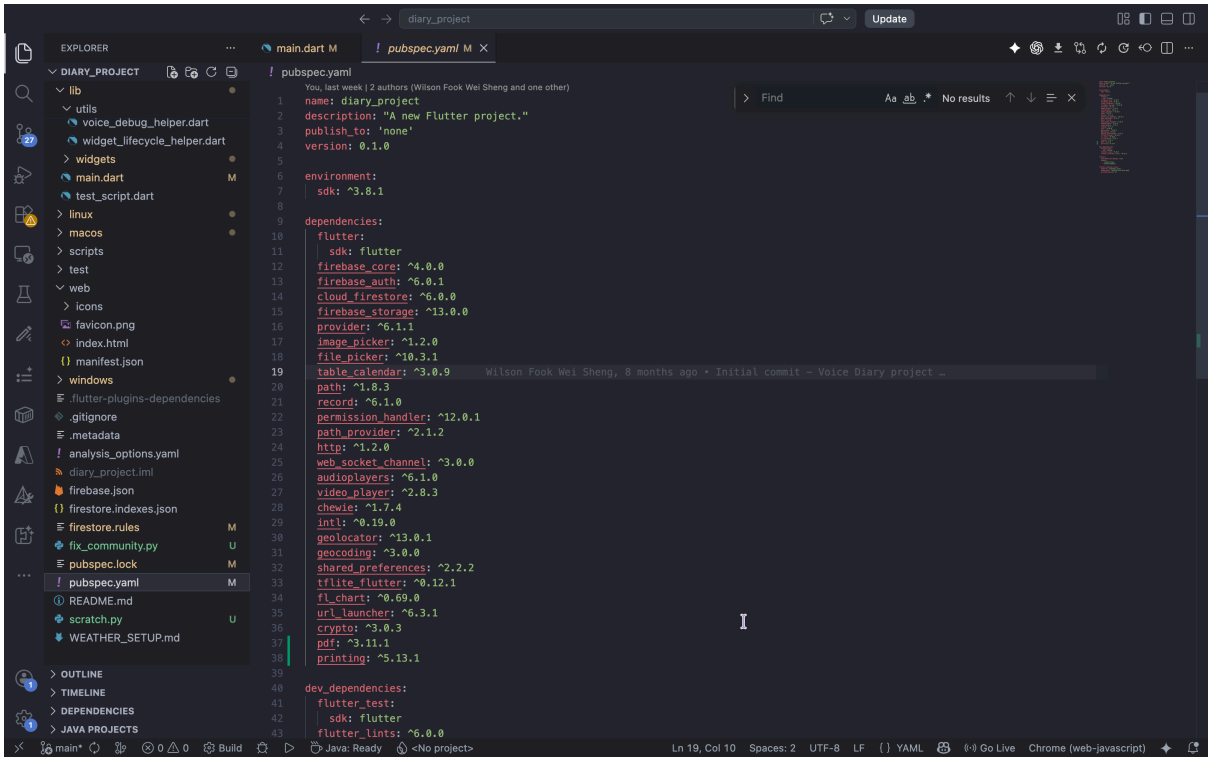


Figure 3.3.1.1 Pubspec.yaml

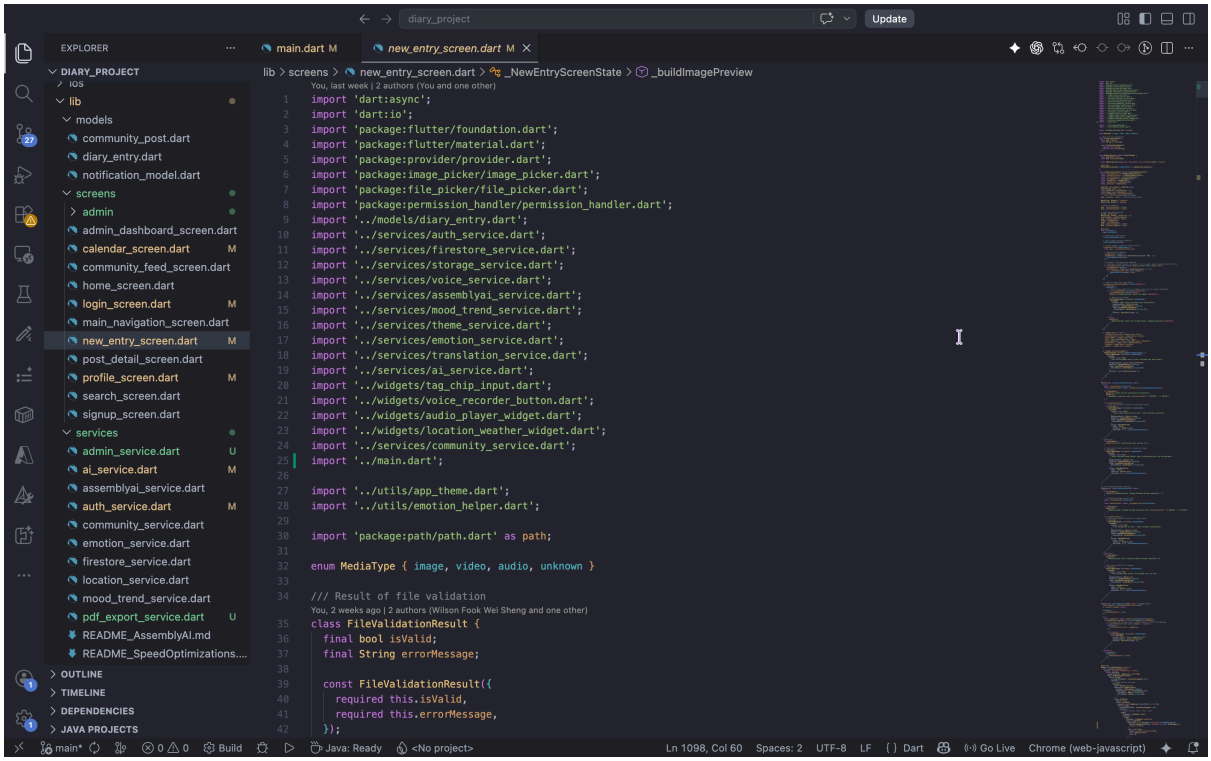


Figure 3.3.1.2 New_entry_screen.dart

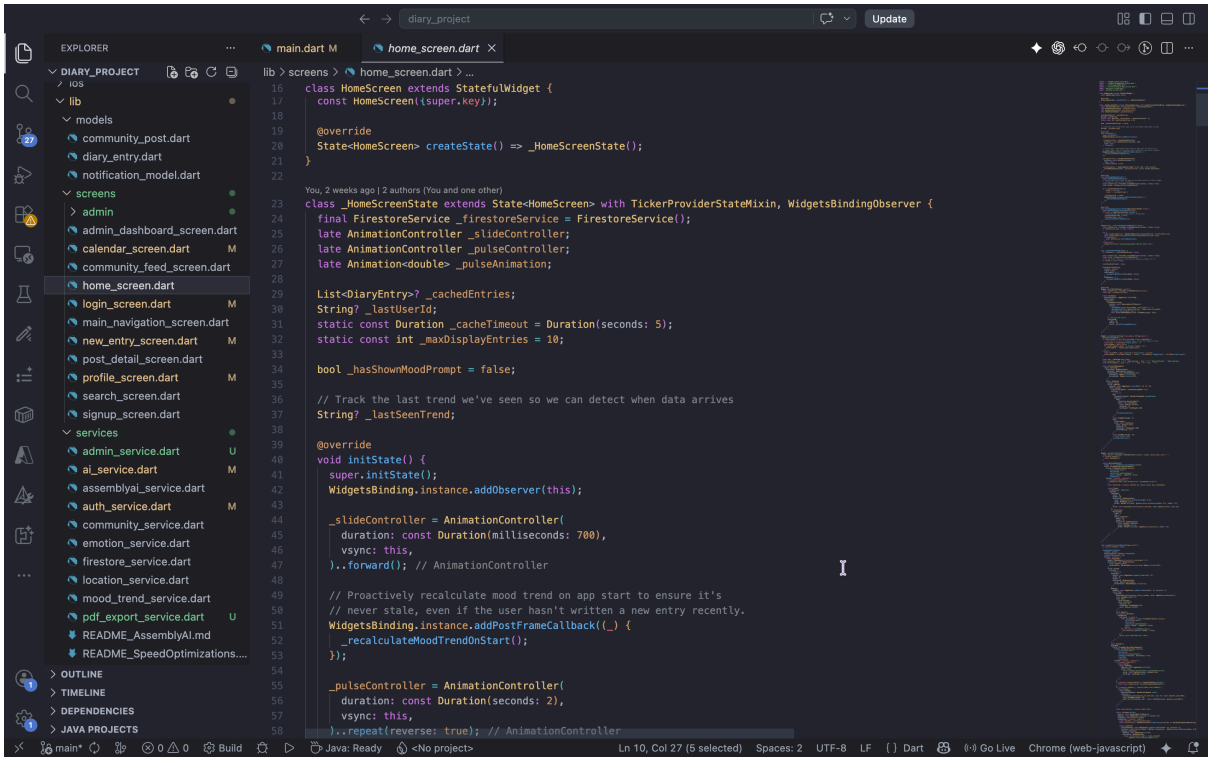


Figure 3.3.1.3 Home_screen.dart

3.3.2 Backend Development

Voice Diary application backend is mostly created using Firebase services, such as Firebase Authentication, Cloud Firestore, and Firebase Storage. In addition to this, these Firebase services provide a scalable and immediate backend environment without the need for any server at all. In addition to this, several APIs have been used within the application from third-party sources, including Gemini AI, Google Translate, AssemblyAI, and OpenWeatherMap. These APIs will be used to enable features related to analysis of the mood of users, translation, text conversion, and weather detection. The logic behind backend processes is mostly defined within the service classes, which can be found in lib/services folder.

Figure 3.3.2.1 show the rules that apply to Firestore database, structured data can be queried in meaningful ways, and the data is updated in real-time on all synchronized devices. As regards the structure of the database, its model will rely on Cloud Firestore collections named users, entries, and posts that show in Figure 3.3.2.2. The former collection will include all information related to the profiles of users, authentication roles, and trends of their moods. All entries of diaries will be saved in the latter collection, together with text information, emotions identified, tags, media URLs, as well as the language and locked status of the entries. As for the collection posts, it will be used for the community feature and will include diary entries shared by users

alongside with likes and comments. Other subcollections will be needed as well, such as notifications and comments.

The steps involve multiple processes working together at the backend. User authentication is one of them, where Firebase Authentication authenticates the user and collects data of the user from Firestore, like role and suspended status of the user. For the creation of diary entries, the system makes use of AI to translate and analyze the text and make mood detection and save the diary entry on Firestore. Trending analysis of the user mood depends on user input and gets updated on the user profile. Voice input of the user gets transcribed with the assistance of AssemblyAI. Community related features include making user post and interact via likes and comments while the administrator can approve, reject, and delete the post made by the user.

Security measures include Firestore rules that ensure users cannot access any other information other than their own. Administrators receive additional privileges to manage the system. However, there are still some areas, such as posts and notifications, that should include additional security measures. API keys, at the moment, are saved on the client side, leading to possible threats.

```
rules_version = '2';
service cloud.firestore {
  match /databases/{database}/documents {

    function isAdmin() {
      return request.auth != null && get(/databases/$(database)/documents/users/$(request.auth.uid)).data.role == 'admin';
    }

    // ── Diary Entries ──────────────────────────────────────────
    match /entries/{entryId} {
      allow read, update, delete: if isAdmin() || (request.auth != null && resource.data.userId == request.auth.uid);
      allow create: if isAdmin() || (request.auth != null && request.resource.data.userId == request.auth.uid);
    }

    // ── Media Files ────────────────────────────────────────────
    match /media_files/{fileId} {
      allow read, update, delete: if isAdmin() || (request.auth != null && resource.data.userId == request.auth.uid);
      allow create: if isAdmin() || (request.auth != null && request.resource.data.userId == request.auth.uid);
    }

    // ── User Profiles (legacy map) ─────────────────────────────
    match /user_profiles/{userId} {
      allow read, write: if isAdmin() || (request.auth != null && request.auth.uid == userId);
    }

    // ── Search History ─────────────────────────────────────────
    match /search_history/{historyId} {
      allow read, update, delete: if isAdmin() || (request.auth != null && resource.data.userId == request.auth.uid);
      allow create: if isAdmin() || (request.auth != null && request.resource.data.userId == request.auth.uid);
    }

    // ── Users & Notifications ──────────────────────────────────
    match /users/{userId} {
      allow read: if request.auth != null;
      allow write: if isAdmin() || (request.auth != null && request.auth.uid == userId);

      match /notifications/{notificationId} {
        allow read, update, delete: if isAdmin() || (request.auth != null && request.auth.uid == userId);
        allow create: if request.auth != null;
      }
    }

    // ── Community Posts ────────────────────────────────────────
    match /posts/{postId} {
      allow read: if request.auth != null;
      allow create: if isAdmin() || (request.auth != null && request.resource.data.authorId == request.auth.uid);
      allow delete: if isAdmin() || (request.auth != null && resource.data.authorId == request.auth.uid);
      allow update: if request.auth != null && (
        isAdmin()
        || resource.data.authorId == request.auth.uid
        || request.resource.data.diff(resource.data).affectedKeys()
            .hasOnly(['likes', 'status', 'reportCount', 'commentsCount'])
      );

      match /comments/{commentId} {
        allow read: if request.auth != null;
        allow create: if request.auth != null;
        allow update, delete: if isAdmin() || (request.auth != null && resource.data.authorId == request.auth.uid);
      }
```

Figure 3.3.2.1 Firestore database's rules

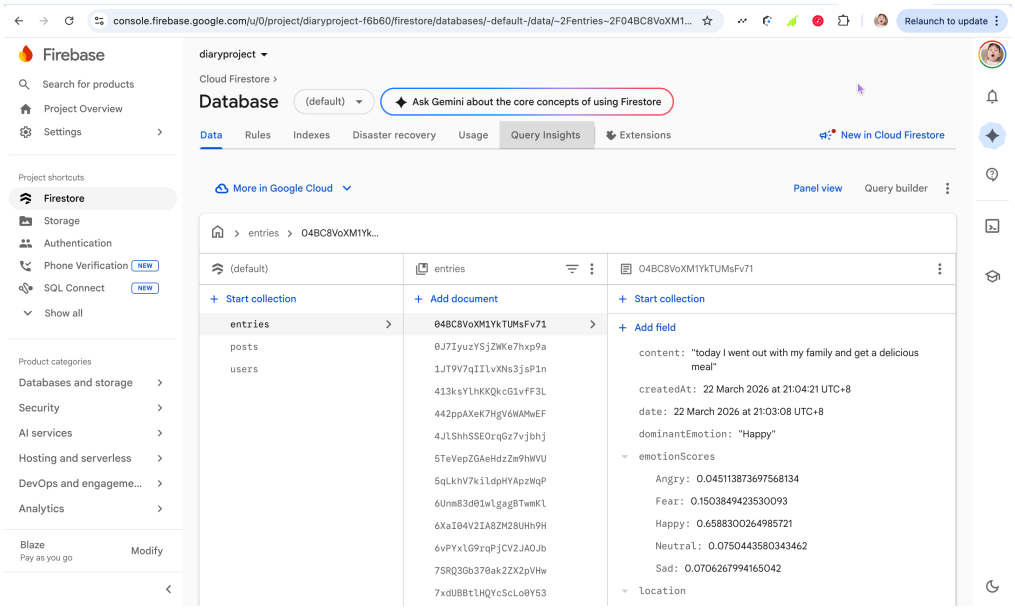


Figure 3.3.2.2 Firebase Cloud Firestore

3.3.3 Login and Signup Authentication Development

After user enter the email address and password, the authentication request is passed to the service layer defined in auth_service.dart that show in Figure 3.3.3.1, which abstracts the Firebase Authentication API and provides three functions: signUpWithEmailAndPassword(), signInWithEmailAndPassword(), and signOut(). These functions make calls to the Firebase underlying methods: createUserWithEmailAndPassword(), and signInWithEmailAndPassword() using async. Upon success, Firebase provides a secure ID token, which is associated with the user session, and it is cached by the Firebase SDK locally. This means that the user will not need to log in again after the app is stopped or poorly closed. The user will remain logged-in until they call the signOut() function.

From a security standpoint, Firebase Authentication handles credentials, including hashing and storage for you, and everything is processed securely on their servers. The app does not store any raw passwords directly. Also, Firebase Auth works with Firestore rules seamlessly, meaning that only authenticated users can read or write their diary entries and media files.

Finally, the sign-out process is design for easily accessible, users simply call signOut() in auth_service.dart, which clears the session and redirect users to login_screen.dart. This means users can always control their sessions, ensure privacy when use shared devices.

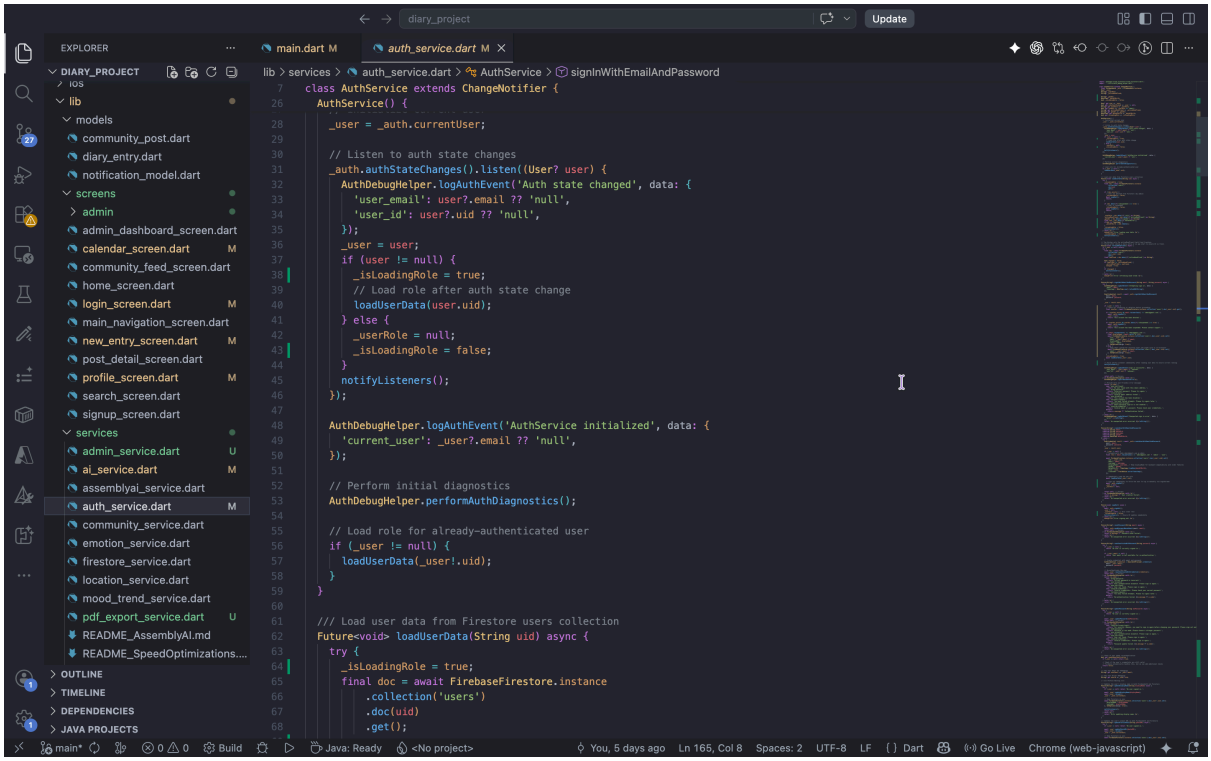


Figure 3.3.3.1 Auth_service.dart

3.3.4 Voice-to-Text Conversion Development

The Voice-to-Text Conversion feature lets users record audio diaries and automatically transcribe them into text. The ability to do this comes from combining Flutter's audio recording capabilities, custom widgets where users can interact with our UI, and the AssemblyAI transcription service via the service layer.

The actual recording and playback are handled in voice_service.dart show in Figure 3.3.4.2. The service abstracts all low-level audio APIs, including starting/stopping the microphone, programmatically storing temporary files for audio to play back, and providing play controls for review.

When a recording is captured, the transcription is soon handled by the assemblyai_service.dart refer to Figure 3.3.4.1 where we have the AssemblyAI integration. The service takes the audio recording and uploads it to AssemblyAI's servers via an authenticated API request. When AssemblyAI is complete processing the audio file through their speech recognition models, they return a transcript of the audio file. The service layer takes care of this process asynchronously. So, the user interface remains responsive while transcribing is taking place.

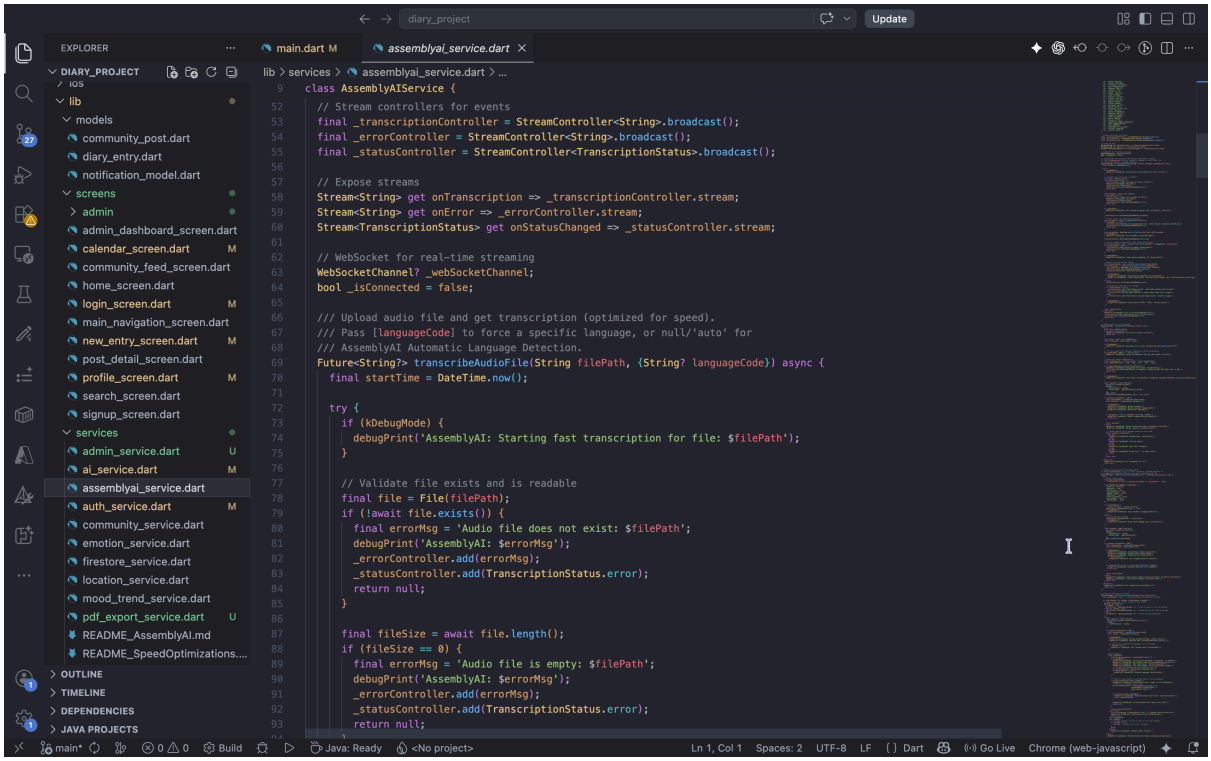


Figure 3.3.4.1 Assembly_service.dart

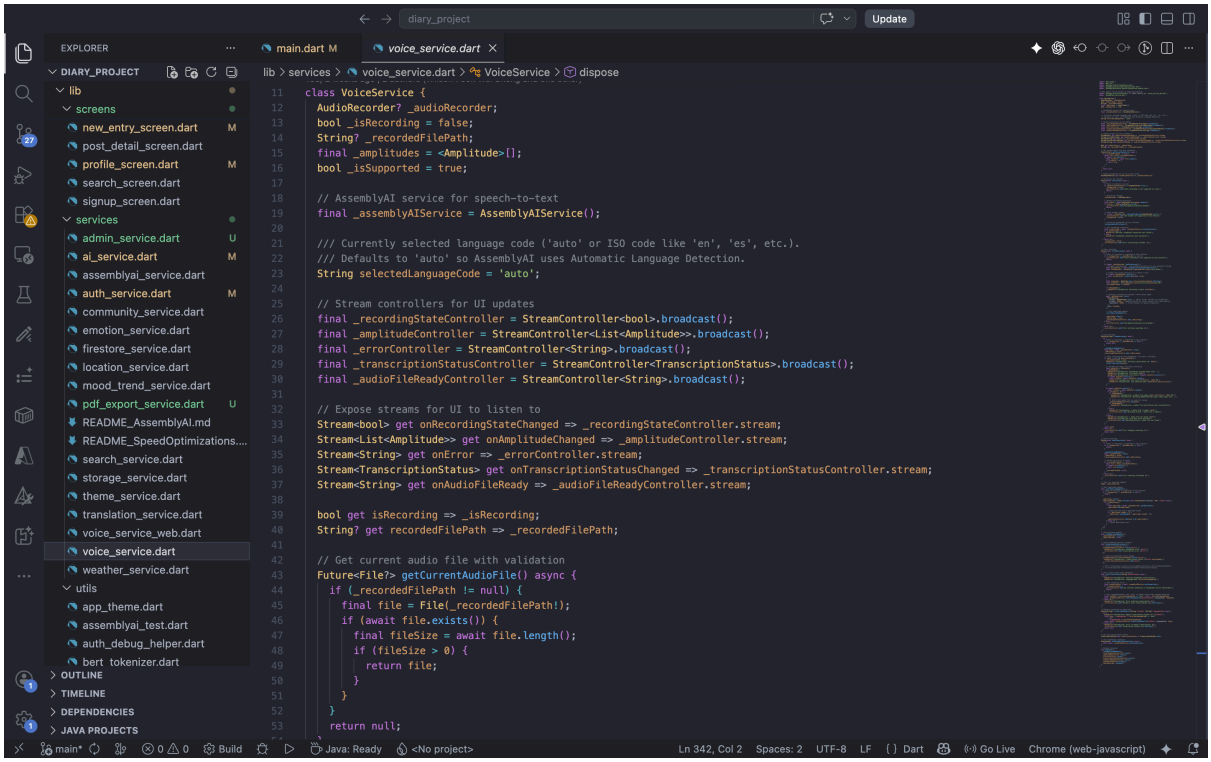


Figure 3.3.4.2 Voice_service.dart

3.3.5 Location and Weather Service Development

The Location and Weather Service will enhance diary entries by automatically associating contextual information about the user's immediate setting. Every time a user writes or records a diary entry, the app will obtain their geographic location as well as current weather conditions.

Refer to Figure 3.3.5.1, the location features function via location_service.dart. The app checks for a permission status and prompt user for permission to access their location through either the device's built-in GPS or via network based location services when the user creates a new diary entry. Once granted, the service accesses the user's latitude and longitude. The weather data is collected by weather_service.dart as shown in Figure 3.3.5.2. Its relies on an outside weather API such as OpenWeatherMap. The service uses the previously acquired latitude and longitude from the Location Service to invoke the API for live weather data.

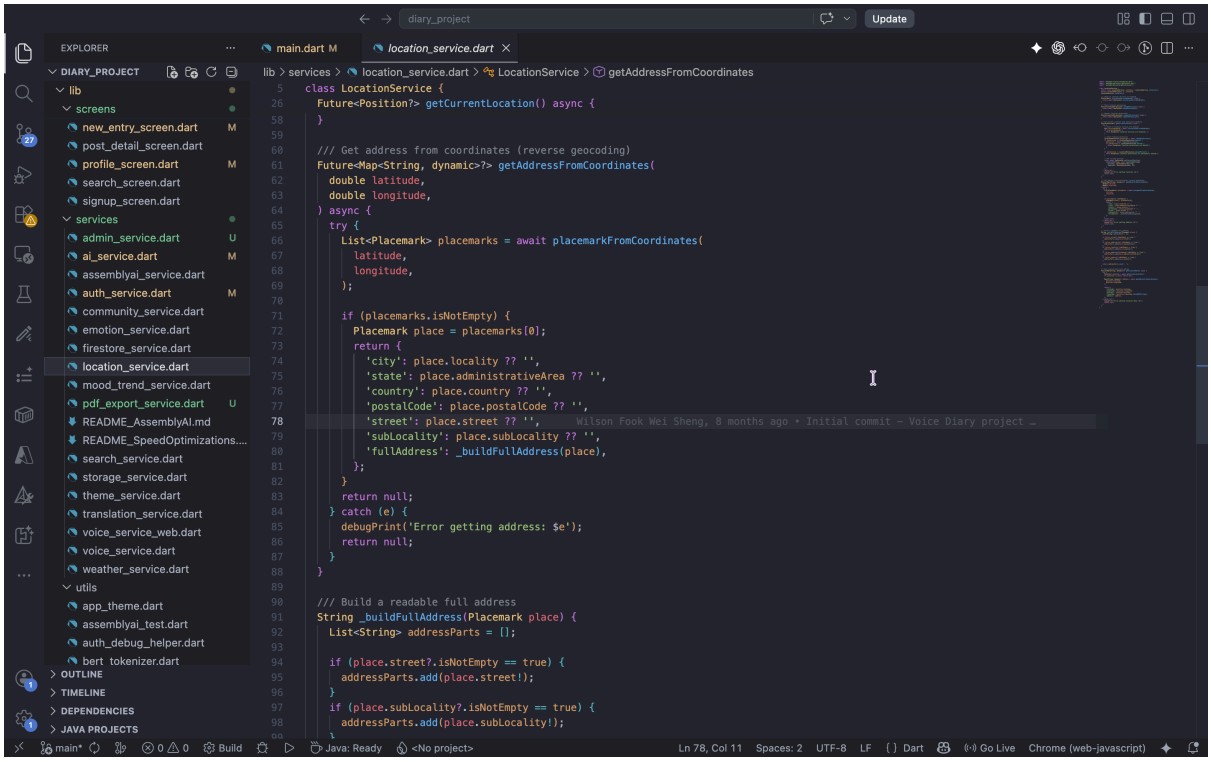


Figure 3.3.5.1 Location_service.dart

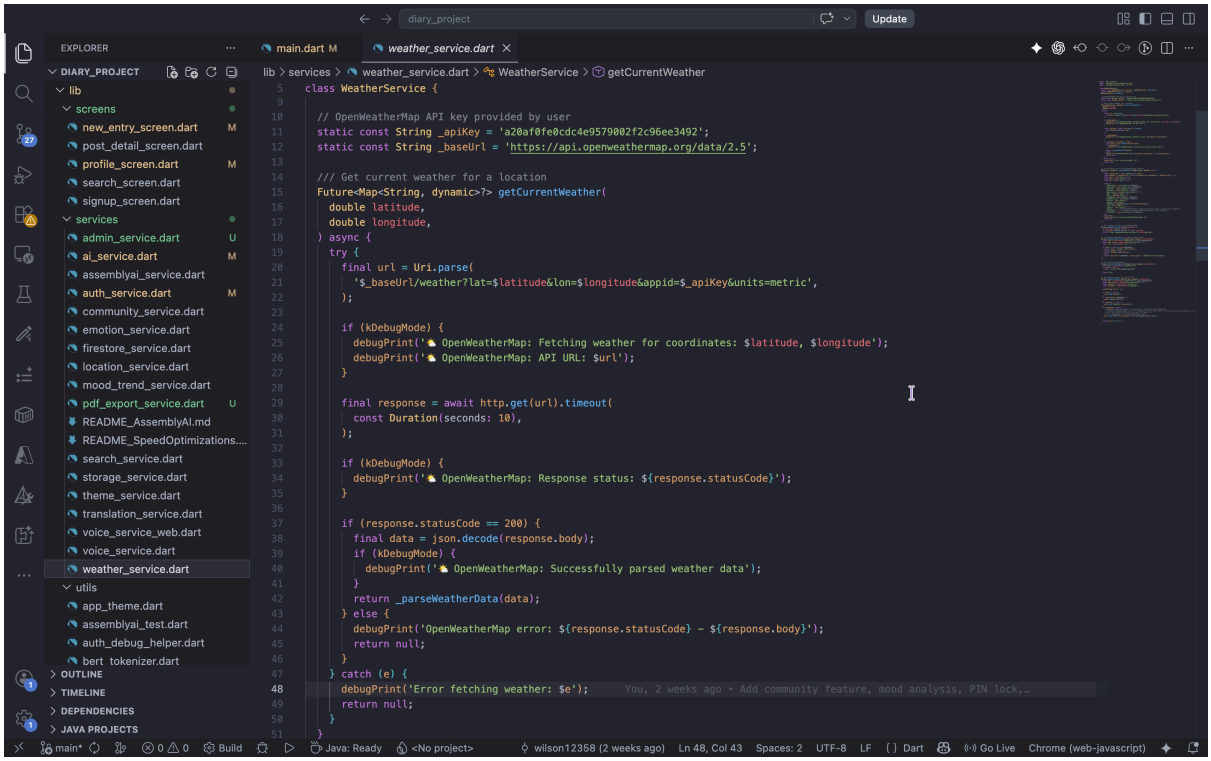


Figure 3.3.5.2 Weather_service.dart

3.3.6 Theme Service Development

The Voice Diary application offers a ThemeService that gives users some control of the user interface by providing a defined number of color themes. Each theme offers a definition of colors and each theme offered follows the same color definition that is statistically obtained and managed by the AppColors class. The AppColors class keeps all color definitions in one class and provides getter methods to return the currently active theme's primary, secondary, and accent colors.

Figure 3.3.6.2 show the ThemeService class is responsible for managing the themes. It implements the ChangeNotifier and uses the Provider pattern for state management to offer reactivity and responsiveness for all dependent widgets. The ThemeService class manages and tracks the currently selected theme for the duration of time it is selected. It initializes the theme every time the app is started in case of a saved theme state and will use shared preferences to persist the theme. On every occasion a user selects a new theme, the service updates the current state, assigns the new palette to the AppColors class refer to Figure 3.3.6.1, saves the selection in local storage, and invokes notifyListeners() such that all widgets relying on it will automatically rebuild with the new look. This ensures theme changes are deployed instantaneously across the entire application without requiring a restart.

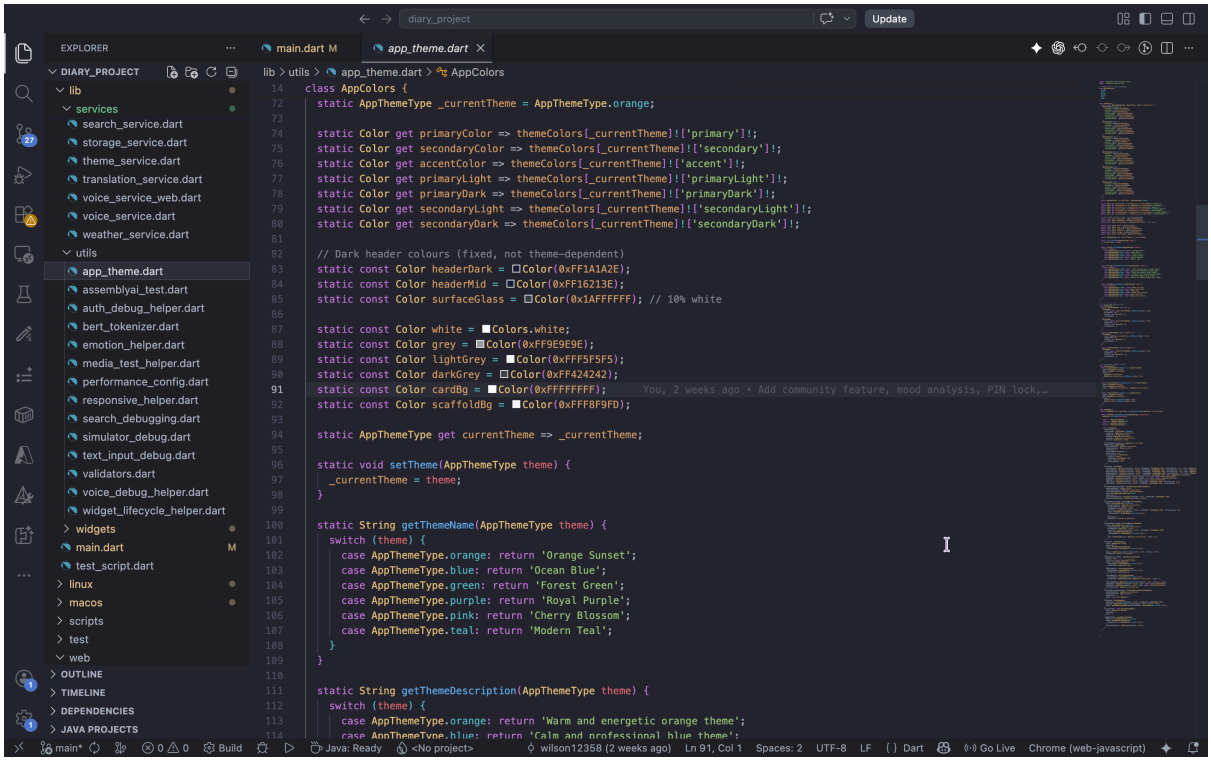


Figure 3.3.6.1 app_theme.dart

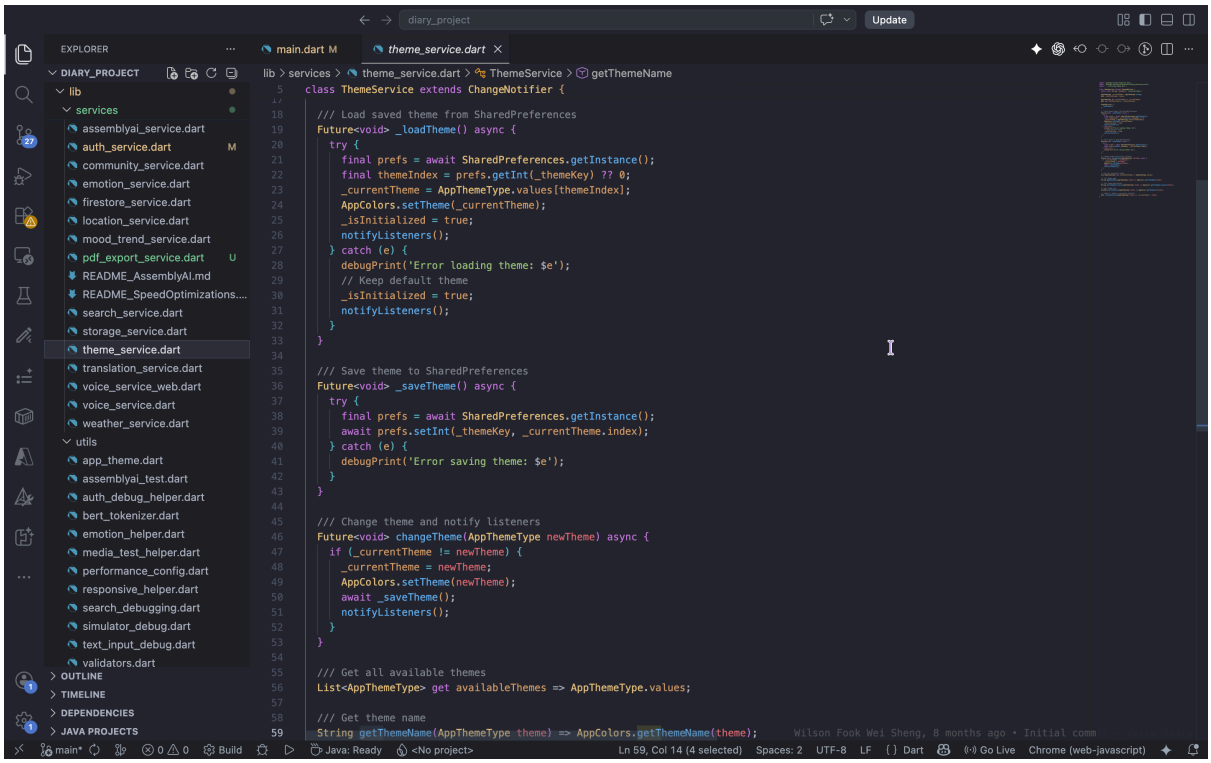


Figure 3.3.6.2 theme_service.dart

3.3.7 Search Service Development

The Voice Diary application's Search Service is implemented at search_service.dart that is designed to provide fast and user-specific retrieval of diary entries in Firebase Firestore. The main function of the Search Service is to allow users to search diary entries based on a title, the content of the entries, and tags. It also includes performance features such as caching and debouncing to improve speed as users search entries.

Figure 3.3.7.1 show the entry point for the Search Service is its main method, searchEntries, which accepts a userId, search query, and optional parameters for strategy, results limit, and cache. The service then forwards the search query to an appropriate smart optimized search or field-specific checks based on the strategy type selected by the user. Figure 3.3.7.2 is the _optimizedFirestoreSearch, which downloads the user's most recent entries and filters client-side. The service will divide the search query into meaningful words and determine matches on any substrings in the title, content, or tags. Multi-word queries need at least 50% of the words to appear, while single-word queries match if they appear in any field that matches. This kind of flexible matching allows users to find entries even with approximate or partial input. Besides, other approaches such as title, content, tags, and full text provide more specific or broad searches depending on the application.

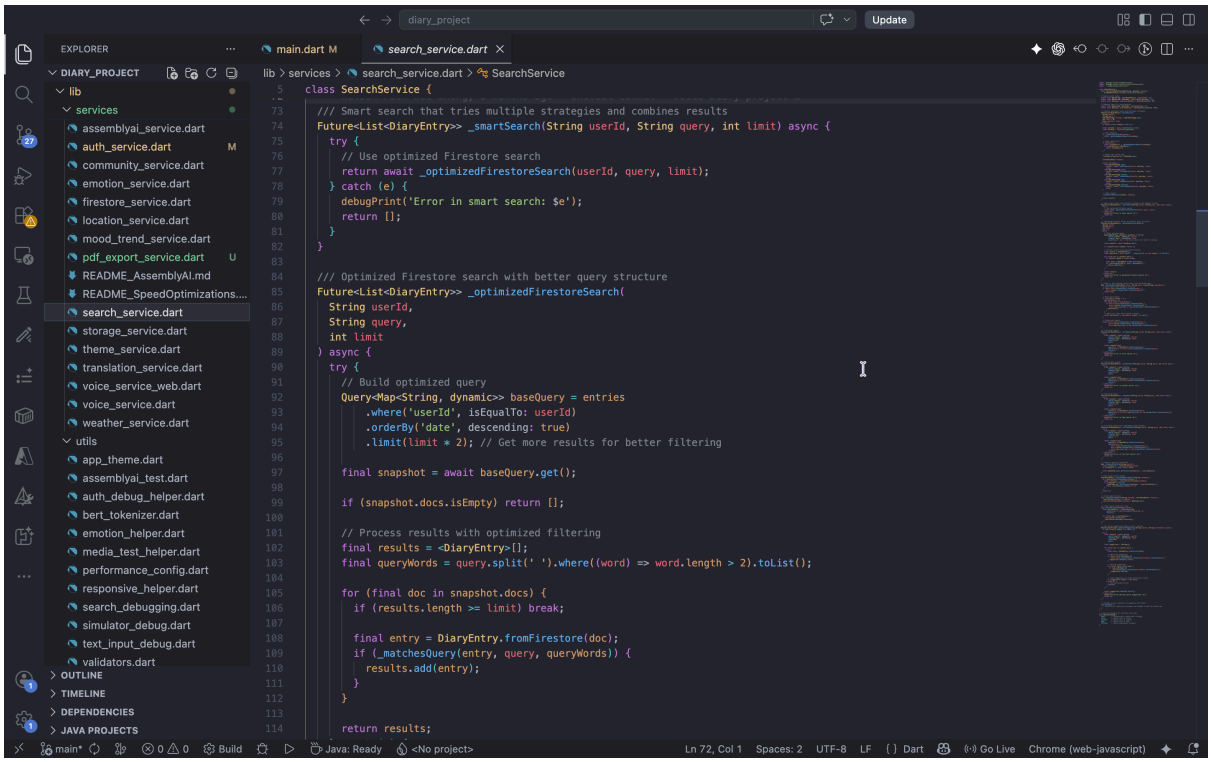


Figure 3.3.7.1 search_service.dart

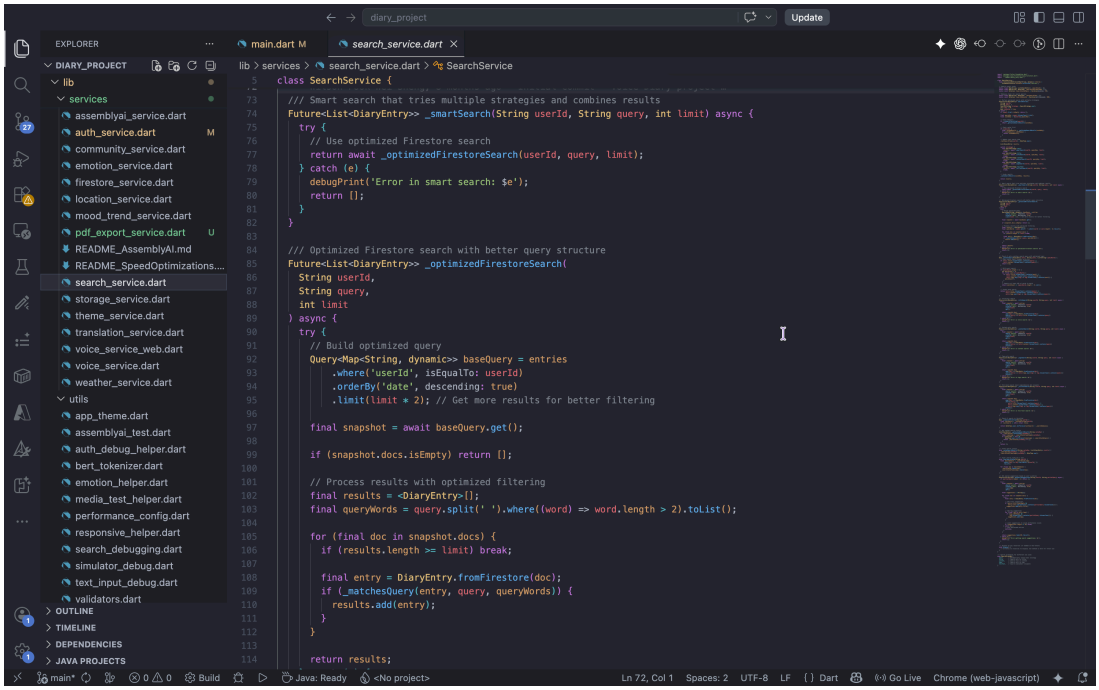


Figure 3.3.7.2 search_service.dart _optimizedFirestoreSearch

3.3.8 AI Mood Detection Development

The AI Mood Detection module is one of the essential intelligent modules incorporated in the Voice Diary application. The primary function of this module is to enable automatic detection of the emotional state of the user based on their input data from the diary entry. Figure 3.3.8.1 show the emotion categorized from 28 mood label into five different states: Happy, Sad, Angry, Fear, and Neutral. This module employs an efficient machine learning algorithm based on the DistilBERT (GoEmotions Student) model, which has been modified to run in TFLite format. The model weights and vocabulary file required for running this model on-device are saved in the local assets of the application, thereby eliminating the need for server interactions.

The emotion detection pipeline starts with the preprocessing of the input text through a specially created tokenizer. The text is cleaned, tokenized, and translated to numerical values such as input IDs and attention masks with a fixed length sequence. The output from the preprocessors is fed to the TFLite model that produces predictions in the form of probabilities on several emotion classes. This information is mapped to the major moods through consolidation of associated emotions and normalization. Multilingual input texts are first translated into English before processing. On the whole, the AI Mood Detection Module is a sophisticated combination of machine learning techniques that offer an intuitive and interactive experience to users.

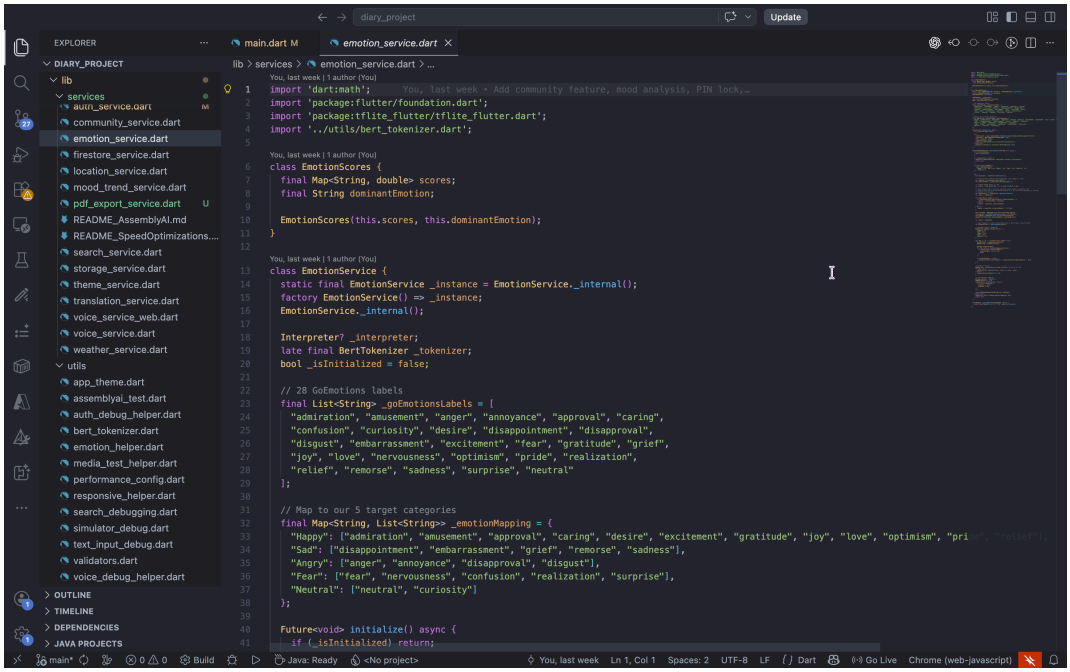


Figure 3.3.8.1 emotion_service.dart

3.3.9 Community Post and Sharing Development

The Community Post and Sharing Module is implemented through a dedicated services layer, which handles all the Firestore database operations. Figure 3.3.9.1 show the main logic of the module is incorporated into the CommunityService class, where all the backend logic related to post creation, getting posts, users, moderations, and notifications. Such design allows one to encapsulate all backend processes in order to have a more manageable solution. In order to initiate sharing the entry created by the user, he or she should press the Share button, which appears after the creation process. The Share functionality is triggered in the New Entry screen, after saving the entry. In this case, the application calls the function for creating a new post with such arguments as post content, detected mood, user details, and media attachments if there are any. The first values of the likes and comments counter should be set at this stage.

The real-time streaming function provided by Firestore is used to display posts on the platform. Users subscribe to data streams that contain specific information based on their level, including approved posts for regular users and pending posts for admin users. Whenever any changes take place in the database, this automatically updates the user interface. User engagement in the form of likes and comments is done in an efficient manner. The function of liking involves editing a field in the document that consists of an array. Users can hence choose to either like or unlike the post by using this method. Commenting involves adding comments to the post's subcollection through a batch write process.

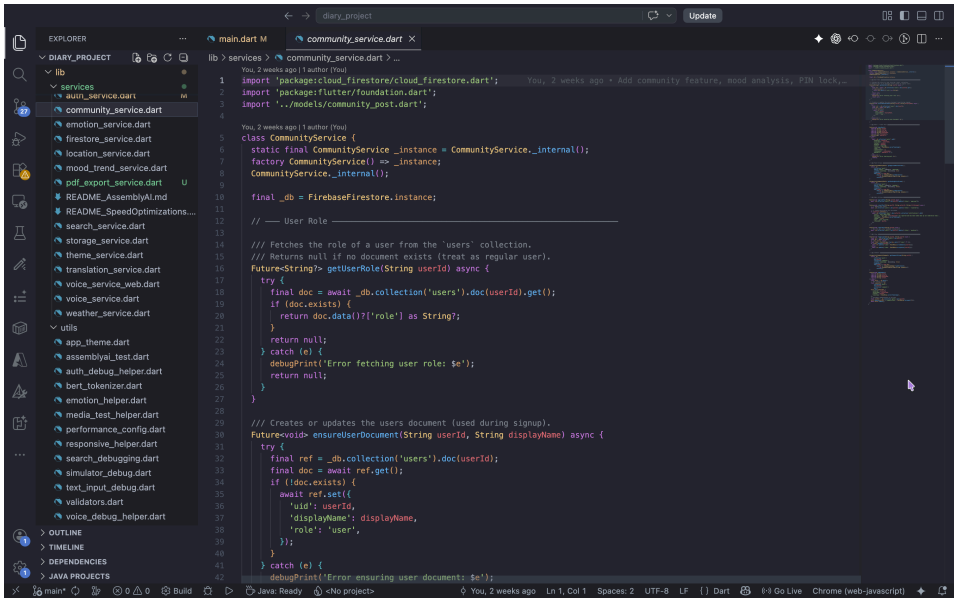
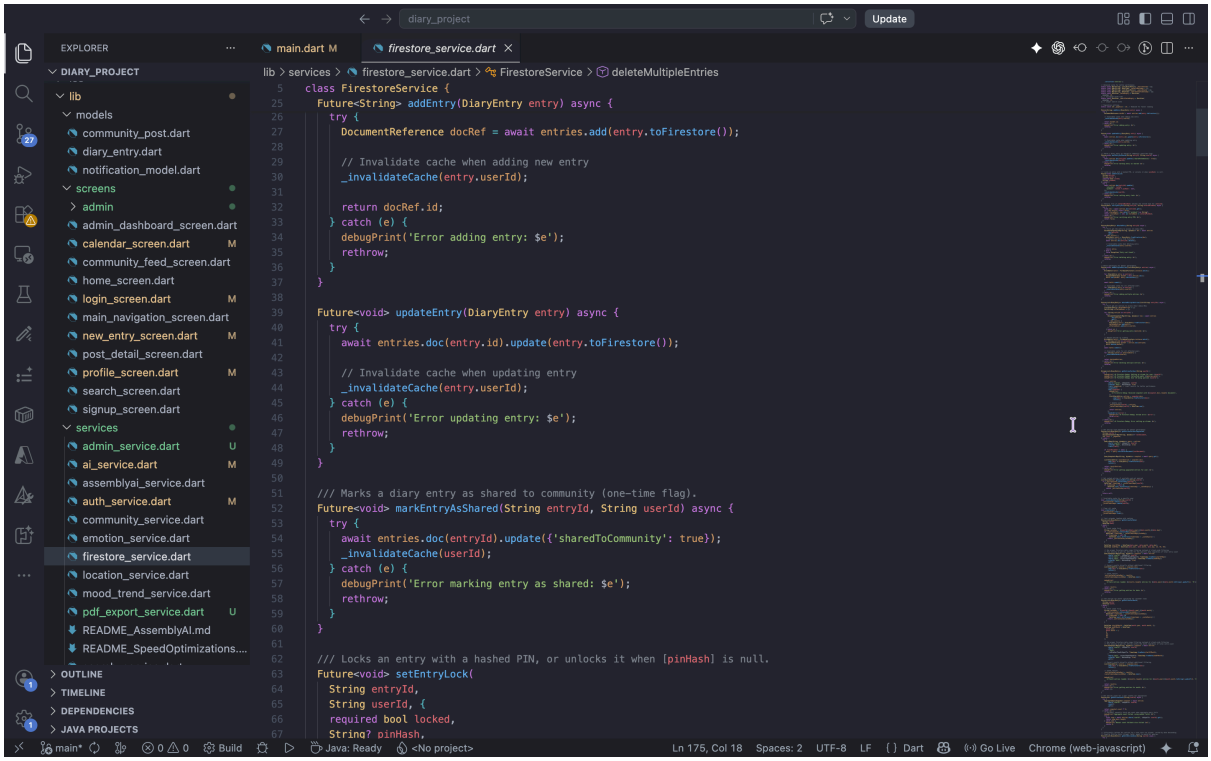


Figure 3.3.9.1 community_service.dart

3.3.10 Diary Entry Module Development

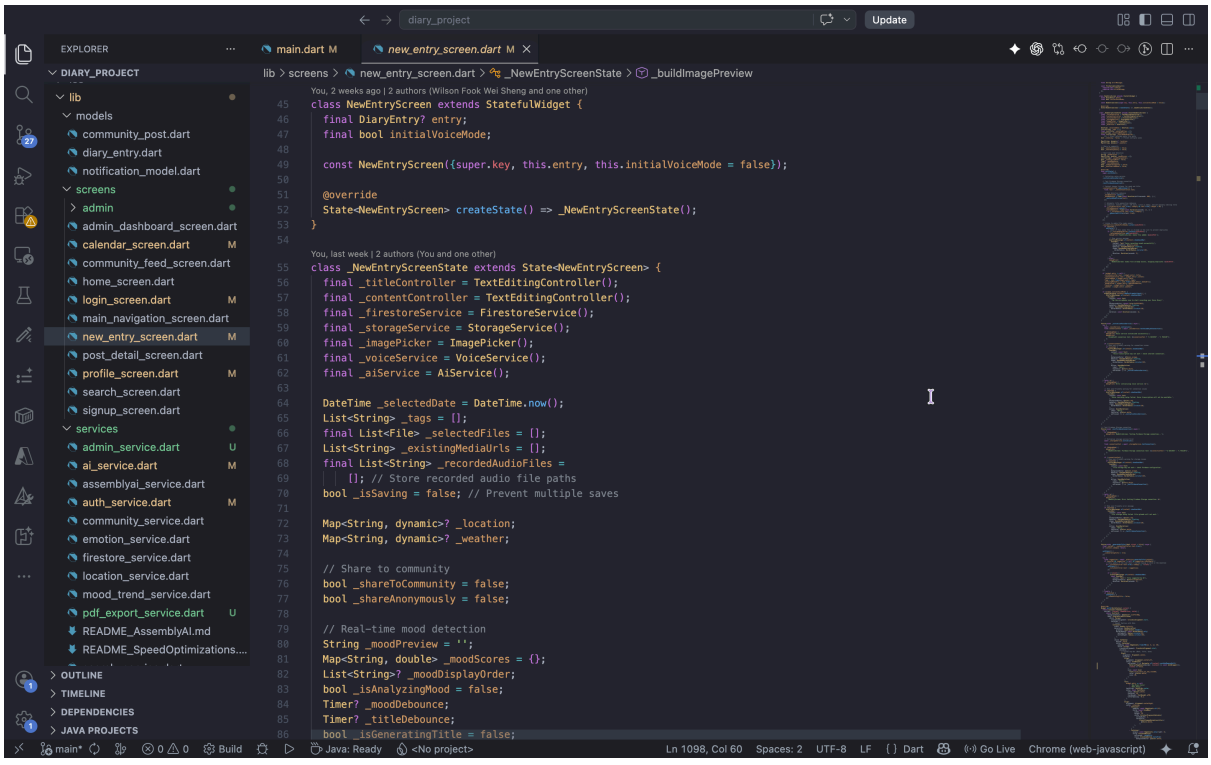
Diary Entry Module is one of the critical parts of the Voice Diary system, and it is used to manage all journaling-related functionality within the application. The structure of the module consists of three tiers, namely data model, service tier, and user interface screens. The data model is called the DiaryEntry, and it contains a wide range of properties for every created entry, including title, content, date, tags, URL of attached media files, emotion of the entry author, and other metadata, like location, weather conditions, and language settings. Figure 3.3.10.1 show FirestoreService is used to connect the app with the database. Hence, it acts as a single point for all the CRUD operations by ensuring the diary entries get saved, updated, deleted, and read effectively. In particular, the FirestoreService allows for creating and storing all created entries in Cloud Firestore and displaying them on the home screen using a data stream.

For the UI aspects, Figure 3.3.10.2 show the NewEntryScreen manages the whole flow of creating and editing an entry. The screen obtains all the necessary details from the user in terms of the title, content, tags, and chosen date, which are collected via form controllers. Other functionalities like media attachment, voice-to-text conversion, location and weather, and more can be added via other calls to services. During the saving flow, the system generates final mood information in terms of either AI-generated emotions or user choices, and a full DiaryEntry is prepared to pass to the FirestoreService for saving.



```
class FirestoreService {
  Future<String> addEntry(DiaryEntry entry) async {
    try {
      DocumentReference docRef = await entries.add(entry.toFirestore());

      // Invalidate cache when adding new entry
      _invalidateCache(entry.userId);

      return docRef.id;
    } catch (e) {
      debugPrint('Error adding entry: $e');
      rethrow;
    }
  }

  Future<void> updateEntry(DiaryEntry entry) async {
    try {
      await entries.doc(entry.id).update(entry.toFirestore());

      // Invalidate cache when updating entry
      _invalidateCache(entry.userId);
    } catch (e) {
      debugPrint('Error updating entry: $e');
      rethrow;
    }
  }

  /// Marks a diary entry as shared to community (one-time flag).
  Future<void> markEntryAsShared(String entryId, String userId) async {
    try {
      await entries.doc(entryId).update({'sharedToCommunity': true});
      _invalidateCache(userId);
    } catch (e) {
      debugPrint('Error marking entry as shared: $e');
      rethrow;
    }
  }

  /// Locks an entry with a hashed PIN, or unlocks it when [pinHash] is null.
  Future<void> setEntryLock(
    String entryId,
    String userId, {
    required bool locked,
    String? pinHash,
```

Figure 3.3.10.1 firebase_service.dart - addEntry



```
class NewEntryScreen extends StatefulWidget {
  final DiaryEntry? entry;
  final bool initialVoiceMode;

  const NewEntryScreen({super.key, this.entry, this.initialVoiceMode = false});

  @override
  State<NewEntryScreen> createState() => _NewEntryScreenState();
}

class _NewEntryScreenState extends State<NewEntryScreen> {
  final _titleController = TextEditingController();
  final _contentController = TextEditingController();
  final _firestoreService = FirestoreService();
  final _storageService = StorageService();
  final _imagePicker = ImagePicker();
  final _voiceService = VoiceService();
  final _aiService = AiService();

  DateTime _selectedDate = DateTime.now();
  List<String> _tags = [];
  final List<File> _selectedFiles = [];
  List<String> _existingMediaUrls = [];
  final List<String> _recordedAudioFiles =
      []; // Store recorded audio file paths
  bool _isSaving = false; // Prevent multiple saves

  Map<String, dynamic>? _location;
  Map<String, dynamic>? _weather;

  // Share to community
  bool _shareToCommunity = false;
  bool _shareAnonymously = false;

  // Real-time mood detection
  String _moodPreview = '';
  Map<String, double> _moodScores = {};
  List<String>? _moodDisplayOrder;
  bool _isAnalyzingMood = false;
  Timer? _moodDebounce;
  Timer? _titleDebounce;
  bool _isGeneratingTitle = false;
```

Figure 3.3.10.2 new_entry_screen.dart – class NewEntryScreen

3.3.11 Diary Entry Module Development

Coding of the Admin Dashboard Module has been done using a modular approach that isolates the code responsible for analytics processing, fetching data, and creating visualizations. The Admin Dashboard Module comprises two modules, namely the analytics dashboard and the moderation dashboard, both of which perform different functionalities. Figure 3.3.11.1 show the analytics dashboard is created using a StatefulWidget, which enables management of the state associated with aspects such as selected date range, interaction with charts, AI-generated analysis, and exportation of data. Real-time data is fetched using the streams provided by the AdminService, especially user data and diaries stored in Firestore. The streams are used by the StreamBuilder widget to fetch the data, after which the data gets updated automatically if any modifications are made on Firestore. Client-side data processing is performed in the absence of backend aggregation. The helper functions used to perform calculations include those responsible for generating analytics related to mood, gender, age, and engagement levels.

The visualization layer employs chart components with each chart type illustrating the particular analytics such as mood distribution, mood trend analysis , and daily usage patterns of users per day. Every single chart can be assigned with a GlobalKey, which enables the application to convert the widget into an image to be saved in a file for exporting purposes. The PDF export function works this way because it uses chart widgets to generate their images and passes them to the specific PDFExportService, which creates an exportable report.

Another functionality that has been introduced to the current module is the AI insight feature via AiService. The AI component analyzes the aggregated data about moods and produces the text insights, which help administrators understand user behavior.

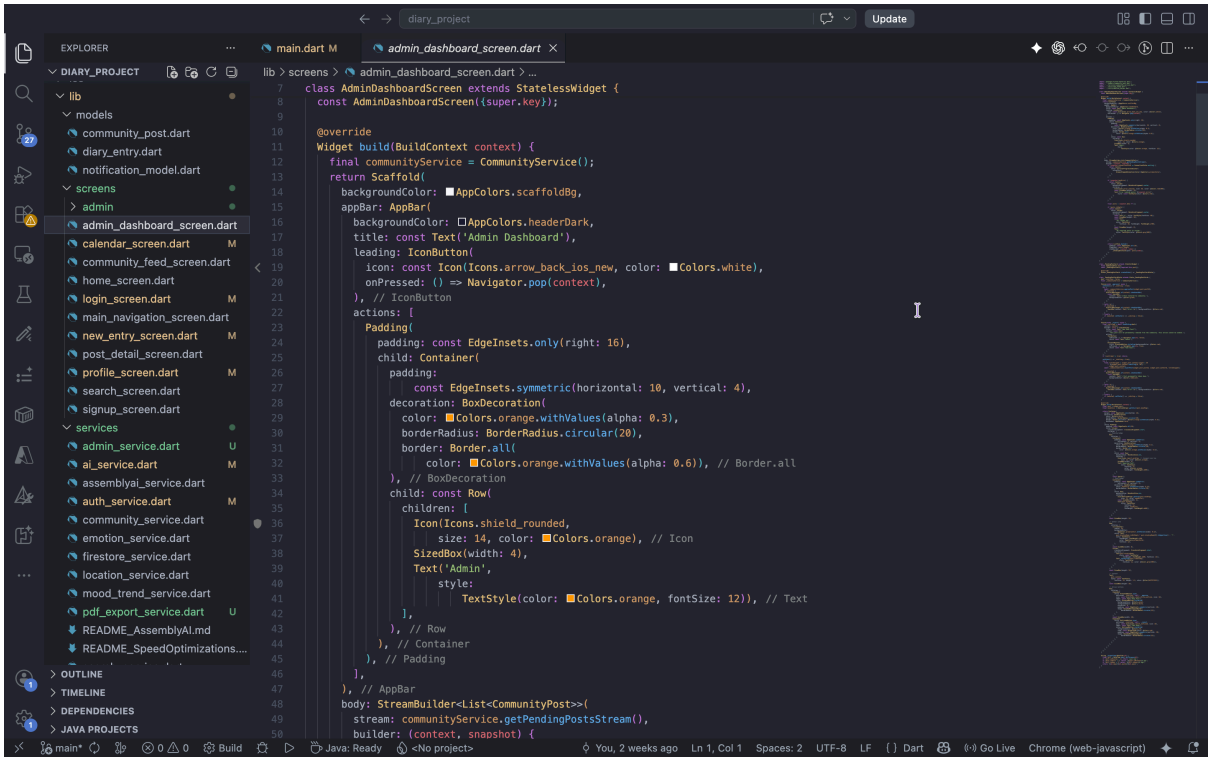


Figure 3.3.11.1 admin_dashboard_screen.dart – StatefulWidget

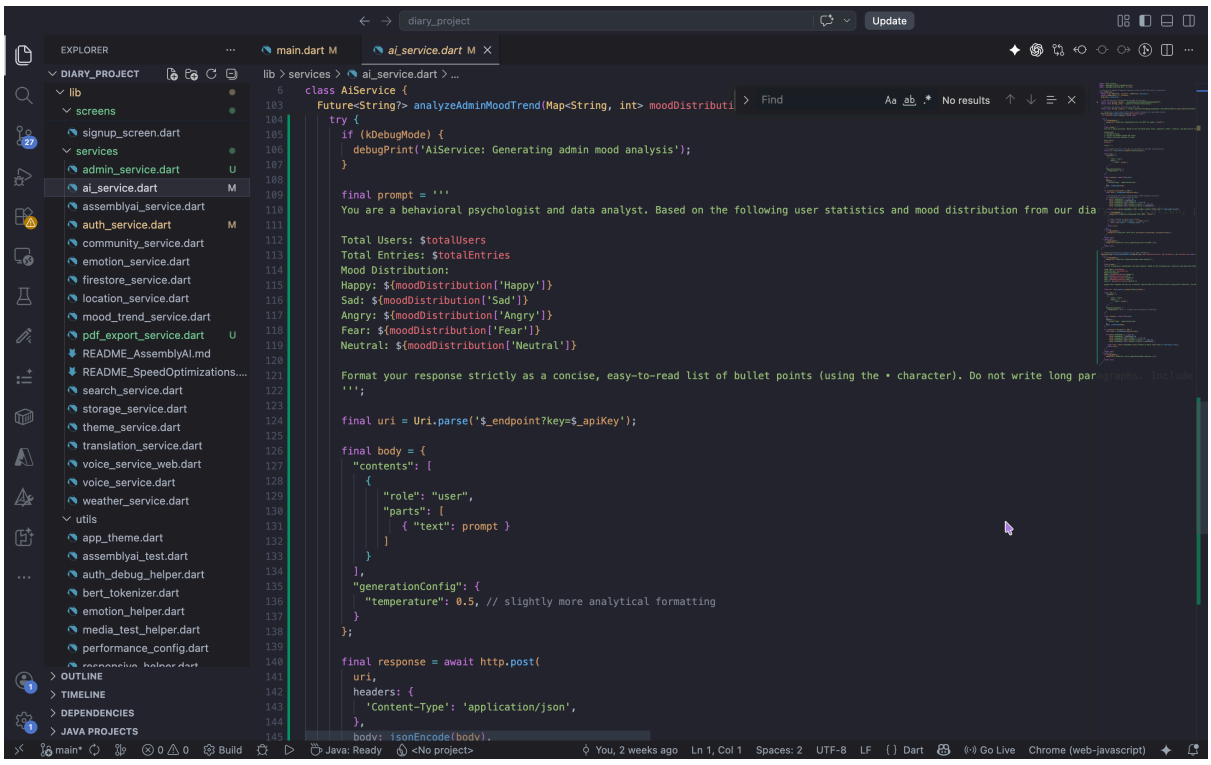


Figure 3.3.11.2 ai_service.dart – analyzeAdminMoodTrend

3.3.12 Entity Relational Diagram

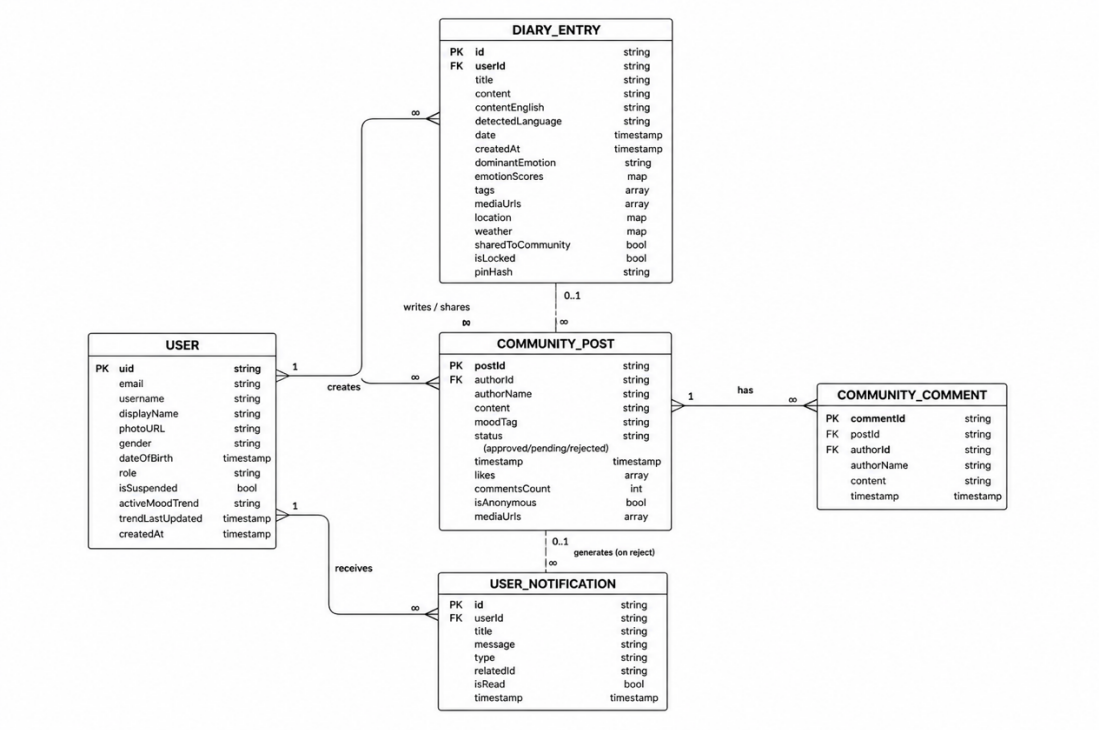


Figure 3.3.12.1 ERD Diagram of Voice Diary

3.4 Implementation Issue and Challenge

One of the major challenges I faced was the accuracy and reliability of speech recognition. First, I went with Vosk because it had the benefit of working on-device without an internet connection. But even after I tested it, in terms of the accuracy of the text it converted, the performance was just not good enough. Particularly displeasing was performance with longer than average sentences, accents, and background noise. Because of these limitations, the user experience was limited because diary entries sometimes needed so much correction that they often were not converted with no edit that the output was incomplete, incorrect or needed a lot of manual correcting and editing. To help address this problem, I explored other options and settled on implement the AssemblyAI API. AssemblyAI offered much better accuracy, solid noise handling, and support for multiple languages, which also made it a more applicable solution for varied users. The main drawbacks of AssemblyAI is that it does require an internet connection to use and also an application that creates some dependency on a third-party service. This transition, from Vosk to AssemblyAI brought out the challenges of working with different solutions in regard to offline capability versus accuracy and ultimately put a priority on user experience.

3.5 Summary

For summary, this chapter illustrates the system design included the use-case diagram and activity diagram that guide users have a well-known about the application function's flows. Entity-relational diagram (ERD) show the straight forward design of database, it provide a visualize of the relationship between entities. Moreover, this chapter provided information regarding the program development structure in which the design of frontend and backend were discussed. For frontend development, Flutter was used with a modular approach whereas for the backend, Firebase authentication service, Firestore, and storage services were used for the database purposes. Furthermore, multiple third-party APIs were used, such as AssemblyAI for converting voice to text, Gemini for AI analysis of text, OpenWeatherMap for weather data retrieval, and DistilBERT model for detecting emotions in the user's voice.

CHAPTER4 METHODOLOGY AND TOOLS

4.1 Overview

This chapter includes the methodology and technology used in development of Voice Diary mobile application. This will include the explanation of software requirements, hardware specifications, and the steps of the methodology used.

4.2 Design Specifications

4.2.1 Methodology

The methodology proposed for this project is Agile Development methodology. Agile is an interactive and incremental software development approach that provides flexible and continuous feedback in the development process [6]. Agile is the most suitable methodology for the development of mobile applications that have many changes in requirements throughout the project lifecycle.

Scrum will be implemented to develop the Voice Diary Android application. Scrum is one of the agile methods that focuses on managing iterative development rather than specific agile practices [7]. There are three phases in Scrum: the initial setup and planning phase, interactive development cycles (which are the sprint cycles), and the project closure phase.

Figure 4.2.1 Agile Methodology by Scrum Framework

4.2.1.1 Initial setup and planning phase

This is the initial phase of the project to identify the objective, vision, and scope of Voice Diary. At the beginning, the idea of creating a voice diary is proposed, and some problem statements

for this topic are found. Literature review of existing projects and applications to analyse the possible user requirements needed to develop a mobile diary application. The product backlog is outlined to have a prioritized list of features and tasks required to complete the application. The technique, tools, and language used to develop this project are planned and identified at this phase.

4.2.1.2 Iterative development phase (Sprint cycles)

Scrum relies on multiple sprint cycles, normally lasting 2 to 4 weeks. Each sprint includes the planning, developing, testing, and review of the ability of the features. At the sprint planning session, tasks will be selected from the product backlog based on priority at the beginning of each sprint. The product backlog for this project includes voice-to-text conversion, emotion detection, multimedia embedding, a tagging system, a search function, a searchable calendar interface, location and weather, theme colour selection, mood analysis dashboard, login and signup, community diary sharing, admin dashboard, mood trend notification, password pin protection. Then have a sprint review to evaluate completed features and gather feedback from users. At the end of each sprint, have a sprint retrospective to find out weaknesses and improvements for the next sprint. Scrum makes the development of the Voice Diary application focused and user-oriented.

4.2.1.3 Project closure phase

At the last phase of development of Voice Diary, final testing for all the modules of the application is tested to ensure proper functionality, stability, and performance. Then do the final adjustment based on the user feedback to have a great user experience. At the same time, the comprehensive documentation, like the user guide and technical notes, should be prepared at this phase to make it easier to manage the app in the future. After the testing and documentation are done, the Voice Diary app is ready to be launched and lets users start to use it to start their journaling process.

4.2.2 Tools to Use

i. Hardware Requirements

Component	Requirements
Model	MacBook Air (M1, 2020)
Operating System	macOS Version 15.4.1 (24E263)

Processor	Apple M1 Chip (8-core CPU, 7-core GPU)
RAM	8.0 GB
Storage	256 GB
Input	Keyboard and mouse
System type	64-bit ARM-based architecture

Table 4.2.2.1 Hardware components and requirements for mobile application development

Component	**Requirement**
Model	Honor Magic 4 Pro (LGE-NX9)
Operating System	Android 15
Processor	Snapdragon 8 Gen 1 (Octa-core)
RAM	8.0 GB
Storage	256 GB
System type	Snapdragon ARM-based, 64-bit

Table 4.2.2.2 Hardware component and requirements for testing

ii. Software Requirement

Component	**Requirements**
Tools	**Android Studio** Android Studio is the main Integrated Development Environment (IDE) use for building, debugging, and testing the Android applications [8].
	Flutter SDK Flutter Software Development Kit (SDK) is an open source framework introduced by Google for building cross-platform software by one codebase [9].

	Firebase Firebase is a platform developed by Google that simplifies mobile app development. It provides many tools and service for building , deploying, and managing applications, including databases, cloud-storage backup and hosting [10].
	Visual Studio Code Visual Studio Code is a multi-language and multi-platform powerful source code editor made by Microsoft. It has syntax highlighting, debugging support, code folding, and line numbering [11].
Languages, Libraries and Frameworks	**Dart** Dart is an object-oriented programming language developed by Google. It is designed for fast and scalable application by using flutter [12].
	Flutter Packages Flutter packages are third-party libraries and plugins that complement the functionality in Flutter apps. Packages help developers incorporate functionalities such as image capture, audio rendering, data visualization, in addition to state management, without coding from scratch.
	AssemblyAI API AssemblyAI is an online Speech-to-Text API that provides accurate transcription from speech to text. The API uses cutting-edge AI models in automatic speech recognition (ASR) [13].

	DistilBERT Base Uncased GoEmotions Student This model is a lightweight version of the BERT architecture, specifically fine-tuned on the GoEmotions dataset developed by Google.
	OpenWeatherMap OpenWeatherMap is a free and commercial weather dataset service that offers current weather conditions, forecasts, and geospatial-based data. We use it in this project to retrieve the user's local weather conditions and store these as part of diary entries.
	Gemini API Gemini is developed by Google and provide powerful natural language processing and generative AI features. Gemini 3.1 Flash Lite is chosen to use in the title generator and overall diary application report.

Table 4.2.2.3 Software components and requirements for mobile applications development

4.3 Timeline

4.3.1 Overview

The timeline is design refer to the methodology that used to develop this project which is Scrum methodology. Firstly, at the first sprint cycle is used for project preparation and setup. It is used to define the project backlog, to list out all the features that should include in the Voice Diary mobile application, such as login, signup, diary storing, voice-to-text conversion, search function, calendar view, diary community, emotion detection, multimedia embedding, community diary sharing, mood trend notification, password PIN protection, and so on. After defined the product backlog, flutter and firebase will be setup for ready to development process. The second sprint will be done for user authentication, the implementation of Firebase Authentication into diary application. Then will start for setup basic user interface for login and signup and create the user profile document in Firestore.

Third sprint cycle will set the goal for create and save basic diary entries. This will used the CRUD operation to create, read, update and delete diary entries. Sprint cycle 4 will be used to implement AssemblyAI API for voice-to-text conversion module. Then is the implementation

of cloud data handling for synchronization between multiple device. Lastly in project 1 will end with the internal testing to test the usability of the application and some bugs will be fix to have a seamless user experience.

For the project 2, it will start with the implement of sketch and drawing canvas, this module will integrate into the diary entry screen. Second scrum cycle will done with the build of diary community module, it will include the design of database structure, and peer testing on community module usability. Move to the sprint cycle 3, sentimental analysis module will implemented by DistilBERT Base Uncased GoEmotions Student, then the accurancy will be tested with sample data. Then will undergo the system improvement at sprint cycle 4. At this cycle Firebase queries will be optimized, conduct code review and static analysis. The sprint cycle 5 will conduct the full system integration testing and user acceptance testing (UAT) to collect feedback. Then an enhancement will be done to have a better user experience. Lastly, sprint 6 is dedicated to prepare project report and presentation slide.

4.3.2 Gantt Chart

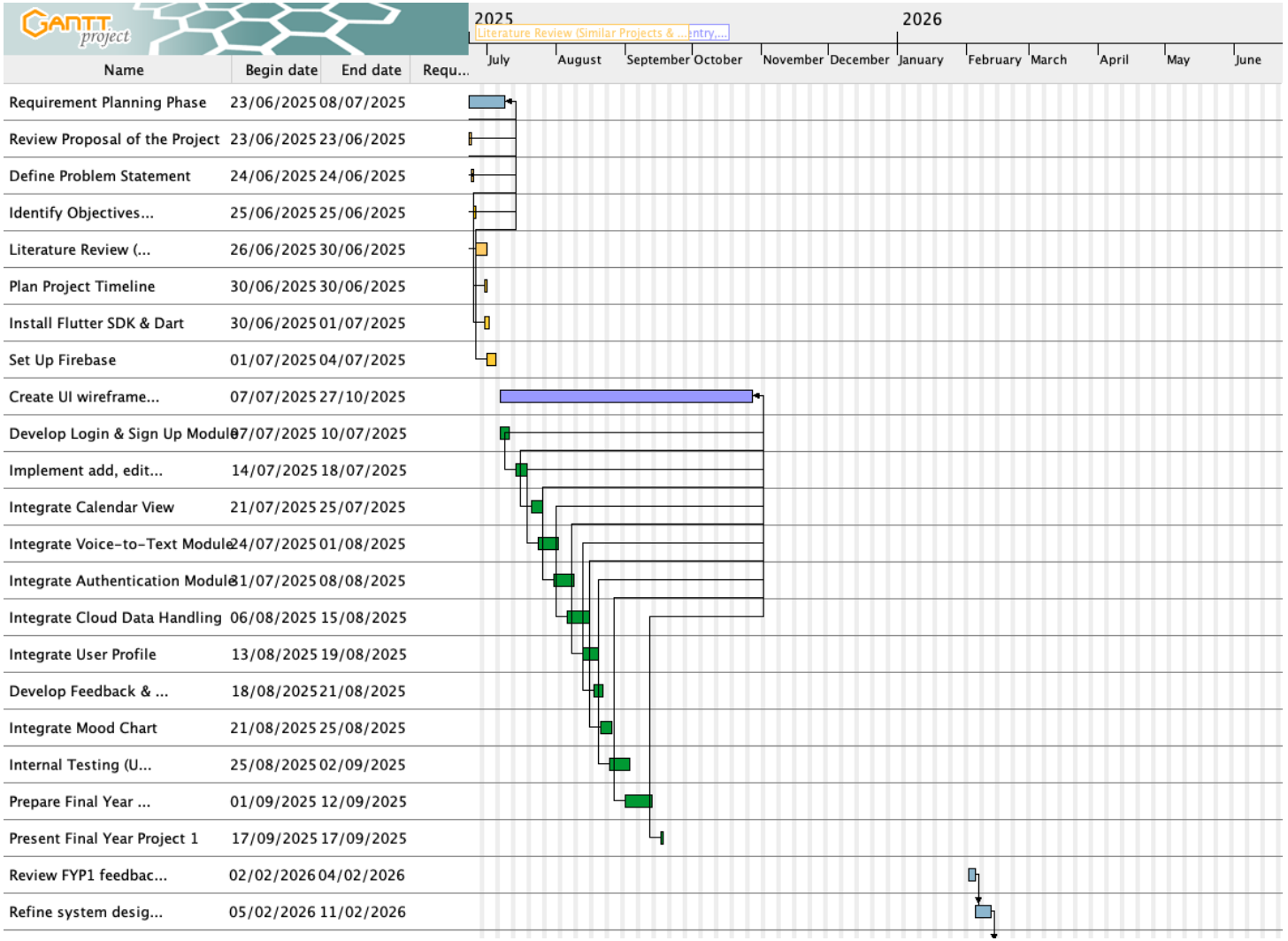


Figure 4.3.2.1 Gantt Chart for the Development Process

Figure 4.3.2.2 Gantt Chart for the Development Process (cont.)

Tasks

Name	Begin date	End date	Requirement Planning Phase
Requirement Planning Phase	23/06/2025	08/07/2025	
Review Proposal of the Project	23/06/2025	23/06/2025	
Define Problem Statement	24/06/2025	24/06/2025	
Identify Objectives and Project Scope	25/06/2025	25/06/2025	
Literature Review (Similar Projects & Existing Websites)	26/06/2025	30/06/2025	
Plan Project Timeline	30/06/2025	30/06/2025	
Install Flutter SDK & Dart	30/06/2025	01/07/2025	
Set Up Firebase	01/07/2025	04/07/2025	
Create UI wireframe (Login, Diary entry, Calendar)	07/07/2025	27/10/2025	
Develop Login & Sign Up Module	07/07/2025	10/07/2025	
Implement add, edit, delete diary entries	14/07/2025	18/07/2025	
Integrate Calendar View	21/07/2025	25/07/2025	
Integrate Voice-to-Text Module	24/07/2025	01/08/2025	
Integrate Authentication Module	31/07/2025	08/08/2025	
Integrate Cloud Data Handling	06/08/2025	15/08/2025	
Integrate User Profile	13/08/2025	19/08/2025	
Develop Feedback & Review Module	18/08/2025	21/08/2025	
Integrate Mood Chart	21/08/2025	25/08/2025	
Internal Testing (Unit Testing + Integration Testing)	25/08/2025	02/09/2025	
Prepare Final Year Project 1 Report	01/09/2025	12/09/2025	
Present Final Year Project 1	17/09/2025	17/09/2025	
Review FYP1 feedback from supervisor	02/02/2026	04/02/2026	
Refine system design & architecture	05/02/2026	11/02/2026	
Revise User Module Development	12/02/2026	19/02/2026	
Enhance Diary Module Development	20/02/2026	27/02/2026	
Implement AI Mood Detection Development	02/03/2026	11/03/2026	
Implement Community Module Development	12/03/2026	25/03/2026	
Implement Admin Dashboard Development	26/03/2026	02/04/2026	
Implement PIN Protection Module	03/04/2026	09/04/2026	
Testing & Debugging	10/04/2026	20/04/2026	
Prepare Final Year Project 2 Report	21/04/2026	27/04/2026	
Prepare Presentation Slide & Demo	28/04/2026	30/04/2026	
Present Final Year Project 2	07/05/2026	07/05/2026	

Figure 4.3.2.3 Gantt Chart Task Name of the Project Timeline

4.4 Conclusion

The method and techniques utilized in creating the Voice Diary mobile application have been detailed in this chapter. The Agile development methodology in combination with the Scrum technique has been employed in the application development process. Following specific development stages, such as the initiation stage, sprint cycles, and completion stage, the application development process has been carried out in a systematic manner but still remains adaptive to changes. The development process has allowed the system to continuously improve based on feedback and testing results.

Moreover, the design specification has been detailed in this chapter. The design specifications included both hardware and software components that were required during the process of developing and testing the mobile application. Modern technology, such as the Flutter SDK, Firebase, AssemblyAI, DistilBERT model, OpenWeatherMap, and Gemini API, has been applied to include innovative functionalities, such as voice-to-text transcription, emotion detection using artificial intelligence, and data enrichment by weather information and other features.

Moreover, the development timeline was designed taking into consideration Scrum sprints, thereby laying out the course of software development starting with basic functionality including authentication and management of diaries through to more complicated modules of community sharing, sentiment analysis, and optimization of the system itself. Using a Gantt chart as the visual aid for development ensured that all milestones of the project would be achieved in accordance with the initially planned schedule.

All things considered, this chapter proves that the selected methodology, tools, and approach to development were quite adequate in helping the developer implement an efficient Voice Diary application.

CHAPTER 5 IMPLEMENTATION AND TESTING

5.1 Overview

In chapter 5 will show the application showcase that have been done in Final Year Project 2. The modules that already develop in this phase are login and signup module, voice-to-text conversion module, emotion detection module, multimedia embedding module, tagging system module, search function module, searchable calendar interface module, location and weather module, theme colour selection module, mood analysis dashboard module, community diary sharing module, admin dashboard module, mood trend notification module, and password pin protection module. The mobile application was named as Voice Diary, and it provide an alternative ways to record entries by using voice. In the application, users can record, view, edit, and delete the diary entries. At the bottom of the application have a navigation bar to let users to select which page that they want to interact with, it arranged with Home, Search, Add, Community, and Profile.

5.2 Login and Sign Up Screen

After user install the application on their phones, the next step would be to type in their email addresses and passwords in order to gain access to the application. At this point, it becomes evident whether the person trying to access the system has the appropriate access credentials. The person will then be able to log into Voice Diary. For new users, there will be the option for signing up through the use of the ‘Sign Up’ link available.

For any login process, there will be the process of validation of input in relation to email address and password. In case the wrong email or password has been provided in any of the two input fields, the process will not work unless the right email or password has been provided. This will be done through the process of Firebase Authentication. Once input validation is complete, login to the system will be done.

In the Sign Up page, users are required to fill out some essential information such as username, password, gender, email address, and date of birth. Validity of this information will be confirmed to ensure correctness. Test case confirms the format of emails and compares whether the input password is equal to that in the confirm password field. Once they have passed these tests, the account is generated and stored in the Firebase database.

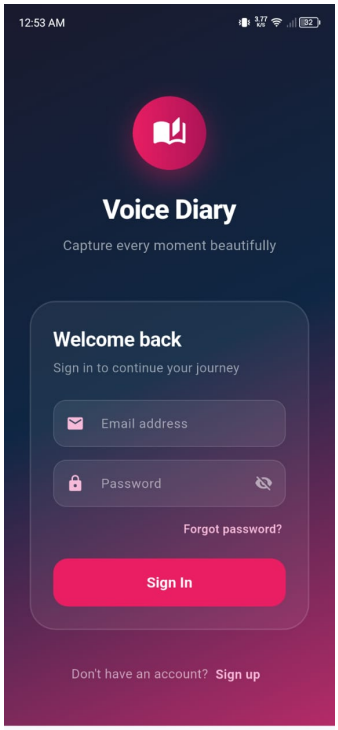


Figure 5.2.1 Login Screen

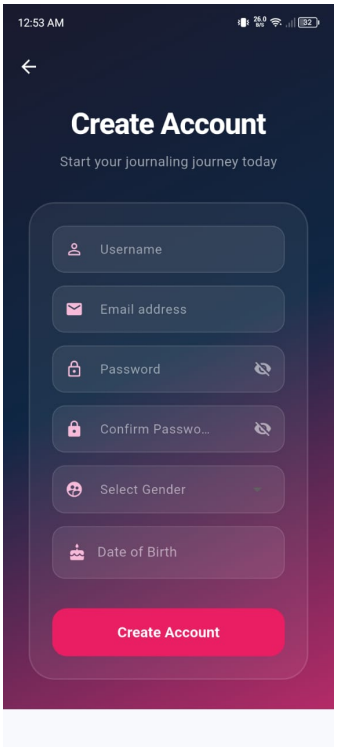


Figure 5.2.2 Sign Up Screen

5.3 Home Screen

Main page that users will see when opening the Voice Diary app is called the Home Screen and represents the main part of the user interface where they will find all needed options. In the upper part of the screen, there will be the name of the app and the user's personalized greeting, including user name in order to make a more personal atmosphere. The two buttons available here are "New Entry" and "Calendar" that let users either start a new entry or go back to previous ones via the calendar page.

Furthermore, there is the option of "Recent Entries". Here, the user will be able to see up to eight most recent entries with the corresponding information – the date, time, name of the topic, description, and current mood. Tapping any of them will show the whole information about the entry. Moreover, each entry will have a three-dot menu with several options, giving an ability to edit the entry, lock it for other people, share in the community, or delete the chosen entry. Lastly, the "+" button placed in the lower-right corner will help users start a new entry fast.

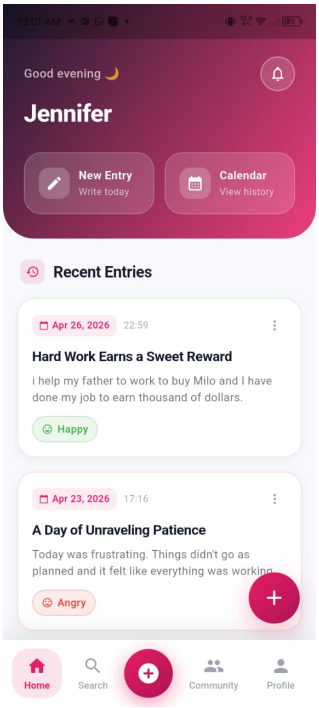


Figure 5.3.1 Home Screen

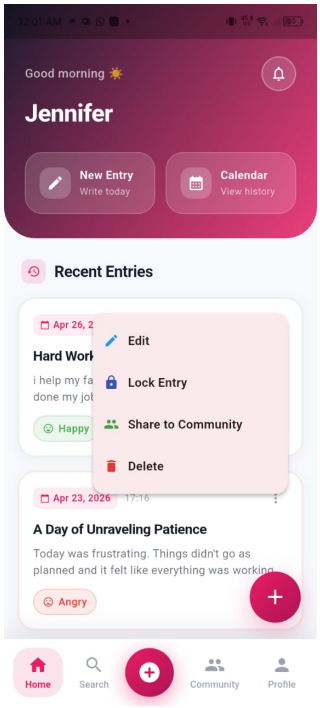


Figure 5.3.2 Manage Diary

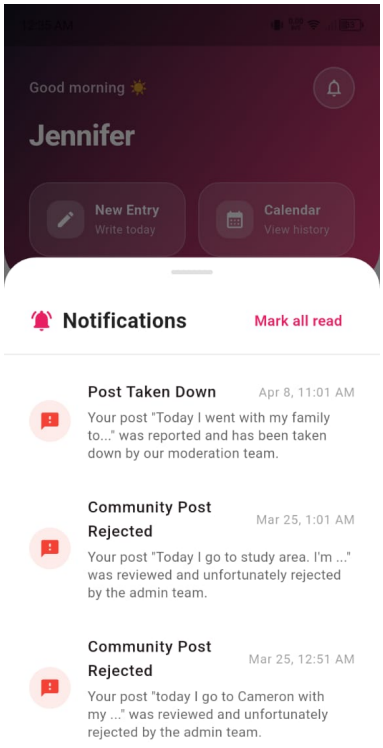


Figure 5.3.3 Notifications Message

5.4 Calendar View Page

At the calendar view page, users can view all their diary entries by just clicking the specific date. Those dates with blue dotted point is the dates contain with diary entries, this will simplify users to recognize which dates that contains of entries. After click on the dates, the diary entries will show in the below for users to review about it. User can switch the calendar display into 1 month, 2 weeks or 1 week, users can select it according to their preference. The calendar screen also support pull-to-refresh function to update the selected dates to ensure that the display entries is accurate.

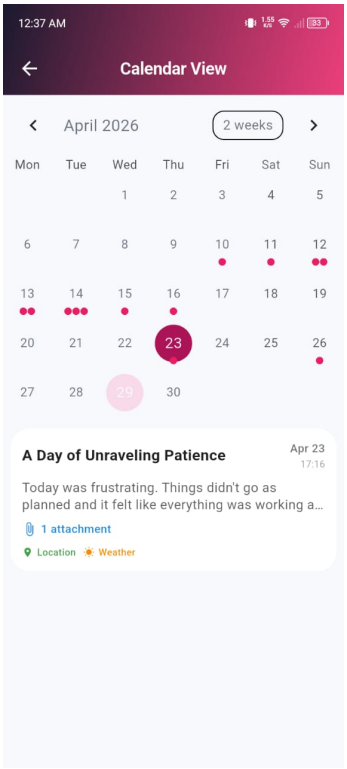


Figure 5.4.1 Calendar View Screen

5.5 New Diary Entry Screen

User can create a new diary entry in this screen. At the top, users can edit the dates of the entry, if not edit it will default as the current date. Next users can input a tittle for the entry. Mood rating system is below of the title section, users can rate their mood between 1-5 using icons and colour for better recognition. There is a location and weather detection based on users' current location, the system will auto include the location and weather data into the diary entries. When writing the diary entry, users can choose to write directly in the text box or use the voice recognition function that transcribe the speech into written text for convenience. After record the voice message, it will include into the diary entries and able to have playback function for future reference. There will also be a tagging notice for users to relevant add tags, which can be used to add useful and organized aspects of their diaries. The last 3 recent tags will at there

to help user quickly select and reuse them. The media will appear in a pre-view section where it will show either existing or new images, videos, or a recording of an audio file. Users can choose use camera to attach new photo or from the albums, the video files only support from gallery, and import audio from files. Finally, users can click the ‘Save’ button on the right top of the screen to save the entry into the database after including all the content. In addition, another feature that can be included in this module is sharing the diary entry with other users in the community through a toggle switch. After entering all the details that are required to be entered in the diary entry, a user simply needs to click on the “Save” button available at the top right side of the screen.

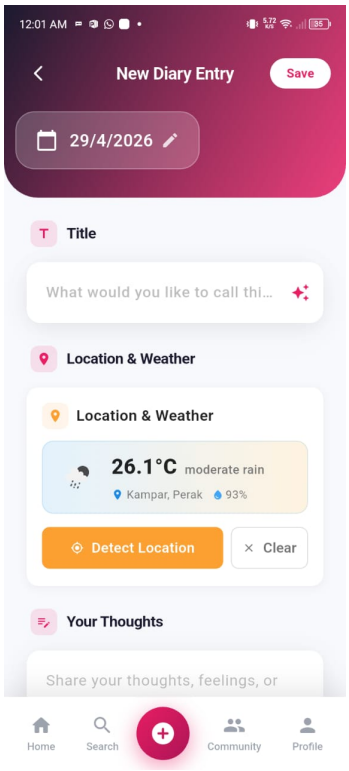


Figure 5.5.1 Diary Entry Screen (Tittle, Location and Weather)

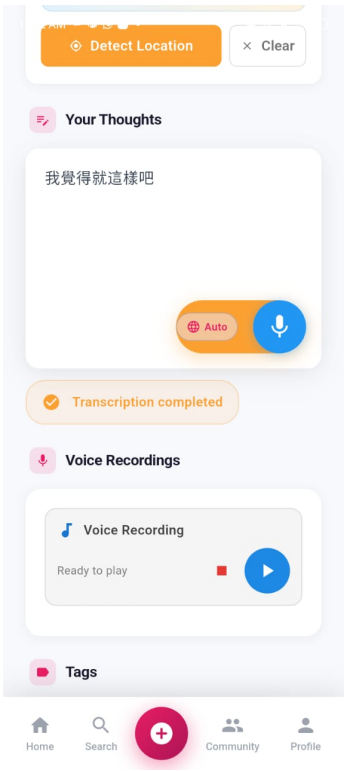


Figure 5.5.2 Diary Entry Screen (Entry Content, Voice Recordings)

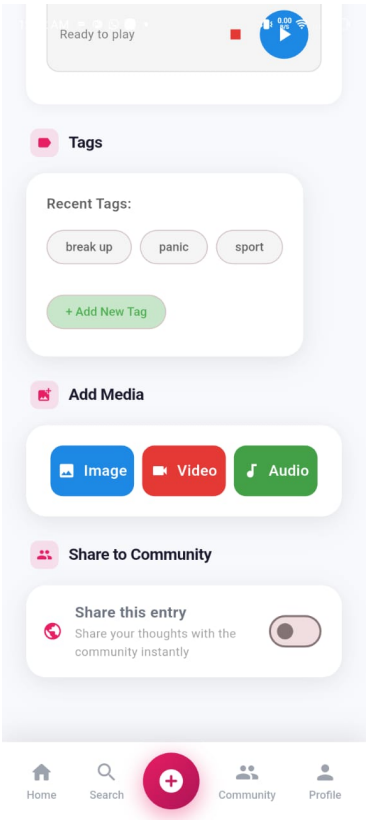


Figure 5.5.3 Diary Entry Screen (Tags, Media, Share to Community)

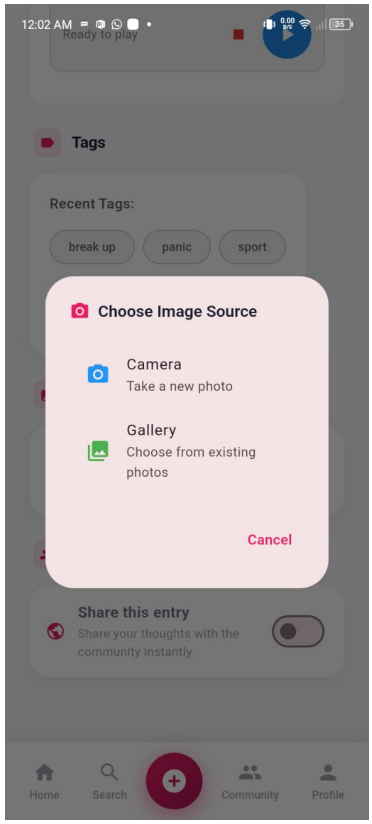


Figure 5.5.4 Selection of Adding Images

5.6 Search Diary Screen

This function can help the user find the desired entries of the diary through the application effectively. To begin, there is a search bar located at the top part of the screen that allows the user to enter keywords related to what they are looking for. In addition to this, the user can perform searches in various ways such as the title, content, and even tags of the diary. There is also another function known as the mood filter. Users can use this particular function to browse the entries depending on their mood states like Angry, Sad, Fear, Happy, and Neutral.

In case the user makes the search in such a way, then the application will display the result in the form of scrollable listing to the user. Each one of these search result listings will consist of information like the date and time of creation, topic of the entry together with the description of its content and finally its mood. However, in case there has been no input from the side of the user in the search box, then the application will automatically display to the user his most recent entries. Such display format will be similar to the format that the entries will be displayed in case of searches.

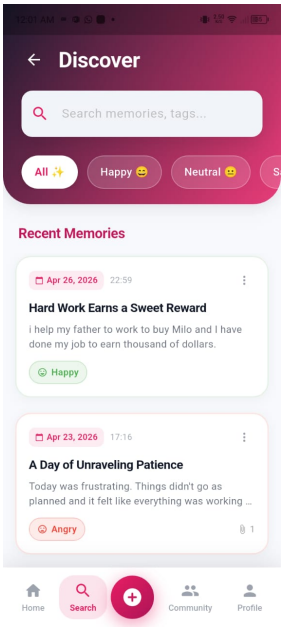


Figure 5.6.1 Search Diary Screen

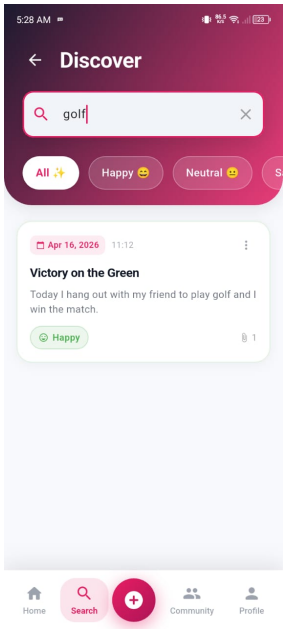


Figure 5.6.2 Search Result

5.7 Profile Screen

On the Profile page, users receive all information regarding personal information, usage statistics, and app settings. Firstly, at the top of the page, there are the profile picture of the user, username, and email. Secondly, users are free to edit their profile image and username here. On the lower part of the page, there is the "Your Diary Journey" widget where users can find all statistics and track their progress. Users are notified of their day streaks, and the total amount of diaries they have created, motivating them to write each day.

The profile also features an advanced mood analysis dashboard where the user's moods are shown graphically. Here the user sees the distribution of their moods depending on some period. Moreover, the page shows a trend graph of moods in recent entries. Correlation insights also appear here – patterns of emotions like frequent appearance of particular feelings on a particular day of the week or at a particular time of the day.

Moreover, Theme Selection is the feature that allows customizing the application in terms of its look and feel. The current theme selected is highlighted, while users are free to choose multiple options according to their preferences. Regarding account settings, there is an option to change the password in case the user desires. Users need to enter their current password first, after which they enter their new one twice for confirmation. There is also the possibility to sign out through the sign-out button at the bottom of the screen.

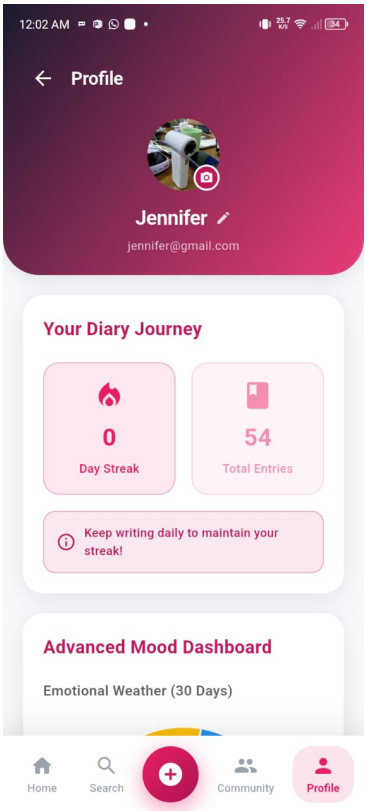


Figure 5.7.1 Profile Page (Profile Details, Diary Journey)

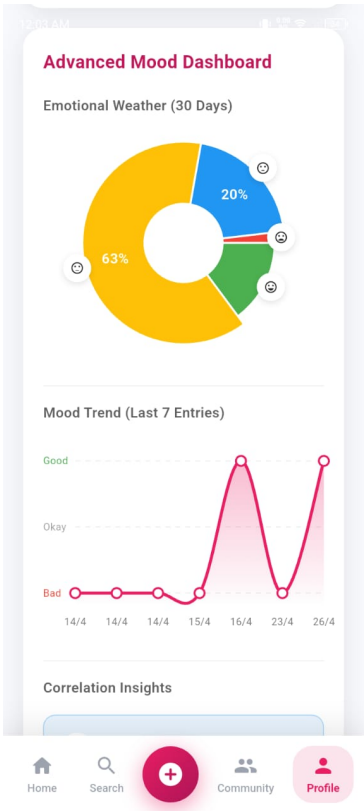


Figure 5.7.2 Profile Page (Mood Analysis Dashboard)

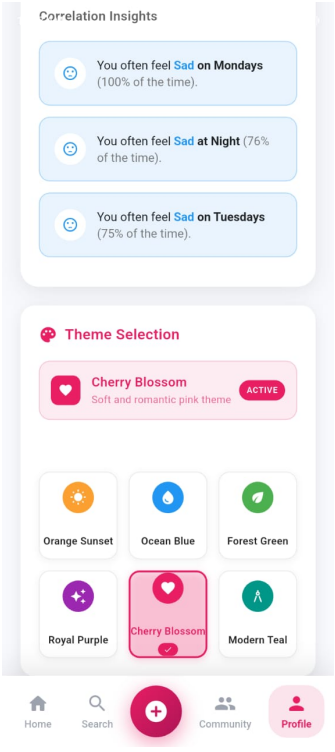


Figure 5.7.3 Profile Page (Theme Selection, Account Setting)

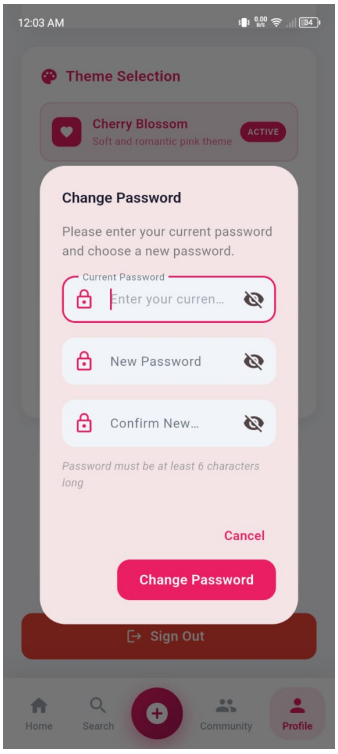


Figure 5.7.4 Change Password

5.8 Community Post Screen

The Community Post Screen provides users with an opportunity to find other people's posts that have been uploaded to the platform. Users will have the chance to interact with others through reading the posts in the community screen that other users have created. The posts appear in scrollable form, where each post comes with information regarding the poster's identity, the posting date, the mood, and text along with any media such as images. The likes and comments section enable users to comment on posts as part of building interactions and emotions among users. When a post is chosen by the user, user gets the complete post, comments from other users, and an opportunity to provide user comment in the comments section below the post.

To improve user experience, the Community screen comes with features of filtering and sorting. Filtering will be done depending on the mood categories including all, happy, sad, angry, fear, neutral. This will enable the users to look for the content that fits their emotional background. Also, there will be a sorting function where users will arrange posts depending on the newest, oldest, most popular, most commented on, or trending posts. In this way, the users will be able to navigate the community content as per their choices. In conclusion, the Community screen acts as a social media platform in the Voice Diary application, whereby users can share their stories and learn from each other.

Figure 5.8.1 Community Post Sharing

Figure 5.8.2 Post Details

5.9 Admin Dashboard Screen

Admin Dashboard is one of the main screens that provide information about all the activities and data available in the Voice Diary application. It was created as a tool for monitoring and analysis of the whole system by the administration team. In the upper part of the screen, there is a summary section where key statistics like the number of users and diary entries are shown, enabling the admin team to evaluate the application's scale of operation. Moreover, the Admin Dashboard contains the Analytics block where administrators have the possibility to select any dates when they want to see data tendencies. Another important characteristic of the Admin Dashboard is the inclusion of Gemini AI Analysis. This tool analyses all the gathered data and presents it as a short report. The report focuses on moods, behavioural characteristics, and users' activity.

Moreover, several data visualization tools are available to assist in analysing user behaviour. These include the mood distribution graph, which visualizes the ratio of various moods that users have experienced. The mood trend for the week graph demonstrates the trends in users' moods over time, while the engagement per day of the week graph displays the level of engagement from users throughout the week. Moreover, demographic analysis is also available, which provides further information about the nature of the users who utilize the application. Demographic analysis entails graphs that demonstrate gender distribution and age distribution, providing insight into the nature of the users who utilize the app. Finally, the Admin Dashboard also includes the feature of exporting the content as a PDF. This allows for the generation of a report of the data analysed via the dashboard in a PDF file, which can then be saved or shared.

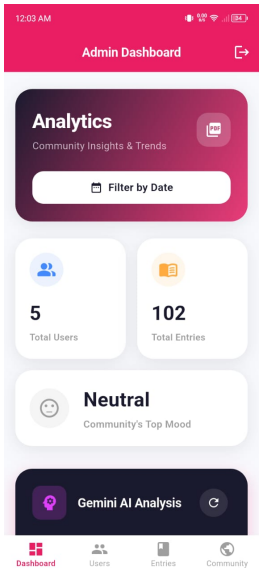


Figure 5.9.1 Admin Dashboard (Analytics Block)

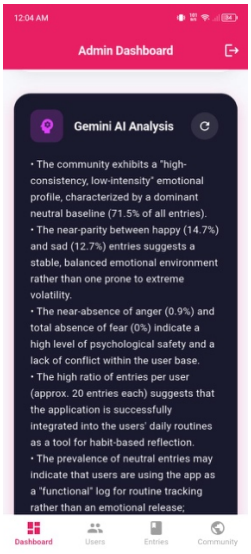


Figure 5.9.2 Admin Dashboard (Gemini AI Analysis)

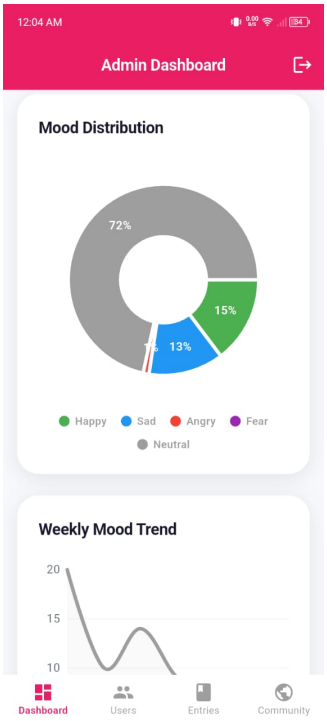


Figure 5.9.3 Admin Dashboard (Mood Distribution, Weekly Mood Trend)

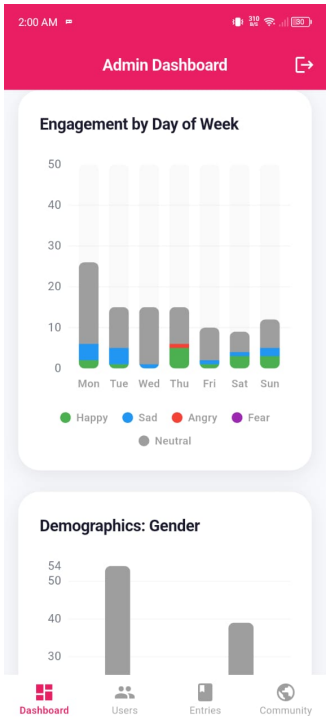


Figure 5.9.4 Admin Dashboard (Engagement by Day of Week, Demographics: Gender)

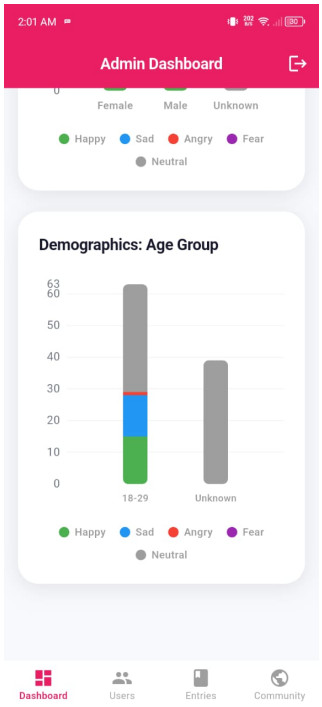


Figure 5.9.5 Admin Dashboard (Demographics: Age Group)

5.10 Manage User Screen

The Manage Users page is used by administrators for tracking and managing users' accounts on the Voice Diary platform. In the list of users displayed here, each user's account is shown in a card form which features basic information about the user, such as name, email address,

gender, birth date, role, and creation date of account. This helps administrators view information of different users in a neat way. The administration panel is equipped with options to manage users' accounts through actions performed on each user card. These actions include either suspending the user's account or deleting the user from the system. Suspension entails temporarily blocking the user from accessing the application whereas deletion results in the user being completely deleted from the system.

Moreover, there is the opportunity to touch on a certain user in order to see some additional details about him or her. For example, users can find the full user profile, which shows the user ID, email, date of birth, gender, join date, and user role. Next to the user profile comes an additional feature through which the users get the option of accessing all their entries from the user's diary. The manage user feature is quite useful as it helps the administrator manage the user account.

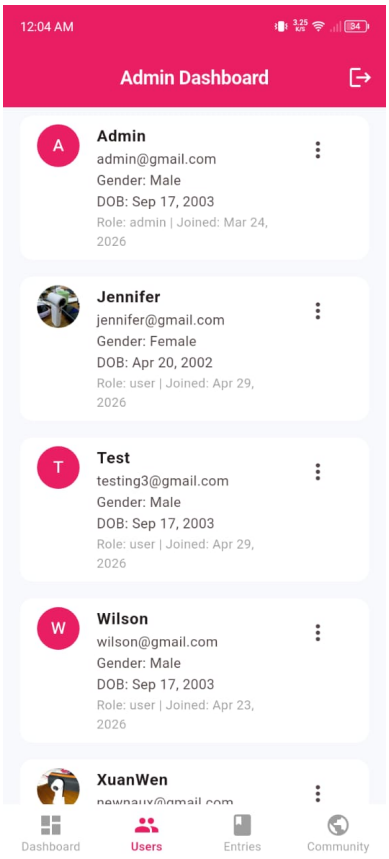


Figure 5.10.1 Admin Dashboard (Manage User)

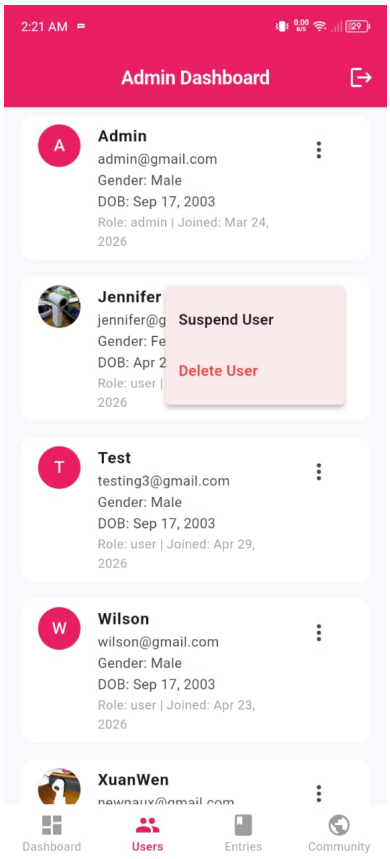


Figure 5.10.2 Admin Dashboard (Suspend, Delete User)

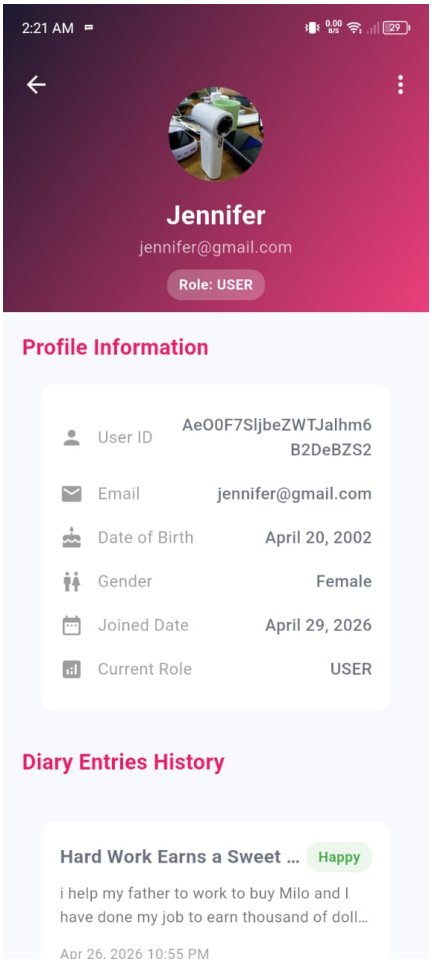


Figure 5.10.3 Admin Dashboard (User Profile Detail)

5.11 Manage Diary Entries Screen

Manage Diary Entries is an area for admins who will be able to view and control all the diary entries made by the app users in the Voice Diary app. This screen shows all entries in the form of cards that display important information like Entry Title, Entry preview text, User ID, Mood, and the Date & Time when the entry was created. Thus, it is easy for managers to monitor all entries from this screen. Moreover, it is possible to use search and filtering functionalities to find the required entry more easily. For example, an admin can use keywords in the search functionality to locate the entry. Alternatively, one may utilize the filtering options and choose a certain Mood (All, Happy, Sad, Fear, Angry, Neutral.) and even enter a certain date to filter the list of entries. Also, the feature comes with a delete icon which helps the administrator to delete any kind of unwanted data from the system.

In conclusion, the Manage Diary Entries interface provides vital means of managing content, helping the administrator to manage the diary entries through search, filtering, reviewing, and deleting the data within the system.

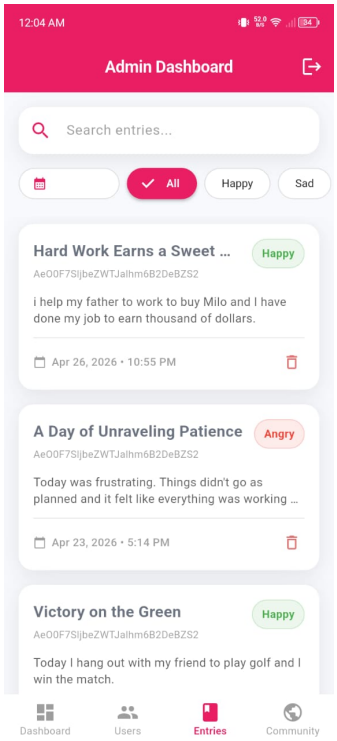


Figure 5.11.1 Admin Dashboard (Manage Diary Entries)

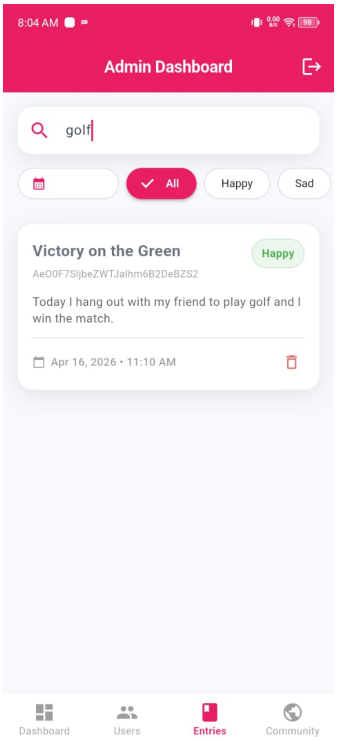


Figure 5.11.2 Admin Dashboard (Search in Manage Diary Entries)

5.12 Manage Community Post Screen

This feature provides a platform for administrators to manage all the posts that appear in the community page of the Voice Diary application. It is a screen from which an administrator will be able to see all the posts shared by members in a particular structure that involves details such as the name of the user who posted, date and time when the post was made, a snippet of the post, and any form of attached media. There is also a searching feature present on the screen which would allow the administrators to look up specific posts by using relevant keywords associated with posts and authors. Another feature available here is a date filter through which posts are identified through time intervals.

This screen is separated into two sections – All Posts and Reported. In the first section, all the posts published by members are listed, and the administrators are able to delete these posts if needed. This gives flexibility to the administrators to manage the content that might appear irrelevant or offensive in the community. Next, the Reported section contains those posts which are flagged by the members for being inappropriate or violating the community's rules. In the section, administrators are given the option to take action regarding these reports, and they can keep or delete a post depending upon whether or not they think a post is in violation of the rules.

Figure 5.12.1 Admin Dashboard (Manage Community Post)

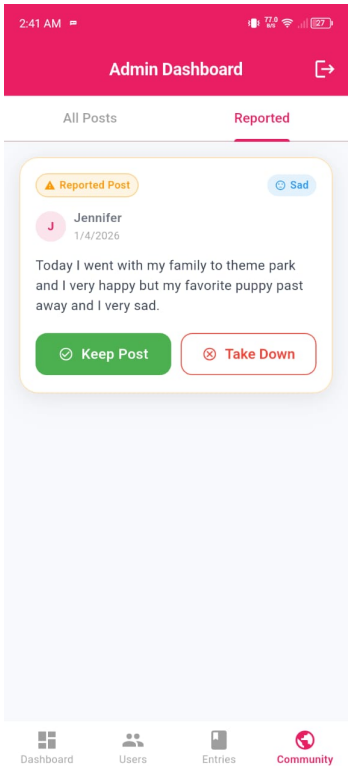


Figure 5.12.2 Admin Dashboard (Manage Reported Post)

5.13 User Testing

No.	Diary Content (Input)	Predicted Emotion	Actual Emotion
1	Today I had a really great time with my friends. We laughed, shared stories, and enjoyed good food together. I feel happy, relaxed, and thankful.	Happy	Happy
2	I feel very tired and unmotivated today. Nothing I do seems to work out, and I keep thinking about my mistakes. It's overwhelming.	Sad	Sad
3	Everything annoyed me today. Small things kept going wrong, and people were not helpful. I felt frustrated the whole day.	Angry	Sad
4	I'm nervous about tomorrow's interview. I keep imagining things going wrong, and it makes me anxious and uneasy.	Fear	Fear
5	Today was quite normal. I attended classes, completed tasks, and spent time on my phone. Nothing special happened.	Neutral	Happy
6	I achieved something important today, and I feel extremely proud. All the effort paid off and motivates me to keep going.	Happy	Happy

7	I feel lonely today. I didn't talk to anyone much, and I miss having meaningful conversations with people around me.	Sad	Sad
8	I got into an argument today, and it upset me. I couldn't control my emotions and now I regret how I reacted.	Angry	Sad
9	I feel scared thinking about my future. There are many uncertainties, and I don't know if I'm making the right choices.	Fear	Fear
10	It was just another routine day. I woke up, went to work, completed tasks, and came home. Nothing unusual happened.	Neutral	Happy
11	I had a peaceful evening watching the sunset. It made me feel calm and reminded me to appreciate small moments.	Happy	Happy
12	I feel disappointed in myself. I didn't perform well and couldn't meet expectations. It makes me question my abilities.	Sad	Sad
13	Someone blamed me unfairly today, and I felt angry. It's hard to accept criticism when it is not true.	Angry	Sad
14	Walking alone at night made me uncomfortable. I kept looking around, feeling like something might happen.	Fear	Fear
15	I spent the day organizing notes and cleaning my room. It was productive but nothing exciting happened.	Neutral	Happy
16	I felt happy helping someone today. Their gratitude made my day better and reminded me of kindness.	Happy	Happy
17	I was really annoyed by delays today. Everything took longer than expected, and it made me impatient.	Angry	Angry
18	I keep worrying about things that haven't happened yet. This fear makes it hard to focus on the present.	Fear	Fear
19	Today was simple and quiet. I stayed home, did some reading, and rested. It felt calm and uneventful.	Neutral	Happy
20	I lost my patience and raised my voice. I regret it now, but I was too frustrated at that moment.	Angry	Angry
21	今天真的很开心，我和朋友一起出去吃饭，还聊了很多有趣的事情。I feel so happy and grateful for this wonderful day.	Happy	Happy
22	Hari ini saya rasa sangat sedih dan letih. Nothing seems to go right, and I just feel like staying alone without talking to anyone.	Sad	Sad

23	我真的很生气，因为事情没有按照计划进行。Everything went wrong and people kept making mistakes, it was frustrating.	Angry	Sad
24	Saya rasa takut tentang masa depan saya. I don’t know what will happen next, and it makes me anxious and uncomfortable.	Fear	Fear
25	今天过得很普通，我只是上课、吃饭，然后回家休息。It was just a normal day without anything special happening.	Neutral	Happy
26	Hari ini sangat seronok, saya berjaya capai sesuatu yang penting. I feel proud and happy because my hard work paid off.	Happy	Happy
27	我今天感觉很孤单，没有人可以聊天。I miss my friends and feel emotionally low throughout the day.	Sad	Sad
28	Saya sangat marah hari ini kerana orang tidak bertanggungjawab. They made things worse and didn’t care.	Angry	Angry
29	我对明天的考试感到很紧张。I’m worried I might fail and that fear keeps bothering me.	Fear	Fear
30	Hari ini biasa sahaja, saya buat kerja rumah dan rehat. Nothing much happened, just a simple and quiet day.	Neutral	Happy

Table 5.13.1 List of AI Mood Detection Testing

The AI Mood Detection tool was tested using 30 diaries, including English and multilingual ones. For each of these diaries, predictions about users' emotional state were made by the system, which were then matched with the true emotions specified in the diary. According to the results of the testing process, the accuracy rate reached 66.67%, which means that 20 out of 30 diaries were classified correctly.

Various trends have been observed from the analysis of the results. First, the model is efficient in identifying definite emotional states, such as happiness and fear. Sentences containing explicit statements such as “happy,” “grateful,” “nervous,” and “scared” have all been accurately predicted by the model. Second, the system has proven to be effective in analyzing multilingual sentences because the translation module has efficiently translated texts written in languages other than English.

Nonetheless, there are some challenges in identifying emotions that belong to similar categories. One such challenge is differentiating between the sadness and anger categories. There are several diary entries filled with frustration or disappointment, but which the model classified

as “Angry” and not “Sad.” The challenge is due to the fact that the model perceives a strong use of negative words as angry language. Additionally, there is the problem of distinguishing neutral from slightly positive emotions. Diary entries about feeling calm or relaxed were identified as “Neutral,” but in fact, they should be categorized as “Happy.”

All things considered, the AI Mood Detection tool has shown decent results and is appropriate enough to identify users' emotions accurately. However, as mentioned above, it works well only with strong or explicitly expressed emotions. To enhance the model's performance, one can develop the classification categories further or improve the training dataset.

5.14 Summary

The use case study has been included in this chapter for illustrating the application showcase of the Voice Diary mobile application developed as part of Final Year Project 2. A number of core and advanced modules have been demonstrated that have been implemented successfully and make an intelligent journaling system. Some of the basic functionalities include user authentication, creation, modification, and deletion of diary entry. Besides these functionalities, there is support for advanced features like voice-to-text conversion, mood detection using AI, multimedia attachment, tagging system, and search functionality. With these functionalities, the system enables users to manage their daily experiences in an efficient manner.

Furthermore, the system includes a calendar view as well as search capability to enable users to search and locate the previous entries. The inclusion of location information as well as weather data makes the reflection even more valuable for the user. There is a profile module in place that assists users to have access to dashboards to perform mood analysis as well as get statistics about their application usage. It is also possible to personalize the use of this software through choosing themes and modifying account settings.

The third major component of the system is the community sharing module, which adds a social element to it by allowing users to share selected diary entries, like and comment on posts from other users, and connect with one another. This is further strengthened by the set of administrative components that include an admin dashboard, user management, diary entry management, and community post management.

Thus, this chapter has illustrated how Voice Diary application has been able to achieve the desired functionality by effectively combining the functionalities of traditional journals with advanced technology like AI, cloud services, and multimedia capabilities. The modules implemented in this chapter increase the efficiency and usability of this software along with improving the level of engagement. All these factors prepare the stage for testing and evaluation that will be discussed in the next chapter.

CHAPTER 6 CONCLUSION

6.1 Project Review, Discussion, and Conclusion

This project started by determining the issues that users meet in traditional diary keeping. Many people have trouble forming a consistent writing habit, whether through lack of time or no motivation. Furthermore, traditional handwritten diary do not have many intelligent features, such as mood detection, context and preparation , media options, that allow users to full engage in self-reflection with maximum opportunity for growth that is sustainable. Thus, this application was made to solve the mentioned problems and provide a more convenient, interactive, and smart journaling tool.

The created program combines several features to make the application efficient and user friendly. The feature that converts the voice to text makes it easier for users to add entries to their diaries without having to type. The feature of AI-based mood detection allows users to understand how they feel in different situations. The function of mood tracking lets users track their moods and see any pattern in the changes. Moreover, contextual information such as location and weather adds to the content of each entry and increases its significance when the users refer to it in the future. Other functions include adding images, tagging, and navigating diaries based on the date. Another useful feature of the application is the community sharing feature. Users can share their entries with other people and leave comments and likes. In addition, the admins will have the capacity to control user activity and also gain some information about user usage statistics via the admin dashboard. Other features that can be added are support for writing messages in different languages, protecting posts using passwords, theme customizations, and searching capabilities.

The project was developed using the Agile approach alongside the Scrum framework to allow for iterative and incremental development of the software based on the specifications of the system. Technologies implemented within the software development process included Flutter SDK for frontend development, Firebase as a backend infrastructure, AssemblyAI for speech recognition, Google Translate as a translator tool, OpenWeatherMap for the provision of weather information, and Gemini API as a provider for intelligent features like title generation and analytics. In conclusion, the proposed application satisfies the basic criteria, namely increasing efficiency, intelligence, and interactivity in journaling. The users can record their diary entries, analyze their emotional patterns, and manage their information via a variety of

intuitive interfaces. Despite not showing great accuracy, the AI Mood Detection feature proved that machine learning technologies are viable in creating personal diary applications.

In conclusion, the basic function of a diary application is developed and the first and third objective of the project is done during the Final Year Project 1, which is voice-to-text conversion modules, and saving of diary entries in application. Then during the Final Year Project 2, will focus on the advanced sentiment analysis, pin protection, , and diary sharing community to provide a smart and personalized journaling solution for users.

6.2 Novelties and Contribution

In turn, the Voice Diary app incorporates a number of innovations and contributions that set it apart from standard diaries and current digital journal apps. First of all, what makes this project unique is the introduction of voice-to-text functionality along with intelligent diary management. As opposed to regular digital journals, the project facilitates users in creating their diary entries orally, thus simplifying and speeding up the process of journaling, while also increasing accessibility of the latter at any time and place. Moreover, another significant contribution made by the project lies in the creation of a mood detection module based on AI and powered by machine learning on-device. In particular, the system can automatically recognize and detect users' emotions, both negative and positive, through analysis of the diary content. Moreover, the combination of sentiment analysis and diary entries allows users to reflect on their own emotions in the long term. Another interesting contribution lies in the multilingual capabilities of the app as the users are allowed to write in various languages due to machine translation.

Apart from this, there is another way in which the app can offer its own contribution to this whole concept, which would be by bringing together all of these aspects in one place. With all the contextual and multimedia information integrated in this manner, the geographical and weather data will automatically give some context to every single point being entered into the diary. It would make the process of analysis and reflection on these entries even easier in the future. In addition, giving the feature of sharing their entries with the rest of the community can take the whole process of journaling to a different level altogether. It is because the feature of sharing will allow the users to interact and connect with each other based on their interests and experiences. Also, the possibility of leaving comments and reacting with likes on posts made by others can contribute to facilitating the interactive process.

From the technological point of view, the app is an example of how modern technologies are successfully used in practice. In particular, Flutter is utilized to develop a cross-platform application, while Firebase provides backend services and third-party APIs are used to implement different features. The modular architecture, proper state management with the help of Provider, and the use of a service-oriented architecture guarantee that the system will be highly scalable and maintainable. To conclude, the Voice Diary application introduces some novelties to mobile application development since it is an advanced and intelligent diary application.

6.3 Future Work

Even though all the major features of the Voice Diary have been realized successfully, there are several directions that require further development and improvement. One of the key aspects in this regard relates to improving the AI Mood Detection algorithm. It should be noted that at this moment the accuracy rate is rather moderate, and it is impossible to distinguish certain emotions from others (e.g., between sadness and anger), as well as neutral moods from mild positive ones. The improvements might include tuning the machine learning algorithm in relation to the larger database, adding context to determine the emotional state of the person, as well as the possibility of classifying multiple emotions instead of just the dominant one.

Moreover, one more improvement in development is multilingual capability. Today, the application is designed to have an ability to analyse emotions in translated texts using a translation API and thus requires analysis in English language only. However, in the future, it can be developed to have a model capable of analysing in different languages.

Moreover, voice recognition technology can further be developed to include offline capability and proper recognition of accents and various conditions. With that, the use of this technology will become much more convenient for users, especially in the regions where there is poor connectivity to the Internet. Additionally, an increase in transcription accuracy and the inclusion of voice feedback would contribute to enhancing the general user experience.

Speaking about the user experience itself, a few more features, such as personalized suggestions, smart reminders, and even habits monitoring could be used to promote regular journaling among users. Namely, the application could offer writing ideas in accordance with moods experienced by users or notify users about their progress. In addition, more advanced analytics, which include emotional trends identification, could be included in the app.

REFERENCES

[1] M. Nazrin, *Mobile Application for Personal Journal/Diary*, Universiti Teknologi PETRONAS, FYP II dissertation, May 2013. [Online]. Available: https://utpedia.utp.edu.my/id/eprint/13569/1/Nazrin_13567.pdf

[2] Y. Shorokhova, *Web Application "Personal Diary"*, Bachelor's thesis, University of Finance and Administration, Prague, 2023. [Online]. Available: https://is.vsfs.cz/th/fncam/Bachelor-thesis-personal-diary.pdf

[3] J. Strietelmeier, "Day One Journal app review – You can turn digital journals into books with this app," *The Gadgeteer*, 2024. [Online]. Available: https://the-gadgeteer.com/2024/03/13/day-one-journal-app-review-you-can-turn-digital-journals-into-books-with-this-app/

[4] K. Dawson, "Keeping a Journal with either Day One or Journey," *Medium*, 2020. [Online]. Available: https://kerry-dawson.medium.com/keeping-a-journal-with-either-day-one-or-journey-e050657a153d

[5] R. Varshney, "[App Fridays] Grid Diary journaling app review," *YourStory*, Dec. 25, 2020. [Online]. Available: https://yourstory.com/2020/12/app-fridays-journaling-app-grid-diary

[6] One Beyond, "Agile Software Development Methodology," *One Beyond*. https://one-beyond.com/process/agile-software-development-methodology/

[7] Scrum.org, "What Is Scrum?," *Scrum.org*, 2020. https://www.scrum.org/resources/what-scrum-module

[8] Pawan Harwani, “What is Android Studio? - Scaler Topics,” *Scaler Topics*, Jul. 18, 2023. https://www.scaler.com/topics/android/what-is-android-studio/

[9] A. Cichocka and Miquido, “What is Flutter SDK?,” *Miquido*, May 24, 2025. https://www.miquido.com/flutter-101/what-is-flutter-sdk/ (accessed Sep. 10, 2025).

[10] A. Manjarres Merino, “Firebase: An all-in-one platform for Web and Mobile application development,” *Sngular Insights*, Sep. 18, 2024. [Online]. Available: https://www.sngular.com/insights/313/firebase

[11] M. Heller, “What is Visual Studio Code? Microsoft’s extensible code editor,” *InfoWorld*, Jul. 08, 2022. https://www.infoworld.com/article/2335960/what-is-visual-studio-code-microsofts-extensible-code-editor.html

[12] A. Javed, “Dart Programming Language: Top Features and Applications,” *Xavor Corporation*, Jul. 04, 2023. https://www.xavor.com/blog/dart-programming-language-features-and-applications/

[13] “AssemblyAI Reviews: Use Cases, Pricing & Alternatives,” *futurepedia*, 2023. https://www.futurepedia.io/tool/assemblyai

[14] S. Loria, “TextBlob: Simplified Text Processing — TextBlob 0.15.2 documentation,” *Readthedocs.io*, 2018. https://textblob.readthedocs.io/en/dev/

POSTER

SMART PERSONAL VOICE DIARY APPLICATION WITH MULTIMEDIA INTEGRATION

INTRODUCTION

- Traditional diaries lack convenience and multimedia support
- Many users prefer mobile-based journaling for flexibility
- Existing diary apps provide only basic text storage with limited intelligent features
- Addresses these gaps by providing a voice-powered, multimedia-enhanced diary application

OBJECTIVE

- Develop a mobile diary application with voice-to-text functionality
- Implement AI-based mood detection using DistilBERT model
- Enable multimedia support (image, video, audio)
- Provide mood analytics and visualization dashboard
- Build a community sharing platform for users
- Develop admin dashboard for system monitoring and management

SYSTEM FEATURES

- Diary Entries: Create, edit, delete, and organize entries.
- Voice-to-Text: Accurate transcription with AssemblyAI.
- Calendar View: Navigate entries by date.
- Community Sharing: Share posts, like, comment, and interact with others.
- Media Attachments: Support for photos, audio, and video.
- Location & Weather: Automatic detection for contextual information.
- AI Mood Detection: Analyze diary content to identify emotions automatically.
- Secure Authentication: Firebase Auth for signup and login.

PROBLEM STATEMENT

- Traditional diaries are inconvenient and text-only.
- Existing apps lack voice input and rich multimedia support.
- Users lack tools for search, organization, and emotional analysis.

MOTIVATION

- Mobile devices are ideal for personal journaling.
- Voice input enables faster, hands-free diary creation.
- Multimedia support makes entries more meaningful.
- Sentiment tracking and contextual data provide personal insights.
- A smart diary promotes reflection and mental well-being.

PROPOSED METHOD

Frontend: Flutter (cross-platform mobile development)
Backend: Firebase (Authentication, Firestore, Storage)
AI Model: DistilBERT GoEmotions (TFLite)
APIs Used:

- AssemblyAI → Speech-to-text
- Gemini API → AI insights & analysis
- OpenWeatherMap → Weather data
- Google Translate → multilingual support

CONCLUSION

The Voice Diary application enhances traditional journaling by integrating voice input and AI mood detection. It provides a convenient, intelligent, and user-friendly way for users to record and reflect on their experiences.

UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
Wholly owned by UTAR Education Foundation
(Co. No. 578227-M)
DU012(A)

BACHELOR OF COMPUTER SCIENCE (HONS)
DONE BY: WILSON FOOK WEI SHENG
SUPERVISED BY: DR TAN JOI SHAN