PROJECT PROPOSAL

INTRODUCTION

A language learning tool using Java programming language will be designed for users to improve their learning efficiency. Our app aims to provide users with a versatile language learning experience through a combination of interactive features and practical exercises as traditional language learning methods often lack convenience and interactivity.

PROBLEM STATEMENTS

Learning a language requires regular practice and suitable tools for remembering new words and understanding grammars. However, the famous apps in the market are often not customizable. They are often designed for beginners who learn with the app only. If the users learnt something from other platforms, such as books videos, they are not capable to make use of those app for revision.

OBJECTIVES

- To aid with users learning and remembering new vocabulary
- To improve users writing ability
- To save users' favourite articles for reading

APP EXAMPLE IN MARKET

DUOLINGO

Duolingo is one of the most popular language learning apps. It uses gamified approach for users to learn by taking small lessons, answering quizzes, and reading stories. Yet, users are not able to customize their learning experience and synchronize with their learning progress from other sources. Meanwhile, free version provided limited functions and there is a quota for users every day.

Although Duolingo provides a comprehensive learning environment. We target at creating a more personalized learning experience so that users can build up their own vocabulary and base on the data stored in the database to generate tests.

ANKI

Anki is designed mainly for building up vocabulary using active recall testing and spaced repetition. It is famous for its flashcards function as users are able to not only create their own flashcards, but also download pre-made combinations uploaded by other users. However, it is mainly used for word memorizing and does not hep with other language ability such as reading and writing.

Our application includes flashcards feature similar to Anki. Yet, we aim at providing more functionality such as writing improvements and allow users to store their favourite readings.

FUNCTIONALITY

CUSTOM FLASHCARDS

Users can create and manage custom flashcards with words, phrases, and sentences tailored to their learning objectives. Meanwhile, the flashcards should be able to be grouped and users can label the theme and level of difficulties.

WEEKLY TESTS

The app will offer weekly language proficiency tests with varying difficulty levels, providing users with an opportunity to assess their progress and identify areas for improvement. The tests are auto-generated based on the flashcards in the database. Besides, the test could also fetch content from the readings stored in order to create questions. Test results and the number of tests took should be displayed at the "home page" for users to review their progress.

WRITING ADVICE

A built-in writing proofreading feature will allow users to practice their writing skills and receive instant feedback on grammar, vocabulary usage, and sentence structure. We are considering using ChatGPT to provide instant response to the users writing as it is a popular natural language processing API which supports multiple languages.

ARTICLE READING

Users can create their own library of articles in their target language, so that they can review the reading afterwards. Besides, weekly test may also obtain content from the readings to generate questions. Apart from the text, users may also add images. It can also serve as a writing journal as users can proofread their writing using writing advice function and store it.

CORE ANDROID COMPONENT

ACTIVITIES

Main activity is the home page, like a dashboard, where users can view the summary of their progress, such as the number of words have been added in flashcards, the number of test taken and the number of writing and reading articles.

Each function will generally has one to three activities. For instance, flashcards may have three activities, one for listing the flashcards, one for creating a flashcards, and one for viewing the flashcards. The writing advice functions may need one activity only where user writes on the bottom-half of the screen and a response will be shown on the top-half.

SERVICES

Our application does not require many services, as it does not need to run in background. Services such as file I/O and networking may be required for storing the articles, flashcards, and allowing to access to the Internet for using API.

CONTENT PROVIDERS

No communication with other application is needed. Yet, data is needed to be stored in through file I/O and network connection is needed to access API. A simple SQLite database is needed to store the flashcards and other relevant data.

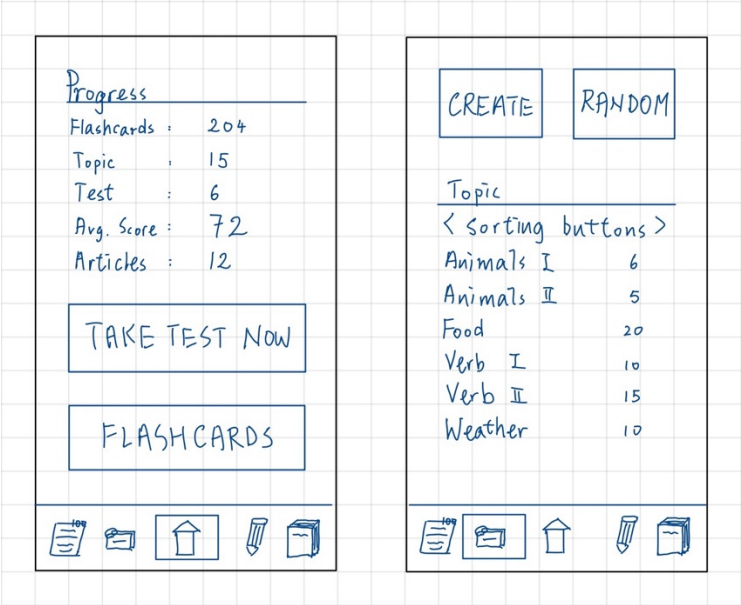
System Broadcasts

Broadcast receiver is needed for sending notification in our application. When the user has not done a weekly test, a notification should be pushed to remind the user. Besides, system notification is also needed to be handled properly in system broadcasts core.

UI Layout Draft

There should be a navigation panel at the bottom for different functions. At home page, users should be able to view their status of learning and some button for quick navigation as shown in the left layout.

The right layout shows briefly about the flashcards page where users can create flashcards or topic, view the list of their flashcards, randomly choose one topic to review, and press any one from the list to check the flashcard topic.



Difficulty Ranking

Hard

Weekly Test: Developing the weekly tests feature requires designing a comprehensive testing framework capable of generating random questions, assessing user responses, and providing feedback on language proficiency based on the database. Furthermore, customizing test parameters such as difficulty level, question types, and scoring criteria adds complexity to the implementation.

Medium

Custom Flashcards: Implementing custom flashcards involves creating a user-friendly interface for card creation and management, integrating features such as text input, image upload, and audio recording. Additionally, storing and retrieving custom flashcard data from a local database is also included.

Writing Advice: Integrating writing proofreading functionality requires natural language processing (NLP) techniques to analyse user-generated text. Yet, we are considering using ChatGPT API which may reduce the workload on NLP. However, the UI and how to display the advice retrieved from API is also a challenge.

EASY

Article Reading: The article reading feature is relatively simple because it does not involve fetching and displaying content from external sources. The major difficulties are to properly store the article content in the local database.