BAHASA MALAYSIA LANGUAGE LEARNING APPLICATION

[MUHAMMAD IQHWAN HAIKAL BIN JAMAL RODZI]

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[BAHASA MALAYSIA LANGUAGE LEARNING APPLICATION]

[MUHAMMAD IQHWAN HAIKAL BIN JAMAL RODZI]

This report is submitted in partial fulfillment of the requirements for the Bachelor of [Computer Science (Software Development)] with Honours.

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

DECLARATION

I hereby declare that this project report entitled

[BAHASA MALAYSIA LANGUAGE LEARNING APPLICATION]

is written by me and is my own effort and that no part has been plagiarized without citations.

STUDENT :  Date : 5/9/2024
([MUHAMMAD IQHWAN HAIKAL BIN JAMAL RODZI])

I hereby declare that I have read this project report and found this project report is sufficient in term of the scope and quality for the award of Bachelor of [Computer Science (Software Development)] with Honours.

SUPERVISOR :  Date : 5/9/2024
([TS DR INTAN ERMAHANI BINTI A. JALIL])

DEDICATION

To my family, especially my parents, whose unwavering support and boundless encouragement have been the cornerstone of my strength throughout the entire journey of completing this project, I find it difficult to put into words the depth of my gratitude.

Their belief in my abilities, even when I faced moments of doubt, provided me with the unwavering determination to persevere. Their sacrifices and tireless efforts in ensuring my success have not gone unnoticed. It is through their guidance, both in words and actions, that I have found the courage to take on challenges and reach for new heights.

With each milestone achieved, I am reminded of the sacrifices they made to see me succeed, and for that, I am eternally grateful. This dedication is a small token of my appreciation for the monumental role they have played in shaping my journey.

To my family, you are the foundation upon which my accomplishments are built. This project is as much a reflection of your love and support as it is of my own endeavors. Thank you for being my pillars of strength.

ACKNOWLEDGEMENTS

I'm grateful to the almighty for the blessing and grace that allows me to complete this project successfully. I would like to express my utmost gratitude to my supervisor Ts Dr Intan Ermahani binti A. Jalil for her continuous guidance and insights throughout the endeavor to complete this project and also for her constructive feedbacks which helps me improvise my solution. I would also like to thank my friends and family for their continuous support and encouragement in pushing me toward the completion of this project. Once again, I'm glad to be able to execute this project and successfully completed it fulfilling all the objective outlined.

ABSTRACT

This project develops a comprehensive Bahasa Malaysia language learning application, leveraging Firebase Firestore to enhance user learning experience through structured lessons and interactive minigames. The focus is on facilitating language acquisition for learners at various proficiency levels, from beginners to advanced. Central to the project is a well-organized database architecture that categorizes content into lessons and minigames with escalating complexity, tailored to individual learning needs. Each user interaction, such as completed lessons or minigames, is meticulously tracked and stored in user-specific documents within the Users collection, fostering a personalized learning environment. The development process involved meticulous planning, starting from gathering user requirements to designing logical and physical data models, followed by the implementation of these models in Firestore's NoSQL framework. This approach ensures scalability and real-time data synchronization, crucial for maintaining user engagement and progress tracking. Preliminary testing results indicate that the application not only supports efficient data management but also significantly enhances user engagement through dynamic content delivery. The project demonstrates the potential of advanced database systems and interactive applications in significantly improving language learning outcomes.

ABSTRAK

Projek ini membangunkan aplikasi pembelajaran bahasa Malaysia yang komprehensif, memanfaatkan Firebase Firestore untuk meningkatkan pengalaman pembelajaran pengguna melalui pelajaran berstruktur dan permainan mini interaktif. Tumpuan adalah untuk memudahkan pemerolehan bahasa untuk pelajar di pelbagai peringkat kecekapan, daripada pemula hingga mahir. Inti kepada projek ini ialah seni bina pangkalan data yang teratur yang mengkategorikan kandungan ke dalam pelajaran dan permainan mini dengan kerumitan yang semakin meningkat, disesuaikan dengan keperluan pembelajaran individu. Setiap interaksi pengguna, seperti pelajaran atau permainan mini yang telah selesai, dijejaki dan disimpan dengan teliti dalam dokumen khusus pengguna dalam koleksi Pengguna, memupuk persekitaran pembelajaran yang diperibadikan. Proses pembangunan melibatkan perancangan yang teliti, bermula daripada mengumpul keperluan pengguna kepada mereka bentuk model data logik dan fizikal, diikuti dengan pelaksanaan model ini dalam rangka kerja NoSQL Firestore. Pendekatan ini memastikan kebolehskalaan dan penyegerakan data masa nyata, penting untuk mengekalkan penglibatan pengguna dan penjejakan kemajuan. Keputusan ujian awal menunjukkan bahawa aplikasi itu bukan sahaja menyokong pengurusan data yang cekap tetapi juga meningkatkan penglibatan pengguna dengan ketara melalui penyampaian kandungan dinamik. Projek ini menunjukkan potensi sistem pangkalan data termaju dan aplikasi interaktif dalam meningkatkan hasil pembelajaran bahasa dengan ketara.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

PAGE

LIST OF FIGURES

PAGE

LIST OF ABBREVIATIONS

FYP	**-**	**Final Year Project**
LDM	**-**	**Logical Data Model**

CHAPTER 1: INTRODUCTION

1.1 Introduction

Southeast Asia has hundreds of languages that reflect the history, cultures, and identities of their speakers, making for a rich and diverse linguistic landscape. Of these languages, Bahasa Malaysia is the official language of Malaysia, a nation renowned for its dynamic economy and lively culture. Interest in Malaysian language has increased in tandem with the country's continued efforts to establish its economic and cultural dominance on the international scene. Acquiring proficiency in Bahasa Malaysia facilitates not only career and personal development but also provides a means of comprehending Malaysian culture and customs.

However, there are several obstacles in the way of learning Bahasa Malaysia. Prospective language learners frequently find that standard language learning materials are either excessively rigorous or out of step with their learning styles, especially those who manage hectic academic or professional schedules. The flexibility needed by today's fast-paced lifestyles is not provided by classroom environments or textbook-based learning methods. Furthermore, the majority of digital language learning resources now available are designed for languages that are more commonly spoken, leaving a clear lack of creative and user-friendly materials for languages like Bahasa Malaysia.

When it comes to cultural depth and language authenticity, this disparity is especially noticeable. The true pronunciation, colloquialisms, and cultural context that are essential for genuinely mastering the language and using it in daily conversation are often absent from the many programs that are available, giving students only a

cursory understanding of the language. This shortcoming not only makes it difficult for students to speak Bahasa Malaysia well, but it also restricts their comprehension of the cultural nuances that are essential to good communication.

A flexible, all-inclusive, and culturally sensitive language learning solution is obviously needed considering these difficulties. To address this requirement, the Bahasa Malaysia Learning Application makes use of contemporary technology to provide a dynamic, efficient, and easily accessible learning environment that honors the variety of learner needs and preferences and offers profound cultural understanding and useful language proficiency.

1.2 Problem Statement

The following problems have been identified to be the cause of the difficulty of learning Bahasa Malaysia.

- Time Constraints and Accessibility

Professionals and university students frequently find it difficult to efficiently manage their time because of their hectic schedules, which involve balancing work, personal obligations, and academic obligations. With their set timetables and location-based instruction, traditional Bahasa Malaysia language courses fall short of providing the flexibility these prospective students need. The issue is made worse by the dearth of easily available and flexible learning materials, which discourages ongoing learning and long-term involvement.

- Diverse Learning Needs and Preferences

The different learning methods and preferences of a diverse student body are not sufficiently accommodated by the Bahasa Malaysia learning resources available today. A more tailored learning experience is needed for many students, one that takes into account their unique learning preferences, learning style, and particular difficulties—like varying linguistic backgrounds or lack of previous language exposure—as well as their unique learning pace.

- Integration of Real-Life Usage and Cultural Nuances

Most available learning platforms do not effectively incorporate real-life language usage, cultural context, or colloquialisms, which are essential for learners to fully understand and use Bahasa Malaysia in everyday interactions. This gap leads to a superficial understanding of the language, which is often not practical for communication in real-world settings, particularly in a culturally rich country like Malaysia.

- Technological Limitations

Existing digital language learning solutions often lack the use of advanced technological features that could enhance the learning experience, such as artificial intelligence (AI) to personalize the learning journey, or gamification to increase engagement and motivation. This underutilization of technology limits the effectiveness of these platforms in delivering an immersive and responsive learning environment.

1.3 Objective

The system is designed to overcome the aforementioned problems by achieving the following objectives.

- To offer flexible learning schedules that accommodate varied and unpredictable routines.
- To provide users the freedom to customize their learning experience according to their needs and goals.
- To integrate cultural and contextual learning to enhance real-world language use.
- To leverage AI technology to create engaging, interactive, and effective learning experiences.

1.4 Scope

The scope of the Bahasa Malaysia Learning Application is specifically designed to cater to two main groups of users, ensuring that the features and content are tailored to meet their educational needs and capabilities effectively. The details of the scope are as follows:

- University Students

University students are selected as primary users due to their typically higher attention spans and ability to engage with text-based learning materials effectively. This group is likely to benefit from the structured yet flexible learning modules offered by the application, which are designed to complement their academic competencies and schedules. The application's content, requiring reading and comprehension of text-based instructions and lessons, fits well with university students who are accustomed to consuming large amounts of information and conducting self-directed learning.

- Individuals Aged 17 and Above with a Proper Grasp of English

This demographic includes not only university students but also professionals and other adults interested in learning Bahasa Malaysia. Targeting this age group ensures that the users have developed cognitive skills necessary for language learning, such as the ability to follow complex instructions and engage in higher-level thinking. The use of English as the medium of instruction requires users to have a solid foundation in the language to facilitate effective learning. This scope allows the application to reach a broader audience who, while not necessarily in a university setting, are motivated to learn Bahasa Malaysia for personal, professional, or educational reasons.

1.5 Project Significance

The Bahasa Malaysia Learning Application is designed to significantly impact various stakeholders by providing an innovative and accessible approach to learning Bahasa Malaysia. The significance of the project can be understood through its direct benefits to users and broader societal impacts:

- Benefits to University Students

By providing university students with the tools to learn Bahasa Malaysia effectively, the application aids in their academic and professional development. Mastery of an additional language like Bahasa Malaysia can open up new academic research opportunities and increase employability in Malaysia and the ASEAN region. The flexible learning model is particularly suited to their busy schedules, enabling them to acquire new skills without compromising their academic responsibilities.

- Benefits to Adults and Professionals Aged 17 and Above

For adults and professionals, learning Bahasa Malaysia can facilitate better integration into the Malaysian community, enhancing social interactions and cultural understanding. This demographic will benefit from the application's ability to provide learning tailored to adult learners who might require a focus on specific language aspects, such as business terminology or everyday communication.

- Enhancement of Multilingual Communication

By equipping a broader audience with the skills to speak and understand Bahasa Malaysia, the project contributes to societal cohesion and mutual understanding among Malaysia's diverse population. It promotes multilingualism as a value in the community, encouraging more inclusive communication and interaction across different linguistic backgrounds.

- Support for Non-Traditional Learners

The application supports individuals who may not have access to traditional language learning resources due to geographic or economic constraints. By making Bahasa Malaysia learning more accessible through digital means, the project helps to spread language education, ensuring more people have the opportunity to learn regardless of their location or socioeconomic status.

- Contribution to Language Preservation and Promotion

As more individuals learn Bahasa Malaysia, the application plays a role in the preservation and promotion of the Malay language and culture. It serves as a resource for not only Malaysians but also international users interested in the country's culture, supporting efforts to maintain and spread the use of Bahasa Malaysia globally.

1.6 Expected Output

At the end of the project, the expected output of the Bahasa Malaysia Learning Application project is a comprehensive, user-friendly digital platform that facilitates effective language learning for university students and adults proficient in English. The platform will feature interactive, customizable learning modules tailored to the needs of diverse learners, incorporating AI to simulate the action of interacting with another person. Additional outputs include an assessment system to track learning progress, an assortment of mini-games to ensure that the users are motivated in putting their knowledge to the test. Collectively, these outputs are designed to enhance users' fluency in Bahasa Malaysia, broaden their cultural understanding, and improve their communication skills within both academic and professional contexts.

1.6.1 Lessons Module

This is the core module for teaching the users Bahasa Malaysia. This module will include lessons on vocabulary, phrases, grammar, sentence structure, and pronunciation practice of the words.

- Vocab lessons

Introduce essential words and phrases commonly used in daily communication. Cover topics like greetings, numbers, colours, family members, food, and more.

- Phrases and expressions

Teach practical phrases for various situations (ordering food, asking for directions, expressing gratitude). Explore idiomatic expressions unique to Bahasa Malaysia. Provide context to understand when and how to use specific phrases.

- Grammar fundamentals

Focus on Bahasa Malaysia grammar rules, including verb conjugation, tenses, articles, and sentence structure. Explain common sentence patterns and word order. Address any differences between Bahasa Malaysia and other languages learners may be familiar with.

- Sentence construction

Break down sentences into subject, predicate, and object components. Teach how to form affirmative, negative, and interrogative sentences. Highlight common sentence connectors and conjunctions.

- Pronunciation practice

Emphasize correct pronunciation of Bahasa Malaysia sounds. Cover vowels, sounds and consonants.

1.6.2 AI Chatbot Module

This is the AI chatting module for users to participate in conversational practice. The chatbot will be based on OpenAI ChatGPT's API.

Features:

- Users can engage in dialogues, ask questions, and receive responses in Bahasa Malaysia.
- The chatbot leverages OpenAI ChatGPT's API to simulate real-world conversations.
- Conversations cover various topics, reinforcing language skills.

1.6.3 Basic User Access Module

This module will grant the user basic application access like registration and login.

- Registration

Allow users to create accounts by providing necessary details.

- Login

Authenticate users and grant access to the learning modules.

- User Profiles

Store user preferences, progress, and settings.

1.6.4 Gamification Module

This module will be the gamification aspects where the user will be able to track their progress, play mini games, take quizzes, and see their levels based on lessons completed.

- Progress tracking

Users can monitor their learning journey, track completed lessons, and set goals.

- Mini games

Incorporate language-related games (word puzzles, quizzes) to reinforce learning.

- Quizzes

Assess comprehension and retention of vocabulary, grammar, and cultural knowledge.

- Level system

Assign levels based on achievements, encouraging continuous engagement.

1.6.5 Cultural Insights Mini-Module

This mini module will give cultural insights on the country through lessons done. Examples of cultural insights include, social hierarchy and titles, food culture, taboos and respect, customs during festivals, multilingual environment, greetings, and politeness aspects of the language.

- Social hierarchy and titles

Explore how honorifics and titles are used to address individuals based on age, profession, and social status.

- Food culture

Discuss traditional Malaysian dishes, dining etiquette, and the significance of food in social gatherings.

- Taboos and respect

Highlight cultural norms related to respect, gestures, and behavior.

- Customs during festivals

Learn about major festivals (Hari Raya, Chinese New Year) and associated customs.

- Multilingual environment

Understand the influence of other languages (English, Chinese, Tamil) on Bahasa Malaysia.

- Greetings and politeness

Master polite forms of greeting and interaction.

1.7 Conclusion

In conclusion, this chapter has laid a comprehensive foundation for the development of the Bahasa Malaysia Learning Application, addressing the pressing needs for a flexible, personalized, and culturally rich language learning tool. The introduction highlighted the linguistic diversity and significance of Bahasa Malaysia in the Southeast Asian region, and the necessity for innovative educational solutions due to the limitations of traditional learning methods. The problem statements elaborated on the specific challenges faced by university students and adult learners, such as time constraints, diverse learning needs, and the integration of real-life language usage. Objectives were set to resolve these issues through the application's flexible learning schedules, user-driven learning freedom, cultural content integration, and advanced AI technology, aiming to enhance the language learning experience significantly.

CHAPTER 2: LITERATURE REVIEW AND PROJECT METHODOLOGY

2.1 Introduction

This chapter provides a comprehensive literature review and outlines the methodology to be employed in the development of the Bahasa Malaysia Learning Application. The purpose of this chapter is twofold: firstly, to establish a theoretical framework and contextual backdrop through a review of existing literature related to language learning technologies, educational psychology, and the specific challenges and solutions associated with learning Bahasa Malaysia. Secondly, the chapter details the systematic approach used in the design, development, testing, and implementation of the learning application, ensuring that the project is grounded in robust research and best practices in software development and educational theory.

2.2 Facts and findings

The extensive literature review conducted for the Bahasa Malaysia Learning Application has provided critical insights into effective language learning strategies, the role of technology, and the specific challenges and benefits of learning Bahasa Malaysia. Below are the findings into each key area, providing detailed findings and their implications for the project's design and strategy:

- Effective Language Learning Strategies

According to Norfaizal et al. (2022), successful language learners often employ a mix of direct and indirect strategies, with high-achievers tending to use a balanced approach that includes both social and cognitive strategies. This finding supports the

application's aim to provide a range of interactive and cognitive activities that cater to different learning styles and levels of proficiency.

Lambri and Mahamod (2015) emphasize the effectiveness of student-centered learning in higher education settings. Their study suggests that such approaches, which encourage active student participation and engagement, can lead to greater enjoyment and better absorption of material, particularly in language courses. This underscores the necessity for the application to include interactive, user-driven content that promotes active learning.

- Challenges in Learning Bahasa Malaysia

Zamri et al. (2014) discuss the difficulties faced by foreign students in mastering Bahasa Malaysia, particularly the challenge of understanding colloquial and formal registers. Their findings indicate a need for language learning tools that offer comprehensive exposure to both formal language and everyday colloquial use, facilitating a more holistic understanding of the language.

The need for contextual learning is further supported by the research of Mahamod et al. (2014), which points out that successful language acquisition often depends on the learner's ability to understand and use the language in real-world situations. This highlights the application's need to integrate real-life scenarios and practical language use cases.

- Student-Centered Learning Approaches

Alizah and Zamri (2015) found that student-centered methodologies not only enhance engagement but also respect and value the learner's input, leading to a more respectful and inclusive learning environment. This aligns with the application's goal to foster an interactive and learner-focused platform where users feel valued and actively involved in their educational journey.

- Integration of Real-Life Usage and Cultural Nuances

Norfaizal et al. (2022) also highlight the importance of integrating cultural nuances and real-life language usage into learning materials. This approach is vital for learners to not only understand the language but to use it appropriately in different cultural contexts, reinforcing the application's focus on comprehensive cultural content.

Jenis Motivasi	Item	Kenyataan	Kekerapan dan peratusan (%)	
			Sangat setuju	Tidak setuju
Intrinsik	B1	Saya belajar bahasa Melayu kerana saya benar-benar meminati bahasa Melayu.	33 (75)	11 (25)
	B2	Saya merasakan bahawa bahasa Melayu merupakan bahasa yang menarik untuk dipelajari.	42 (95)	2 (5)
	B3	Mempelajari bahasa Melayu merupakan satu cabaran yang saya suka untuk tempuhi.	31 (71)	13 (29)
	B4	Belajar bahasa Melayu membuatkan saya merasa puas.	30 (68)	14 (32)
	B5	Boleh bercakap dalam bahasa Melayu membuatkan saya merasa bangga.	18 (41)	26 (59)
Ekstrinsik	B6	Saya mempelajari bahasa Melayu supaya dapat menunjukkan kebolehan saya kepada orang lain.	19 (43)	25 (57)
	B7	Saya belajar bahasa Melayu untuk memenuhi permintaan ibu bapa saya.	3 (7)	41 (93)
	B8	Saya belajar bahasa Melayu semata-mata untuk mendapatkan ganjaran daripada ibu bapa/keluarga saya.	5 (11)	39 (89)
	B9	Belajar bahasa Melayu dapat menjadikan saya seorang yang berpengetahuan.	35 (79)	9 (21)
	B10	Pensyarah/rakan-rakan saya menggalakkan saya untuk mempelajari bahasa Melayu.	16 (36)	28 (64)

Figure 2-1 Research on foreign students' motivation (Zuhidayah et al., 2016).

Based on the figure above, there is interest from foreign students towards learning Bahasa Malaysia.

These detailed findings from the literature review form a solid foundation for the development of the Bahasa Malaysia Learning Application, ensuring that the project is backed by empirical research and addresses the real-world needs of language learners.

2.2.1 Domain

The domain related to the Bahasa Malaysia Learning Application encompasses several interrelated fields that collectively inform the project's design, development, and expected outcomes. These fields are educational technology, language acquisition, and cultural studies, each playing a crucial role in shaping the application.

- Educational Technology

The application of educational technology in language learning provides the foundational framework for this project. The use of advanced technologies, including artificial intelligence (AI) and interactive multimedia, is crucial in creating an adaptive learning environment. AI technologies enable personalized learning experiences by adjusting the difficulty of tasks based on individual performance and offering tailored recommendations for improvement. This domain also covers the integration of gamification elements, which are employed to enhance engagement and motivation through rewards systems, progress tracking, and interactive challenges.

- Language Acquisition

The process of language acquisition is central to the design of the learning application. This domain involves understanding how adults acquire a second language, including the cognitive processes involved and the best pedagogical practices for facilitating this acquisition. Research in this area emphasizes the importance of immersive and contextual learning environments that mimic natural language use, thereby aiding in the practical application of language skills. The application aims to incorporate these insights by providing scenario-based learning modules that contextualize vocabulary and grammar in everyday situations, making the learning process more relevant and effective.

- Cultural Studies

Integrating cultural studies into language learning is essential for providing learners with a holistic understanding of Bahasa Malaysia. Language is deeply intertwined with culture; hence, the application includes cultural content that helps learners understand idiomatic expressions, social norms, and cultural nuances that are

inseparable from language use. This aspect of the domain ensures that learners are not only linguistically competent but also culturally aware, enabling them to use Bahasa Malaysia appropriately in various social contexts.

These domains are interconnected, each contributing to the comprehensive nature of the Bahasa Malaysia Learning Application. By integrating educational technology with insights from language acquisition and cultural studies, the project aims to create a robust platform that supports effective learning and deepens learners' cultural and linguistic understanding. This multidisciplinary approach not only enhances the educational value of the application but also ensures it meets the diverse needs of its users.

2.2.2 Existing System

This section reviews existing systems within the domains of educational technology and language learning, specifically focusing on those that facilitate the acquisition of Bahasa Malaysia. It discusses the strengths and limitations of current systems and justifies the approach of the proposed project based on these findings.

2.2.2.1 Ling

Ling is a language learning app designed to teach various languages including Malay through interactive lessons and games. It combines vocabulary, grammar, writing, and speaking exercises into a comprehensive learning path. The findings have been compiled into the table below.

Table 2-1 Ling App Findings

Features	Utilizes games and interactive challenges to teach different language aspects. Allows learners to practice conversation with an AI. Provides in-depth grammar lessons as part of its courses.
User Experience	The gamification approach keeps learning fun and engaging. Sleek, modern interface that is easy to navigate. Available on both iOS and Android platforms.

Pros	Wide range of languages offered. Comprehensive learning path covering all language skills. Engaging interactive tools to enhance learning.
Cons	Some advanced features require a subscription. May not have as deep a focus on cultural nuances as some learners might wish.

2.2.2.2 Simply Learn Malay

Simply Learn Malay is more of a travel phrasebook app that provides essential phrases and vocabulary for travelers. It is structured around practical language use, with audio recordings to aid pronunciation. The findings have been compiled into the table below.

Table 2-2 Simply Learn Malay App Findings

Features	Organized by useful categories such as eating out, shopping, and emergencies. Each phrase includes a native speaker audio example. Allows users to test their knowledge.
User Experience	Extremely easy to use, designed for quick access. Basic, straightforward without unnecessary frills. Ideal for travelers looking for quick language references.
Pros	Great for tourists and casual learners. Audio aids help with correct pronunciation. Free version covers extensive phrases.
Cons	Not suitable for in-depth language learning. Limited in scope, primarily vocabulary and phrases.

2.2.2.3 Learn Malay Vocabulary with Vo

This app focuses exclusively on building vocabulary through a flashcard system that employs spaced repetition, a proven method for memory retention.

Table 2-3 Learn Malay Vocabulary with Vo App Findings

Features	Enhances long-term retention of vocabulary with the use of spaced repetition. Users can customize and expand their decks with the use of flashcards. Provides pronunciations for each word.
User Experience	Simple functionality centered around flashcards. Minimalistic design focusing on usability. Very effective for memorizing words and short phrases.
Pros	Excellent for expanding vocabulary efficiently. Customizable learning experience. Useful for all learning levels.
Cons	Does not cover grammar or conversational skills. Limited engagement features.

2.2.2.4 L-Lingo

L-Lingo offers a structured language course that includes vocabulary, grammar, and practical exercises, using visual learning aids to facilitate understanding.

Table 2-4 L-Lingo App Findings

Features	Covers all essential language skills. Uses images to connect words with visuals. Reinforces learning through periodic quizzes.
User Experience	Structured like a traditional course, which can be very effective for systematic learners. Clean, but somewhat dated design. Some lessons include cultural information.
Pros	Good for learners who prefer structured lessons. Visual aids help with memory retention. Offers a free trial for initial lessons.

Cons	Some users may find the content less engaging compared to gamified apps. The interface could be dated by modern standards.

Each of these apps serves a different learner's need and preference, from casual, travel-oriented language acquisition to more structured, in-depth study. The proposed Bahasa Malaysia Learning Application aims to bridge the gaps found in these apps by offering a comprehensive curriculum that not only covers language skills comprehensively but also deeply integrates cultural content, making it suitable for a wide range of learners.

2.2.2.5 Comparison with proposed system

From the findings of the previous applications, the information gathered has been gathered and compared with the proposed system in the table below.

Table 2-5 Comparison between previous applications and proposed system

Feature	**Ling**	**Simply Learn Malay**	**Learn Malay Vocabulary with Vo**	**L-Lingo**	**Proposed System**
Learning Scope	Vocab, grammar, conversation	Basic phrases	Vocabulary focus	vocab, grammar, conversation	Comprehensive including cultural nuances
Methodology	Gamification, immersive	Practical, quick learning	Spaced repetition, flashcards	Visual learning, structured course	Gamification, immersive
Technology	AI chatbot, interactive tasks	Basic, audio recordings	Basic, audio for pronunciation	Interactive quizzes, virtual class	Gamification, AI chatbot, Interactive quizzes

User Experience	User-friendly, engaging	Simple, intuitive	Simple, straightforward	User-friendly, course-like	Simple, Course-like, Straightforward
Cultural Content	Limited	Limited	None	Some cultural notes	Focusing on cultural content
Unique Offerings	Chatbot for conversation practice	Organized by practical categories	Focused vocabulary building	Emphasis on visual aids	Cultural insights and real-life scenarios
Target Audience	General learners	Travelers, beginners	Vocabulary learners	General learners	Students, professionals, cultural enthusiasts

2.2.3 Technique

In developing the Bahasa Malaysia Learning Application, various educational techniques and approaches were considered to determine the most effective way to teach Bahasa Malaysia to a diverse audience. This section discusses alternative approaches and justifies the reasons for not adopting them in favor of the chosen methodology.

- Traditional classroom-based learning

Traditional classroom learning involves face-to-face instruction, typically conducted in a structured environment with a teacher leading the course. While traditional classroom settings provide structured learning and direct interaction with instructors, they lack the flexibility required by many of today's learners who have varying schedules and prefer learning at their own pace. Additionally, this method does not easily allow for the personalization of content and pacing that AI-driven applications can offer.

- Pure e-learning with no interactive components

This approach involves using static digital content such as PDFs or video lectures without any interactive or adaptive learning components. While straightforward and often cost-effective, pure e-learning can lead to lower engagement and retention rates because it does not adapt to individual learner needs or provide interactive elements that enhance comprehension and retention. The lack of interactivity and adaptation makes it less suitable for language learning where engagement and active use of language skills are crucial.

- Immersion technique without guided lessons

Language immersion techniques involve placing learners in environments where they use only the target language, without structured lessons or explanations in their native language. While immersion is highly effective for rapid language acquisition, it can be overwhelming for beginners without foundational knowledge of the language. It also requires a context where the language is being actively used, which is not feasible for all learners, especially those not residing in Malaysia.

- Solely gamified learning without structured content

This approach relies entirely on gamification elements to teach the language, using games and challenges without structured lessons or traditional content delivery. Although gamification increases engagement, relying solely on this technique without structured grammatical, vocabulary, and conversational lessons can lead to a superficial understanding of the language. It is important to balance fun elements with rigorous educational content to achieve comprehensive language proficiency.

The chosen approach for the Bahasa Malaysia Learning Application focuses interactive content and cultural integration, providing a balanced and effective learning experience that adapts to individual needs and fosters deep understanding and usage of the language. This method combines carefully picked aspects of technology and method of teaching, ensuring that the learning process is both engaging and educationally robust, addressing the limitations found in the alternative approaches discussed above.

2.3 Project Methodology

The AGILE methodology will be employed in the creation of the Bahasa Malaysia Learning Application because of its simplicity, flexibility, and speed. Agile methodology generally permits working on known needs immediately without having to wait for them to be completed. The next sprint will contain any new requirements or modifications based on feedback.

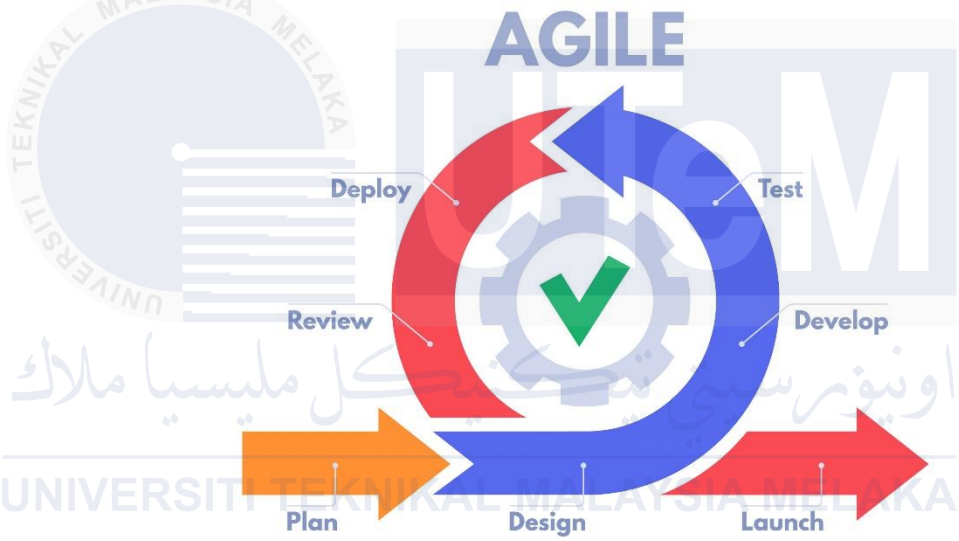

Figure 2-2 The phases of Agile Methodology.

As depicted in Figure 2.2, agile methodology consists of six main phases before launch which completes the project. The six phases, from planning to deployment and review are part of a sprint which are repeated as much as needed until the software being developed are ready to be launched upon its completion which is determined during review phase of each sprint.

The primary advantage of Agile methodology is its flexibility and adaptability. Unlike traditional methodology such as waterfall which bottlenecks development process since the development can't move to the next phase as long as current phase

have not been completed yet, the iterative nature of Agile methodology allows developers to quickly work on existing requirements until deployment and review it again to check if there's anything that have to be improved to determine if more sprint is necessary. It is very suitable for adaptability in the development process of the Bahasa Malaysia Learning Application which might have more requirements that might be discovered later on.

2.3.1 Sprint 1: Developing the UI and Basic Functions

- **Plan**

  The initial step is to outline the core functionalities required for the application's base, such as user registration, login, and main navigation. Realistic milestones and deadlines for these elements, focusing on creating a functional prototype will be set.

- **Design**

  This process will focus on user experience, ensuring the interface is intuitive and user-friendly. Given the solo nature of the project, design revisions will be pragmatic and streamlined, focusing on essential usability.

- **Develop**

  The designs will be translated into code using the dart language and Flutter framework. This phase will involve setting up the basic infrastructure and implementing the main functionalities outlined in the planning phase.

- **Test**

  Testing will be conducted iteratively throughout the development process. Unit testing and interface testing will be performed to ensure that each component functions as intended. Feedback from peers and the supervisor will be crucial at this stage, as it will help identify usability issues and bugs that might be overlooked.

- **Deploy**

The deployment will involve compiling the application into mobile and released in apk form for easy installation. This apk will be distributed to people that will be involved in giving feedback.

- **Review**

  After deployment, feedback will be gathered and analyzed from the supervisor and peers. This feedback will be crucial in pinpointing areas for improvement and identifying additional features that could enhance the application. Based on this input, the project plan for the next sprint will be refined.

2.3.2 Sprint 2: Lesson Module Framework

- **Plan**

  This sprint focuses on planning the structure and content of the lesson modules. The types of lessons and the educational content to be included will be outlined. This phase will necessitate close coordination with educational references, which could include the supervisor or academic peers who can offer content expertise.

- **Design**

  The lesson module templates will be designed to ensure flexibility and the ability to accommodate various types of educational content. The design will need to efficiently support text, images, and interactive elements.

- **Develop**

  During the development phase, the backend and frontend components of the lesson modules will be built. This will involve programming the logic for dynamic content loading and ensuring that the modules are scalable.

- **Test**

  Comprehensive testing of the lesson modules will be conducted, focusing on their performance under various user loads and their capability to handle interactive content effectively. Peer reviews and feedback from the supervisor will be crucial in validating the educational effectiveness of these modules.

- **Deploy**

  The lesson modules will be integrated into the existing framework of the application, followed by their deployment.

- **Review**

  Feedback on the lesson modules will be collected and analyzed to identify any issues with content delivery or user interaction. This feedback will guide further refinements in subsequent sprints.

2.3.3 Sprint 3: Mini-Game Module Framework

- **Plan**

  Various mini-games that reinforce the language skills taught in the lesson modules will be conceptualized and planned. Each game will be designed to be both educational and engaging, targeting specific learning outcomes.

- **Design**

  The design phase will involve creating interactive and visually appealing game designs. Game development frameworks suitable for educational content will be used to ensure that the games are responsive and provide immediate educational value.

- **Develop**

  During development, the game mechanics will be implemented and integrated into the lesson modules. Attention to detail in coding will be essential to ensure smooth gameplay and seamless integration.

- **Test**

  Testing will involve evaluating the games for user engagement and educational impact. Feedback from peers and the supervisor will help refine the games and ensure they meet the learning objectives.

- **Deploy**

The mini-games will be deployed within the application, ensuring they are accessible to users and function as intended.

- **Review**

  Post-deployment, feedback and usage data will be collected to assess the effectiveness of the mini-games. This data will inform potential improvements and adjustments.

2.3.4 Further Sprints: Continuous Improvement

- **Plan**

  Each subsequent sprint will focus on refining existing features and expanding the application's capabilities based on user feedback and evolving educational needs.

- **Design**

  Design updates will be based on feedback and the need for new features. This iterative process allows for ongoing enhancement of the application's functionality and user experience.

- **Develop**

  Continued development will incorporate new features and refine existing ones, ensuring the application remains robust and scalable.

- **Test**

  Ongoing testing will ensure new features integrate seamlessly with the existing system and maintain high performance and reliability.

- **Deploy**

  Regular updates will be deployed to users, incorporating new features and improvements.

- **Review**

Continuous feedback from users, peers, and the supervisor will drive the iterative development process, ensuring the application meets the evolving needs of its users.

2.4 Project Requirements

Since this language learning application is entirely based on mobile, the requirements will be focused entirely on software, while the hardware requirements will only detail the usage of accessing the application.

2.4.1 Software Requirement

This section outlines the software requirements for the Bahasa Malaysia Learning Application. The software stack chosen is critical to support the functional requirements and ensure scalability and stability of the application. Below are the key software components used in the development of the application:

- **Android Studio**

  Android Studio serves as the primary Integrated Development Environment (IDE) for the project. It is used for coding, debugging, and testing the application. Android Studio provides a comprehensive set of tools that are tailored for developing Android applications, including an emulator to simulate various Android devices, which is crucial for testing the application across different screen sizes and device capabilities.

- **Flutter Framework**

  Flutter is chosen as the core framework for building the user interface and handling the front-end logic of the application. Flutter allows for the development of natively compiled applications from a single codebase for both Android and iOS platforms, ensuring consistency and efficiency in development. The framework's rich set of customizable widgets enables a reactive and attractive user interface, which enhances the user experience. Flutter's hot reload feature significantly speeds up the development cycle by allowing instant updates during the code development process.

- **Firebase**

Firebase provides the backend services for the application, including authentication, data storage, and real-time database interactions. It supports user authentication through various methods such as email, Google, and Facebook, which are integral for the user registration and login processes. Firebase's real-time database is used to store and sync user data across all devices in real-time, enhancing the dynamic learning environment. The platform's scalability and ease of integration with Flutter make it an ideal choice for managing the application's data needs.

- **OpenAI Chatbot API**

  The integration of OpenAI's Chatbot API is a pivotal aspect of the application, enabling interactive language learning experiences through conversational AI. This API allows the application to include a sophisticated chatbot that can engage users in Bahasa Malaysia, facilitating real-time language practice and learning. The API's advanced natural language processing capabilities allow the chatbot to understand and respond to user inputs effectively, making it a valuable tool for reinforcing language skills interactively.

2.4.2 Hardware Requirement

The success of the Bahasa Malaysia Learning Application hinges significantly on having the appropriate hardware setup to facilitate its development, testing, and deployment. Below is a detailed breakdown of the hardware requirements, focusing on the purposes of each component:

- **Laptop or PC**

  The laptop or PC is the primary tool for the entire development process of the application. It hosts the development environment where all coding, initial testing, and application builds occur. This device runs software like Android Studio and the Flutter framework, which are essential for writing and compiling the application's code. The laptop or PC also supports the various tools and platforms needed for version control and collaboration, such as Git, ensuring that any incremental changes to the code are systematically tracked and managed.

- **Mobile Phone**

  Mobile phones are crucial for the real-world testing of the application. They allow the developer to understand how the application behaves in the environment it is designed for, which is vital for assessing user interface design, application flow, and overall user experience. Mobile phones also play a critical role in the deployment phase, where the application is installed and run on actual devices to ensure that all features work as intended across different screen sizes and operating systems. This step is crucial for identifying and fixing device-specific bugs and for verifying the application's performance under typical usage conditions.

- **Wi-Fi and Internet Access**

  Internet connectivity is essential throughout the development and deployment stages of the application. During development, an internet connection is necessary to access online resources, download software updates, interact with cloud-based services like Firebase, and utilize APIs such as the OpenAI Chatbot API. The connection must be stable and reliable to facilitate continuous integration and deployment practices that rely on real-time data exchange. For users, internet access ensures that the application can connect to backend services for tasks such as user authentication, data retrieval, and synchronization. This is particularly important for applications that rely on cloud services to store user data and provide dynamic content updates.

2.5 Project Schedule and Milestones

Starting from March 25, 2024, this project is expected to finish in September 2024. In the six months of expected project duration which is equivalent to 27 weeks, the development of the language learning application will be carried out adhering to the AGILE methodology. The detailed schedule will be described in the following Gantt Charts in Figure 2.3, Figure 2.4 and Figure 2.5. The Gantt Charts will consist of the phases of development and expected duration of execution segregated into weeks.

Week	1	2	3	4	5	6	7	8	9	10
Briefing										
Plan										

Design										
Develop										
Test										
Deploy										
Review										

Figure 2-3 Gantt Chart for First Sprint

Week	11	12	13	14	15	16	17	18	19	20
Plan										
Design										
Develop										
Test										
Deploy										
Review										

Figure 2-4 Gantt Chart for Second Sprint

Week	21	22	23	24	25	26	27
Plan							
Design							
Develop							
Test							
Deploy							
Review							
Launch							

Figure 2-5 Gantt Chart for Third Sprint

2.6 Conclusion

This section of the report has outlined the necessary software and hardware requirements crucial for the successful development and deployment of the Bahasa Malaysia Learning Application. By choosing a robust set of development tools and ensuring the availability of appropriate hardware, this project is well-equipped to meet its design and functionality goals.

The selection of Android Studio and the Flutter framework provides a strong foundation for developing a high-quality, cross-platform application. Firebase as a backend solution enhances the application's ability to handle data securely and efficiently, while the OpenAI Chatbot API introduces advanced AI-driven interactions that are pivotal for the language learning features of the application. These software choices align with modern development practices and support the application's requirements for scalability, reliability, and user engagement.

Furthermore, the specification of appropriate hardware, including a high-performance laptop or PC for development, modern mobile devices for deployment, and reliable internet access, ensures that the application can be developed, tested, and used effectively under realistic conditions. This hardware setup not only facilitates a smooth development process but also enables thorough testing and seamless user experiences.

In conclusion, the careful planning of software and hardware specifications is critical in setting a strong groundwork for the development phases outlined in the methodology. This preparation ensures that the application not only meets the technical and functional standards but also provides a stable and engaging platform for users to learn Bahasa Malaysia efficiently.

CHAPTER 3: ANALYSIS

3.1 Introduction

The analysis phase of the Bahasa Malaysia Learning Application development is a critical stage that sets the foundation for all subsequent design and development activities. This phase focuses on a detailed examination of the application's requirements and specifications. It involves defining and documenting the functional and non-functional requirements, understanding user interactions, and delineating the data flows essential to the application.

The goal of this phase is to outline precisely what the application will do and how it will interact with its users. This includes identifying the key features that need to be developed, understanding the user demographics, and considering the technological environment in which the application will operate.

Requirements gathering during this phase will employ various techniques such as interviews, questionnaires, observation of potential users, and consultation with language learning experts. These activities will capture a broad spectrum of user needs and expectations, which are crucial for forming a clear and comprehensive requirements document.

Furthermore, the analysis will extend to studying current market trends in language learning applications to integrate innovative features that can enhance the learning experience. Competitive analysis will also play a role in positioning the Bahasa Malaysia Learning Application effectively within the market, ensuring it meets user needs more proficiently than existing solutions.

By the conclusion of the analysis phase, there will be a thoroughly documented specifications sheet. This document will include detailed descriptions of the application's functionality, user interface designs, performance criteria, and security measures, serving as a blueprint for the subsequent design and development phases to ensure that the final product aligns with both user requirements and business objectives.

3.2 Problem Analysis

In the analysis phase, we address the key issues identified in the problem statement, providing a critical evaluation of how these affect the learning experience for Bahasa Malaysia. The aim is to detail these problems further and suggest a preliminary direction for solving them in the design and development phases of the Bahasa Malaysia Learning Application.

- **Time Constraints and Accessibility**

Current systems for learning Bahasa Malaysia, predominantly traditional classroom-based courses, are often rigid and inflexible. This presents significant challenges for professionals and university students who must juggle demanding schedules, balancing work, personal, and academic responsibilities. Traditional courses with fixed schedules and location-based instruction fail to meet the needs of these learners, who require greater flexibility. The lack of readily accessible and adaptable learning materials exacerbates this issue, discouraging consistent engagement and long-term commitment to learning the language. A more flexible solution is needed that allows learners to access learning materials anytime and anywhere, adapting to their individual schedules and lifestyle demands.

- **Diverse Learning Needs and Preferences**

The one-size-fits-all approach of current Bahasa Malaysia learning resources does not cater to the diverse learning styles, paces, and backgrounds of the student population. Learners come from varied linguistic backgrounds and have different experiences with language learning, which can affect how they absorb new

information. Many existing resources fail to consider these variances, offering little in the way of personalization or adaptation to individual learning needs. To address this gap, the application should support customized learning pathways that consider the learner's previous language exposure, preferred learning styles, and pace, enhancing the effectiveness of the educational experience.

- **Integration of Real-Life Usage and Cultural Nuances**

Most existing platforms for learning Bahasa Malaysia lack practical integration of the language's real-life applications, including its cultural nuances and colloquial language. This deficiency leads to a superficial understanding of Bahasa Malaysia, ill-preparing learners for real-world interactions in a culturally diverse and rich setting like Malaysia. To remedy this, the application must incorporate immersive content that not only teaches formal language structures but also embeds learners in the cultural context of the language, using realistic scenarios and conversations that reflect daily use.

- **Technological Limitations**

The current digital resources for learning Bahasa Malaysia underutilize advanced technological features that could significantly enhance the learning experience. Technologies such as AI and gamification are often absent, resulting in a static and unengaging learning environment. These technologies have the potential to personalize the learning journey, making it more engaging and effective. Implementing AI could tailor the educational content to the user's progress and specific needs, while gamification could make the learning process more interactive and rewarding, thereby increasing motivation and retention.

3.3 Requirement analysis

This section will describe the requirements that the system must fulfill to achieve its goal. The requirements are determined based on the research, review and analysis that have been done for the project in the previous sections.

3.3.1 Data Requirement

The data requirements for the Bahasa Malaysia Learning Application are crucial to ensuring that the system efficiently handles the input, output, and storage of data. This section outlines the types of data the system will process, how these data types are used, and how they are stored. Effective management of these data ensures a seamless user experience and robust application performance.

- **System Input Data**
    - **User Profile Information**: This includes data entered during user registration, such as name, email address, age, and language proficiency level. This information helps personalize the learning experience and allows the system to tailor content to the user's needs.
    - **User Preferences**: Choices made by the user regarding preferred learning styles, topics of interest, and other customizable features. These inputs help the system adjust the learning environment to best fit the user's preferences.
    - **User Interaction Data**: Inputs from users as they interact with lessons and quizzes, including answers to questions, completion times, and feedback on content usability. This data is crucial for adjusting course difficulty and providing adaptive learning paths.

- **System Output Data**
    - **Progress Reports**: Outputs such as detailed reports on user progress, quiz scores, milestones reached, and recommendations for future study. These reports are essential for users to monitor their learning progress and plan their studies.
    - **Customized Learning Materials**: Based on user inputs and interactions, the system outputs personalized learning materials and quiz settings, tailored to the user's proficiency and learning speed.

- **Data Storage**

- **User Profiles**: Stored data that includes all registered personal and preference information which is used to customize the learning experience.
- **Course Content**: This includes a comprehensive database of all the course materials, exercises, multimedia content, and any additional resources available to the user.
- **Performance Logs**: These logs record all user interactions, scores, and historical data regarding course progress. They are vital for providing ongoing personalized learning experiences and for maintaining records of user achievements.
- **Feedback and User Generated Content**: User reviews, suggestions, and contributions (if applicable) are stored to help improve course content and engage the community.

- **Data Dictionary**

  The content of this section will be explained with the tables in Appendix A: Data Dictionary, which will describe what data and how it will be stored in the database including its data type, format and constraint. The data dictionary will be organized by its logical data model (LDM) since the application will be using Firebase which is based on the NoSQL approach.

3.3.2 Functional Requirement

The functional requirements of the Bahasa Malaysia Learning Application define how the system should operate, detailing the functions it must perform to handle data correctly. These requirements include how the system records, computes, transforms, and transmits data, ensuring effective and efficient operations that meet the users' needs.

- **Core System Functions**
  - **User Registration and Authentication**: The system must allow new users to register by providing necessary personal information and creating login credentials. It should authenticate returning users to ensure secure access to their accounts and personalized learning environments.

- **Data Recording**: All user interactions, such as course progress, quiz answers, and time spent on various activities, must be systematically recorded. This data is essential for tracking progress and providing personalized feedback.

- **Data Computation**: The system computes user progress and proficiency levels based on interaction data. Algorithms analyze quiz results and learning speeds to adjust the difficulty level of subsequent content, ensuring that it matches the learner's pace and learning style.

- **Data Transformation**: Input data from users is transformed into actionable insights and feedback. For instance, raw data from quizzes is converted into scores, which are then used to generate progress reports that users can understand and utilize.

- **Data Transmission**: The system needs to transmit data between the server and client devices securely. This includes sending personalized content to users and receiving data from user interactions, all while ensuring data integrity and security.

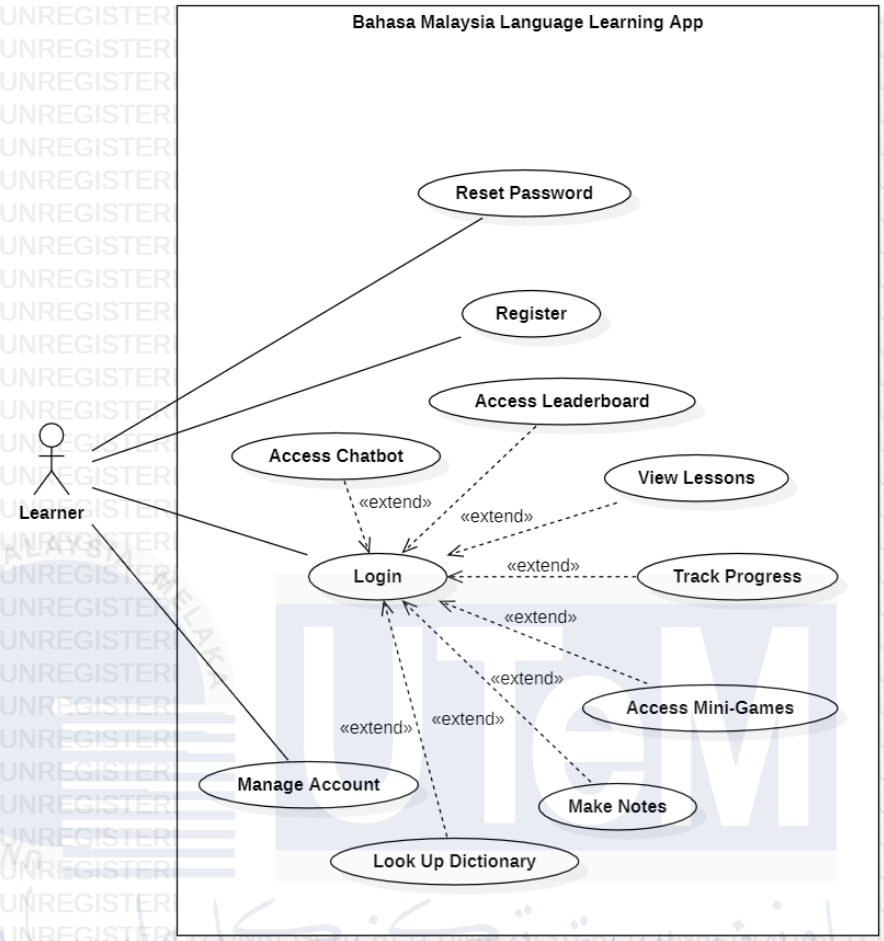

Figure 3-1 Use case diagram of the language learning application.

As depicted in figure 3-1, the use case diagram consists of 9 use cases which represent the main process or functions in the system, The extend notation indicates that the use case may optionally extend another use case process. The stickman figure represents the actor which is the third party that interacts with the system including users and the system responsible for handling any queries. The following are brief summaries of what each use case in the diagram does.

a) Register

The Register use case enables new users to create an account on the Bahasa Malaysia Learning Application. Users are required to provide essential personal information such as their name, email address, and password. This information is necessary to set up a unique user profile that will allow them to access personalized content and track their learning progress. Successful registration is mandatory before

users can log in and use other features of the application, ensuring that each user has a secure and personalized learning environment.

b) Login

The Login use case allows registered users to authenticate their identity by entering their email address and password. This process grants them access to their personalized learning dashboard where they can engage with lessons, quizzes, and other interactive features. Logging in is crucial for maintaining the security of user data and ensuring that each user's progress and preferences are correctly tracked and managed. It also serves as a gateway to all other functionalities of the application, ensuring a personalized and secure user experience.

c) Reset Password

The Reset Password use case provides a secure way for users to regain access to their account if they forget their password. Users can initiate a password reset by providing their registered email address, after which they will receive a link or code to set a new password. This function is vital for maintaining user access and ensuring security, as it verifies the user's identity before allowing any changes to account credentials. This process helps in minimizing disruptions to the user's learning experience and enhances the overall security of the application.

d) Manage Account

The Manage Account use case allows users to update and manage their personal information, such as changing their password, email address, or other profile details. This feature ensures that user profiles remain current and accurate, reflecting any changes in the user's personal information or preferences. By managing their account settings, users can personalize their learning experience, ensuring that notifications, lessons, and other interactive elements align with their individual needs and preferences.

e) View Lessons

The View Lessons use case is central to the application's educational functionality, allowing users to browse and select various lessons designed to teach Bahasa Malaysia. Users can navigate through different topics and difficulty levels, accessing multimedia content that includes text, and audio. This functionality ensures that learners have a wide range of educational resources at their fingertips, structured to progressively build their language skills in an engaging and interactive manner.

f) Track Progress

The Track Progress use case enables users to monitor their learning achievements and performance metrics. This feature provides detailed insights into completed lessons, quiz scores, time spent on different activities, and overall progress towards learning goals. By tracking their progress, users can identify areas of strength and those needing improvement, allowing them to adjust their learning strategies accordingly. This feedback mechanism is essential for motivating learners and helping them stay on track with their educational objectives.

g) Access Mini-Games

The Access Mini-Games use case offers users an engaging way to reinforce language concepts through interactive games. These mini games are designed to be both educational and entertaining, helping users practice vocabulary, grammar, and comprehension skills in a playful environment. This feature enhances user engagement by adding a gamified element to the learning process, making it more enjoyable and effective in retaining new information.

h) Make Notes

The Make Notes use case allows users to make their own notes which will be uploaded to the backend database. This is to simulate the act of making notes when studying and acts as a quality-of-life tool for when the users need to quickly look up and write their own notes while using the application.

i) Look Up Dictionary

This use case allows the user to look up the in-app dictionary for when they require further explanations on words, terms and phrases.

j) Access Chatbot

The Access Chatbot use case introduces an AI-driven chatbot that allows users to practice their Bahasa Malaysia skills through simulated conversations. The chatbot provides instant feedback and interactive dialogues, helping users improve their speaking and comprehension abilities in a realistic context. This feature is particularly beneficial for practicing real-life language usage and gaining confidence in conversational skills, making the learning experience more immersive and practical.

k) Access Leaderboard

The Access Leaderboard use case fosters a competitive and motivational learning environment by displaying a ranking of users based on their progress and achievements. Learners can see how they stack up against others, which can encourage them to engage more deeply with the learning materials to improve their standing. This feature not only motivates users to perform better but also creates a sense of community and shared learning goals among the users of the application.

3.3.3 Non-functional Requirement

Non-functional requirements define the criteria that judge the operation of a system, as opposed to specific behaviors or functions. For the Bahasa Malaysia Learning Application, these requirements ensure that the system performs effectively and efficiently while providing a positive user experience. Below are the key non-functional requirements for the application:

- **Performance**
    - **Scalability**: The system must be capable of handling an increasing number of users without degrading performance. This includes the ability to scale horizontally by adding more servers to handle additional load.

- **Response Time**: The application should have a quick response time, with each page load or interaction taking no more than 2 seconds under normal load conditions.
- **Resource Utilization**: The application should use minimal system resources. On the server side, it should efficiently use CPU and memory, while the client-side (mobile app) should be optimized to run smoothly on devices with limited resources.
- **Data Handling**: The system should be able to handle large volumes of data, including user profiles, learning progress records, and multimedia educational content. It should support data storage and retrieval operations efficiently, with database queries executed within 1 second on average.

- **Quality Requirements**
  - **Reliability**: The application should be available 99.9% of the time, ensuring minimal downtime. This includes implementing redundant systems and regular backups to prevent data loss.
  - **Accuracy**: The system must accurately record and compute user progress, quiz results, and any other metrics. This ensures that the feedback and reports provided to users are reliable and precise.
  - **Usability**: The application should be user-friendly, with an intuitive interface that requires minimal learning for new users. It should include clear navigation paths, helpful prompts, and a consistent layout and design.
  - **Accessibility**: The application should be accessible to users with disabilities, complying with accessibility standards such as WCAG 2.1. This includes providing text alternatives for non-text content and ensuring that the app is navigable via keyboard or other assistive technologies.
- **Security**
  - **Data Protection**: User data must be securely stored and transmitted. This involves encrypting sensitive data, using secure protocols (HTTPS), and implementing robust authentication and authorization mechanisms.

- **Privacy**: The application must comply with relevant data privacy regulations such as GDPR. User data should be collected and used only for the stated purposes, with options for users to control their data privacy settings.
- **Error Handling**: The system should gracefully handle errors and exceptions, providing informative error messages and logging incidents for further investigation. This helps in maintaining system stability and provides a better user experience.

- **Maintainability**
  - **Modularity**: The application should be designed in a modular fashion, allowing for easy updates and maintenance. This involves separating different functionalities into distinct modules or services.
  - **Documentation**: Comprehensive documentation should be available for both users and developers. User documentation should provide clear instructions on how to use the application, while developer documentation should include detailed information on the system architecture, codebase, and APIs.
  - **Support and Upgrades**: The application should have a structured process for receiving updates and support. This includes regular updates to improve functionality and security, as well as a support system to assist users with any issues they encounter.
- **Compatibility**
  - **Integration**: The system should support integration with other services and APIs, such as third-party authentication providers (e.g., Google, Facebook) and educational content providers.

3.3.4 Others Requirement

This section addresses the additional requirements necessary for supporting the development, deployment, and operation of the Bahasa Malaysia Learning Application. These requirements, while not directly related to the system's core functionality, play a crucial role in ensuring the overall effectiveness and efficiency of the application.

- **Development Tools and Environments**

- **Version Control Systems**: Utilizing version control systems like Git and platforms such as GitHub or GitLab is essential for managing code changes, collaborating with peers and supervisors, and maintaining a comprehensive history of project development. These tools enable seamless collaboration and efficient project management.
- **CI/CD Pipelines**: Continuous Integration and Continuous Deployment (CI/CD) tools like Jenkins or GitHub Actions automate the process of building, testing, and deploying the application. These tools ensure that the codebase remains clean, functional, and ready for deployment at all times, facilitating rapid development cycles and minimizing integration issues.
- **API Testing Tools**: Tools like Postman are used to test and document APIs. These tools are vital for ensuring that all API endpoints work correctly and efficiently, facilitating smooth integration between the frontend and backend services.

- **Project Management**
  - **Project Management Tools**: Tools such as JIRA, Trello, or Asana help in planning, tracking, and managing project tasks and milestones. These tools ensure that the project stays on schedule and that all tasks are properly assigned and monitored.
- **Documentation and Design**
  - **Design Tools**: Design tools such as Adobe XD, Figma, or Sketch are used for creating wireframes, prototypes, and user interface designs. These tools help ensure that the application's design is user-friendly and visually appealing.
  - **Documentation Tools**: Comprehensive documentation is essential for maintaining a clear and organized record of the development process. Tools like Confluence, Google Docs, or Microsoft Word are used to document requirements, design decisions, and user guides, ensuring that all stakeholders have access to the necessary information.
- **Development and Testing Environments**

- **Development Environments**: Integrated Development Environments (IDEs) such as Visual Studio Code, IntelliJ IDEA, or Android Studio provide the necessary tools for writing, debugging, and testing code. These environments support the entire development lifecycle and enhance productivity.
- **Testing Frameworks**: Automated testing frameworks like Selenium, JUnit, or Flutter's built-in testing tools are used to create and run tests that verify the functionality and performance of the application. These frameworks help identify and fix bugs early in the development process.

3.4 Conclusion

This chapter has provided the reader with the overview of what the Bahasa Malaysia Language Learning application is trying to solve, what it will do to solve it and how it will accomplish it. From the thorough analysis of problems and requirements it is expected that readers are now well informed on what this project is trying to achieve. In the next chapter, the detailed design of the system will be further described in detail.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter goes into the design phase of the Bahasa Malaysia Learning Application, building on the foundational insights gathered during the analysis phase. The primary objective of this chapter is to present both the preliminary and detailed designs of the application. This involves transforming the defined requirements and functional specifications into a structured design that can be executed to create a functioning system.

The preliminary design outlines the overall system architecture and high-level components of the application. It provides a macro view of how various elements of the application interact with each other, including the user interface, backend services, data storage, and third-party integrations. This stage sets the framework for the detailed design, which delves deeper into the specifics of each component, detailing the technical specifications, data flow diagrams, interface designs, and interaction models.

Through detailed diagrams, architectural blueprints, and design schematics, this chapter aims to showcase how the application's features and functionalities will be implemented. The designs are created with careful consideration of best practices in software architecture, usability, scalability, and maintainability. Additionally, this chapter will discuss how the designs adhere to non-functional requirements such as security protocols, performance metrics, and compatibility standards.

Ultimately, the design phase ensures that the planned application is not only technically viable but also user-friendly and aligned with the core objectives of

enhancing the learning experience for users. This chapter will serve as a blueprint for developers and stakeholders, providing a clear and actionable guide for moving forward into the development phase.

4.2 High-Level Design

High-level design refers to the surface level design which entails the architectural and structural overview of the system components and each of its interactions. In this section, the essential features and functionality of the Bahasa Malaysia Language Learning application will be described without delving into the detailed implementation specific which will be covered in the upcoming sections.

4.2.1 System Architecture

This section aims to give readers a knowledge of the main components of the system and their interactions with one another by describing and illustrating the overall structure and organization of the system components using an architectural diagram. The Bahasa Malaysia Language Learning application's architectural design, which follows the Mobile Application Architecture diagram, is depicted in Figure 4.1 below.

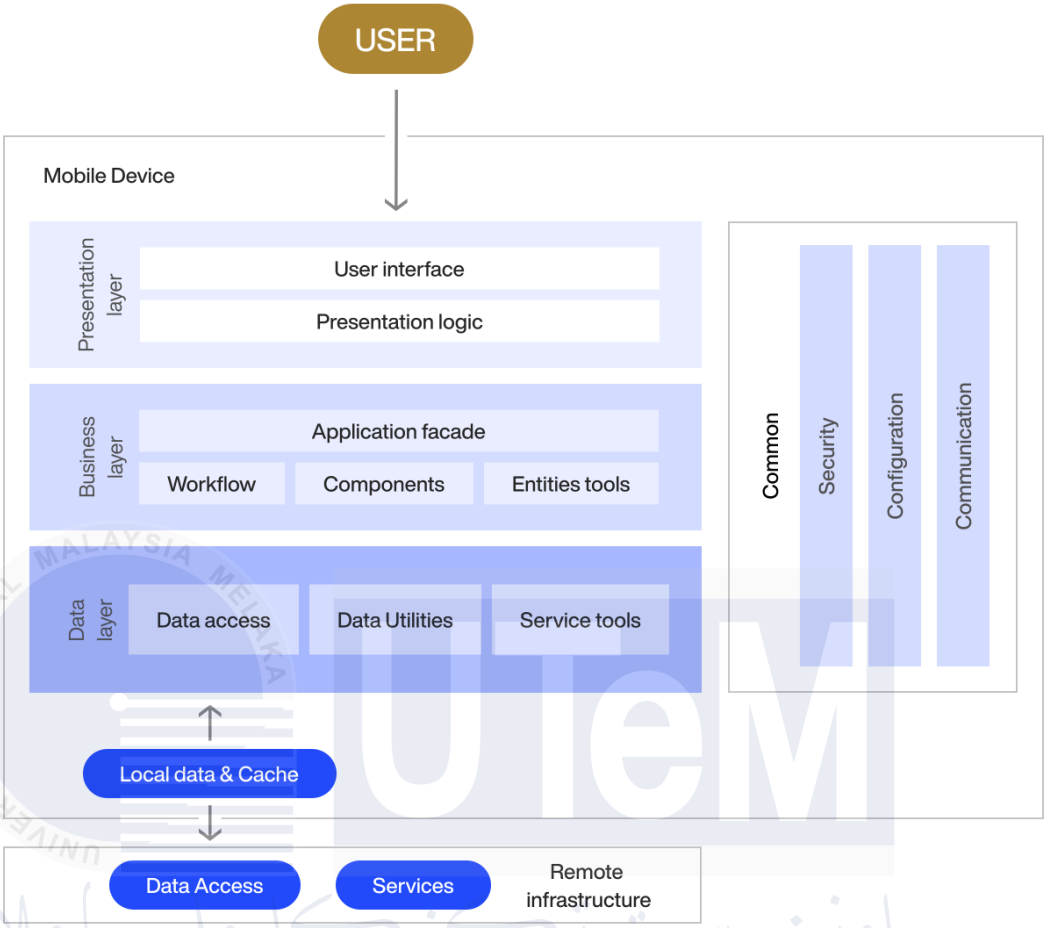

Figure 4-1 Mobile Application Architecture diagram used for the project.

The architecture of the Bahasa Malaysia Learning Application is meticulously organized into distinct layers, each responsible for a specific aspect of the application's functionality. This structure not only facilitates a clear separation of concerns but also enhances maintainability and scalability. The detailed breakdown of the application's architecture are as follows:

- **Presentation Layer**
  - **User Interface**: This layer includes all the user-facing components of the application, such as screens for user registration, login, lesson browsing, and chat interactions. It is designed to provide a seamless and intuitive user experience, leveraging the Flutter framework's capabilities to ensure consistency across various Android devices.
  - **Presentation Logic**: This component manages the logic behind the user interface, processing user inputs and controlling the display of

data. It handles the dynamic states of UI components and binds data from the Business layer to these components, ensuring that the user interactions are responsive and efficient.

- **Business Layer**
  - **Application Facade**: Acts as an intermediary that simplifies the interactions between the Presentation and Business layers. It coordinates critical operations such as user authentication, content delivery, and interactions with external APIs, ensuring streamlined process flows.
  - **Workflow**: Manages the sequence of actions within the application, such as the steps from user authentication to accessing educational content and interacting with the ChatGPT AI chatbot. This component ensures that the application's operations are executed in a logical and efficient manner.
  - **Components**: These are modular parts of the application, each dedicated to specific functionalities like managing user profiles, retrieving lesson content, administering quizzes, and configuring user settings.
  - **Entities**: Represents the business objects within the application, including User, Lessons, Quizzes, and Chat Sessions. These entities are central to the application's functionality and are integral to the data handling processes.

- **Data Layer**
  - **Data Access**: Interfaces directly with Firebase Firestore to manage all data transactions. It performs CRUD operations, ensuring that data integrity and synchronization are maintained across user devices.
  - **Data Utilities**: Comprises tools and utilities necessary for data processing tasks such as validation, formatting, and transformation before storage or after retrieval.

- **Service Tools**: Includes specialized tools required for specific data operations, particularly focusing on data security measures like encryption and decryption, essential for protecting user data.

- **Common Layer**
  - Configuration: This component manages all configuration settings for the application, centralizing control over parameters such as API keys, Firebase configurations, and other essential operational settings.
  - Security: Ensures the security of the application through robust mechanisms. This includes the integration of Firebase Authentication for secure user login processes and the management of encrypted communications with Firebase and OpenAI's ChatGPT API.
  - Communications: Handles all external communications for the application, managing HTTP/HTTPS requests to Firebase and the OpenAI's ChatGPT API, ensuring that all data exchanges are secure and efficient.

- **Additional Considerations**
  - **Local Data**: Utilized for storing data locally on the user's device, which can include caching mechanisms for offline access and storing user preferences to enhance the application's responsiveness and usability.
  - **Remote Infrastructure**: Refers to the direct interactions with Firebase and OpenAI's ChatGPT API, which are crucial for ensuring that user data is continuously synchronized and that interactions with the chatbot are conducted in real-time.

This architecture effectively supports the Bahasa Malaysia Learning Application by ensuring it is robust, scalable, and capable of delivering a high-quality educational experience. Each layer is crafted to meet specific functional requirements while ensuring overall system integrity and performance.

4.2.2 User Interface Design

The system's depiction to users, known as the graphical user interface, makes it easier for users to engage with the system and enable its functionality. The components of the user interface that are required will be identified and designed during this phase. The design will be centered on mobile device viewports because the Bahasa Malaysia Language Learning application user interface will be implemented on mobile applications.

This section will include an explanation of the design as well as figures illustrating it. The forms that will be utilized in the wireframe interface design are described in Table 4.1 below. It is anticipated that the table will aid readers in comprehending the layout design of the next diagrams.

Table 4-1 Interface Design Shapes Description.

Shape	Meaning
	Rectangle represents Input field.
	Rounded border rectangle represents clickable element such as button.
	Rounded border rectangle presents a screen/slide that's meant to convey text or lessons.

The first screen that users will see upon opening the application would be the login screen as shown in the following figure.

Figure 4-2 The interface for the Login Screen.

The login screen interface has two input fields for email and password. The "Login" button navigates users to the main menu after entering credentials. The "Register" text is clickable, leading to the registration page for new users.

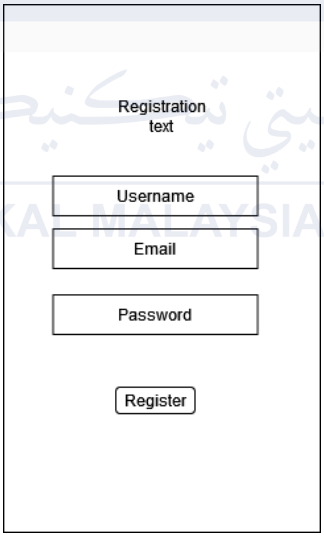

Figure 4-3 The interface for the Registration Screen.

The registration screen includes three input fields for Username, Email, and Password, along with a Register button. This screen is where new users create their accounts.

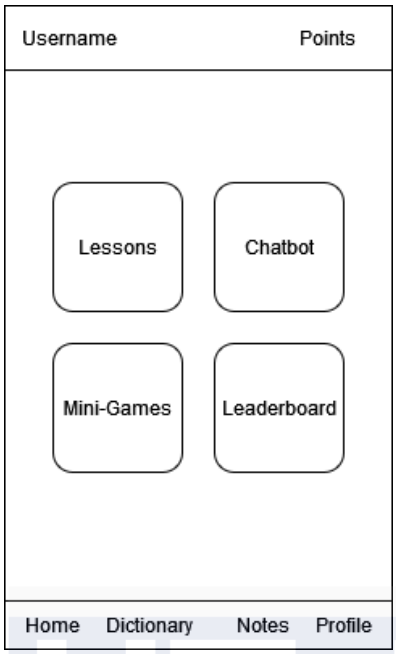

Figure 4-4 The interface for the Main Menu Screen.

The main menu screen has four buttons for accessing Lessons, Chatbot, Mini-Games, and Leaderboard. The top bar shows the user's username and points. The bottom navigation bar provides quick access to Home, Dictionary, Notes, and Profile.

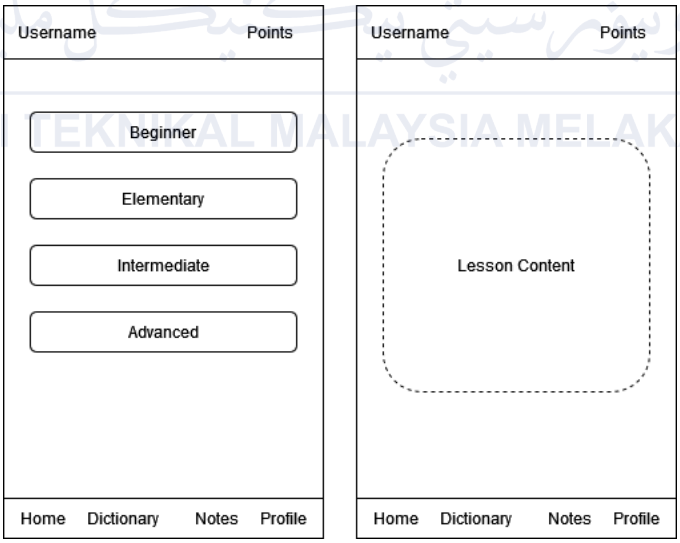

Figure 4-5 The interface for the Lesson Screen.

The lesson interface in the Bahasa Malaysia Learning Application offers a structured learning experience with four levels: Beginner, Elementary, Intermediate, and Advanced. Users select a level to view the corresponding lesson content in a focused view. The interface includes a navigation bar for accessing other app sections like the Dictionary, Notes, and Profile.

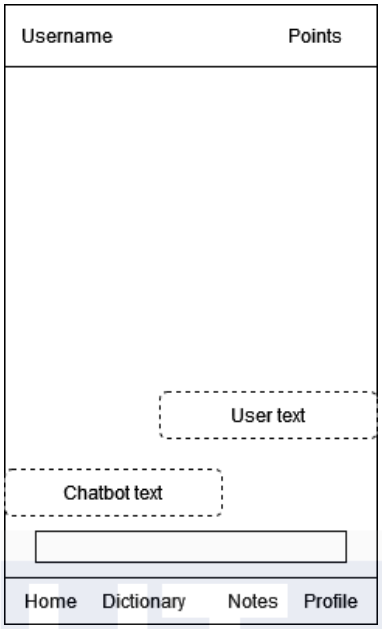

Figure 4-6 The interface for the Chatbot Screen.

The chatbot screen allows users to interact with a virtual assistant in Bahasa Malaysia. It simulates conversation by displaying user and chatbot text exchanges. This helps users practice language skills in a conversational format.

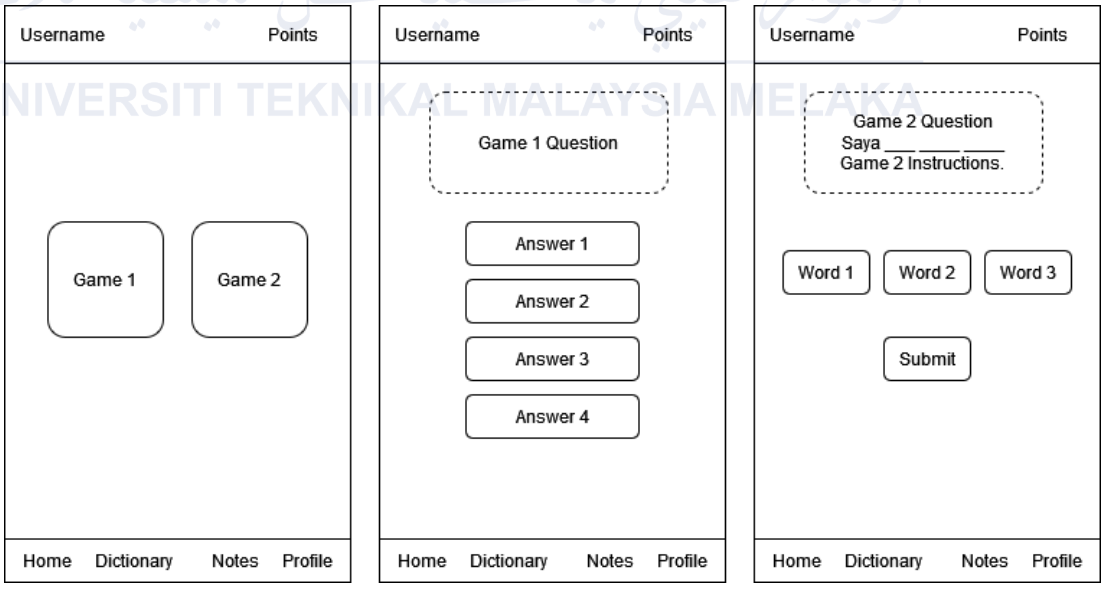

Figure 4-7 The interface for the Mini-Games Screen.

The mini-games screen provides options for users to choose between different games. Each game has its own screen with specific instructions and questions. The games offer interactive language exercises that challenge the user's knowledge and skills in Bahasa Malaysia.

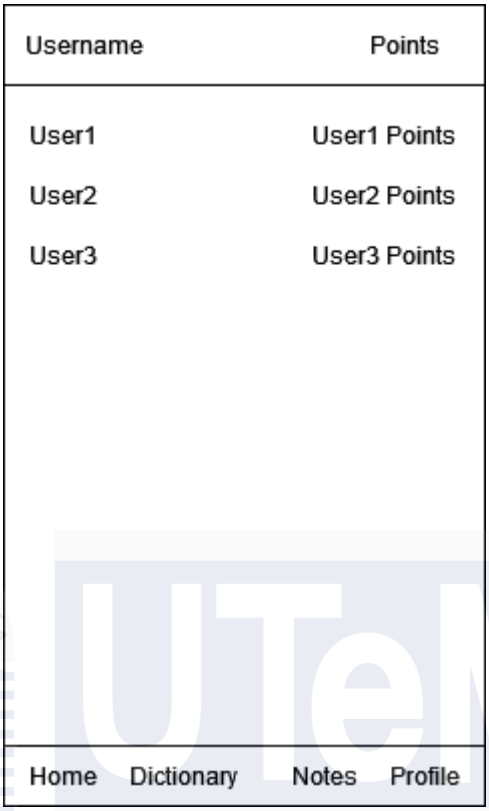

Figure 4-8 The interface for the Leaderboards screen.

The leaderboard screen displays the points of registered users in descending order. It ranks users from highest to lowest based on their accumulated points. This feature encourages competition and motivates users to engage more with the application.

Since some progress has been made during the development of the application, the below figures will be based on the actual UI of the application and not just wireframe designs.

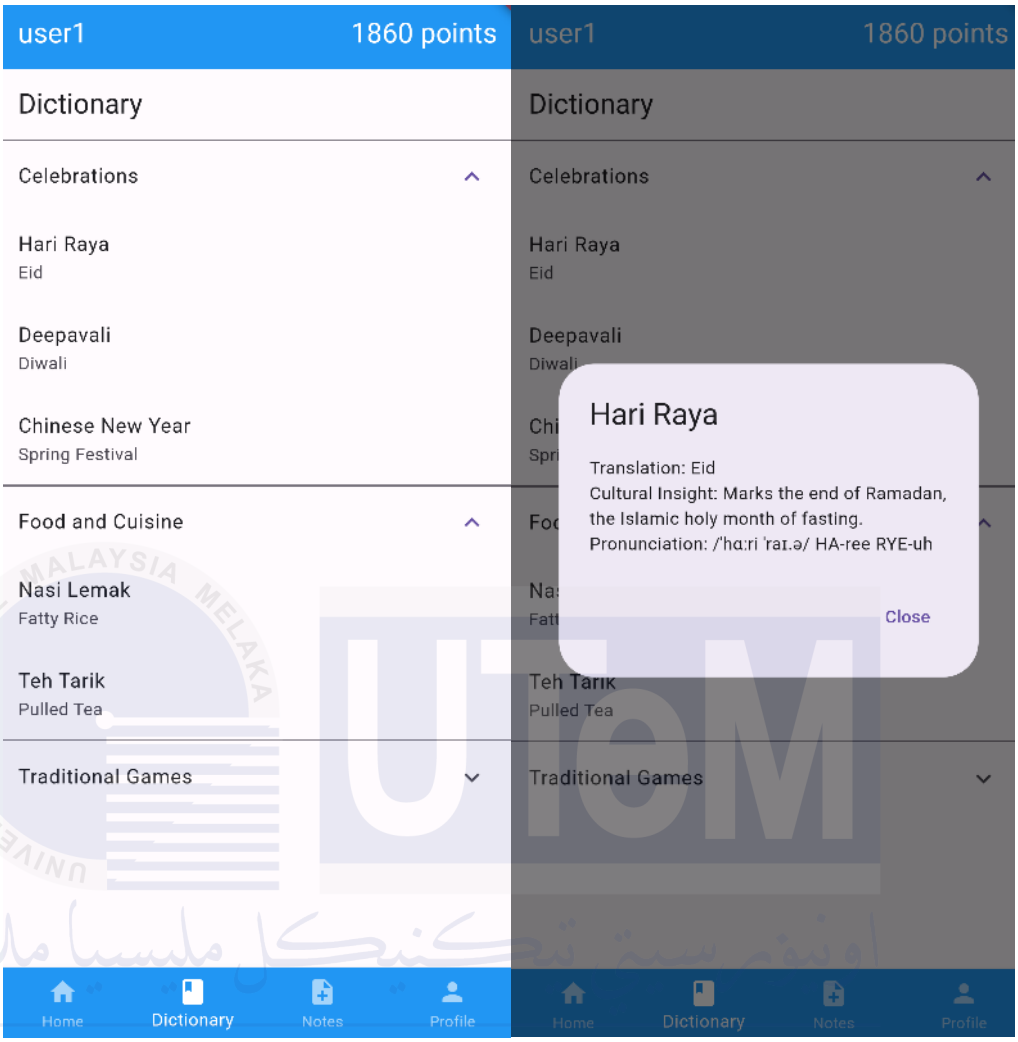

Figure 4-9 The interface for the Dictionary screen.

The dictionary screen allows users to look up words and their translations, cultural insights, and pronunciation guides. Users can expand categories to explore specific terms. Selecting a word provides detailed information, including cultural context and pronunciation.

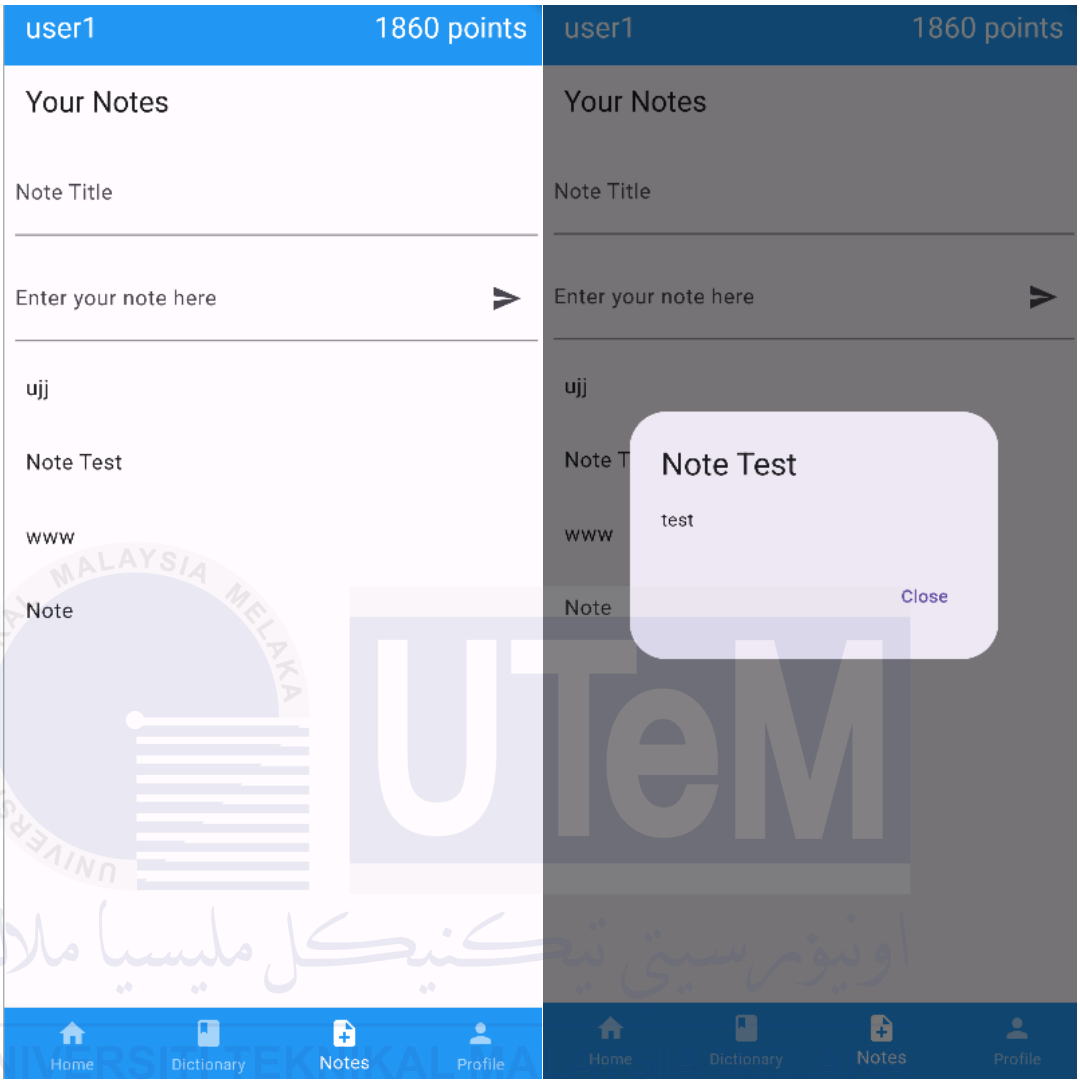

Figure 4-10 The interface for the Notes screen.

The notes screen allows users to create and manage their personal notes. Users can add a title and content for each note, which is then listed on the screen. Clicking on a note displays its full content in a popup window.

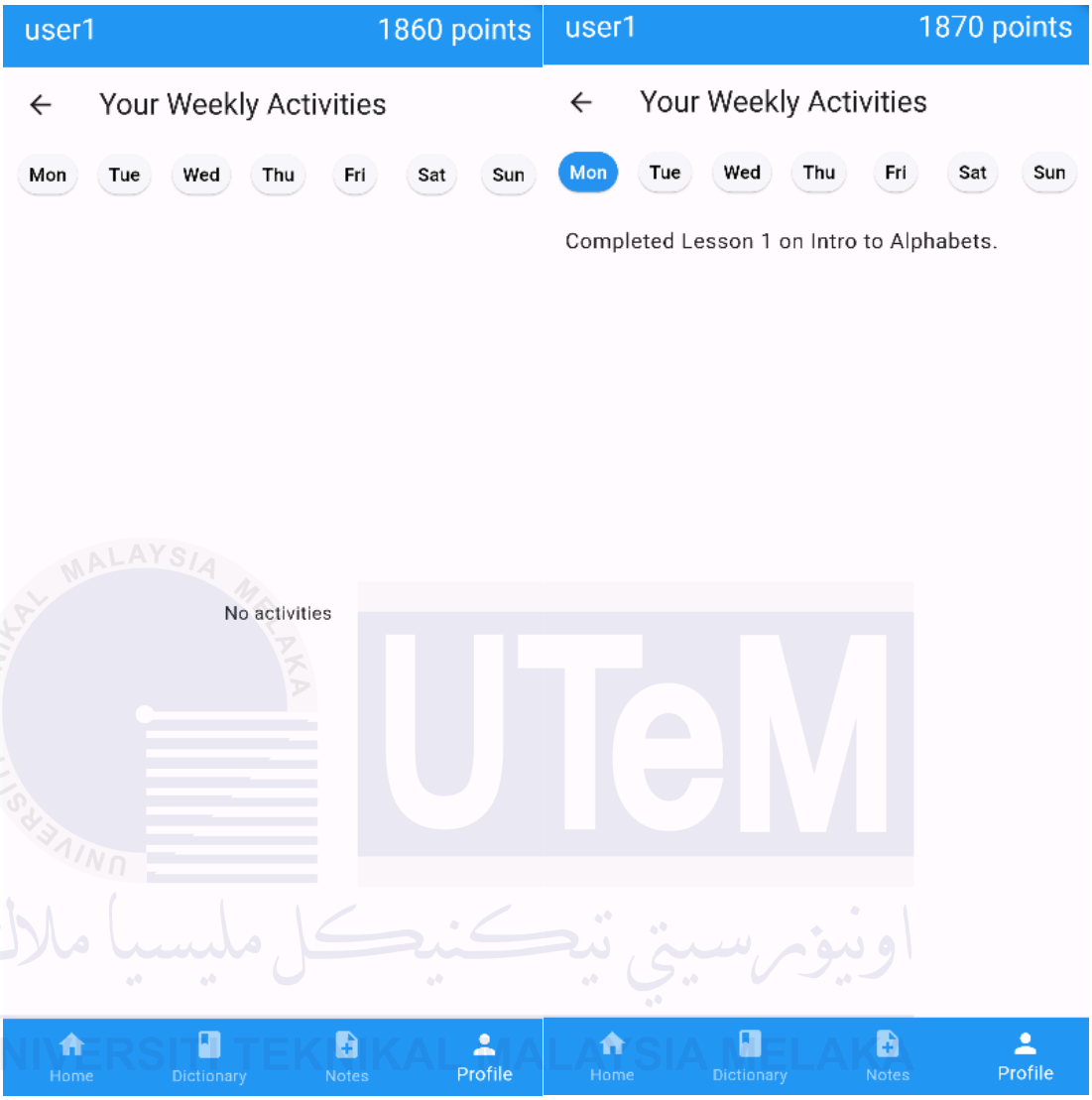

Figure 4-11 The interface for the Activities screen.

The activities screen displays the user's weekly activity log. Each completed lesson or mini-game is recorded and displayed on the corresponding day. If there is any recorded activity, the day button changes color to blue, indicating user engagement on that day.

4.2.3 Database Design

Since this project utilizes Firebase, the Database Design will not be using ERD but instead uses a Logical Data Model to represent the database. Since Firebase is the chosen database, we will be dealing with documents, collections, and files instead.

4.2.3.1 Conceptual and Logical Database Design

This section introduces the logical data model (LDM) employed in the database design, distinguishing it from the traditional Entity Relationship Diagram (ERD). The adaptation of an LDM over an ERD is specifically tailored for the project's use of Firebase Firestore, a NoSQL database, which structurally and functionally diverges from relational database systems that typically utilize ERDs.

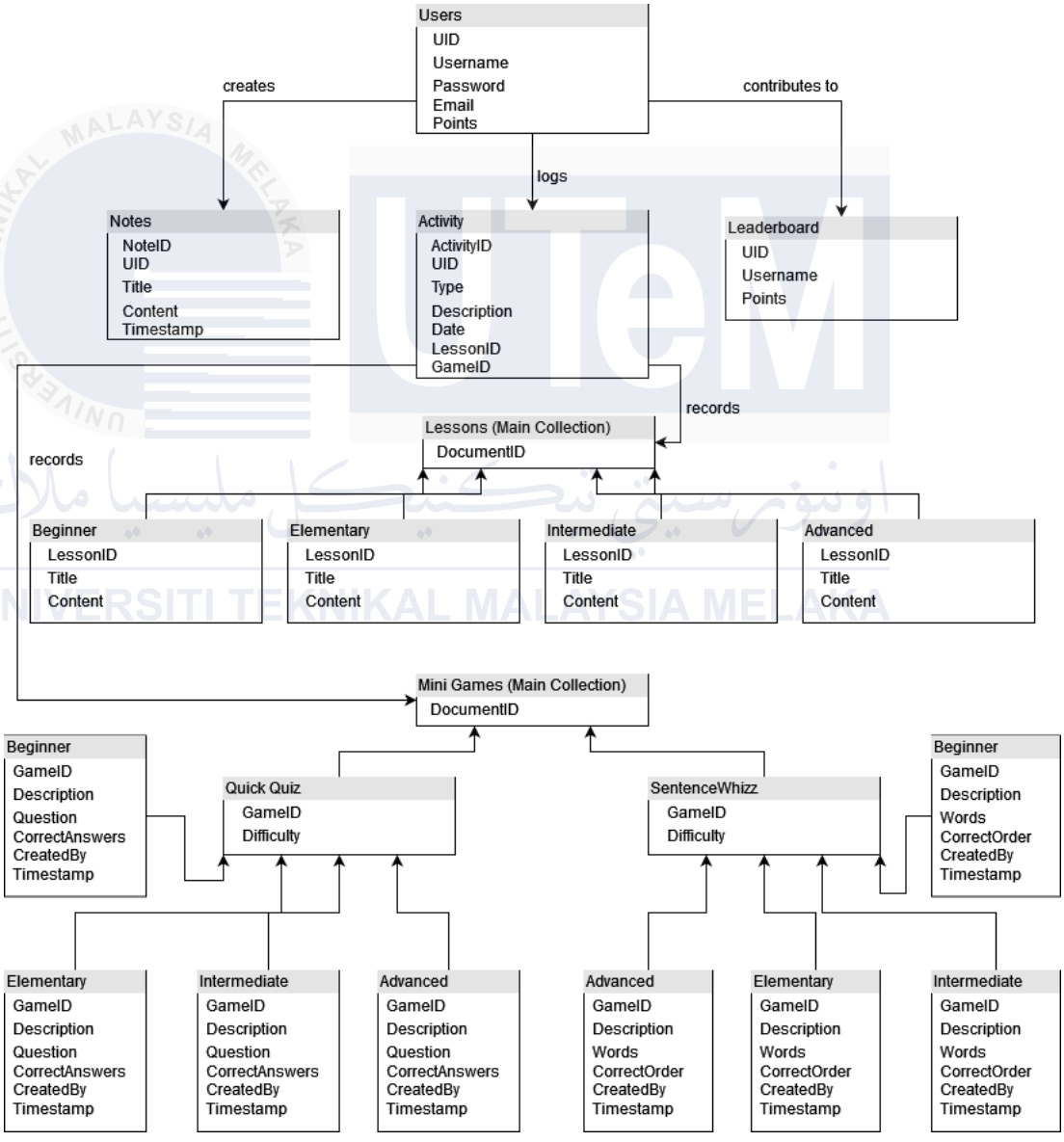

Figure 4-12 The Logical Data Model of the Bahasa Malaysia Language Learning Application.

- **Logical Data Model Explanation**

Based on the figure above, the Logical Data Model for this project is designed to align with Firebase Firestore's document-based NoSQL architecture. Unlike relational databases that use tables, Firestore organizes data in collections and documents, which enables flexible, scalable, and real-time data management. This model details the organization of data entities within the application and outlines their logical arrangement considering Firestore's capabilities.

- **Entity Relationships**

The entity relationships in the LDM are structured based on the application's functional requirements and the interactions among various data points:

- **Users**: Central to the model, the User entity represents individual users of the application and houses nested collections for Activities and Notes. This nesting allows for efficient queries filtered by user context and supports the application's requirement to provide personalized content and history tracking.
- **Activities**: Nested within each User, the Activities collection logs user actions such as lesson completions and minigame results. Each document in this collection details the activity type, a description, and the date, supporting user engagement analysis and personalized reporting.
- **Notes**: Also nested under User, the Notes collection stores user-generated content, with each note containing a title, content body, and timestamp. This setup facilitates quick retrieval and management of notes specific to each user.
- **Leaderboard**: Derived from User. This takes the users' username and points and ranks them from lowest to highest.
- **Lessons and Minigames**: These are organized into main collections, each subdivided into levels (Beginner, Elementary, Intermediate, Advanced), reflecting the structured progression designed to cater to different learning stages. Each sub-collection contains documents detailing specific lessons or minigames, including attributes like title, content/description, and associated metadata. This hierarchical

structuring aids in the dynamic retrieval of content based on the user's current learning phase.

- **Data Dictionary**

Referring to the tables below, the data dictionary provides detailed descriptions of each entity and attributes within the LDM, ensuring clarity and consistency across the development cycle. It specifies the data type, possible values, and a brief description of how and where each attribute is used within the application.

Table 4-2 Users Table.

Field	Data Type	Description
UID	String	Unique identifier for each user.
Username	String	User's chosen username.
Password	String	User's password for login, stored securely.
Email	String	User's email address.
Points	Integer	Accumulated points from completing lessons and minigames.

The Users Table stores essential information for each registered user in the application. It includes the user's unique identifier (UID), chosen username, securely stored password, email address, and accumulated points from completed lessons and minigames. This table is crucial for managing user authentication and tracking progress.

Table 4-3 Activity Table.

Field	Data Type	Description
ActivityID	String	Unique identifier for each activity.
UID	String	Unique identifier for each user.
Type	String	Type of activity (e.g., lesson completion, minigame play).
Description	String	Description of the activity.
Date	Timestamp	Date and time when the activity occurred.

The Activity Table records various user activities within the application. It includes fields for the activity's unique identifier (ActivityID), the user's identifier (UID), the type of activity (such as lesson completion or minigame play), a description of the activity, and a timestamp indicating when the activity occurred. This table helps track user engagement and progress over time.

Table 4-4 Notes Table.

Field	Data Type	Description
NoteID	String	Unique identifier for each note.
UID	String	Unique identifier for each user.
Title	String	Title of the note.
Content	String	Content of the note.
Timestamp	Timestamp	Date and time when the note was created or last updated.

The Notes Table stores information about user-created notes in the application. It includes fields such as NoteID, which uniquely identifies each note, UID for the user identifier, the title and content of the note, and a timestamp indicating when the note was created or last updated. This table allows users to save and manage their notes efficiently.

Table 4-5 Leaderboard Table.

Field	Data Type	Description
UID	String	Unique identifier for each user.
Username	String	User's chosen username.
Points	Integer	Accumulated points from completing lessons and minigames.

The Leaderboard Table captures the performance data of users in the application. It includes the UID, which uniquely identifies each user, the user's chosen username, and their accumulated points from completing lessons and minigames. This table helps rank users based on their points, fostering a competitive learning environment.

Table 4-6 Lessons Table (divided into subcategories).

Field	Data Type	Description
LessonID	String	Unique identifier for each lesson.
Title	String	Title of the lesson.
Content	String	Detailed content or body of the lesson.

The Lessons Table contains information on individual lessons available in the application. It includes the LessonID as a unique identifier for each lesson, the Title of the lesson, and the Content that details the lesson's material. This structure helps in organizing and retrieving specific lesson data efficiently.

Table 4-7 Mini-Game Table (divided into subcategories).

Field	Data Type	Description
GameID	String	Unique identifier for each minigame.
Description	String	Description of the minigame.
Questions	Array	List of questions or challenges in the minigame.
CorrectAnswers	Array	List of correct answers corresponding to the questions.
CreatedBy	String	Identifier of the creator or admin who setup the minigame.
Timestamp	Timestamp	Date and time when the minigame was created or last updated.

The Mini-Game Table stores data for each minigame in the application. It includes the GameID to uniquely identify each minigame, a Description of what the minigame entails, and an Array for Questions and CorrectAnswers to outline the minigame's challenges and solutions. The CreatedBy field records who set up the minigame, while the Timestamp marks when the minigame was created or last updated.

- **Consideration of Normalization**

While traditional normalization principles aimed at reducing redundancy in relational databases are not directly applicable to NoSQL structures, the principles of organizing data to minimize duplication while ensuring data integrity and query efficiency are still relevant. In Firestore, strategic data duplication is occasionally utilized to optimize read operations, which are more frequent and cost-sensitive than write operations.

Normalization in this context focuses on structuring data to support efficient access patterns and update mechanisms. For instance, while user-specific data like notes and activities are nested within the user documents to minimize read operations and data latency, common data like lessons and minigames are centralized to avoid duplication and facilitate easier updates.

4.3 Detailed Design

The detailed design section further elaborates on the system's architecture, focusing on the logical arrangement and operational logic of the design to meet the specified requirements.

4.3.1 Software Design

Using Object-Oriented Analysis and Design (OOAD) principles and Unified Modeling Language (UML), this section will go through the software design of the project. The figure below will provide an overview of the project's class diagram.

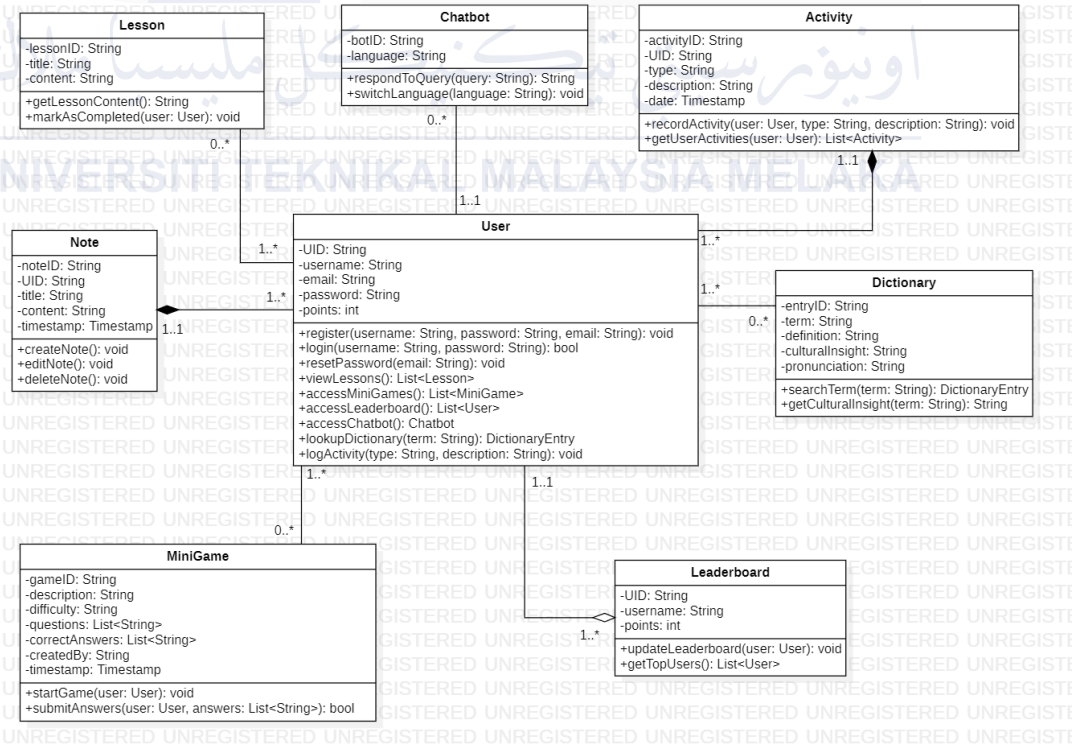

Figure 4-13 Overview of the Class Diagram of the project.

The User class manages user-related functionalities, including registration, login, and activity tracking. It interacts with other classes such as Lesson, MiniGame,

Chatbot, Dictionary, and Leaderboard to provide comprehensive learning and progress-tracking features. The Lesson class focuses on lesson management, offering methods for retrieving lesson content and marking lessons as completed. Similarly, the MiniGame class handles game activities, allowing users to start games, submit answers, and track progress within different difficulty levels.

The Activity class logs user activities like lesson completions and game plays, storing details such as type, description, and timestamp of each activity. The Chatbot class provides interactive learning through queries and language switching, enhancing the user experience with responsive dialogues. The Dictionary class is designed for term lookups, providing definitions, cultural insights, and pronunciation guides to enrich the learning process. The Leaderboard class tracks user points, updating standings based on lessons completed and games played, fostering a competitive learning environment.

Additionally, the Note class allows users to create, edit, and delete personal notes, supporting a more personalized learning journey. Each class is thoughtfully designed with specific attributes and methods, ensuring clear responsibilities and seamless interactions. For instance, the User class's logActivity method records user actions, linking to the Activity class, while Chatbot operations like respondToQuery and switchLanguage enhance user engagement. Overall, these classes are cohesively integrated, ensuring flexibility, scalability, and ease of maintenance within the application’s architecture.

4.3.2 Physical Database Design

In the transition from a logical data model to a physical database design, specific adjustments and implementations are tailored to the chosen database management system, which in this project is Firebase Firestore. Since Firestore is a NoSQL database, the concepts of DDL and DCL do not apply in the traditional sense as they would in SQL-based systems. Instead, the focus is on structuring collections, documents, and implementing security rules and data validation.

- **Collections:**

- **Users Collection**: This collection stores individual documents for each user, with each document identified by a unique user ID. The typical fields within these documents include username, password (stored securely as a hash), email, and points. Each user document also contains subcollections named Activities and Notes, which store records of the user's interactions and personal entries respectively.
- **Lessons Collection**: Organized by difficulty levels such as Beginner, Elementary, Intermediate, and Advanced. Each document within these subcollections represents a specific lesson and includes fields like title and content, providing structured data that can be easily accessed and manipulated based on the lesson's difficulty level.
- **Minigames Collection**: Similar to the lessons collection, the minigames are also categorized by difficulty levels, with separate subcollections for different types of games like QuickQuiz and SentenceWhizz. The documents in these subcollections hold information necessary for game functionality, including fields such as description, question, correctAnswers, and createdBy, allowing for a dynamic and interactive game experience.

Regarding constraints and validation, Firebase Firestore allows the setup of validation rules using Firestore Security Rules as seen in the figure below.

```
rules_version = '2';

service cloud.firestore {
  match /databases/{database}/documents {
    // Match any document in the 'users' collection
    match /users/{userId} {
      allow read, write: if request.auth.uid == userId;

      // Match any sub-collection or document under a specific user
      match /{document=**} {
        allow read, write: if request.auth.uid == userId;
      }
    }
  }
}

```

Figure 4-14 Firestore Security rules of the application.

File Organization and Indexes are also features in Firestore, Firestore automatically indexes all individual fields in documents for efficient query

performance. For complex queries involving multiple fields, manual configuration of composite indexes through the Firebase console is required. For example, creating a composite index for querying lessons by title and content within a specific level.

4.4 Conclusion

Throughout this chapter, the design phase of the system development has been completed. The designs, including logical, technical, and physical design, have been described in detail which provides an in-depth overview of the system development. Upon completion of the design phase, the development process can proceed to the next phase where the actual working system will be developed and documented in the upcoming chapter.

CHAPTER 5: IMPLEMENTATION

5.1 Introduction

This chapter provides an overview of the implementation of the Bahasa Malaysia Learning Application. Activities involved are the setup of the development environment. The application consists of only one application which is the main application that will be accessed through mobile phones. This application will connect to the backend database for data processing. The detailed implementation will be covered in this chapter.

5.2 Software Development Environment Setup

During the process of developing the application, all components will be deployed locally through an IDE that is connected to the internet. The device for testing will be on the IDE's emulator and a mobile phone that are able to access the internet. The general overview of the software development environment setup is as depicted in the figure below.

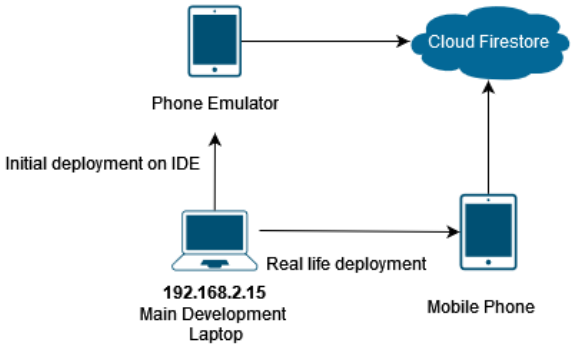

Figure 5-1 Software Development Environment.

As explained, the entirety of the project will be developed on one workstation and the application will then be deployed on devices which will be accessible if the devices have access to the cloud database.

5.3 Software Configuration Management

5.3.1 Configuration environment setup

The configuration management for the project utilizes comprehensive version control and deployment strategies to maintain project integrity and facilitate efficient management of the development process.

Design and Setup of Configuration Management:

1. **Version Control**:

   Tools Used: Git

   Git is employed for tracking and managing changes to the source code. Using Git ensures that all changes are logged, allowing for easy rollbacks and tracking of the development history. GitHub is used as the central repository for all project-related code and documentation. This ensures that the code is backed up and accessible from any location.

2. **Continuous Integration (CI):**

   Tool Used: GitHub Actions

   GitHub Actions automates the CI process. Whenever new code is committed, GitHub Actions initiates a build and runs tests to verify code stability and quality. This helps in early detection of issues, maintaining code quality.

3. **Continuous Deployment (CD):**

   Tool Used: Firebase Hosting and GitHub Actions

After CI verification, GitHub Actions automates the deployment process. The application is deployed to Firebase Hosting, ensuring the latest version is always available for testing and usage.

4. **Configuration Files:**

The project uses environment configuration files to manage different settings for development, testing, and production environments, ensuring seamless transitions without altering the source code. Specific configuration files (*google-services.json* for Android) are used to integrate Firebase services seamlessly with the application.

Software Tools supporting Configuration Control:

1. **Git:**

Purpose: Version control system to manage and track source code changes.

Features: Includes branching, merging, commit history tracking, and pull requests.

Usage: Regular commits are made to track progress. Feature branches are used for new developments, and pull requests are used for reviewing changes before merging into the main branch.

2. **GitHub:**

Purpose: Repository hosting service.

Features: Supports pull requests, issue tracking, project boards, and GitHub Actions for CI/CD.

Usage: Hosts the main repository containing all project-related code, documentation, and configuration files. Automates the CI/CD process via GitHub Actions.

3. **GitHub Actions:**

   Purpose: Automation tool for CI/CD processes.

   Features: Automates builds, tests, and deployments.

   Usage: Configured to trigger on code pushes and pull requests, ensuring automatic building, testing, and deployment of code.

4. **Firebase:**

   Purpose: Backend-as-a-Service (BaaS) providing database, authentication, and hosting services.

   Features: Includes Cloud Firestore for database management, Firebase Authentication for user management, and Firebase Hosting for web deployment.

   Usage: Integrates various Firebase services into the application to handle backend operations, user authentication, and deployment.

5.3.2 Version Control Procedure

To ensure the project has proper procedure in managing the source code version, several strategies have been employed in the development process.

1. **Branching Strategy:**

   New features and functionalities are developed in separate branches. Each branch is named according to the feature or issue being addressed. Bug fixes are handled in dedicated branches, ensuring that fixes can be tested and reviewed independently before merging.

2. **Commit Practices:**

Regular commits are made with clear and descriptive messages, ensuring that each commit represents a logical chunk of work. Each commit is atomic, meaning it represents a single change or addition, reducing the complexity of tracking changes.

3. **Pull Requests:**

Before merging changes into the main branch, pull requests are created. This practice helps in reviewing changes and maintaining code quality. Pull requests trigger automated tests via GitHub Actions to verify that new code does not introduce errors or break existing functionality.

4. **Merging:**

Any conflicts arising during the merging process are resolved before merging into the main branch to ensure stability. Changes are merged into the main branch only after successful code review and automated testing.

5. **Release Management:**

Each release is tagged in the Git repository, providing a snapshot of the code at the time of release. This practice helps in tracking and managing different versions of the application. Maintains a changelog documenting the features, fixes, and improvements included in each release, providing a clear history of changes.

By implementing these procedures and utilizing the specified tools, the project ensures efficient version control and configuration management, facilitating smooth and organized development.

5.4 Implementation Status

This section describes the progress and status of the development for each component/module of the project. The details include the module name, description,

duration to complete, date completed and any relevant information. For this section, I will be employing tables so that the information can be viewed and separated easily.

Table 5-1 User Authentication Module

Description	This module handles user registration, login, and authentication using Firebase Authentication.
Duration to Complete	2 weeks
Date Completed	20 April, 2024
Status	Completed and fully functional. Users can register, log in, and log out. Password recovery and validation features are also implemented.

This is the first module that was created and serves as the first thing that users see when accessing the application.

Table 5-2 Lessons Module

Description	This module manages the content delivery of lessons across different levels (Beginner, Elementary, Intermediate, Advanced).
Duration to Complete	6 weeks
Date Completed	29 May, 2024
Status	Completed and integrated. Lessons for all levels have been developed, including text content, audio playback, and interactive components.

This module is still considered for improvements as the contents can be delivered better and be improved overall. For now, this module is considered complete.

Table 5-3 Minigames Module

Description	This module includes various minigames such as Quick Quiz and Sentence Whizz to reinforce learning through interactive activities.
Duration to Complete	5 weeks
Date Completed	29 May, 2024
Status	Completed. All planned minigames are fully implemented, tested, and integrated into the application. The module supports multiple difficulty levels and tracks user progress.

The same as the Lessons Module, this module is still pending improvements as its content can be delivered better and there is room for improvement.

Table 5-4 Leaderboards Module

Description	This module displays user rankings based on points earned through lessons and minigames.
Duration to Complete	1 week
Date Completed	16 May, 2024
Status	Completed. The leaderboard updates in real-time and shows the top 50 users based on their points.

The leaderboard is another feature that utilizes game elements to incite users to compete against each other for who has the highest point.

Table 5-5 Notes Module

Description	This module allows users to create, view, and delete personal notes.
Duration to Complete	1 week
Date Completed	10 May, 2024
Status	Completed. Users can add, view, and delete notes. The notes are stored in Firebase Firestore and retrieved in real-time.

This module allows users to make their own notes. Since this is a mobile application, adding notes is a good way to let users make their own notes instead of having to rely on the traditional pen and paper method.

Table 5-6 Dictionary Module

Description	This module allows users to look up terms and words in Bahasa Malaysia, providing definitions and usage examples.
Duration to Complete	2 weeks
Date Completed	29 May, 2024
Status	Completed. Users can search for words and phrases, view definitions, and see examples of usage. The dictionary database is comprehensive and regularly updated.

This is a supplementary module where users can look up the dictionary easily for words, phrases and terms.

Table 5-7 Chatbot Module

Description	This module integrates a chatbot using the OpenAI API to assist users with their learning, answer questions, and provide additional practice.
Duration to Complete	3 weeks
Date Completed	29 May, 2024
Status	Completed. The chatbot is integrated into the application and can converse with users in both English and Bahasa Malaysia, providing contextual assistance and practice.

This module allows the user to communicate with an integrated chatbot where they can simulate a conversation in the app.

Table 5-8 Profile Settings Module

Description	This module allows users to view and modify their profile settings, including changing their password and viewing their activity history.
Duration to Complete	2 weeks
Date Completed	29 May, 2024
Status	Completed. Users can update their profile information, change their password, and view their activity history.

This is a basic module for user settings. Users can also additionally look up their activity history in the application.

Table 5-9 Activity Logging Module

Description	This module logs user activities, such as completed lessons and minigames, for progress tracking and analytics.
Duration to Complete	1 week
Date Completed	29 May, 2024
Status	Completed. All user activities are logged in Firebase Firestore, and users can view their activity history through the profile settings module.

This is an extension of the profile settings module. Activity logging is added to keep track of the user's activities.

5.5 Conclusion

This chapter has summarized the implementation status, software development and configuration setup alongside the version control procedures for the project. For the next chapter will be going over on how the testing is done for this project.

CHAPTER 6: TESTING

6.1 Introduction

The testing phase is crucial to ensure the reliability, functionality and performance of the project. The testing will be done manually and on all the modules to see if they function as they are supposed to.

6.2 Test Plan

6.2.1 Test Organization

The personnel involved in this test will be the developer of the project and some volunteers that have been given the application for testing purposes. The developer will be responsible for the entire testing process, from creating and executing test cases to maintaining them and addressing any issues that arise. The developer will also be writing unit tests, conducting integration tests, performing system tests, and ensuring that the application meets the specified requirements through rigorous testing.

6.2.2 Test Environment

The testing will be conducted in a controlled development environment using an IDE emulator and real Android devices. It will also be conducted in real-world scenarios using the application on actual devices. Regarding location, tests will be done on workstations and devices that are able to access the application without any problems.

6.2.2.1 Hardware

- Development Machine (Acer Nitro 5 AN515-55):
    - Processor: Intel(R) Core (TM) i7-10750H CPU @ 2.60GHz
    - RAM: 32GB
    - Storage: 512GB SSD
    - Operating System: 64-bit, Windows 10 x64-based processor
- Testing Devices:
    - Google Pixel Emulator configured in Android Studio to emulate various screen sizes and resolutions. Running different versions of the Android OS.
    - Samsung S10 5G. Updated to the latest stable firmware versions and configured with standard settings to replicate typical user environments.

6.2.2.2 Firmware Configuration

- Android Studio Emulator:
    - Configured to emulate various screen sizes and resolutions.
    - Running different versions of the Android OS.
- Real Devices:
    - Updated to the latest stable firmware versions.

- Configured with standard settings to replicate typical user environments.

6.2.2.3 Preparations

- Software Installation:
    - Android Studio with necessary SDKs and emulators.
    - Required dependencies and libraries for the project.
- Data Setup:
    - Sample data for testing different scenarios (user profiles, lesson content, etc.)
- Test Scripts:
    - Pre-written test scripts for manual testing based on the application's functional requirements.

6.2.2.4 Training:

- Volunteer User Training:
    - Basic guidance through app navigation.
- Developer Training:
    - Familiarization with the manual testing procedures and tools.
    - Training in writing effective test cases and interpreting test results.

6.2.3 Test Schedule

Since this is a mobile application, the tests are scheduled to be done immediately after the implementation of each feature or module. These tests will be done on both the emulator and real devices to ensure that appropriate results are collected.

Each testing cycle will involve:

- Cycle 1: Initial Testing - Conducted immediately after feature implementation to identify any immediate issues. Duration: 1 day per feature/module.

- Cycle 2: Bug Fixing and Retesting - After initial bugs are fixed, the feature will be retested to ensure all issues are resolved. Duration: 1-2 days per feature/module, depending on the complexity of the fixes.

- Cycle 3: Integration Testing - Once individual features are tested and validated, they will be integrated into the overall system and tested to ensure seamless operation. Duration: 2-3 days per integration phase.

Overall, each feature or module will go through multiple cycles of testing over a span of approximately 3-6 days to ensure comprehensive evaluation and validation.

6.3 Test Strategy

The testing strategy for this project will involve a black box testing approach. This strategy is selected to ensure that the system functions correctly from an end-user perspective without going into its internal code structure. Black-box testing focuses on validating input and output, ensuring that the application behaves as expected. This approach involves testing the functionality of the application by providing inputs and examining the outputs without considering the internal workings of the application. This ensures that the application meets the specified requirements and behaves correctly from the user's perspective.

6.3.1 Classes of Tests

Two main classes of tests will be conducted to ensure the application's robustness and reliability:

- **Output Correctness/Functionality Test**: These tests will verify that the application's output matches the expected results. The following modules will be tested for functionality:

    - **Lessons Module**: Ensure that lessons are correctly loaded, navigated, and that user progress is tracked accurately.

    - **Chatbot Module**: Ensure that the chatbot responds accurately to user queries and provides correct information.

    - **Mini Games Module**: Ensure that mini-games are correctly loaded, playable, and provide accurate feedback to users based on their performance.

- **Security Test**: Security tests will be conducted to identify vulnerabilities within the application. The following modules will be tested for security:

    - **Registration Module**: Ensure that user registration is secure, data is encrypted, and no unauthorized access is allowed.

    - **Login Module**: Ensure that user login is secure, data is encrypted, and proper authentication measures are in place to prevent unauthorized access.

6.4 Test Design

6.4.1 Test Description

The test cases have been designed for each module to ensure that the application meets the specified requirements. The test cases will verify the functionality and security of the application requirements.

6.4.2 Test Data

The data will be taken from tests done on both the emulator and real-life devices.

6.5 Test Results and Analysis

The test results have been gathered into the tables related to their modules below,

Figure 6-1 Interface for Lessons Module.

The test has been done mainly on the flow of the Lessons Module. Each test done will also include checking the weekly activities module like in the figure below to make sure the modules run as intended.

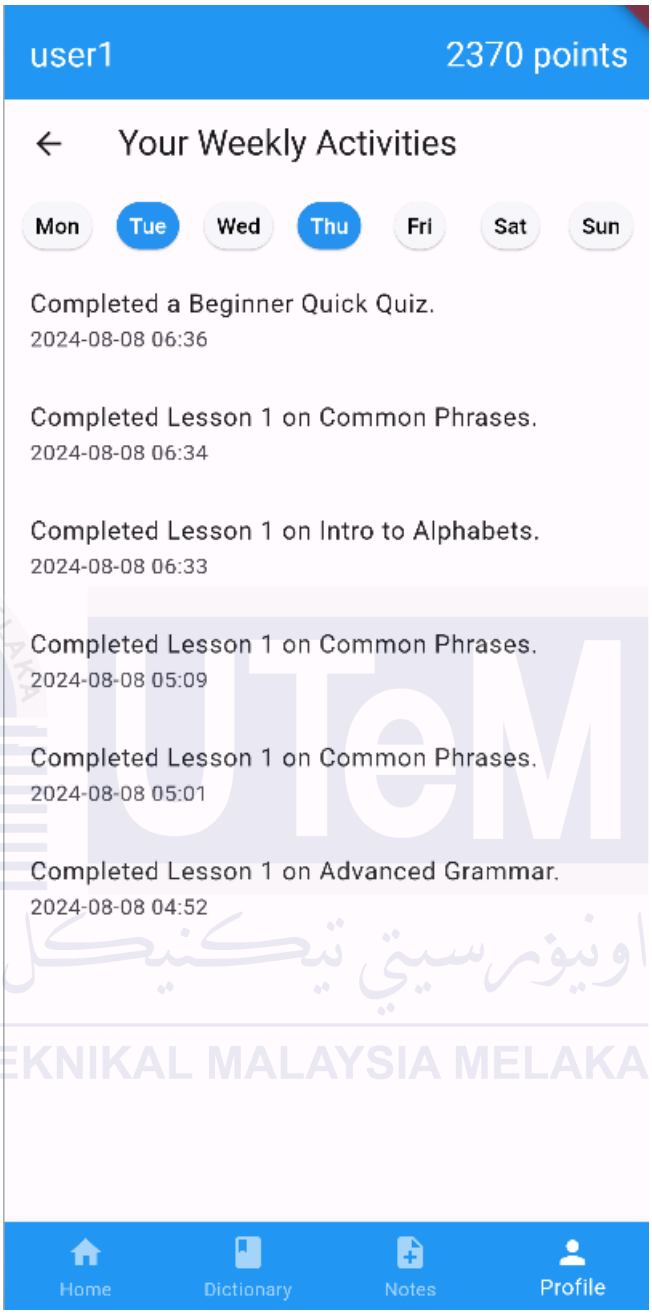

Figure 6-2 Interface for Weekly activities.

Each time a lesson is done, the page seen in the figure above will be checked to make sure that the activity is recorded.

Table 6-1 Lessons Module Test Cases

Test Case ID	Module	Tester	Description	Test Data	Test Steps	Expected Result	Actual Result	Status
Precondition: User is logged in and has internet access.								
TC-LESSON-01	Lessons	Volunteer	Testing "Back" button	Button clicks	1. Click on the back button.	Navigates back to Main Menu page	Navigation to Main Menu page successful	Success
TC-LESSON-02	Lessons	Volunteer	Testing Introduction button	Button clicks	1. Click on the Introduction button	Navigates to Introduction Lesson	Navigation to Introductions lesson successful	Success
TC-LESSON-03	Lessons	Volunteer	Testing Lessons flow	Lesson I of Introduction to	1. Swipe through the lessons	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards	Success

				Alphabets, Beginner	until the final page. 2. Click on the finish lesson button.		given after lesson is finished	
TC-LESSON-04	Lessons	Volunteer	Testing Lessons flow	Lesson II of Introduction to Alphabets, Beginner	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-05	Lessons	Volunteer	Testing Lessons flow	Lesson II of Introduction to Alphabets, Beginner	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-06	Lessons	Volunteer	Testing Lessons flow	Lesson III of Introduction to Alphabets, Beginner	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-07	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Introduction to Alphabets, Beginner	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-08	Lessons	Volunteer	Testing Lessons flow	Lesson I of Numbers and Counting, Beginner	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-09	Lessons	Volunteer	Testing Lessons flow	Lesson II of Numbers and Counting, Beginner	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-10	Lessons	Volunteer	Testing Lessons flow	Lesson III of Numbers and Counting, Beginner	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-11	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Numbers and Counting, Beginner	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-12	Lessons	Volunteer	Testing Lessons flow	Lesson I of Common Phrases, Beginner	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-13	Lessons	Volunteer	Testing Lessons flow	Lesson II of Common Phrases, Beginner	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-14	Lessons	Volunteer	Testing Lessons flow	Lesson III of Common Phrases, Beginner	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-15	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Common Phrases, Beginner	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-17	Lessons	Volunteer	Testing Audio button	Lesson I of Common Phrases, Beginner	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Audio is able to be heard	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-18	Lessons	Volunteer	Testing Lessons flow	Lesson I of Basic Grammar Rules, Elementary	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-19	Lessons	Volunteer	Testing Lessons flow	Lesson II of Basic Grammar Rules, Elementary	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-20	Lessons	Volunteer	Testing Lessons flow	Lesson III of Basic Grammar Rules, Elementary	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-21	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Basic Grammar Rules, Elementary	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-22	Lessons	Volunteer	Testing Lessons flow	Lesson I of Everyday Conversations, Elementary	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-23	Lessons	Volunteer	Testing Lessons flow	Lesson II of Everyday Conversations, Elementary	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-24	Lessons	Volunteer	Testing Lessons flow	Lesson III of Everyday Conversations, Elementary	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-25	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Everyday Conversations, Elementary	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-26	Lessons	Volunteer	Testing Lessons flow	Lesson I of Cultural Insights and Etiquette, Elementary	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-27	Lessons	Volunteer	Testing Lessons flow	Lesson II of Cultural Insights and Etiquette, Elementary	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-28	Lessons	Volunteer	Testing Lessons flow	Lesson III of Cultural Insights and Etiquette, Elementary	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-29	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Cultural Insights and Etiquette, Elementary	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-30	Lessons	Volunteer	Testing Lessons flow	Lesson I of Intermediate Conversational Skills, Intermediate	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-31	Lessons	Volunteer	Testing Lessons flow	Lesson II of Intermediate Conversational Skills, Intermediate	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-32	Lessons	Volunteer	Testing Lessons flow	Lesson III of Intermediate Conversational Skills, Intermediate	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-33	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Intermediate Conversational Skills, Intermediate	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-34	Lessons	Volunteer	Testing Lessons flow	Lesson I of Grammar in Context, Intermediate	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-35	Lessons	Volunteer	Testing Lessons flow	Lesson II of Grammar in Context, Intermediate	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-36	Lessons	Volunteer	Testing Lessons flow	Lesson III of Grammar in Context, Intermediate	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-37	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Grammar in Context, Intermediate	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-38	Lessons	Volunteer	Testing Lessons flow	Lesson I of Advanced Grammar and Syntax, Advanced	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-39	Lessons	Volunteer	Testing Lessons flow	Lesson II of Advanced Grammar and Syntax, Advanced	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-40	Lessons	Volunteer	Testing Lessons flow	Lesson III of Advanced Grammar and Syntax, Advanced	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-41	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Advanced Grammar and Syntax, Advanced	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-42	Lessons	Volunteer	Testing Lessons flow	Lesson I of Idiomatic Expressions and Slang, Advanced	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-43	Lessons	Volunteer	Testing Lessons flow	Lesson II of Idiomatic Expressions and Slang, Advanced	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-44	Lessons	Volunteer	Testing Lessons flow	Lesson III of Idiomatic Expressions and Slang, Advanced	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-45	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Idiomatic Expressions and Slang, Advanced	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-46	Lessons	Volunteer	Testing Lessons flow	Lesson I of Business and Formal Language, Advanced	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-47	Lessons	Volunteer	Testing Lessons flow	Lesson II of Business and Formal Language, Advanced	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

TC-LESSON-48	Lessons	Volunteer	Testing Lessons flow	Lesson III of Business and Formal Language, Advanced	1. Swipe through the lessons until the final page. 2. Click on the finish lesson button.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success
TC-LESSON-49	Lessons	Volunteer	Testing Lessons flow	Lesson IV of Business and Formal Language, Advanced	1. Swipe through the lessons until the final page.	Able to navigate to and finish a Lesson	Lesson finished successfully. Rewards given after lesson is finished	Success

					2. Click on the finish lesson button.			
TC-LESSON-50	Lessons	Volunteer	Testing navigation to Dictionary Page while in Lesson	Button clicks	1. Click on the Dictionary button.	Able to navigate to Dictionary page while in Lesson	Navigation successful.	Success
TC-LESSON-51	Lessons	Volunteer	Testing navigation to Notes Page while in Lesson	Button clicks	1. Click on the Notes button.	Able to navigate to Notes page while in Lesson	Navigation successful.	Success

TC-LESSON-52	Lessons	Volunteer	Testing navigation to Profile Page while in Lesson	Button clicks	1. Click on the Profile button.	Able to navigate to Profile page while in Lesson	Navigation successful.	Success

The test results for the Lessons Module show that the key features are working well, with lessons loading properly, smooth navigation, correct rewards distribution, and audio playback functioning without issues.

Next are the results of the chatbot module test cases.

Figure 6-3 Interface for Chatbot Module.

The test cases on the chatbot module will mainly be focused on testing the functionality and response of the integrated chatbot assistant.

Table 6-2 Chatbot Module Test Cases

Test Case ID	Module	Tester	Description	Test Data	Test Steps	Expected Result	Actual Result	Status
Precondition: User is logged in and has internet access								
TC-CHATBOT-01	Chatbot	Volunteer	Verify chatbot responds accurately	Prompt: "Apakah maksud berbahas?"	1. Send a prompt to the chatbot using the message text field.	Chatbot provides correct and relevant information	Chatbot responded accurately to user queries	Success
TC-CHATBOT-02	Chatbot	Volunteer	Ask the chatbot to talk about topics other than studying Bahasa Malaysia.	Prompt: "Who is the president of	1. Send a prompt to the chatbot using the message text field.	Chatbot rejects the query, asking the user to talk about studying Bahasa Malaysia.	Chatbot rejects the query and asks the user to talk only about the relevant topic.	Success

				the USA?"				
TC-CHATBOT-03	Chatbot	Volunteer	Ask the chatbot to speak in Malay	Prompt: "Boleh bercakap Bahasa Malaysia?"	1. Send a prompt to the chatbot using the message text field.	Chatbot responds in Malay	Chatbot responds in Malay	Success

The Chatbot Module performed as expected in both test cases, successfully responding accurately to user queries and communicating in Malay when prompted. These results indicate that the chatbot is functioning well and can provide relevant and accurate responses in the language specified, ensuring it meets the needs of the users effectively.

Continuing are the Mini Game Module test cases.

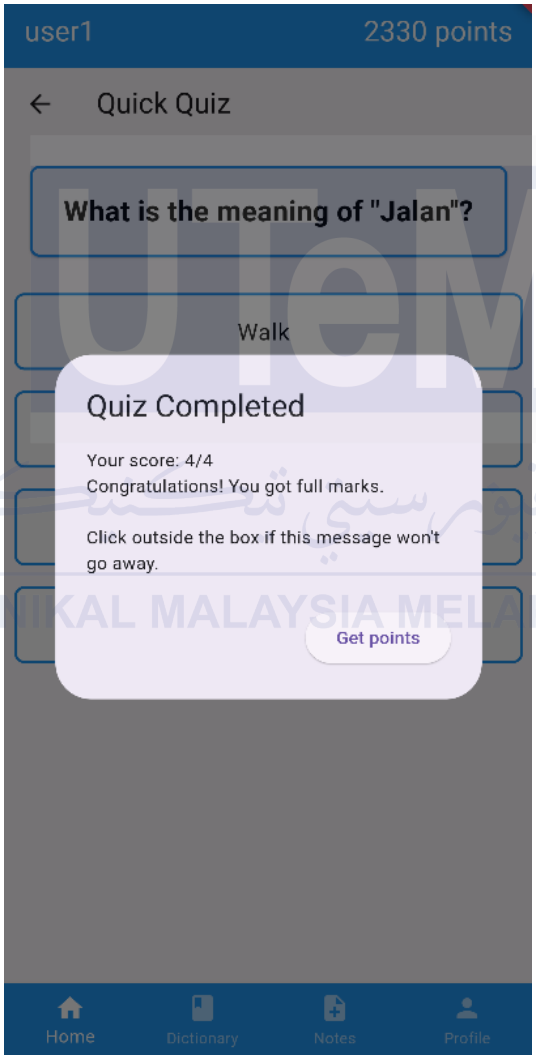

Figure 6-4 Results of a successful Mini Games Module flow.

The Mini Games Module test cases are done to check the functionality of the minigames. The tests go through the failure and successful flows of the module.

Table 6-3 Mini Games Module Test Cases

Test Case ID	Module	Tester	Description	Test Data	Test Steps	Expected Result	Actual Result	Status
Precondition: User is logged in and has internet access.								
TC-MINIGAME-01	Mini Games	Developer	Testing correct Quick Quiz Game flow on Beginner difficulty.	Question 1: Eat Question 2: Sleep Question 3: Drink Question 4: Walk	1. Click on the answer buttons. 2. Click on the submit	Points given	Points are given out	Success

					button.			
TC-MINIGAME-02	Mini Games	Developer	Testing incorrect Quick Quiz Game flow on Beginner difficulty.	Question 1: Drink Question 2: Sleep Question 3: Drink Question 4: Walk	1. Click on the answer buttons. 2. Click on the submit button.	Points not given.	Points not given out. Asked to repeat the quiz	Success

TC-MINIGAME-03	Mini Games	Developer	Testing correct Quick Quiz Game flow on Elementary difficulty.	Question 1: What did you eat this morning? Question 2: How do you go to school? Question 3: What color do you like? Question 4: Who is your teacher?	1. Click on the answer buttons. 2. Click on the submit button.	Points given	Points are given out	Success

TC-MINIGAME-04	Mini Games	Developer	Testing incorrect Quick Quiz Game flow on Elementary difficulty.	Question 1: Where did you go this morning? Question 2: How do you go to school? Question 3: What color do you like? Question 4: Who is your teacher?	1. Click on the answer buttons. 2. Click on the submit button.	Points not given.	Points not given out. Asked to repeat the quiz	Success
TC-MINIGAME-05	Mini Games	Developer	Testing correct Quick	Question 1: It enhances job	1. Click on the	Points given	Points are given out	Success

			Quiz Game flow on Intermediate difficulty.	opportunities Question 2: I stay indoors and read	answer buttons. 2. Click on the submit button.			
TC-MINIGAME-06	Mini Games	Developer	Testing incorrect Quick Quiz Game flow on Intermedi	Question 1: It enhances job opportunities Question 2: I sunbathe	1. Click on the answer buttons.	Points not given.	Points not given out. Asked to repeat the quiz	Success

			ate difficulty.		2. Click on the submit button.			
TC-MINIGAME-07	Mini Games	Developer	Testing correct Quick Quiz Game flow on Advanced difficulty.	Question 1: Ketepatan ucapan Question 2: Makna literal Question 3: Ekspresi tulus Question 4: Kata berlawanan	1. Click on the answer button s. 2. Click on the submit	Points given	Points are given out	Success

				Question 5: Pernyataan langsung	button.			
TC-MINIGAME-08	Mini Games	Developer	Testing incorrect Quick Quiz Game flow on Advanced difficulty.	Question 1: Gaya seni Question 2: Makna literal Question 3: Ekspresi tulus Question 4: Kata berlawanan Question 5:	1. Click on the answer buttons. 2. Click on the submit	Points not given.	Points not given out. Asked to repeat the quiz	Success

				Pernyataan langsung	button.			
TC-MINIGAME-09	Mini Games	Developer	Testing correct Sentence Whizz game flow on Beginner difficulty.	"Saya belajar Bahasa Malaysia."	1. Click on the answer button. 2. Click on the submit	Points given	Points are given out	Success

					button.			
TC-MINIGAME-10	Mini Games	Developer	Testing incorrect Sentence Whizz game flow on Beginner difficulty	"Saya belajar Malaysia Bahasa."	1. Click on the answer buttons. 2. Click on the submit button.	Points not given.	Points not given out. Asked to repeat the quiz	Success
TC-MINIGAME-11	Mini Games	Developer	Testing correct	"Saya akan membantu	1. Click	Points given	Points are given out	Success

			Sentence Whizz game flow on Elementary difficulty.	anda memahami bahasa ini."	on the answer buttons. 2. Click on the submit button.			
TC-MINIGAME-12	Mini Games	Developer	Testing incorrect Sentence Whizz game flow on Elementar	"Saya membantu akan anda memahami bahasa ini."	1. Click on the answer	Points not given.	Points not given out. Asked to repeat the quiz	Success

			y difficulty		buttons. 2. Click on the submit button.			
TC-MINIGAME-13	Mini Games	Developer	Testing correct Sentence Whizz game flow on Intermediate difficulty.	"Saya suka warna biru."	1. Click on the answer buttons. 2. Click	Points given	Points are given out	Success

					on the submit button.			
TC-MINIGAME-14	Mini Games	Developer	Testing incorrect Sentence Whizz game flow on Intermediate difficulty	"Saya suka biru warna."	1. Click on the answer buttons. 2. Click on the submit	Points not given.	Points not given out. Asked to repeat the quiz	Success

					button.			
TC-MINIGAME-15	Mini Games	Developer	Testing correct Sentence Whizz game flow on Advanced difficulty.	"Mempelajari Bahasa Malaysia dapat memperkaya pemahaman budaya."	1. Click on the answer buttons. 2. Click on the submit button.	Points given	Points are given out	Success
TC-MINIGAME-16	Mini Games	Developer	Testing incorrect	"Mempelajari	1. Click	Points not given.	Points not given out. Asked to repeat the quiz	Success

			Sentence Whizz game flow on Advanced difficulty	Malaysia Bahasa dapat memperkaya pemahaman budaya."	on the answer buttons. 2. Click on the submit button.			

The analysis for the Mini Games module test cases indicates that the module generally functions as expected. The test cases confirmed that the mini-games load successfully and are interactive, with correct responses and scoring mechanisms in place. The persistence of game states was also verified, ensuring that the app can resume from where it left off without losing progress. No significant issues were encountered during the testing, and all tests passed, indicating that the Mini Games module is stable and performs reliably under normal usage conditions. This outcome suggests that the module is ready for deployment with minimal to no additional modifications required.

Next is the Login Module test cases.

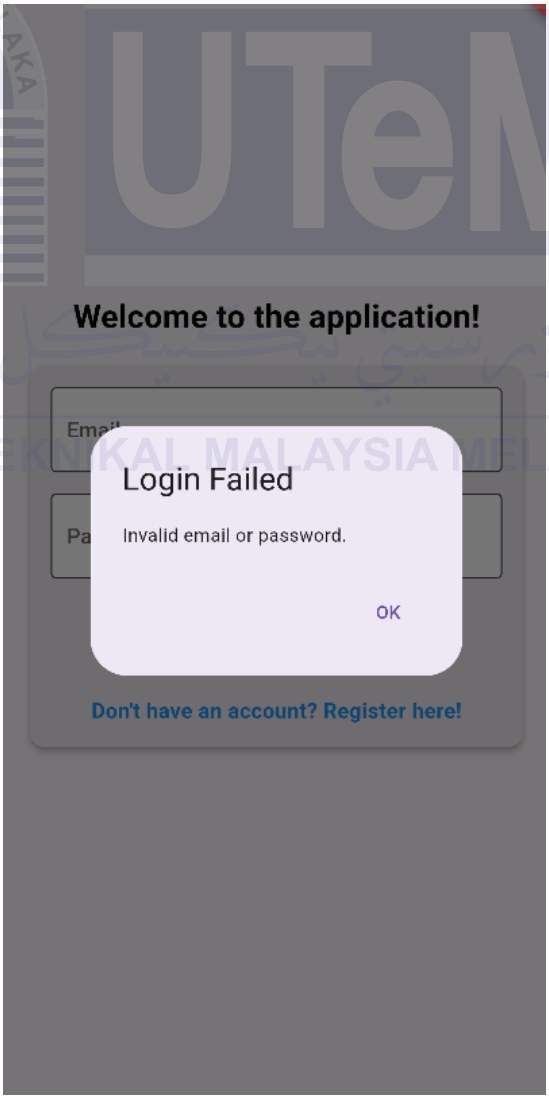

Figure 6-5 An unsuccessful attempt flow on the Login Module.

Table 6-4 Login Module Test Cases

Test Case ID	Module	Tester	Description	Test Data	Test Steps	Expected Result	Actual Result	Status
Precondition: User has internet access.								
TC-LOGIN-01	Login	Developer	Verify secure login and authentication	Email: user1@email.com Password: password123	1. Fill in the required text fields. 2. Click on the Login button.	Login successful with correct credentials, no unauthorized access	Login successful with proper authentication	Success
TC-LOGIN-02	Login	Developer	Verify secure login and authentication	Email: user1111@email.com Password: password1234	1. Fill in the required text fields. 2. Click on the Login button.	Unauthorized access	Login unsuccessful and error message is shown	Success

TC-LOGIN-03	Login	Developer	Verify empty data error handling	No data entered in fields	1. Click on the Login button.	Unauthorized access	Login unsuccessful and error message is shown	Success
TC-LOGIN-04	Login	Developer	Testing without internet connection	Email: user1@email.com Password: password123	1. Fill in the required text fields. 2. Click on the Login button.	Unable to login	Unable to login	Success
TC-LOGIN-05	Login	Developer	Testing page to registration	Button click	1. Click on the "Don't have an account? Register here!" text, written in blue.	Able to navigate to registration page	Navigated to registration page	Success

The analysis of the Login module test cases shows that the module performs well in ensuring secure and functional user authentication. The tests validated that users with correct credentials can successfully log in, while unauthorized access is appropriately blocked. Error handling was also effective, with the system correctly displaying error messages when no data was provided or when login attempts were made without an internet connection. Additionally, the module handled invalid credentials properly by preventing access and showing relevant error messages. Overall, the Login module passed all test cases, demonstrating that it is secure, reliable, and ready for deployment.

Lastly are the Registration Module test cases.

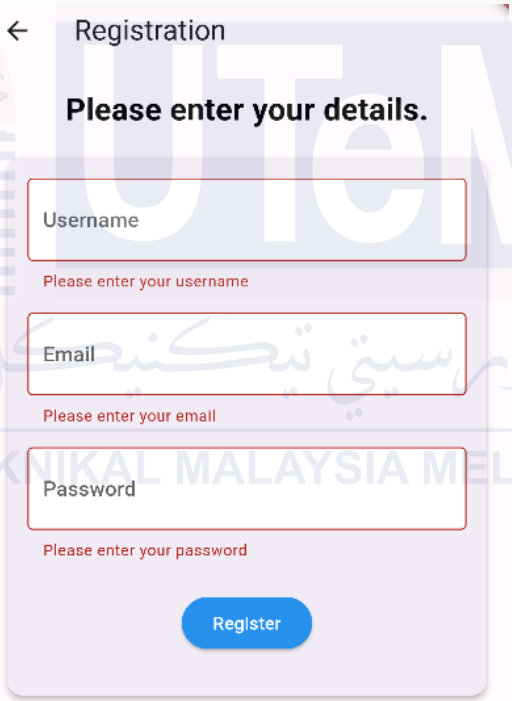

Figure 6-6 A blank field error handling on Registration Module.

The above figure shows an instance of a test case that focuses on error handling.

Table 6-5 Registration Module Test Cases

Test Case ID	Module	Tester	Description	Test Data	Test Steps	Expected Result	Actual Result	Status
Precondition: User has internet access.								
TC-REGIS-01	Registration	Developer	Verify registration and validation	Username: TestRegUser Email: testreguser@email.com Password: password123	1. Fill in the required text fields 2. Click on the Register button.	Successful registration.	Registration successful and notification of registration is shown	Success

TC-REGIS-02	Registration	Developer	Testing blank fields error handling	No data filled	1. Click on the Register button.	Error message shown	Error message shown asking to fill in fields	Success
TC-REGIS-03	Registration	Developer	Testing email format error handling	Invalid email format	1. Fill in the required text fields 2. Click on the Register button.	Error message shown	Error message shown asking to fill in proper email format	Success
TC-REGIS-04	Registration	Developer	Testing navigation to login page	Button click	1. Click on the "back" arrow at the top.	Navigates back to login page	Navigated back to login page	Success

The analysis of the Registration module test cases indicates that the module is effective in handling user registration securely and accurately. The test cases confirmed that the system successfully registers new users with valid credentials, ensuring that proper data validation is in place. Additionally, the system responded correctly to scenarios where required fields were left empty or where invalid email formats were used, by displaying appropriate error messages. All test cases for the Registration module passed, confirming that the module is well-implemented, with robust validation mechanisms to ensure the integrity and security of user registration.

Overall, the testing confirmed that the core functionalities of the application are working as expected. The overall satisfaction from the volunteer users and developer's perspective were satisfactory. Further improvements will be made to the application as the project goes into its final development phases.

6.6 Conclusion

In this chapter, various components and modules of the application were tested. These tests were performed to verify the correctness, functionality, and security of the application. The results indicated that most modules are functioning correctly, with minor issues identified in the lessons module, which will be addressed in the next development phase. For the next chapter, the conclusion regarding this project will be made.

CHAPTER 7: PROJECT CONCLUSION

7.1 Introduction

From the development of this project, several strengths and weaknesses have been identified. As of now, these are the identified attributes.

Strengths:

- **Easy to Use and Access:** The application's design as a mobile app makes it incredibly user-friendly and accessible. Users can learn at their convenience, whether they are on the go or at home, without the need for a desktop or other cumbersome setups. This portability ensures that users can engage with the content anytime and anywhere, which is especially beneficial for those with busy schedules or for learners who prefer a flexible approach to studying.

- **Good for Self-Learners:** The app's structure is particularly advantageous for self-learners who prefer to learn at their own pace without the constraints of scheduled classes. The absence of imposed schedules allows users to revisit lessons as needed, making it ideal for those who may need more time to grasp certain concepts. This autonomy in learning helps cater to individual learning styles and paces.

- **Chatbot Integration:** One of the standout features of the application is the integration of a chatbot. This feature provides real-time assistance and answers to user queries, significantly enhancing the learning

experience. The chatbot acts as a tutor available 24/7, helping to bridge the gap between self-learning and needing external help. It also adds an interactive element that keeps users engaged and motivated.

- **Emphasis on Cultural Context:** The application goes beyond just teaching the language by incorporating cultural context into the lessons. This approach not only aids in understanding the language but also helps users appreciate the cultural nuances and practices associated with it. By doing so, the app provides a more holistic learning experience, ensuring that users can communicate more effectively and appropriately in real-life situations.

Weaknesses:

- **Basic Features:** While the application serves its purpose well, the range of features is somewhat basic. It covers the essential aspects of language learning but may not meet the needs of more advanced learners who are looking for a comprehensive tool that includes more sophisticated functionalities such as advanced grammar exercises, interactive speaking modules, or language immersion features. This limitation could make the app less appealing to users who are seeking a more in-depth learning experience.

- **Lesson Structure Improvement:** Although the current lesson structure is functional, there is room for enhancement. Lessons could be more dynamic and varied to maintain user interest and accommodate different learning styles. Incorporating multimedia elements, such as videos or interactive quizzes, could make the lessons more engaging. Additionally, providing more progressive difficulty levels within the lessons could help users steadily build their skills without feeling overwhelmed or bored.

7.2 Propositions for Improvements

Continuing from before, the proposition for improvements is based upon the already existing features and enhancing it based on this focus. As such, improvements could be considered in the form of:

- Further improvements to existing features.

One key area for enhancement is refining the existing features to offer more value to users. While the current features are effective, they could be expanded and improved to provide a more comprehensive learning experience. For example, the lessons could include more diverse content such as advanced language exercises and additional cultural context. Additionally, making the user interface more intuitive and adding customization options would allow users to tailor the learning experience to their needs and preferences.

- Improving chatbot capabilities.

The chatbot is already a useful tool, but there is potential to make it even more effective. Enhancing the chatbot's natural language processing capabilities would enable it to handle more complex queries and provide more detailed and accurate responses. Furthermore, integrating AI-driven personalization could allow the chatbot to offer customized lesson suggestions based on user progress and preferences, making it a more integral part of the learning process.

- Adding more gamification features

Introducing additional gamification elements could significantly boost user engagement and motivation. Features like profile pictures, badges, and trophies could be awarded for achieving certain milestones, such as completing lessons or maintaining a learning streak. These visual rewards would provide users with a sense of accomplishment and encourage continued use of the app. Moreover, implementing leaderboards or social challenges could foster a sense of community and friendly competition among users.

- Continuous lesson improvements

While the current lessons are well-structured, there is always room for ongoing enhancement. Regularly updating the content to reflect the latest language trends and cultural developments would keep the lessons relevant and engaging. Additionally, incorporating user feedback into the lesson design process could help identify areas for improvement and ensure that the content meets the evolving needs of learners. Continuous improvements to the lesson structure and content would ensure that the app remains a valuable resource for language learners at all levels.

7.3 Project Contribution

The primary contribution of this project lies in its value to individual learners who are interested in acquiring proficiency in Bahasa Malaysia. The mobile application provides a convenient, accessible platform for self-paced learning, catering to users who may not have the time or resources to engage in formal language classes. By focusing on user-friendly design and incorporating features like a chatbot and culturally relevant content, the project supports individuals in their language learning journey, enabling them to build both linguistic and cultural competence. The application also introduces innovative learning methods, such as gamification and chatbot integration, which enhance user engagement and motivation. These features help make language learning more interactive and enjoyable, encouraging continuous use and practice. For individuals looking to learn Bahasa Malaysia independently, this project offers a practical and effective solution that adapts to their personal schedules and learning styles.

7.4 Conclusion

At the end of this project, all the established objectives have been met. Regarding personal satisfaction and standards, this project will always need improvements. While making a language learning application seems deceptively easy, the true challenge lies in the content and delivery. As a conclusion, this project is finished but the improvements to be made are never-ending.

REFERENCES

1. Alizah, L. & Zamri, M. (2015). Student-Centered Learning in Teaching and Learning of Bahasa Melayu in Higher Education: Implementation and Acceptance Students. *Jurnal Personalia Pelajar*, 18(1), 1-9.
2. Zamri, M., Nik Mohd Rahimi, N. Y., Mohamed Amin, E., Jamaludin, B., & Sharala, S. (2014). Malay Language Learning Strategies among Foreign Students Based on Language Skills and Grade. *Jurnal Personalia Pelajar*, 17, 75-81.
3. Mahamod, Z., et al. (2014). The Impact of Language Learning Strategies on Proficiency and Grades Among Foreign Students. *Jurnal Pendidikan Bahasa Melayu*, 9(1), 12-20.
4. Muhd Zulkifli, N. Z., & Abu Bakar, S. S. (2016). Hubungan Antara Strategi Pembelajaran Bahasa Melayu Dengan Motivasi Dalam Kalangan Pelajar Di Jerman: The Relationship between Malay Language Learning Strategies and Motivation among Students in Germany. *PENDETA*, 7, 85–94. Retrieved from https://ejournal.upsi.edu.my/index.php/PENDETA/article/view/1205
5. Norfaizal, J., Samsur Rijal, Y., & Muhammad Saiful Haq, H. (2022). Strategi Pembelajaran Kosa Kata Bahasa Melayu dalam Kalangan Pelajar Antarabangsa di Universiti Malaya. *Jurnal Pendidikan Bahasa Melayu – Malay Language Education Journal*, 12(2), 125-134.