LearnByRepetition

Documentation

Konrad Wasilewski

1. Project overview

The project aims to develop a mobile application that facilitates language learning by allowing users to create and manage flashcards for studying English and Polish. The app implements a local SQLite database to store and manage the flashcard sets and individual flashcards. The user interface is designed with multiple views, ensuring a user-friendly and seamless experience. The app leverages the capabilities of the Android operating system to provide an efficient and effective tool for language learning.

2. Used technology

The app was built in Android Studio using Kotlin programming language as well as Room library for communication with the built in SQLite database.

3. App contents

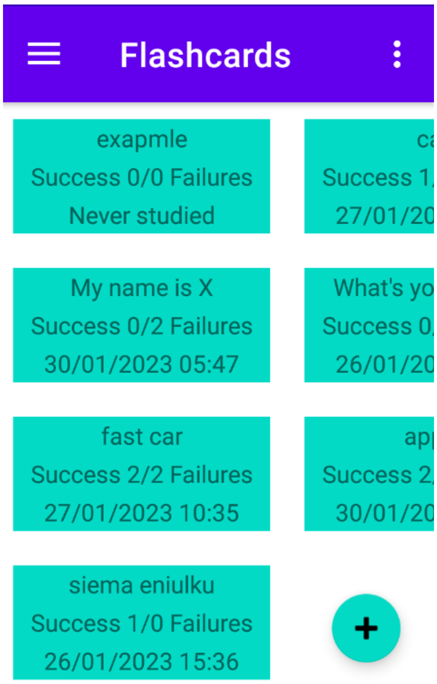
This app provides 4 main tabs used for moving between 4 main functionalities of application. There is “Home” that is the starting point when application is initially started, then there is “Flashcards” and “Flashcard sets” that are used for navigating through database contents. And at the end there is “Random” that allows for quick learning sessions with random flashcards. Also, there is a popup menu where you find options to switch between themes and languages.

4. Used UI components

This app makes use of:

- Recycler Views – They allow for easy presentation of long lists of data.
- Floating Buttons – Their non-intrusive characteristics allow for additional functionality without affecting any underlying views.
- Popup menu – Its’ small size allows for storing less important functionalities without taking up the precious screen space.
- “Slide out” menu – Is a stylistically pleasing content that provides access to many functionalities in a detailed manner, its presence is a must in a big part of many modern mobile applications.
- Fragments – This type of component was used to implement easy movement between menu items. It’s easy to implement and it provides all required functionalities

5.UI example screenshots



exapmle
Success 0/0 Failures
Never studied

ca
Success 1,
27/01/20

My name is X
Success 0/2 Failures
30/01/2023 05:47

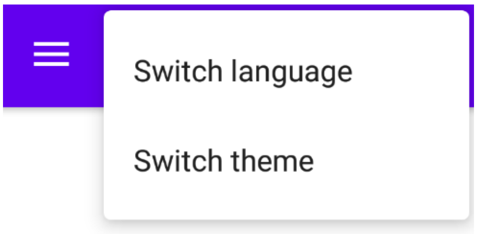
What's yo
Success 0,
26/01/20

fast car
Success 2/2 Failures
27/01/2023 10:35

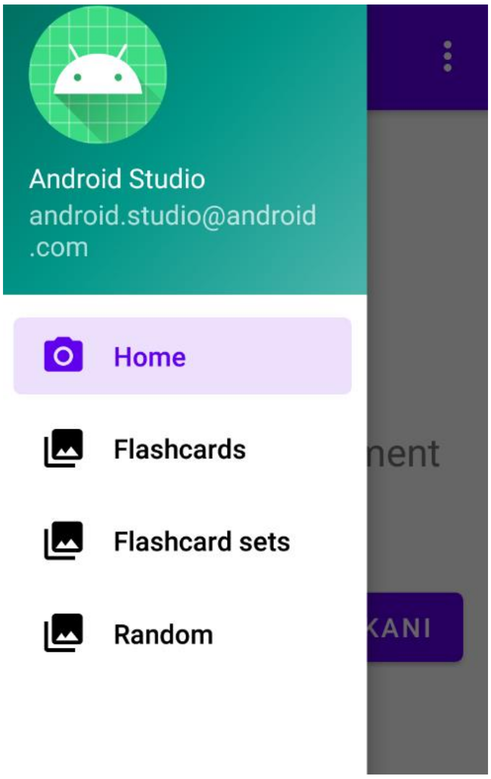
ap
Success 2,
30/01/20

siema eniulku
Success 1/0 Failures
26/01/2023 15:36

Img. 1 Recycler view with flashcards as well as a floating button



Img.2 Pop-up menu



Img.3 Slide out menu

6. Hardware resources

This application’s use of hardware is fairly simple and limited:

- Accelerometer is used for detection of shaking action to access random flashcards inside the “Random” tab
- Proximity sensor is used for disabling in app touch when it reads a distance below 1cm, it’s made to prevent accidental inputs when for example: The user accidentally doesn’t block his screen and puts their phone into a pocket

7. Database

The database used is very simple, it makes use of the Room library as well as the built in SQLite database. Here are the data models and their fields:

- Flashcard
  - englishText – Simply English meanig of the word
  - polishMeaning – Simply Polish meaning of the word
  - dateLastStudied – Date and time when the flashcard was last studied
  - successCount – Count of successful guesses of the flashcard
  - failureCount – Count of failed guesses of the flashcard
  - isWord – If true then the flashcard is considered a word otherwise it's a sentence
  - isDataDefault – Used to differentiate between flashcards used for prepopulating the database and flashcards created by the user
  - id_flashcard – Flashcard's identifier and primary key
- FlashcardSet
  - name – It's the set's name
  - id_set - Set's identifier and primary key
- IntermediateFlashcardsSets – Used as a join between the previous 2 tables
  - id_set – Simply set id
  - id_flashcard – Simply flashcard id

8. Most interesting solution

In order to implement proximity sensor functionality across all activities I've coded it inside my own class which inherited from application, later on the application class was registered inside the AndroidManifest wchich allowed me to implement the functionality by just copying and pasting a 7 line dispatchTouchEvent override in each activity.

9. Conclusion

In the end the app came out to be a good prototype for a possible future development, its' process of development had a greatly affected my understanding of Android applications, their coding standards as well as their differences in UI design. Even though the application has its' problems it provides all functionalities necessary for full usability, it can be used for learning and creating learning sets which was my original goal.