DESIGN SPECIFICATION

for

Foreign Note iOS App

Version 1.0

Authors:
Dara Beng
Brittney Witts
Veronica Sumariyanto

March 10, 2019

Contents

1 Abstract **3**

2 Architecture Overview Diagram **4**

3 Entity Relationship Diagram **5**

4 Activity Diagram **6**
- 4.1 Create Vocabulary 6
- 4.2 Delete Vocabulary Collection 7
- 4.3 Update Tag 8
- 4.4 Search and Filter 9

5 Message Sequence Diagram **10**

6 State Diagram **11**

7 Class Diagrams **12**

8 Prototype **13**
- 8.1 Horizontal 13
- 8.2 Vertical 14
  - 8.2.1 Add Vocabulary 14
  - 8.2.2 View Vocabulary 15
  - 8.2.3 Add Vocabulary Relation 16
  - 8.2.4 List of User's Vocabulary Collection 17
  - 8.2.5 Search and Filter Vocabulary 18

9 Technology Stack **19**

1 Abstract

The purpose of this design specification is to give an overview of how the system will be implemented. It goes over the implementation through the use of many diagrams such as: system diagrams, entity relationship diagrams, activity diagrams, message sequence diagrams, state diagrams, and a class diagram.

Additionally, this document provides prototypes on how the user interface of the app will look. These prototypes give a visual representation of how the user can navigate from one state of the app to the next. Finally, it lists and describes the technology stack that will be used to build and deploy the app.

2 Architecture Overview Diagram

This diagram illustrates the architecture of the system. It lists the external systems or entities that the system will interact with.

For more detail on how each technology is used, see Technology Stack.

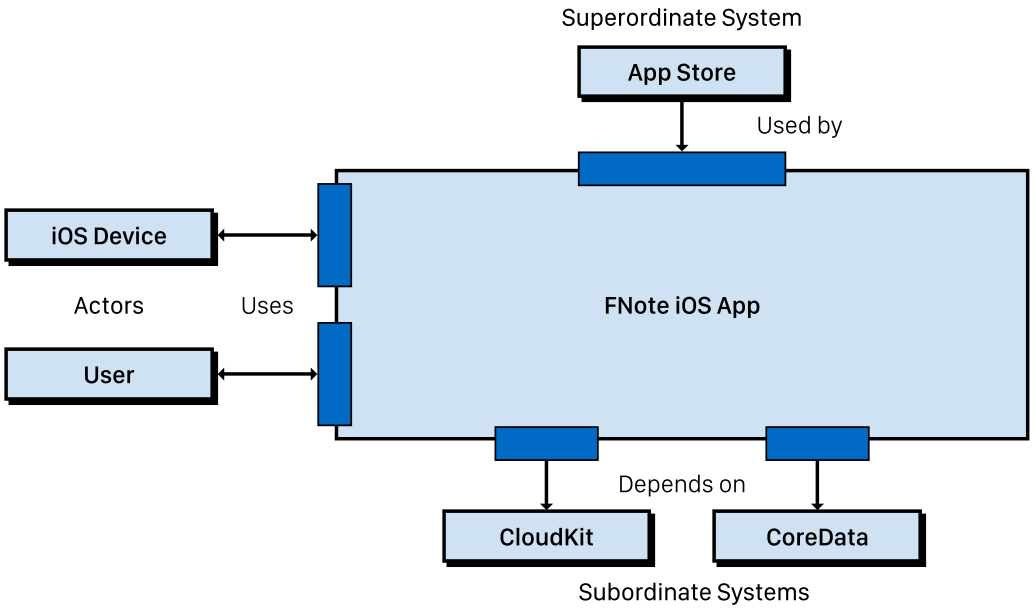

Figure 2.1: Architecture Overview Diagram

3 Entity Relationship Diagram

This diagram illustrates the relationships between objects that are being used and stored in the local and cloud database which are CoreData and CloudKit respectively. It also briefly explains the behavior of an object's method. Additionally, it describes some of the available metadata that a local or cloud record contains.

Every entity is stored both on user's device and iCloud. `CKRecord` is a `CloudKit Object` and it is used to offload entity data to the cloud. This allows the data to be synced across all of the user's iOS devices.

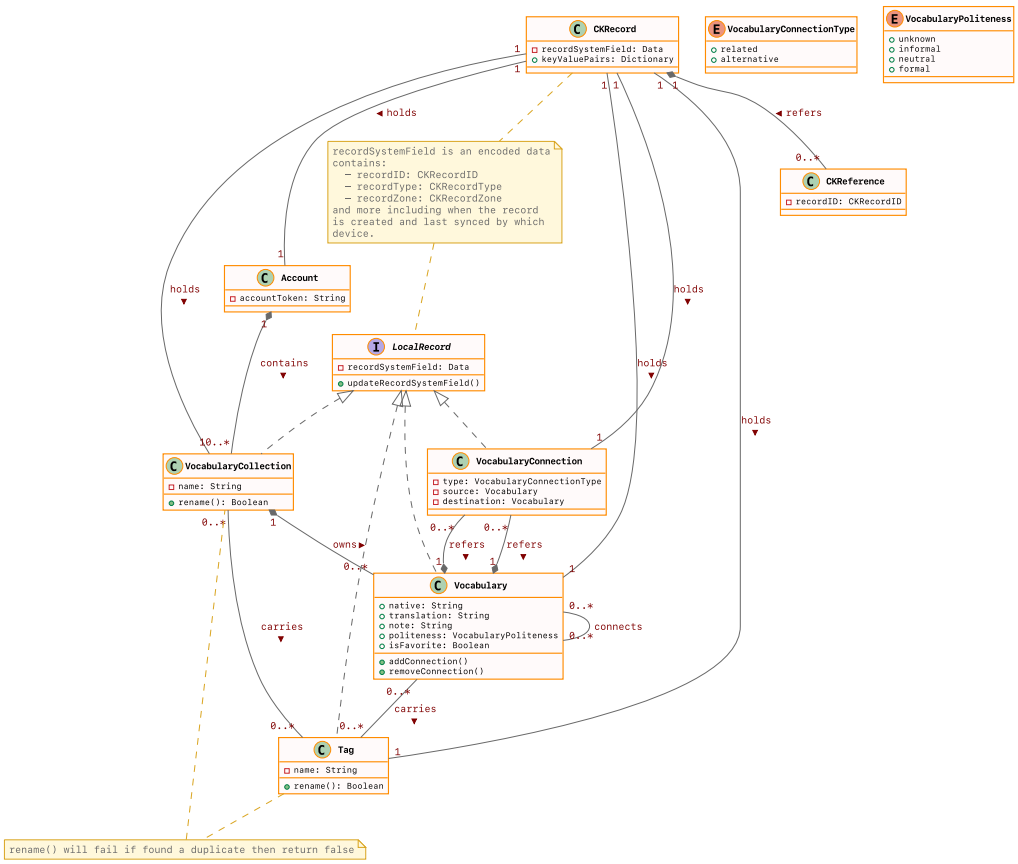

Figure 3.1: Entity Relationship Diagram

4 Activity Diagram

4.1 Create Vocabulary

This diagram illustrates the activities of the system and choices that the user will make when creating a vocabulary.

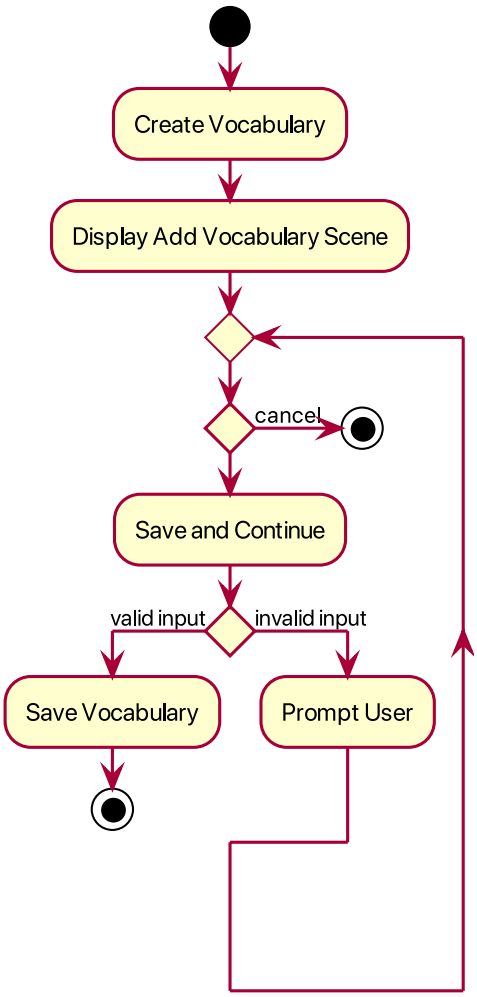

Figure 4.1: Create Vocabulary Activity Diagram

4.2 Delete Vocabulary Collection

This diagram illustrates the activities of the system and choices that the user will make when deleting a vocabulary collection.

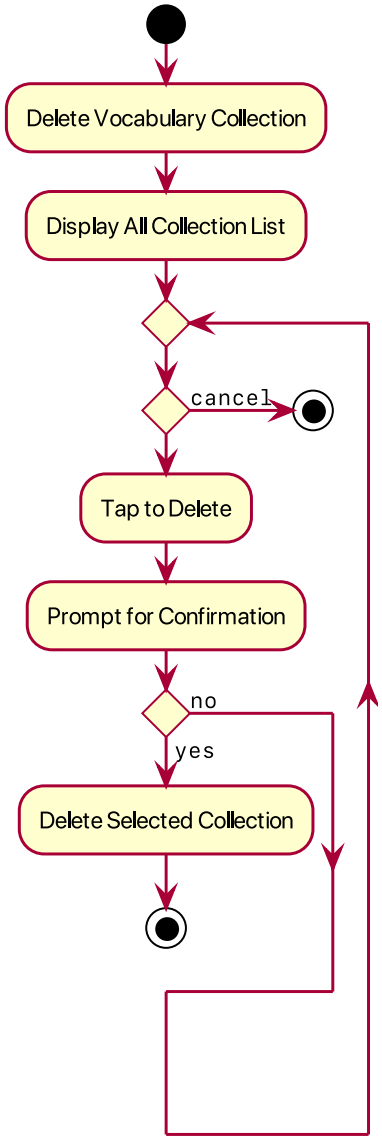

Figure 4.2: Delete Vocabulary Collection Activity Diagram

4.3 Update Tag

This diagram illustrates the activities of the system and choices that the user will make when updating a tag.

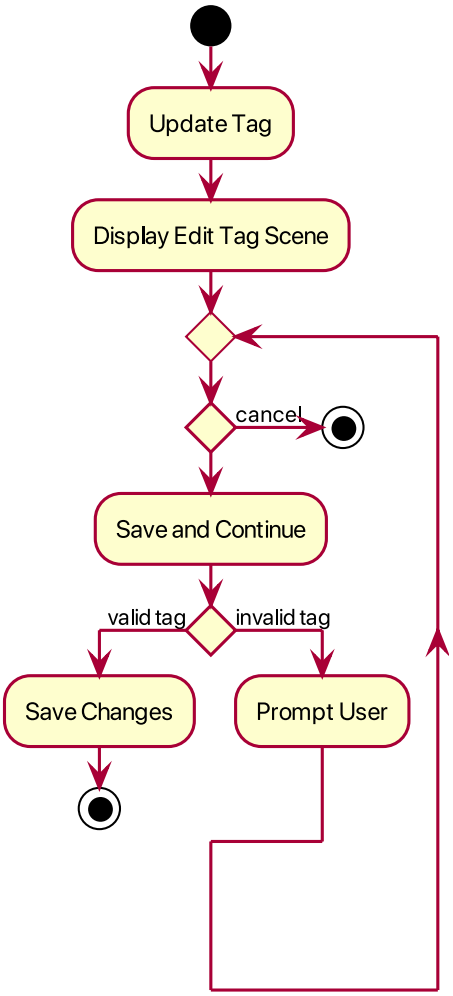

Figure 4.3: Update Tag Activity Diagram

4.4 Search and Filter

This diagram illustrates the activities of the system and choices that the user will make when searching and filtering for results. Users can select one of the live-search results to view the vocabulary without having to finish typing their query. Users can also refine the search result with filter options.

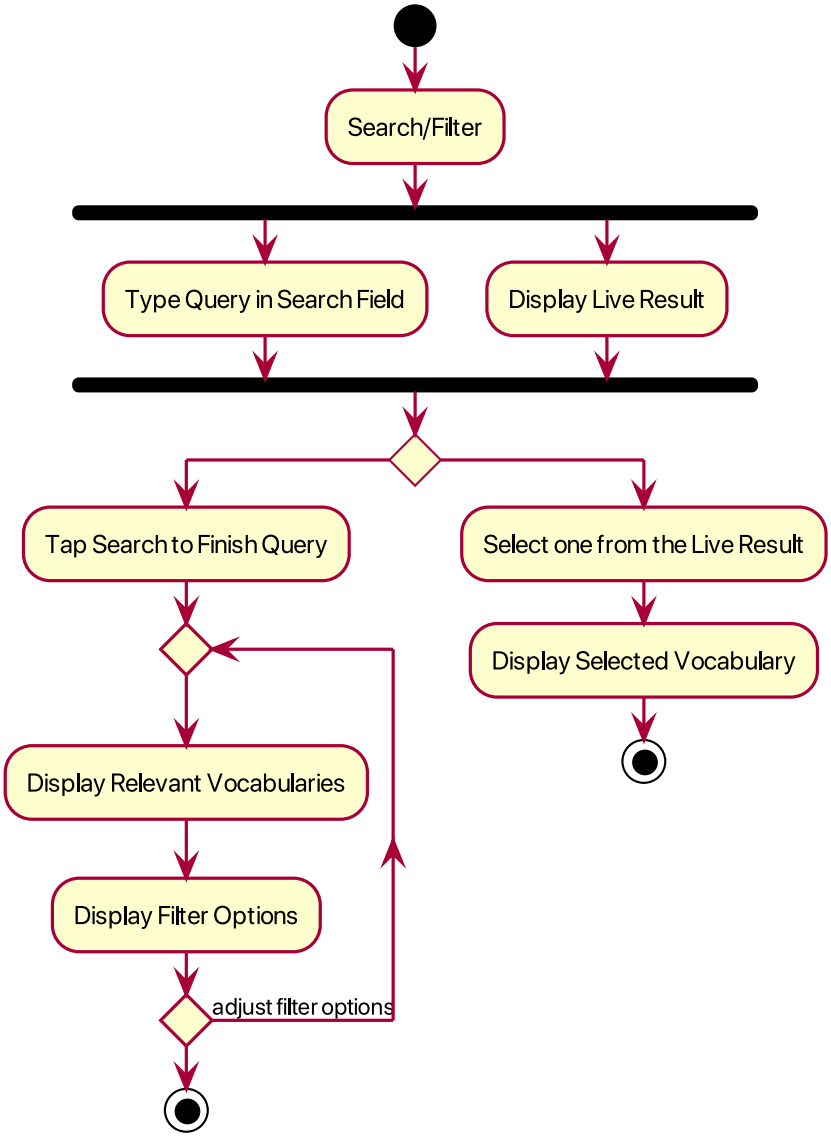

Figure 4.4: Search and Filter Activity Diagram

5 Message Sequence Diagram

This sequence diagram illustrates the process of how a user can create a new vocabulary. Within the vocabulary collection view controller, the user can tap on the add button to create a new vocabulary.

For the visual representation, see Add Vocabulary prototype.

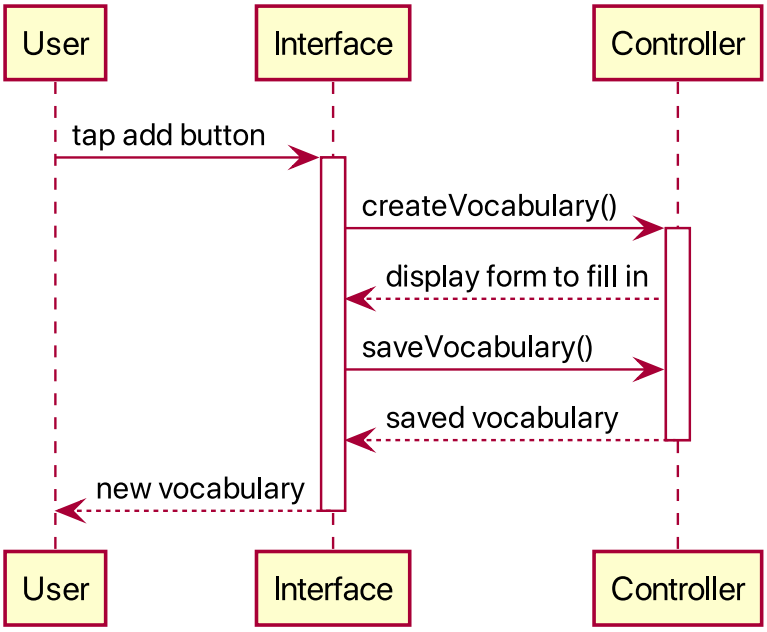

Figure 5.1: Architecture Overview Diagram

6 State Diagram

This diagram illustrates the possible transitioning states the user can be in while using the system.

- Studying - user views all the vocabularies in a selected collection.
- Selecting - user selects one of the vocabularies.
- Adding - user adds a new vocabulary to the collection.
- Searching - user searches for existing vocabularies.
- Editing - user edits an existing vocabulary.
- Deleting - user deletes an existing vocabulary

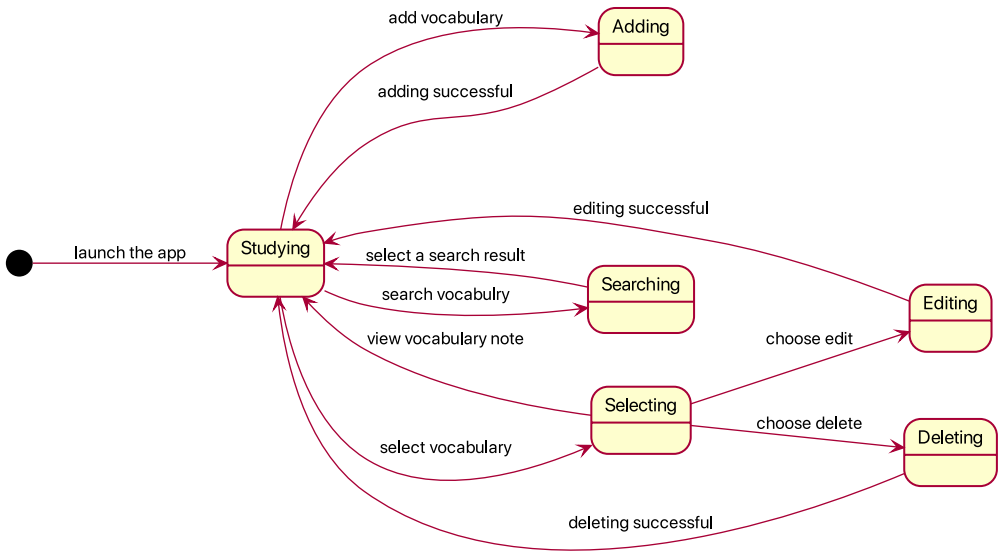

Figure 6.1: Architecture Overview Diagram

7 Class Diagrams

This diagram illustrates the flow of the application and how the view controllers communicate with its coordinator to execute the methods the coordinator implements. It also illustrates which view controllers and object models are being used by the coordinator to carry out the given tasks.

The diagram starts with `MainTabBarViewController` where it prepares all the initial data and then lets the `VocabularyCollectionCoordinator` takes over the initial flow of the application.

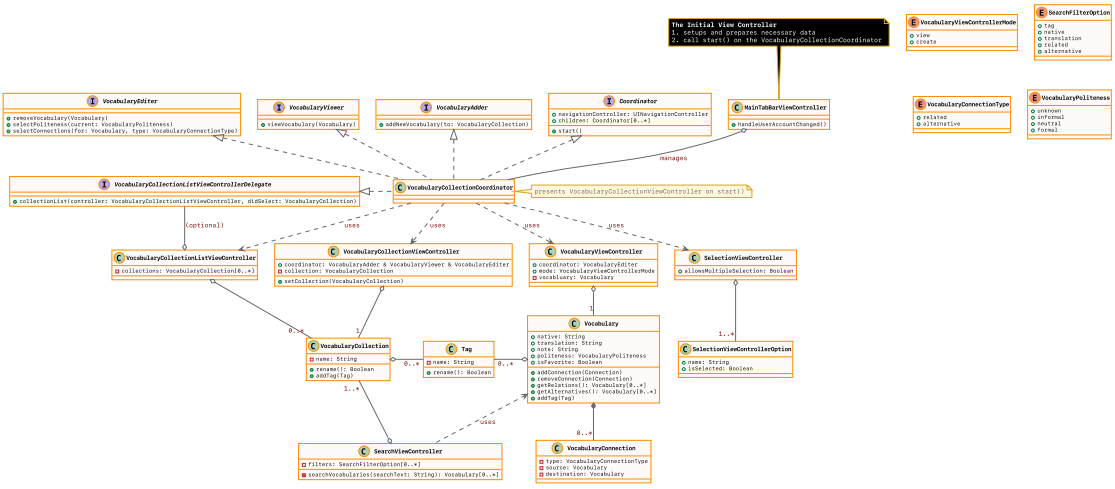

Figure 7.1: Class Diagram

8 Prototype

8.1 Horizontal

This diagram illustrates the horizontal prototype user interface in *FNote* app. From left to right are:

- Collection List View Controller - shows all user vocabulary collections
- Collection View Controller - shows all vocabularies in a collection
- Search View Controller - searches for vocabularies based on user's input and filter options.

For more detail, see Vertial Prototype.

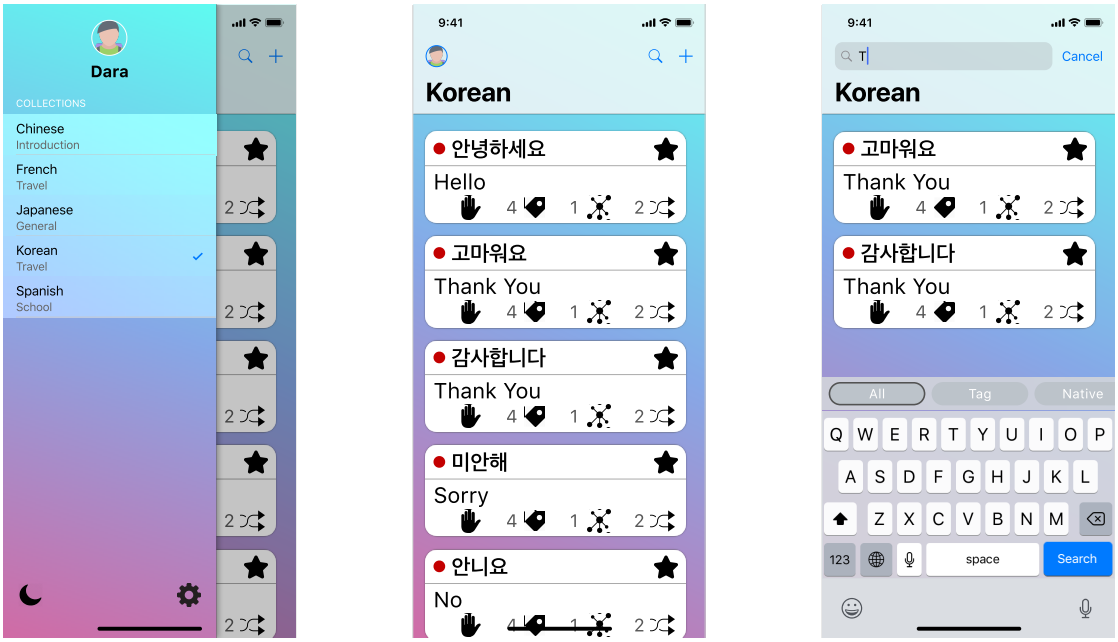

Figure 8.1: Horizontal Prototype

8.2 Vertical

8.2.1 Add Vocabulary

A prototype for adding a new vocabulary to the current vocabulary collection. User begins by tapping on the + sign button. Then a vocabulary scene will be displayed. User will need to fill in the required fields and then tap **Add**. Then the new vocabulary will be added to the vocabulary collection.

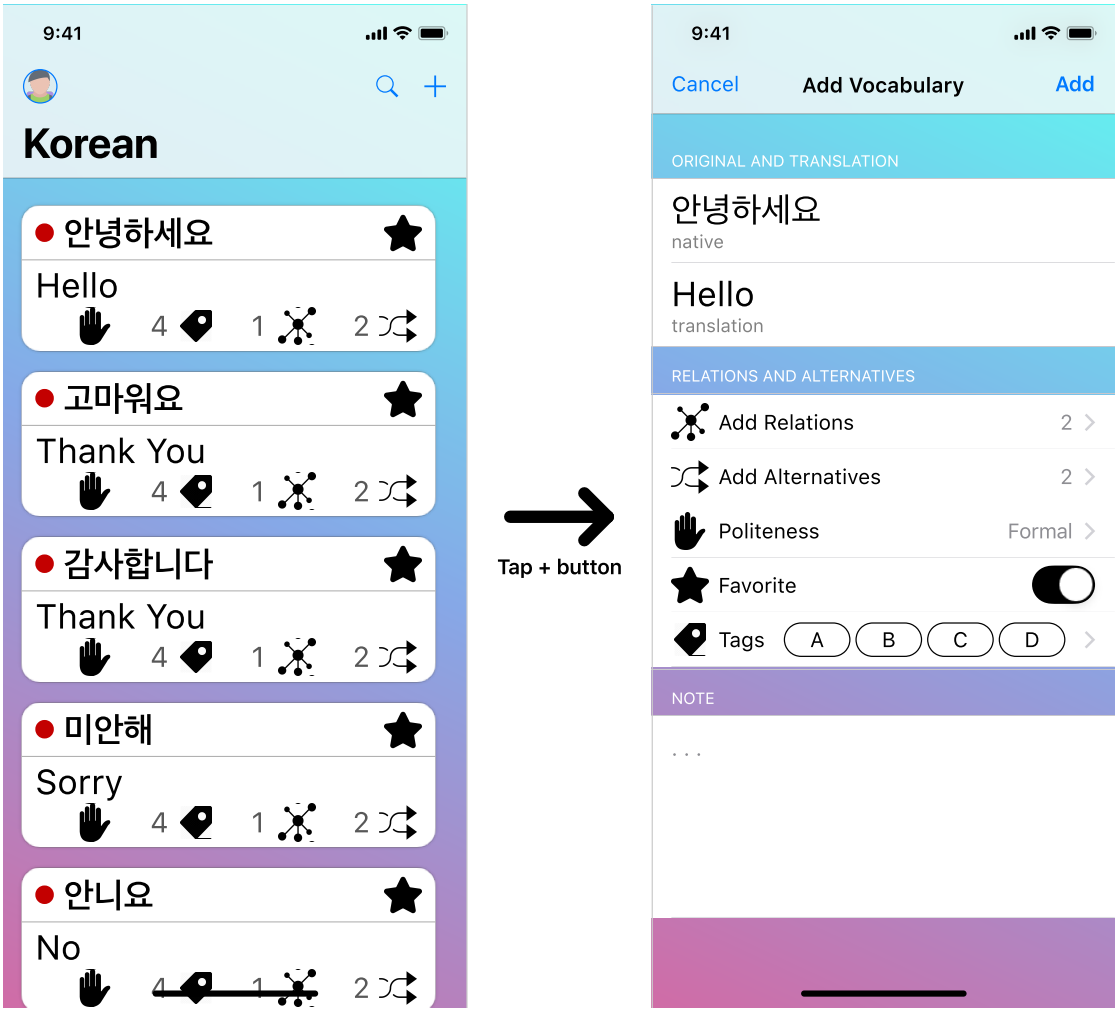

Figure 8.2: Add Vocabulary

8.2.2 View Vocabulary

A prototype for viewing a vocabulary in the current vocabulary collection. The user taps on one of the vocabulary to go to view vocabulary scene. The user can edit any fields and save to update the vocabulary.

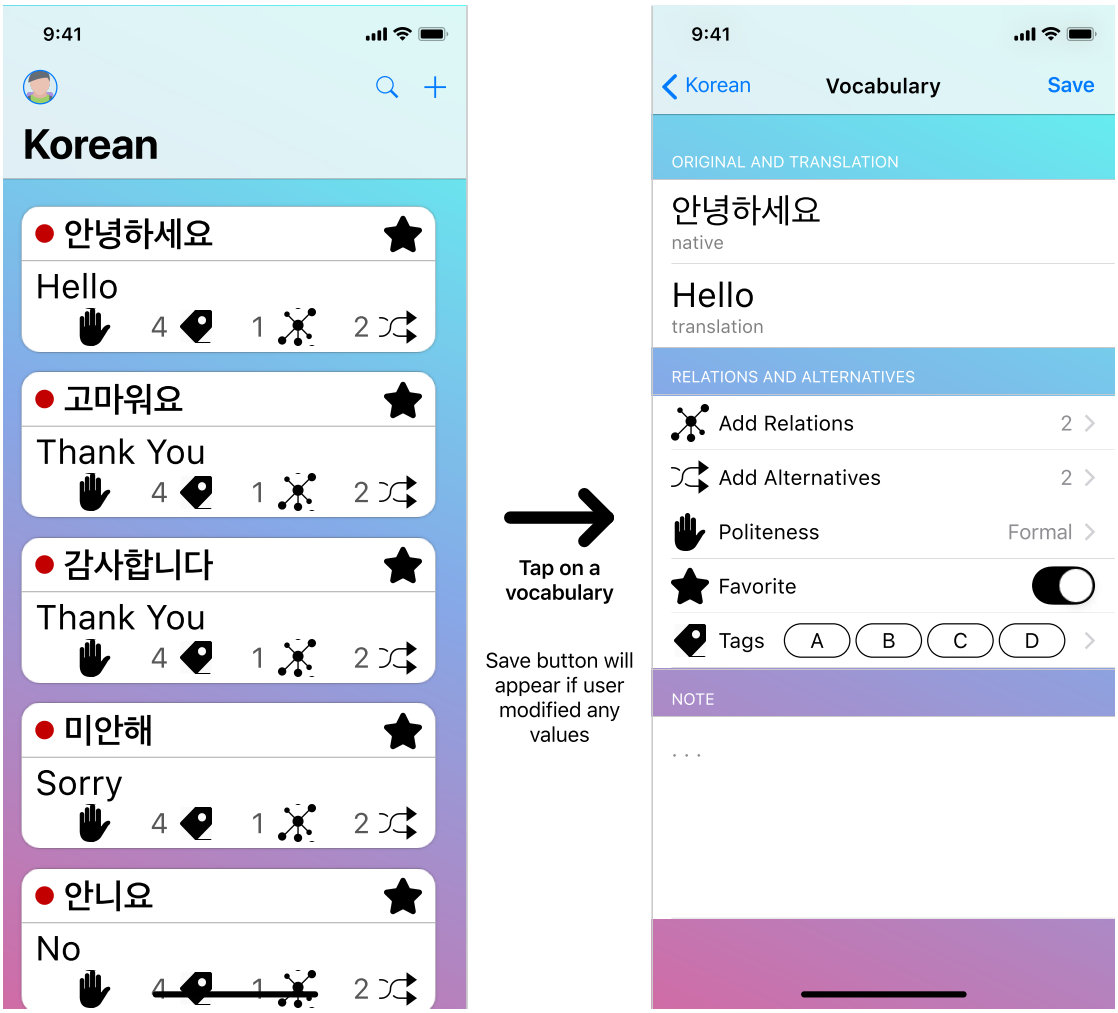

Figure 8.3: View Vocabulary

8.2.3 Add Vocabulary Relation

A prototype for adding connections to a vocabulary. In this case, the connection is of type *related* connection. Users can mark, by tapping on a vocabulary, which vocabularies are related to the current vocabulary.

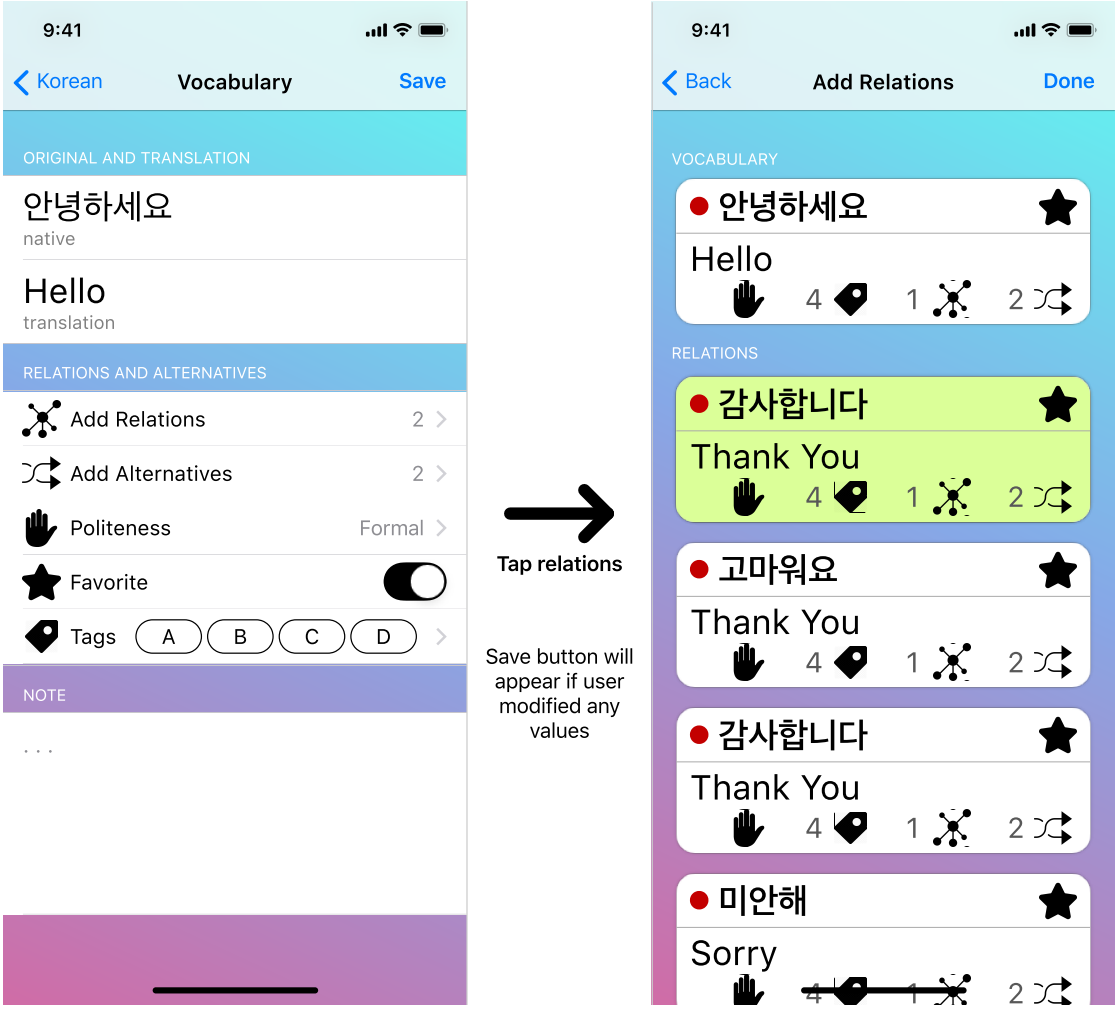

Figure 8.4: Add Vocabulary Relation

8.2.4 List of User's Vocabulary Collection

A prototype for viewing all user's vocabulary collections. Users can select one of the collections from the list to switch to that collection. To access the collection list, users will need to tap on the `Profile` icon

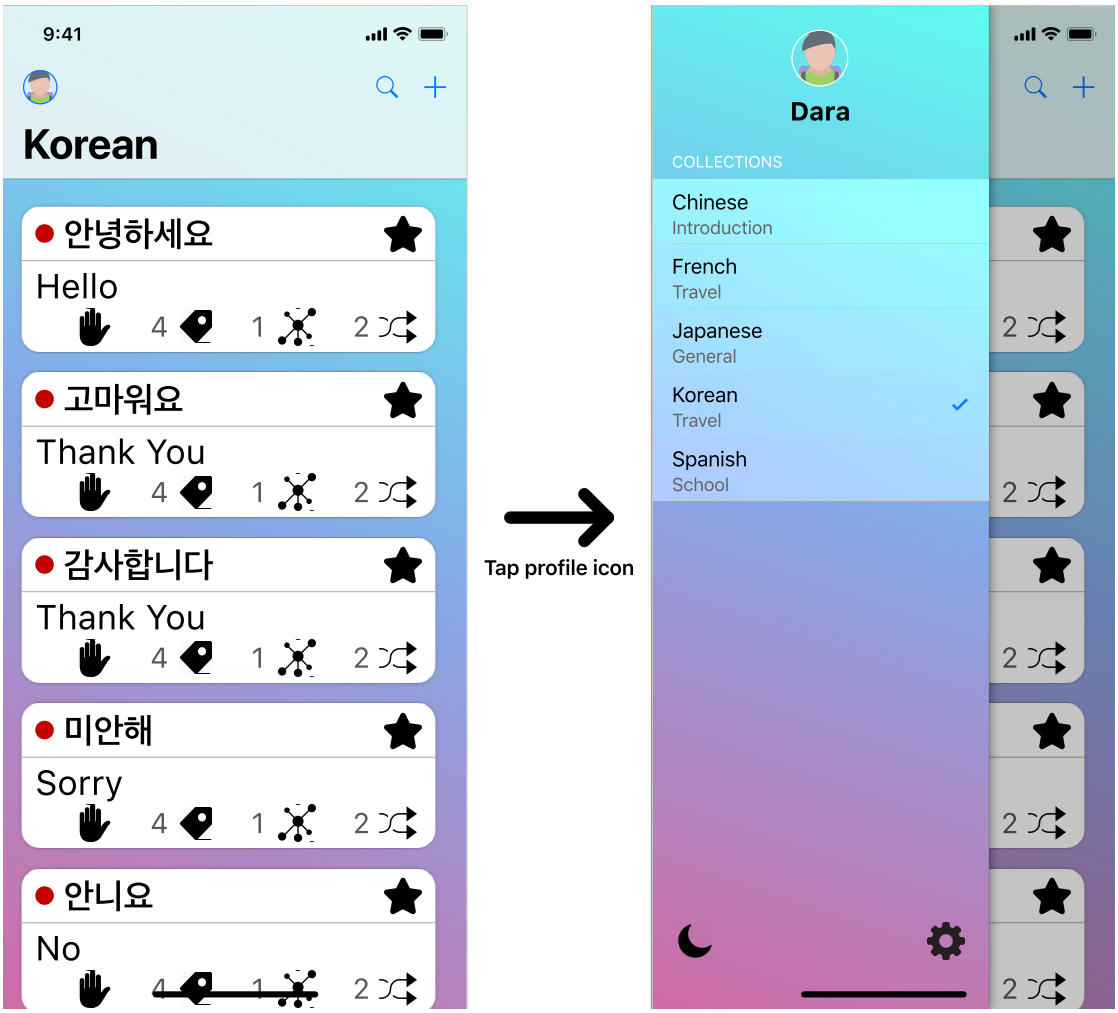

Figure 8.5: List of User's Vocabulary Collection

8.2.5 Search and Filter Vocabulary

A prototype for searching vocabulary. The search is based on user's input and filter options. Users can type in the search text to begin searching. Users can also filter the result to refine the search.

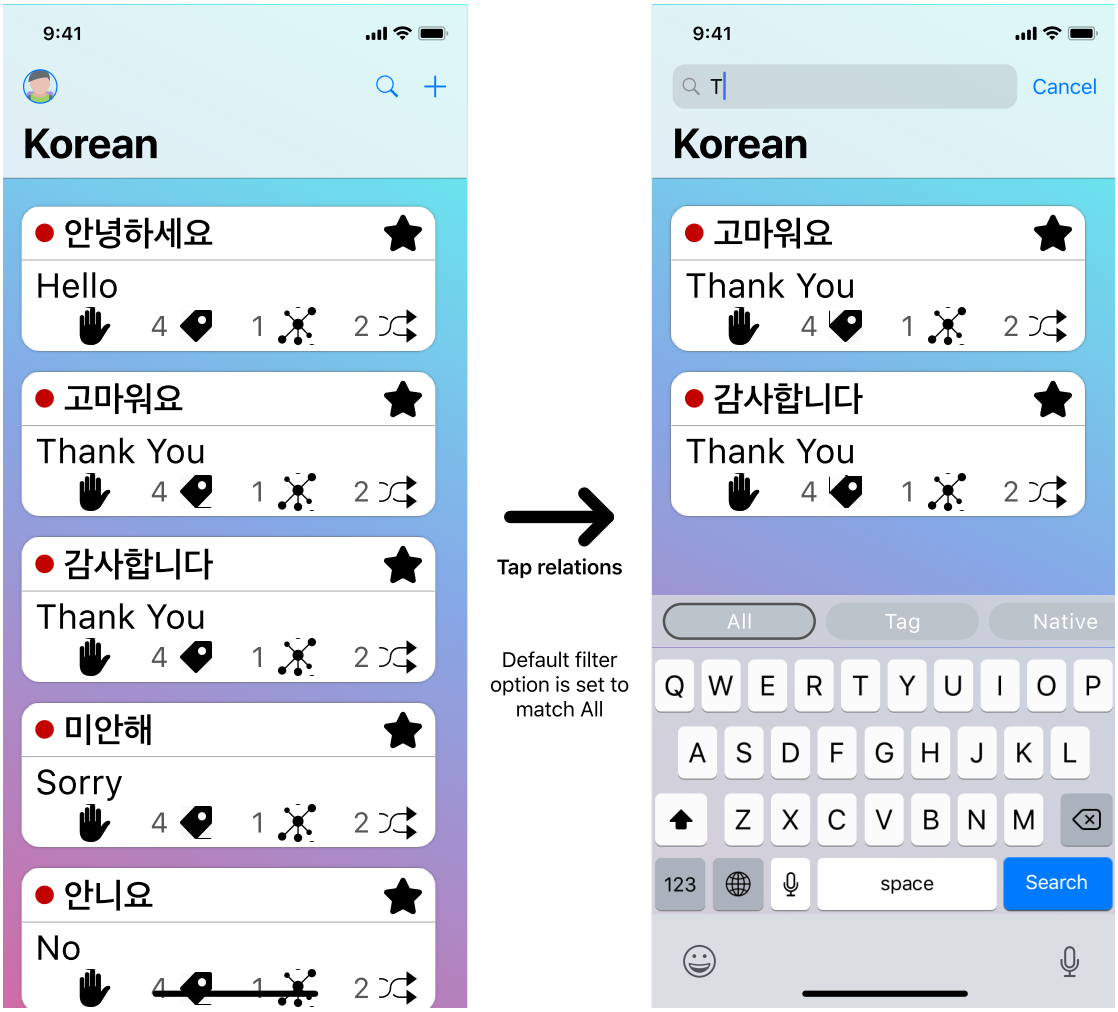

Figure 8.6: Search and Filter Vocabulary

9 Technology Stack

This diagram presents the technology stack needed to build and deploy the application.

- App Store - Apple's app distribution platform. *FNote* app will be published there.
- iOS Devices - the application will run on iPad and iPhone.
- iCloud - user's data will be saved to their iCloud account.
- CloudKit - an API, provided by Apple, used to communicate with iCloud.
- CoreData - a local database, provided by Apple, used to store data on the device.
- Swift - the programming language used to write the application.

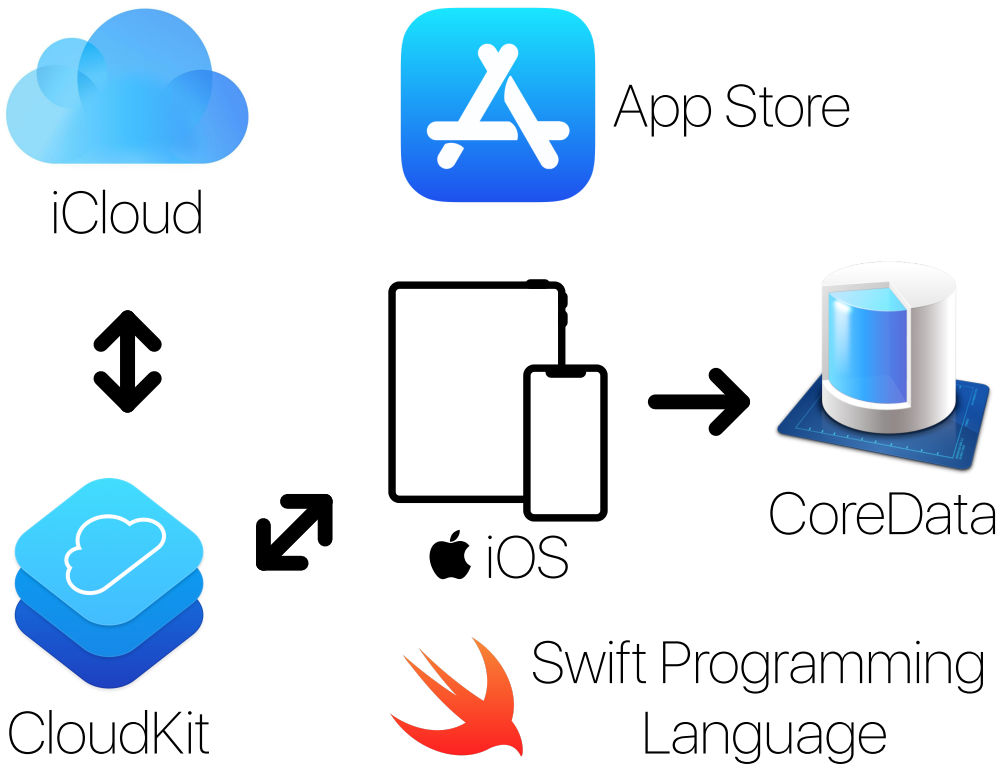

Figure 9.1: Technology Stack used to build and deploy *FNote* app