

IM3180 Design and Innovation Project

(AY2023/24 Semester 1)

Project Report

Title: Duelingo

Github: <https://github.com/Suu-ly/Duelingo>

Submitted by: Group 7

Choy Zi Jie Victor

Gems Seah Gui Yang

Jeremy Lee Qi Jie

Josiah Chua Ing Peng

Leong Hong Wei

Lim Xian Long

Pua Jun How Austin

Shental Ang

Wong Yu Hao

Yeo Chan Hoang Lance

Supervisor: Assoc Prof Teh Kah Chan

Chapter 1: Introduction.....	3
1.1: Background Research.....	3
Chapter 2: Objective and Scope.....	5
Chapter 3: Literature Review.....	6
3.1: Duolingo.....	6
3.2: Other Applications On The Market.....	8
3.3: Review.....	12
Chapter 4: Design and Implementation.....	13
4.1: Considerations.....	13
4.1.1: Feature Considerations.....	13
4.1.2: Frontend Software Consideration.....	15
4.1.3: Backend Software Consideration.....	16
4.2: Final Design and Implementation.....	19
4.2.1 Frontend Development.....	19
4.2.1.1: Visual Development and Testing using Figma.....	19
4.2.1.2: Building the Frontend Using React Native.....	20
4.2.2: Backend Development.....	21
4.2.2.1: Authentication.....	21
4.2.2.2: Storing of User Information.....	22
4.2.2.3: Storing of Questions.....	23
4.2.2.4: Storing of Multiplayer Sessions.....	23
4.3: Main Features of the Application.....	24
4.3.1: Quiz.....	24
4.3.2: Multiplayer.....	24
4.3.3: Lives.....	25
4.3.4: Leaderboard.....	25
4.3.5: Friends.....	26
4.3.6: Home Screen.....	27
4.3.7: Text To Speech.....	27
Chapter 5: Conclusion and Recommendation.....	28
5.1: Conclusion.....	28
5.2: Recommendations for Future Works.....	28
5.2.1: AI Handwriting Analysis.....	28
5.2.2: AI Voice Recognition Pronunciation Analysis.....	28
Chapter 6: Screenshots of Application.....	30
Chapter 7: Resources.....	32

Chapter 1: Introduction

The introduction of technology such as mobile applications has transformed the way we learn new languages. Nearly 1.5 billion people globally are learning a new language [1]. The global language learning market size was estimated to be worth USD 52 Billion in 2022 and has a CAGR of 20% between 2023 and 2032 indicating USD 337.2 Billion [2]. This indicates that there is a considerable market and demand for these applications.

Gamification, the use of game design features outside of games, has emerged as a valuable tool in a variety of sectors, including language learning. It takes use of inherent human propensity for play and competition and turns the challenging work of language learning into an engaging and enjoyable experience. When gamification is applied effectively, it can dramatically improve the language learning process by making it more interesting, entertaining and ultimately, more effective.

1.1: Background Research

Language acquisition is a multidimensional process that combines cognitive, emotive and social elements. Current traditional language learning strategies such as textbooks and flashcards lacks the social and interactive aspects of learning. For many students, this made the procedure dull and repetitious. Furthermore, these alternatives are often confined by time, place, and money, making genuine and diversified language access challenging for learners. As a result, there is a need for a more creative and effective technique to engage learners in language acquisition, which might also give them additional opportunities for social involvement and collaboration.

One potential option is to use technology, particularly mobile applications to improve language learning. Convenience, accessibility, flexibility, personalization and interactivity are just some of the benefits of mobile application over traditional methods. Mobile apps enable students to study whenever and wherever they choose, at their own speed, level, and preferences, while getting quick feedback and incentives. Audio, videos, and graphics are examples of multimedia components that may be used to make language learning more interesting and inspiring. The

following are some characteristics that might make the learning language application more appealing to users:

1. A clear and relevant learning objective that guides the learner's progress and provides a sense of achievement.
2. A variety of learning activities that could challenge the learner's skills and knowledge, and offer differing modes of input and output, such as listening, speaking, reading and writing.
3. A robust feedback system that gives the learners timely and constructive feedback on their performance and areas of improvement.
4. Reward system that recognizes learner's efforts and accomplishments. Incentivising them to continue learning. This would include points, levels, badges, leaderboards and certificates.
5. Story or narrative that contextualises the language content, creating a more immersive and emotional connection.
6. Social or multiplayer component that allows the user to interact and collaborate with other learners and native speakers, fostering a sense of community and competition

Gamification boost learners' motivation, engagement, and retention by providing a pleasant and fun learning experience while also meeting their intrinsic and extrinsic demands, such as autonomy, competence, and relatedness. Gamification also help students improve their cognitive and metacognitive abilities, such as problem-solving, critical thinking, creativity, and self-regulation, by exposing them to challenging and meaningful activities and giving scaffolding and feedback.

There is a market for a language learning application with gamification and multiplayer elements, since these features may make the program more enticing to prospective users and help it stand out from the crowd. According to Grand View Research (2020), they highlighted some of the important market growth factors and trends, such as:

1. The globalised world's rising need for multilingualism and cross-cultural communication, particularly in business, tourism, education, and diplomacy.
2. The growing popularity and accessibility of online and mobile language learning platforms, particularly among younger and more tech-savvy generations that desire flexible, accessible, and customised learning experiences.
3. The incorporation of modern technology, such as artificial intelligence, natural language processing, voice recognition, and augmented and virtual reality, into language learning systems in order to improve efficiency, efficacy, and interaction.
4. Adoption of creative and engaging pedagogical techniques with language learning platforms, such as gamification, storytelling, and social learning, to boost student motivation, retention, and satisfaction.

Finally, this background and motivation lay the groundwork for the proposed innovation in language learning applications, which aims to not only improve the effectiveness of language acquisition but also to transform the learning process into a collaborative and socially engaging experience.

Chapter 2: Objective and Scope

The objective of this application, Duelingo, is to provide a gamified and multiplayer learning platform that can help users learn a new language in a fun and engaging way. It does this by mimicking the features of Duolingo, such as points, levels, and leaderboards to motivate and reward users on their language learning progress. Duelingo also adds a storyline that teaches the vocabulary of the new word first, before reinforcing the knowledge through short tests, creating an immersive and emotional connection that would help them learn the language in a more meaningful context. Lastly, a multiplayer feature that allows users to connect and duel with each other, enabling user interaction and creating a better user experience. This would foster a sense of community and competition among the users, and provide them with opportunities to practice and apply their language skills.

Chapter 3: Literature Review

In this section, we examine the limitations of existing measures to enhance and improve these features.

3.1: Duolingo

Duolingo is an educational technology company based in the United States that develops learning apps and offers language certification. Duolingo offers lessons on music, maths, and over 30 languages, like English, French, and Spanish to less typically studied languages such as Welsh, Irish, and Swahili.

Duolingo takes a gamified approach to language learning, with lessons that feature translating, interactive exercises, quizzes, and stories to make learning more engaging and fun as shown in the figure below.

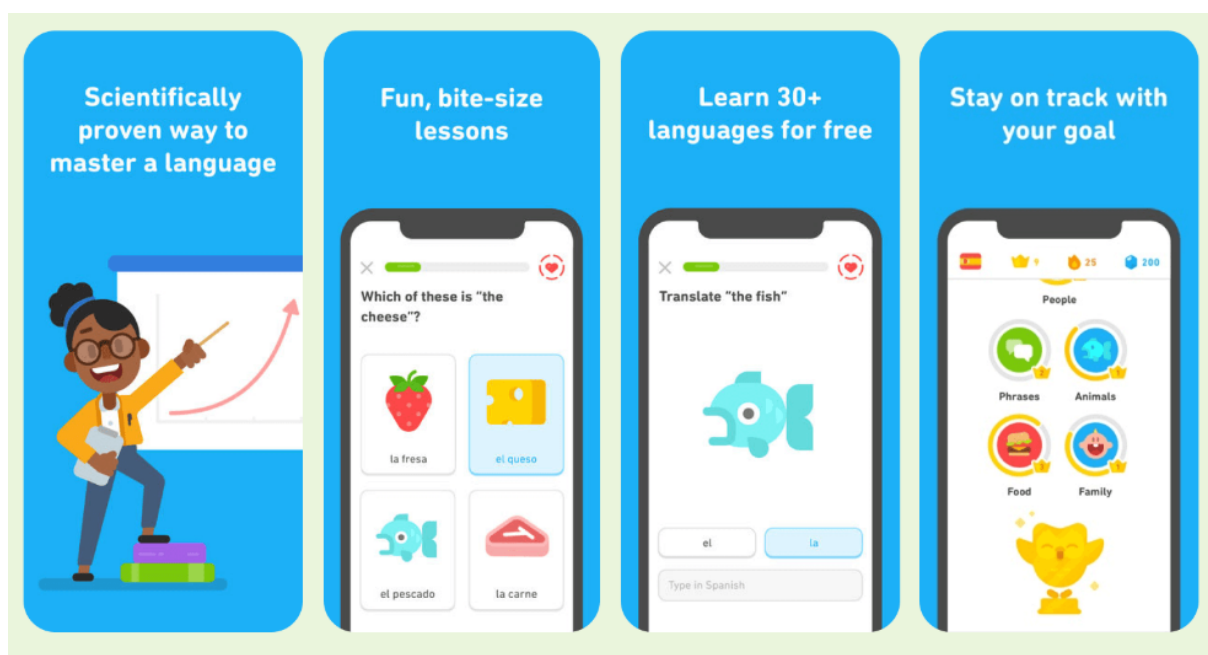


Figure 1. Duolingo's promotional material

A 2017 study found no significant difference between elementary pupils learning Spanish through the "gamification" of Duolingo and those studying in classroom situations, with both groups displaying a similar rise in achievements and self-efficacy.

Another study on adults in 2022 using Duolingo as their only language learning tool, published in the journal *Foreign Language Annals*, found that the participants who completed a course had similar reading and listening proficiency to university students after four semesters of study, concluding that Duolingo could be an effective tool for language learning.

As of February 2023, Duolingo was the most popular language learning app in the world based on monthly downloads, with approximately 13.4 million users installing the program that month.

3.2: Other Applications On The Market

Despite its popularity, this award-winning program, like other competitive language apps on the market, have their shortcomings. The table below shows the feature comparison between different language learning apps.

Language Learning Mobile Applications	Social & Collaborative features	Speech Recognition & Pronunciation	Interactive Multimedia Content	Gamification Elements: Rewards systems
Explanation	Evaluate features facilitating language exchange between user	Evaluate the app's ability to provide constructive feedback on user's pronunciation	Evaluate the inclusion of diverse multimedia elements like audio clips, videos, and images/avatars	Evaluate the effectiveness of gamification features in motivating users
Duolingo	Minimal	Minimal	Minimal	Minimal
Babbel	No	No	Minimal	No
Mondly	No	No	Minimal	No

Drops	No	No	Yes	No
Ling App	No	No	Minimal	No
Busuu	No	Yes	Minimal	No
Rosetta Stone	No	No	Yes	No
Pimsleur	No	No	Minimal	No
Memrise	No	No	Minimal	No
LingoDeer	No	No	Minimal	No

Rubrics:

	Yes	Minimal	No
Social & Collaborative features	<ul style="list-style-type: none"> Active community engagement features (e.g., Add friends, Multiplayer) to enhance the language exchange experience. 	<ul style="list-style-type: none"> Basic community features that may need expansion. 	<ul style="list-style-type: none"> Lack of community engagement features.
Speech Recognition & Pronunciation	<ul style="list-style-type: none"> High accuracy in recognizing and interpreting user speech. Detailed feedback on user pronunciation, offering constructive suggestions. 	<ul style="list-style-type: none"> Generally accurate but may have occasional errors or limitations. Basic feedback provided, but may lack depth or specificity. 	<ul style="list-style-type: none"> Poor accuracy, hindering effective speech recognition. Limited or no feedback on pronunciation.
Interactive Multimedia Content	<ul style="list-style-type: none"> Diverse range of multimedia content, including videos, audio clips, images, and interactive exercises. Highly interactive multimedia content that engages users actively in the learning process. 	<ul style="list-style-type: none"> Some variety in multimedia content, but there is room for expansion. Some interactive elements, but improvements could enhance engagement. 	<ul style="list-style-type: none"> Limited multimedia content, impacting the richness of the learning experience. Lack of interactivity, making the content less engaging for users.
Gamification Elements: Rewards systems	<ul style="list-style-type: none"> A diverse range of rewards, including badges, points, levels, and other incentives. Rewards effectively motivate users to engage in language learning activities. 	<ul style="list-style-type: none"> Some variety in rewards, but there is room for expansion. Some motivational impact, but improvements could enhance engagement. 	<ul style="list-style-type: none"> Limited diversity in rewards, impacting motivation and engagement. Limited motivation from the rewards system.

Based on the table above, we can clearly see that most language learning mobile applications in the market fall short on their “social & collaborative features”, “Interactive Multimedia Content”, “Rewards system” and “Speech Recognition & Pronunciation”. Hence, these are the features that require improvement

3.3: Review

We discovered flaws and limitations that our Duellingo enhancement app can fill. To begin, it will include a real-time multiplayer mode in which users will be able to compete on the app and trigger incoming challenges. Second, a compelling story line is essential for a language-based app. A compelling storyline can capture the attention of users and keep them engaged with the app. Finally, there is a speech recognition and pronunciation feature that employs artificial intelligence to provide users with feedback on their pronunciation.

Chapter 4: Design and Implementation

4.1: Considerations

4.1.1: Feature Considerations

All three improvement features are exciting to develop. However, due to the project's time constraints, we must consider which two features are best to develop. This section compares the type of improvement features as shown in Table 1.

	Real-time Multiplayer	Story-line	AI Powered Speech Recognition & Pronunciation
Project goals & objectives	High - Identified as a fundamental aspect of the project, essential for defining the overall direction.	High - Identified as a fundamental aspect of the project, essential for defining the overall direction.	Medium - Not identified as a fundamental aspect of the project, not essential for defining the overall direction.
Technical Feasibility	High - Acknowledges the team's technical programming knowledge, ensuring the feasibility of implementing main features.	Medium - Team members' minimal experience in curating a story line. It could be a challenge, but could also be rewarding.	Medium - Team members not experienced in AI programming. Recognizes a potential challenge in the technical design aspect.

Resource Availability	High - Sufficient manpower available for the feature.	High - Sufficient manpower available for the feature.	Low - insufficient manpower available for the feature.
Time Constraint	High - Time allocated based on feature priority, ensuring the main features receive sufficient attention.	High - Time allocated based on feature priority, ensuring the main features receive sufficient attention.	Low - Acknowledges not a main feature. Less time allocated based on feature priority
Impact on User Experience	High - Recognized as a main feature with a significant impact on user experience.	High - Further enhances the interactive media, adding a competitive element.	Medium - Considered a good addition to the user experience, providing a speech & recognition feedback feature

For this project, the real-time multiplayer and storyline are suitable as the features that are built are relatively achievable within the time constraints of the project scope. Real-time multiplayer and storyline are vital in increasing user experience in a social context, while AI Powered Speech Recognition & Pronunciation feedback is considered a good addition to user experience as a supporting feature. As a result, because of its ease of development and ability to encourage real time social language learning experience, real-time multiplayer and storyline features are the best choice of our project.

4.1.2: Frontend Software Consideration

Native (Java)		React Native (TypeScript)		Flutter	
Pros	Cons	Pros	Cons	Pros	Cons
Best Performance	Team has less experience with native coding	Large community makes debugging easier	Worst Performance	Able to deploy to iOS as well	Smaller community with less support available
Full access to lower level features of the device	Lack of hot reloading makes debugging and iterating slower	Wide range of libraries available	Inability to implement complex, platform specific features	Better performance than React Native	Worse performance than React Native
	Only available for Android	Able to deploy to iOS as well			Team is unfamiliar with the Dart programming language
		Team has prior experience with web development and React			Less libraries available

		Hot Reloading is available			
--	--	----------------------------	--	--	--

We chose React Native, due to the large community that it has, making it easier to troubleshoot and find solutions should we encounter problems throughout development. Features such as hot reloading also speed up development time by making it easier for us to view our changes and debug the app. Our team also has members with prior experience using React, React Native and web development, while we lack experience with the other solutions. The cons of React Native also do not apply much to us, as our app does not require access to lower level features of the device such as GPS, and it is also not computationally heavy, meaning performance will not be impacted as much.

When considering the programming language used for React Native, we chose TypeScript over Javascript because its static typing feature improves code readability and maintainability. Code written by other members is more easily understandable and type-related errors are raised by the compiler during development.

4.1.3: Backend Software Consideration

Firebase		Amazon Web Services (AWS)		Microsoft Azure	
Pros	Cons	Pros	Cons	Pros	Cons
Provides robust	Smaller community,	Large community	Resources are	Large community	Requires server

authentication features	but popular among beginners		limited by geographic location.		management
Designed for mobile app development	Difficult to migrate data to another platform	Better uptime and availability	Less mobile app deployment options	Designed for enterprises	Not optimised for personal users
Good support on real-time and NoSQL database	No support on relational database	Supports different types of database engines	Limited security options	Strong security and recovery strategies	Worse documentation
Good analytics features		Wide range of tools available	Can be overwhelming for beginners	Good hybrid capabilities	
No server management needed		Larger storage		Better scalability	
Good documentation		Good documentation			

After careful consideration, we decided to go with Firebase because it is tailored for mobile app development. Although our team is generally not familiar with cloud services, its clear and beginner-friendly documentation should allow us to pick up relevant skills in a reasonable time. Firebase also handles the creation and authentication of user accounts for us. Furthermore, Firebase comes with a database solution, which is suitable for storing our user data and question data, and Realtime Database, which allows its data to be synchronised across users in realtime, making it ideal for the multiplayer feature.

4.2: Final Design and Implementation

4.2.1 Frontend Development

4.2.1.1: Visual Development and Testing using Figma

Our team first created a user flow in order to map out how the user would be able to traverse the app first before creating any designs.

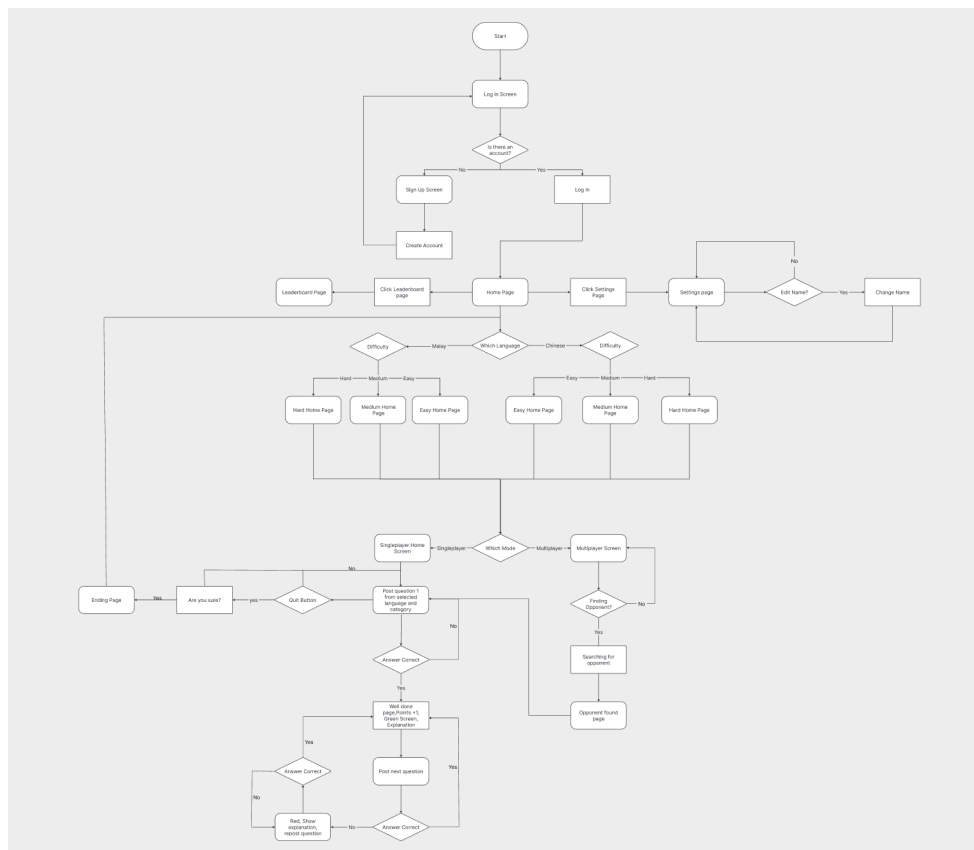


Figure 2. User flow diagram

Next, we can design each of the screens, referencing Google's Material UI 3 design guidelines in order to create a professional and consistent look and feel for the app.

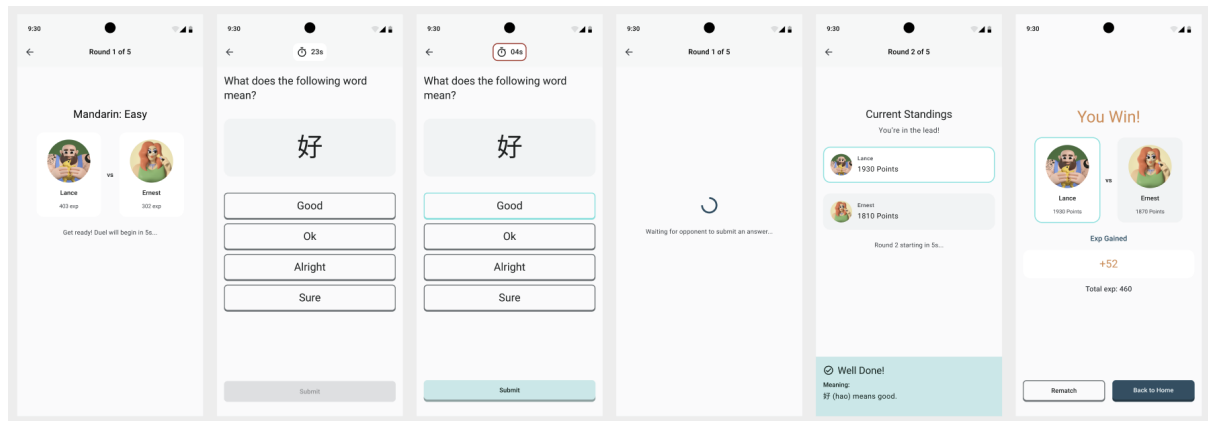


Figure 3. Screens designed in Figma

We focused on giving the app a fun feel, using bright colours, cute avatars and animated buttons to give our app a pleasing and playful UI. This design language communicates that learning a new language should be fun.

4.2.1.2: Building the Frontend Using React Native

The frontend was mainly built using React Native with the React Navigation and React Native Paper libraries. React Native is a UI framework that allows for the generation of natively rendering UI across a multitude of platforms from a single JavaScript codebase. It allows for the creation of components, which allow for effective code reuse. UI elements that are used across different screens are made into components, accepting properties that are able to customise each component for use in different pages. This allows for UI components to have a consistent look across different pages, and makes the code more maintainable as we only need to update the component in order to update all its instances.

Navigation within the app is handled by the React Navigation library. We used a stack navigator, which pushes a screen onto the stack when navigating to a new page, and pops the stack when going back. We also used a tab navigator for the home screen, which displays different screens based on the selected tab.

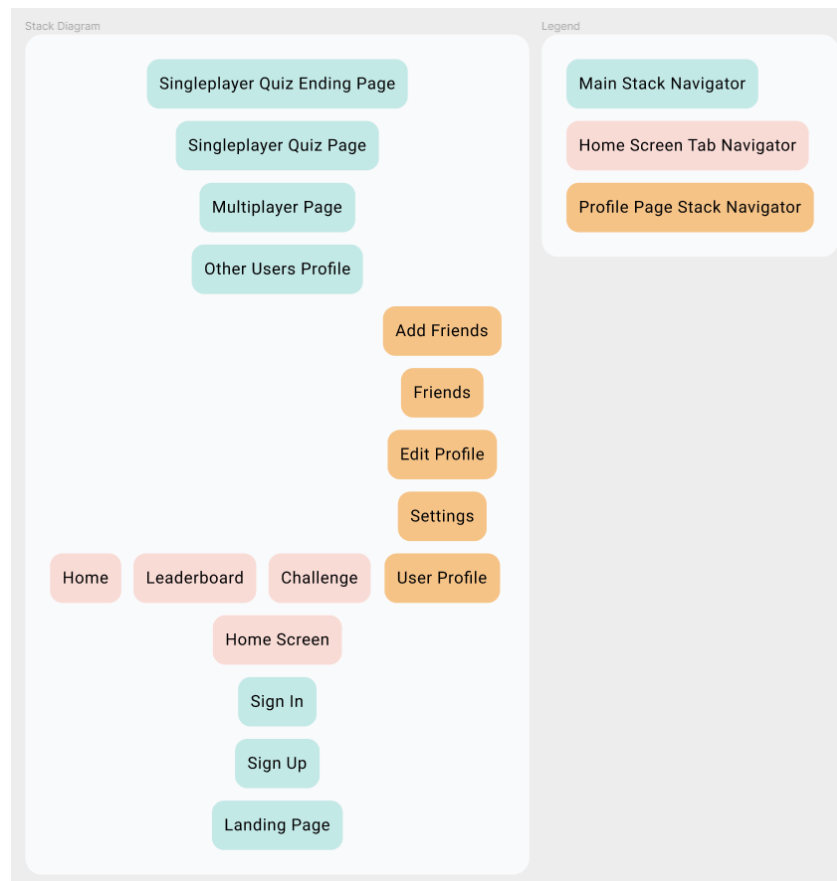


Figure 4. React Navigation Navigator Structure

As we adhere to Google's Material UI guidelines, we used the React Native Paper library, which implements the latest Material 3 specifications for various UI elements such as dialogs and buttons. We are able to use a custom theme with the library, enabling all components to automatically use colours as defined in our theme file.

4.2.2: Backend Development

4.2.2.1: Authentication

The user authentication of our app is implemented using the Firebase Authentication service. Login details are checked to verify the identity of a user. During sign-up, email and username are validated to ensure their uniqueness before a unique ID is generated and their authentication details stored. This unique ID is then used to create a user-specific document to store user information, which will be explained further in the following section.

```

export const signUp = async (
  email: any,
  password: any,
  username: any,
  displayName: any,
) => {
  let userCredential = await auth().createUserWithEmailAndPassword(
    email,
    password,
  );
  await createUser(email, username, displayName, userCredential.user.uid);
  signIn(email, password);
};

```

Figure 5. Code snippet that creates a user when they sign up

4.2.2.2: Storing of User Information

To store the information of each user, we used Cloud Firestore, a NoSQL document database provided by Firebase. Firestore stores data in JSON-like documents. Each document contains a set of key-value pairs, where the values can be various data types, including strings, numbers, arrays and nested objects. Documents are organised into collections. A collection is a group of documents, and each document within a collection has a unique identifier.

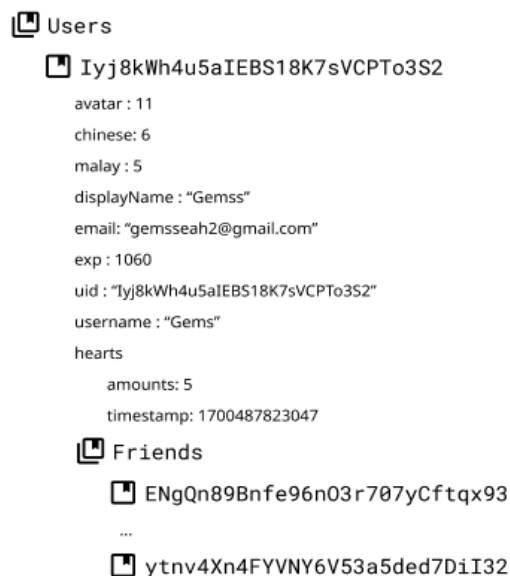


Figure 6. Structure of Cloud Firestore database to store user data

Next, using the input details and unique id generated during sign-up, we designed an API to create user-specific documents in our Firestore database. The data fields

used in these documents are modelled on the visual representation we did on Figma, and their values can be read and updated using other APIs we implemented.

4.2.2.3: Storing of Questions

The questions are stored using a similar structure in the Firestore database. The questions are stored based on the hierarchy - Quiz, Language, Module and Multiplayer, Topic. The structure of the database has been chosen in order to lower the loading time when calling for data by optimising how the data is retrieved.

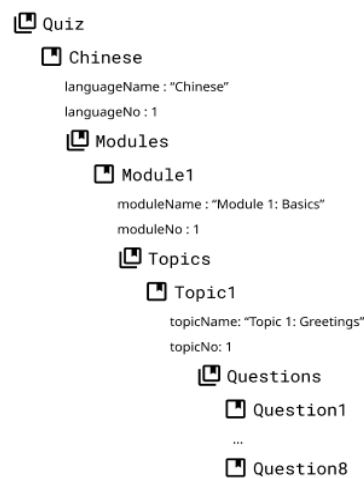


Figure 7. Structure of Cloud Firestore database to store questions

4.2.2.4: Storing of Multiplayer Sessions

The multiplayer sessions are stored in Firebase's Realtime Database which allows for the database to be synced across a multitude of different users in realtime, making it ideal for multiplayer sessions which happen in realtime. We store the following variables in order to keep track of the state of the game.



Figure 8. Structure of realtime database during a multiplayer game

4.3: Main Features of the Application

4.3.1: Quiz

We display the questions according to the format present in the JSON file returned from the database. There are two different formats, the first being an introduction to the word that is being used, and the next is a multiple choice question meant to test the users on the words taught. Once all questions are answered, they will move to the ending page, where the exp received by the users is calculated based on the difficulty and the score and then updated in Firestore.

4.3.2: Multiplayer

In order to initiate a multiplayer game, we first challenge a player. We will refer to the challenging player as the host. This creates a new game lobby as well as a new entry in Firebase's Realtime Database with the user's ID with parameters such as the language and difficulty chosen. We have a listener in the database that will call a function when it detects a challenge issued and show a dialog asking to accept.

When both players detect that they have accepted the request, they are navigated to the multiplayer screen. Once navigated, players load the randomised questions from the database as well as calculate their offset from the server timestamp. We set up listeners for the realtime database so we are able to know when any changes are made. Once the questions are loaded, they indicate that they are ready for the game to start. The host will then set the current timestamp to the server's timestamp in order to trigger the countdown for the game to start. Both players receive the server timestamp and include their local offset in order to sync the time between different devices and have the countdown trigger simultaneously.

Once the game has started, we keep track of whether each player has answered and update the realtime database with the current state of the game. We update the points based on the time taken to answer the question.

We only reveal the answers once we see that both players have answered. The host then triggers a countdown to move to the next question. When all questions are answered, we update the exp of both players based on the difficulty and the points scored.

Players are able to rematch each other, implemented similar to how the challenge mechanism is.

4.3.3: Lives

Users have a maximum of 5 lives and every time a question in a single player quiz is answered wrongly, we deduct one life. We store both the number of lives and a timestamp of when we last regenerated a life. The app will regularly check against this timestamp to see if a heart needs to be regenerated.

```
//increase lives based on when user last heart refill
export const increaseLives = async (userID: string) => {
  const currentTime = Date.now();
  const userDoc = await firestore().collection('Users').doc(userID).get();

  if (userDoc.exists) {
    const userData = userDoc.data();
    if (userData.hearts) {
      const timestamp = userData.hearts.timestamp;
      const lives = userData.hearts.amount;
      const differenceInSeconds = Math.floor((currentTime - timestamp) / 1000);
      const twoMinutesInSeconds = 120;
      const heartsToRefill = Math.floor(
        differenceInSeconds / twoMinutesInSeconds,
      );
      await firestore()
        .collection('Users')
        .doc(userID)
        .update({
          'hearts.timestamp': currentTime,
          //if lives + heartsToRefill is more or equal to five, lives should increased to 5.
          'hearts.amount': firebase.firestore.FieldValue.increment(
            lives + heartsToRefill >= 5 ? 5 - lives : heartsToRefill,
          ),
        });
    }
  }
};
```

Figure 9. Code snippet showing logic of how lives are regenerated

4.3.4: Leaderboard

From our database, we make a query for the top 50 users ordered by the number of exp they have accumulated.

```

export const getLeaderboardData = async () => {
  let leaderboardData: FirebaseFirestoreTypes.DocumentData[] = [];
  await firestore()
    .collection('Users')
    .orderBy('exp', 'desc')
    .limit(50)
    .get()
    .then(querySnapshot =>
      querySnapshot.forEach(documentSnapshot => {
        leaderboardData.push(documentSnapshot.data());
      }),
    );
  return leaderboardData;
};

```

Figure 10. Code snippet of how leaderboard data is retrieved

The current top 3 users would be given custom styling, to show their status.

4.3.5: Friends

Users are able to search for other users either by their username or their display name. By adding other users as friends, their online status and learning progress can be checked and they can be challenged for a real-time multiplayer quiz.

In the Firestore database, friends are stored as a subcollection of documents embedded in a user-specific document. Friends' IDs are read from the subcollection before retrieving their user information to be displayed as a card list in the frontend.

```

export const getFriendData = async () => {
  const friendList = await getFriendList();
  if (friendList.length > 0) {
    let friendData: FirebaseFirestoreTypes.DocumentData[] = [];
    while (friendList.length) {
      const batch = friendList.splice(0, 10);
      await firestore()
        .collection('Users')
        .where('uid', 'in', batch)
        .get()
        .then(querySnapshot => {
          querySnapshot.forEach(documentSnapshot => {
            friendData.push(documentSnapshot.data());
          });
        });
    }
    return friendData.sort((a, b) => b.exp - a.exp);
  }
  return [];
};

```

Figure 11. Code snippet of how friend's data is retrieved

4.3.6: Home Screen

The home screen pulls the module and topic names as well as the user's completed number of modules for the respective language from the database and presents it to the user. The backend queries have been reduced so that it takes less time to call the data and only loads data for other languages if needed. The respective topic buttons will be enabled depending on the progress of the user.

4.3.7: Text To Speech

Users are able to hear the pronunciation when they press on the options button in the testing question and the sound button in the teaching question using the text-to-speech function.

```
const containsEnglish = (text: string): boolean => {  
  // Regular expression to match English characters  
  const englishRegex = /^[a-zA-Z ]/;  
  return englishRegex.test(text);  
};  
  
const optionContainsEnglish = containsEnglish(question.options[0]);  
  
const init_tts = async () => {  
  Tts.setDefaultPitch(1);  
  Tts.setDefaultLanguage(optionContainsEnglish ? 'en-US' : 'zh-CN');  
  Tts.setDefaultRate(optionContainsEnglish ? 0.3 : 0.5);  
};  
  
init_tts();  
  
const readText = async (text: string) => {  
  Tts.stop();  
  Tts.speak(text);  
};
```

Figure 12. Code snippet of how text to speech is implemented

Chapter 5: Conclusion and Recommendation

5.1: Conclusion

In this design innovation project, a language learning mobile application named *Duelingo* was developed. The development of the *Duelingo* has been a valuable and rewarding experience. The language learning mobile application provides an innovative and convenient way for users to learn new languages on the go. The features of the application, such as real-time multiplayer and leaderboard offers a unique experience for users of all skill levels to connect with users around the world and challenge with one another on their language competency at no financial cost. Additionally, the application's user-friendly interface and intuitive design make it accessible to a wide range of users. The development process involved extensive research, testing, and feedback from potential users, resulting in a high quality and reliable product. Moving forward, the application has potential to expand and evolve with the changing needs of its users, and continue to promote gamified online language learning without international border limitations, where cultural exchanges can be achieved remotely.

5.2: Recommendations for Future Works

5.2.1: AI Handwriting Analysis

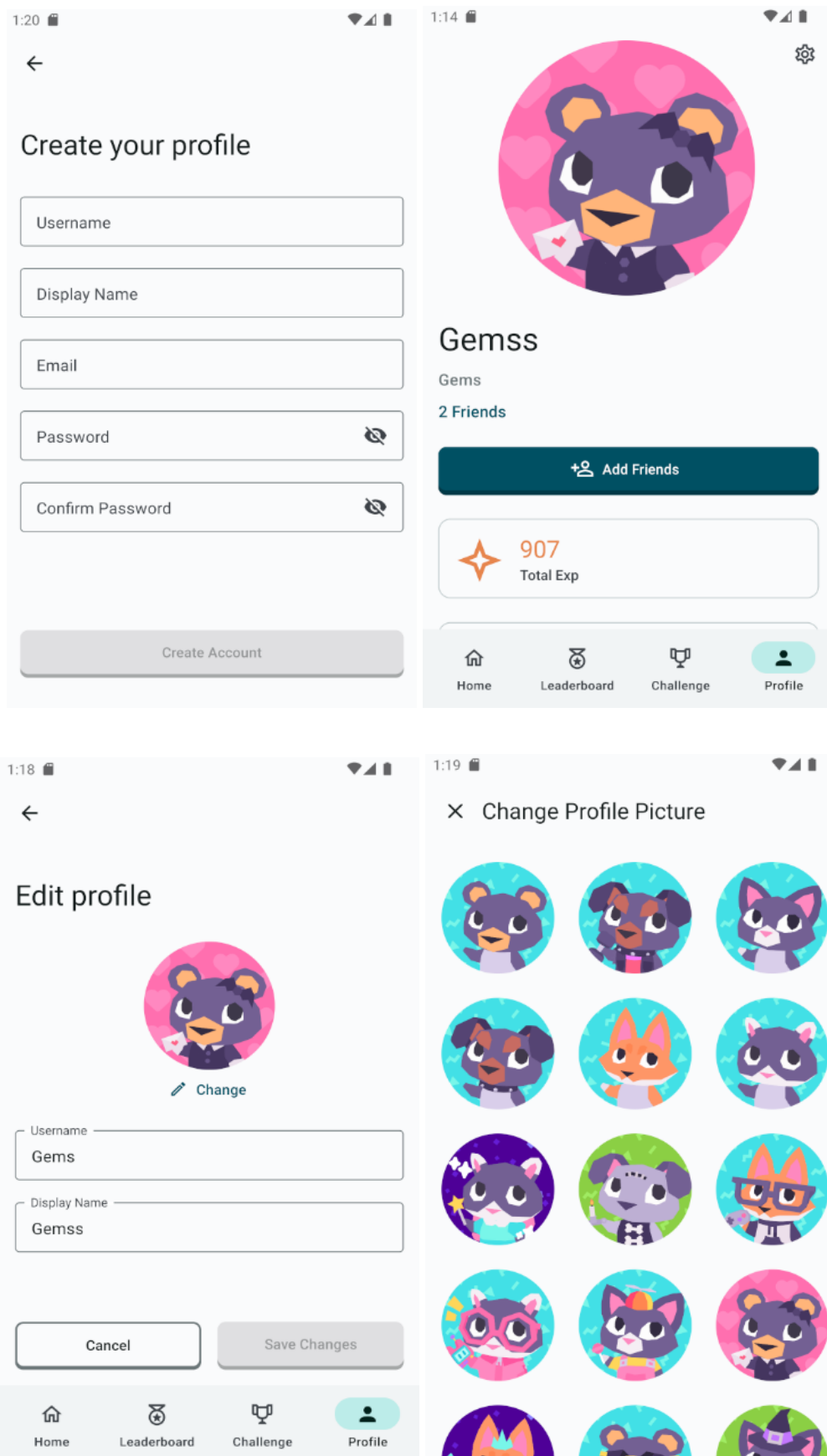
By incorporating this feature, users will be able to receive tailored feedback on their writing, which will help them improve and advance their proficiency with language writing. With the help of this creative addition, users will be encouraged to hone their writing abilities and build a comprehensive language proficiency that goes beyond verbal communication. Users can anticipate a thorough learning process that covers both verbal and written aspects with our AI Handwriting Analysis feature, enabling them to communicate successfully in their target language.

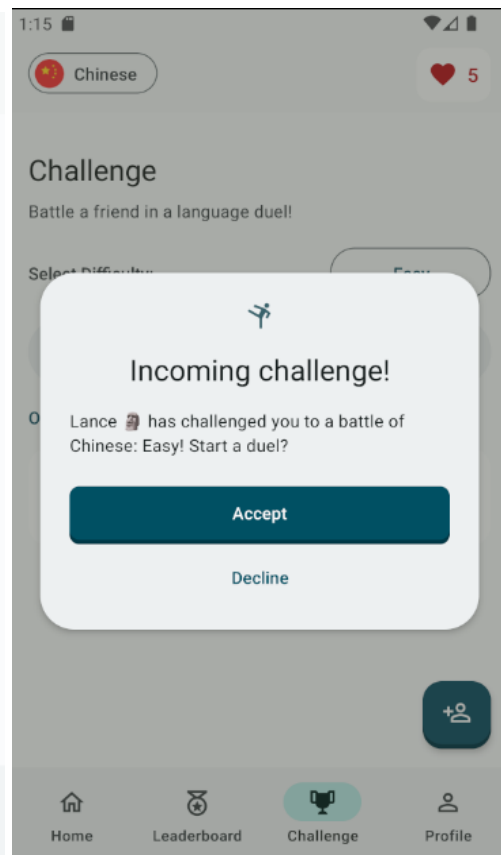
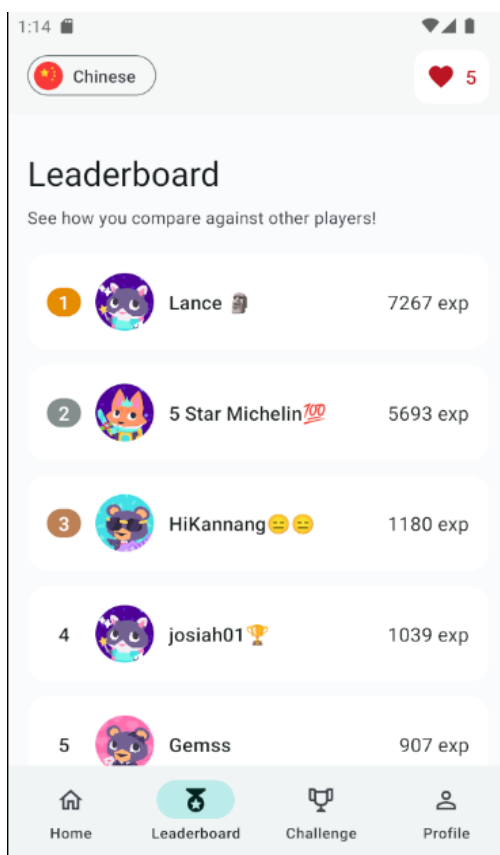
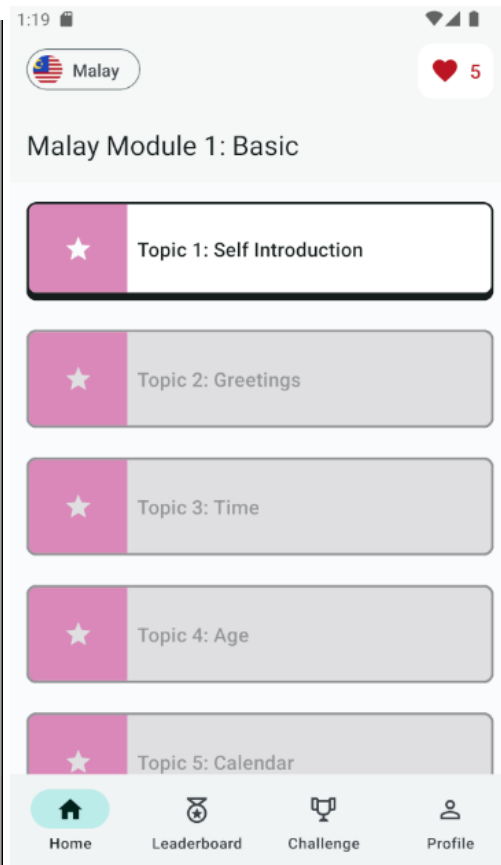
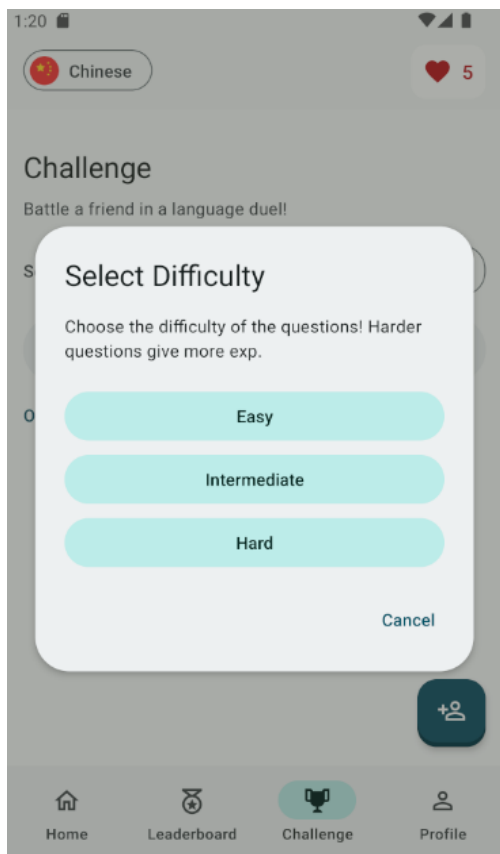
5.2.2: AI Voice Recognition Pronunciation Analysis

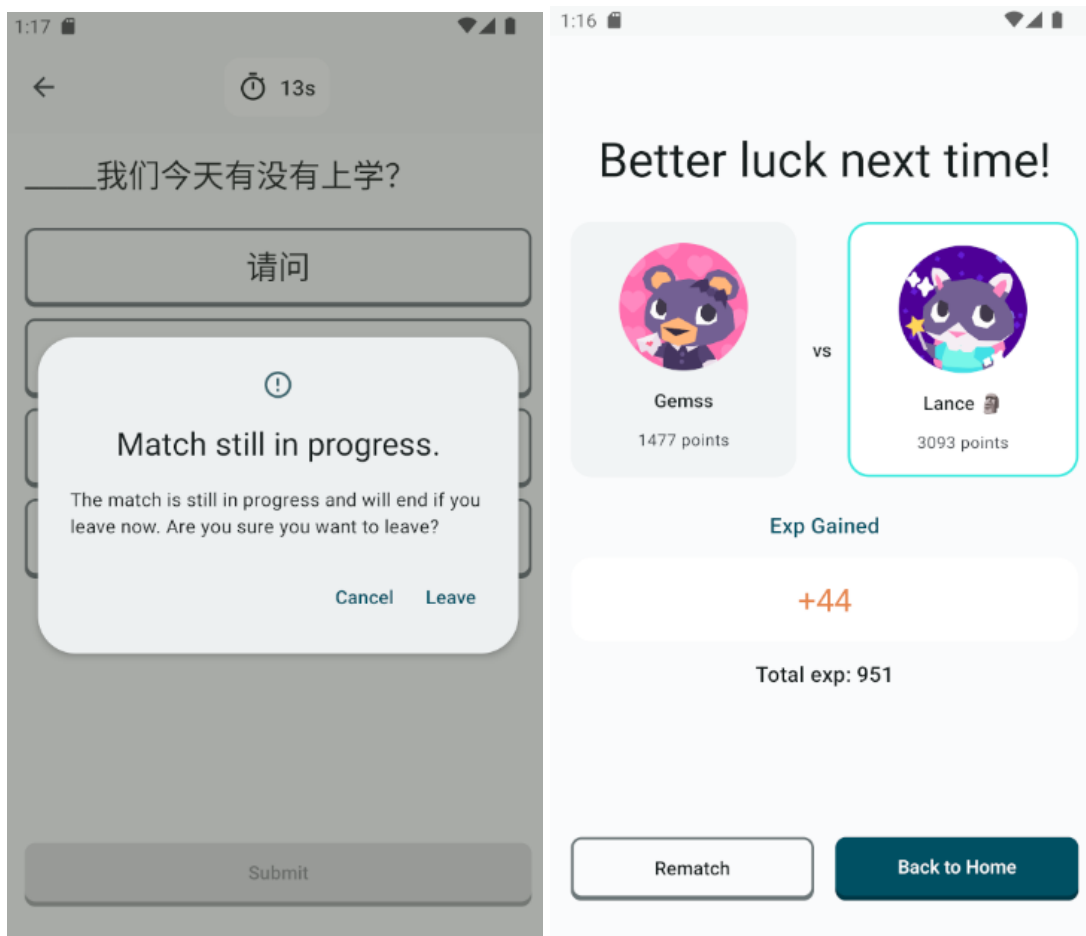
By incorporating this feature, users will be able to receive real-time feedback on their language pronunciation. which will make the experience more interactive and

individualised while allowing users to actively gauge and improve their verbal communication skills. By embracing such an approach, the AI voice recognition system not only becomes more technically advanced but also reinforces its role as an accessible and user-friendly tool in the realm of language interaction and learning.

Chapter 6: Screenshots of Application







Chapter 7: Resources

- [1] Abdalslam, "The Ultimate List of Language Learning Statistics, Trends And Facts 2023 & 2024," Abdalslam, Aug. 16, 2023. <https://abdalslam.com/language-learning-statistics>
- [2] Al-Dosakee, K., & Ozdamli, F. (2021). Gamification in Teaching and Learning Languages: A Systematic Literature Review. *Revista Romaneasca pentru Educatie Multidimensionala*, 13(2), 559-577
- [3] Wikipedia Contributors. (2019, May 8). Duolingo. Wikipedia; Wikimedia Foundation. <https://en.wikipedia.org/wiki/Duolingo>