PROJECT REPORT

ABSTRACT

A language learning tool on android platform using Java programming language is designed for users to improve their learning efficiency. Currently, the app provides two ways of learning, free-writing and flashcard sets. In the remaining time, writing revision using AI technology and learning dashboards will be implemented to comprehend the learning experience of users.

HOW TO USE

Due to the ongoing development of dashboards and setting functions, only flashcards and writing functions are shown below:

WRITING

To start writing freely, navigate to "writing" page by tapping the menu icon on the top-left corner. A navigation bar shall be shown immediately afterward (Figure 1).

After that, users can type in the title and the content of the article. When writing is finished, simply tap "SAVE" button and the article will be immediately shown on the lower screen.

To edit or delete the article, users can single tap or hold on the article. For editing, the data will be put in the input fields on the upper screen. The current data will be clear and therefore a confirm message box is popped up asking the users to confirm the action (Figure 3); For deleting, users can hold on the designated article and a message box pops up to ask for confirmation (Figure 4).

Below the purple dividend line with "notes" on it is a sorting dropdown menu. Users are able to sort the articles by either date (latest first) or title (alphabetic order).

FLASHCARDS

The flashcard function allows users to store multiple sets of flashcards. To create a set, navigate to "Flashcard" page. The "CREATE A SET" button is used for adding a new set while the "VIEW RANDOM" button is used for randomly selecting a set of flashcards for review. Below the "Topic" bar shows a list of created flashcard-sets (Figure 5).

Inside the create set page, users are required to input a title (the topic) of the set. Then, by clicking the "+" button, a purple card is generated below the button and users can input the front-text and back-text of the card. Click "SAVE" button to confirm creating the set (Figure 6).

NOTE: view and add photo are not implemented yet.

To view the set of flashcards, simply click the title of the set and users is navigated to the view flashcards page. Topic, current flashcard index and total number of flashcards are shown at the top purple bar. Users can click “EXIT” button to return to the flashcards page. Front-text will be shown first by default and users can control the flashcards using the “PREV”, “FLIP”, and “NEXT” buttons to view the previous flashcard, flip the current flashcard, and view the next flashcard respectively. Figure 7 and 8 shows the front and back state of the flashcard.

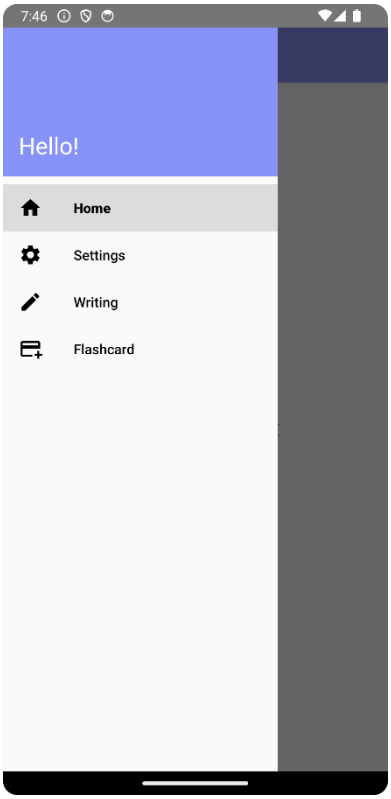

Figure 1

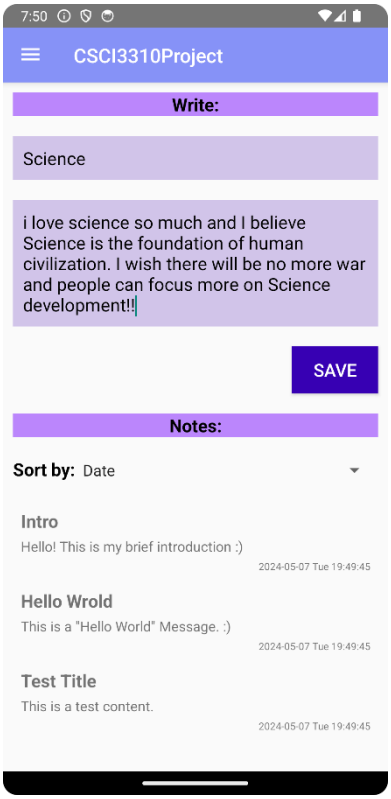

Figure 2

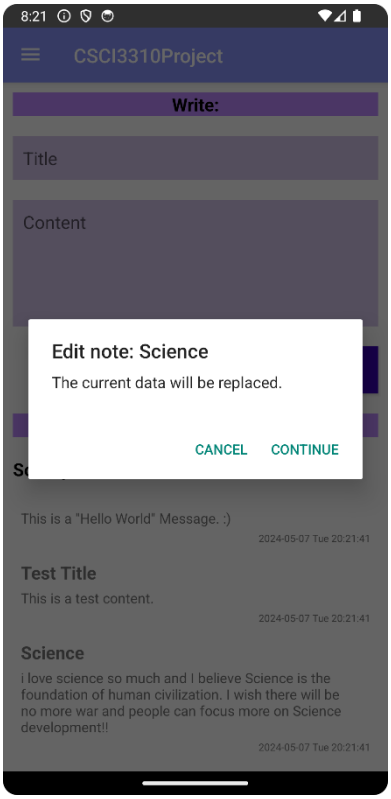

Figure 3

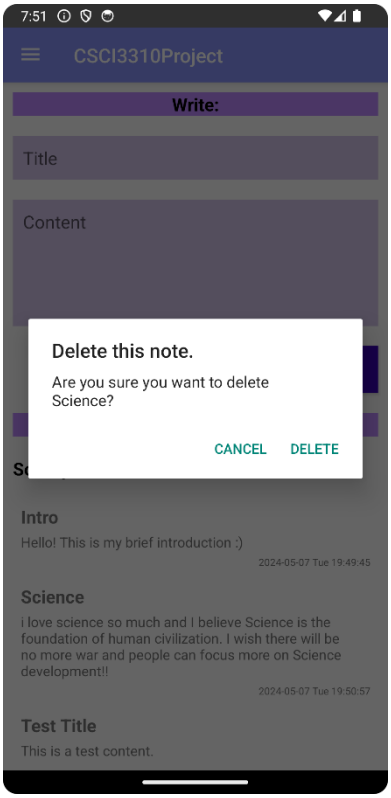

Figure 4

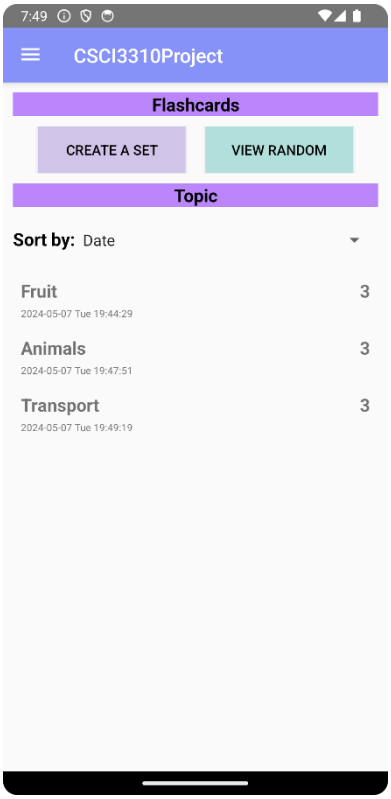

Figure 5

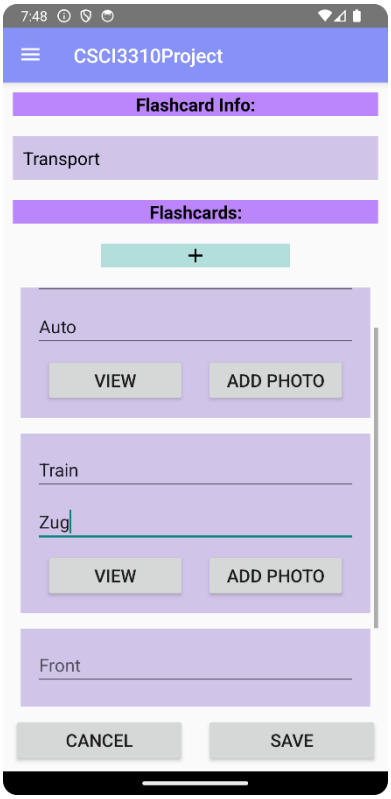

Figure 6

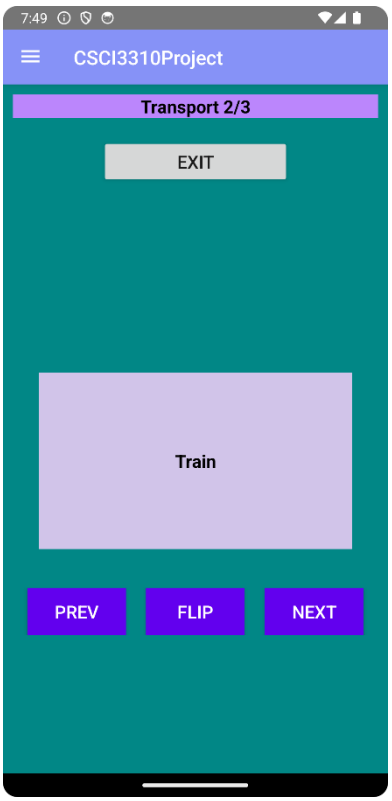

Figure 7

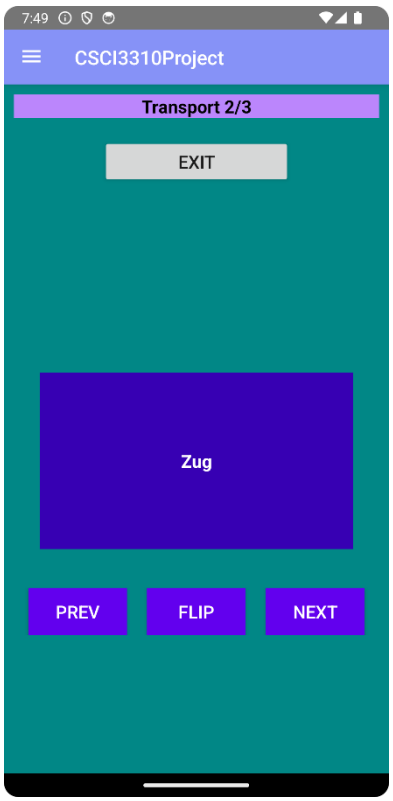

Figure 8

BUILD AND TEST

NOTE: To be completed

IMPLEMENTATION DETAILS

NAVIGATION

NavigationView is used for navigation sidebar. Instead of using *Activity*, *Fragment* is used to display pages because it can easily create a smoother page transaction effect. To navigate outside the navigation sidebar, developers can use the following methods:

```
public void NavigateToFragmentByNavID(int id);
public void NavigateToFragmentByFragment(Fragment fragment);
```

WRITING

Each article is a java class *Note.java*, which contains title, content, and datetime. In *WriteFragment.java*, note data are saved in *SharePreference*. Currently, there is no public method open for other classes to be used. For dynamically populating articles, a *LinearLayout* inside a *ScrollView* is used. A whole article UI *Layout* is stored in *layout* folder (note_item.xml) in order to be added into the *LinearLayout*.

FLASHCARD

The flashcard function consists of 3 fragments, main, create and view, as well as two java helper classes *Flashcard* and *FlashcardSet.* The sidebar navigates to main fragment. The list below showing flashcards set is similar to the one in writing function. In create fragment, Each set of cards has a unique *SharePreference* XML file while the set itself, is stored in a common *SharePreference* (SetPrefs.xml). In Figure 9, there are 3 flashcard-sets and the set information are stored in *SetPrefs*. For passing the selected set information, an additional *FlashCardSetTempPrefs* is created so that data consistency can be assured.

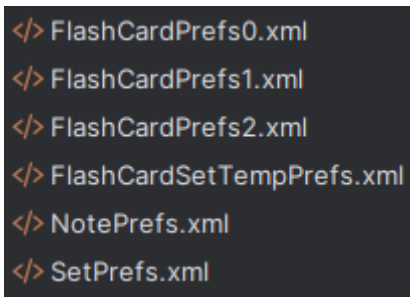

Figure 9

AI REVISION AND DASHBOARD

To be written…

REFLECTION

It is noted that the UI design would be changed in the future. Yet, the core functions will remain unchanged and we would try our best to finish the remaining 2 features.