GitHub Username: whoeverlovely

MyDict

Description

MyDict is an vocabulary learning app which schedules daily vocabulary learning and reviewing list based on your customization. If a word confuses or reminds you of other words, you can easily link them together. When you review it, check them together to reinforce your memory.

Intended User

Students who are preparing for exams such as IELTS, TOEFL, GMAT , GRE , SAT etc.
People who want to expand their vocabulary and improve their English.

Features

- Generate daily vocabulary learning and reviewing list.
- Easily link a vocabulary with other vocabularies which might cause confusions to reinforce the memory of all the vocabularies.
- Provide widget to display the vocabularies of the day.

User Interface Mocks

Screen 1

Screen for vocabulary learning/review

If you already knew the vocabulary when the vocabulary appears in the learning list or you confirm you've learned it when it appears in the review list, click "Archive", the vocabulary will not show up in the review list in the future.
If there's other vocabulary you feel confused with, click the "Link" button.
If there's a vocabulary linking group containing the world, click "View Links" will display all other words.

Screen 2


Screen for vocabulary linking

In the example, when the user is learning "exploit", he feels it looks like "explode", but he is not sure what "explode" means. So he enters "explode" in the search bar, click button "Link it to Exploit" which will link "exploit" with "explode".

Screen 3

Screen for vocabulary group
When the user clicks "View Links", all vocabularies linked to "Exploit" will be listed.
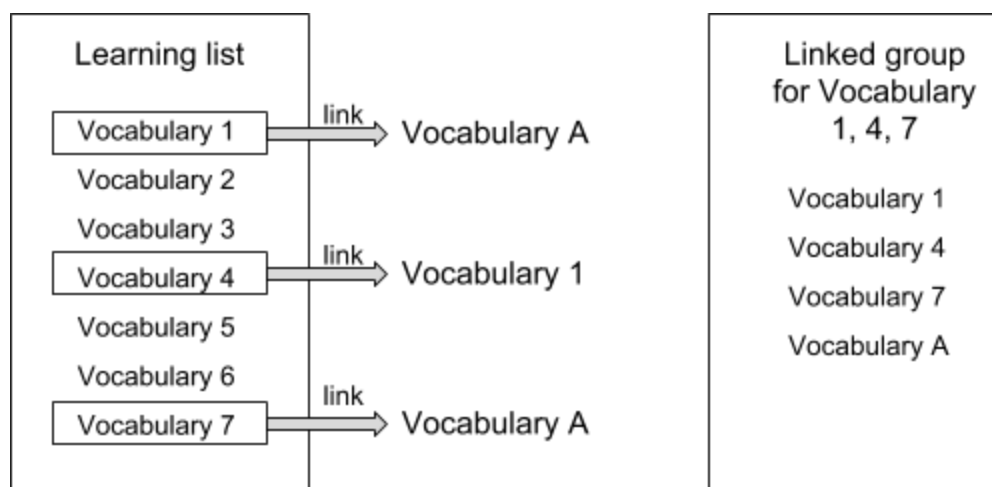
As the chart above illustrated, the words in the linked group could be linked when the user is learning "Exploit", or when the user is learning those words and linked them to "Exploit". In this example, the user linked "explode" while learning "exploit", then linked "exploit" when learning "explore".

Widget

Widget updates on a daily basis and displays the learning and review list of the day.

Key Considerations

What language will you use to write the app?
The app will be written solely in Java Programming Language.

How will your app handle data persistence?
There'll be a data server on Google Cloud Storage. Users download specific vocabulary list according to user's settings. Import the downloaded data to android sqlite database. The database includes a vocabulary table which contains columns including word, phonetic, definition (in English), translation (in Chinese), tag etc. Access local data via content provider.

Describe any edge or corner cases in the UX.
TBC

Describe any libraries you'll be using and share your reasoning for including them.
- Timber
  -- to make log and debug simpler and more convenient
- Butter knife
  -- to access views more efficiently
- Firebase JobDispatcher
  -- to schedule widget daily updating jobs in background

Which tools and libraries version will you use?
- Android Studio
  -- 3.0.1
- Gradle

-- 4.1
- minSdkVersion
  -- 15
- targetSdkVersion
  -- 27
- Timber
  -- 4.1.2
- Butter knife
  -- 8.8.1

Describe how you will implement Google Play Services or other external services.
- Google Translate
  -- will be used when linked word is not available in local database
- Google Cloud Storage
  -- will be used to store vocabulary lists centrally and allowing users to download as requested

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

Data resource and server setup
- Split the existing vocabulary list into several smaller lists such as " GRE vocabularies, TOELF vocabularies etc, based on the tags on each entry.
- Create a project on Google Cloud Storage and upload prepared lists.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity (Screen 1 above)
- Build UI for LinkActivity and LinkFragment (Screen 2 above)
- Build UI for GroupActivity and GroupFragment(Screen 3 above)

Task 3: Local database setup

- Create local database to store vocabulary list

●   Create content provide

Task 4: Preference Settings

●   Implement App Settings activity using PreferenceFragment. User is able to select which vocabulary list he/she wants to learn in the Settings and app starts a download task accordingly.

Task 5: Daily Learning and Reviewing list generation
●   Implement daily learning and reviewing list logic
●   Update database entries accordingly

Task 6: Widget

Create a widget to list learning and reviewing list of the day
●   Create widget layout
●   Implement AppWidgetProvider

Submission Instructions
●   After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    ○   Make sure the PDF is named "**Capstone_Stage1.pdf**"
●   Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:
●   Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
●   Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"