How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

GitHub Username: deansgit

Boda

Description

Collection of tools to help build your Korean vocabulary! Build your ability to sight-read syllables on the fly in multiple fonts. Learn numbers, objects, and common phrases.

Intended User

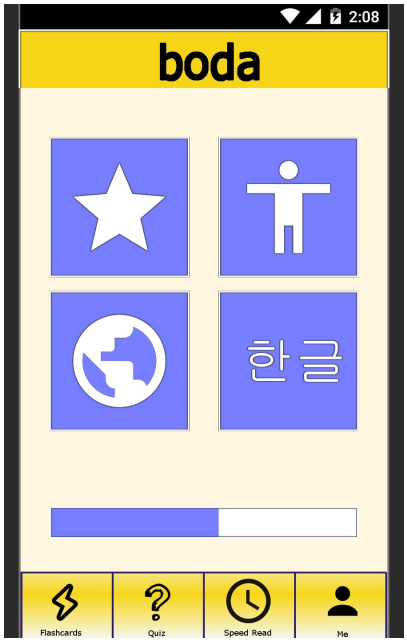
Great for beginner to intermediate students of the Korean language. Appropriate for all ages.

Features

- Select from a variety of typed and handwritten fonts
- Track progress and achievements
- Speed quizzes test your reading and typing skills.
- Categorized flashcards help learn contextually; incl. animals, foods, and idioms
- Pictures help round out the learning experience

User Interface Mocks

Main



Main screen with categories and progress bar. Progress bar fills according to ratio of objects viewed:total objects.

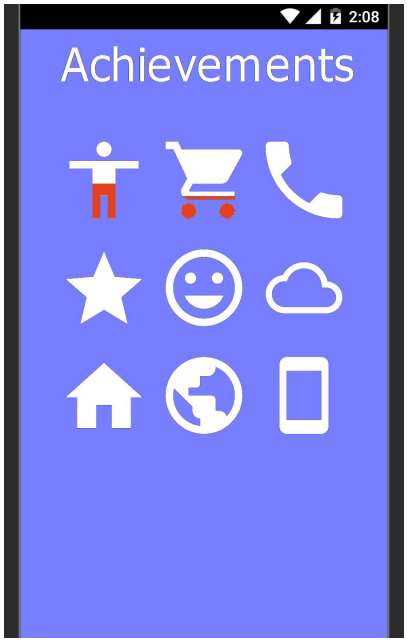
Flash Card and Syllable Speed Reader

- If image flash card, bottom text allows user to cycle through different fonts and shows English/Pronunciation when tapped.
- If syllable block, view disappears after 1 second and dialog box pops up asking for user to rewrite syllables shown and then displays both for comparison.

Quiz

Multiple choice quiz. Correct answers denoted with white and gold checks.

Achievements



Icons fill with color as each category is cleared.

Key Considerations

How will your app handle data persistence?

Create a library of mapped values for ~6,000 most common words and all 11,000+ combinations of characters and their romanization/pronunciation.

Use SQLite and SharedPreferences to store values in database, and store user data.

Describe any corner cases in the UX.

- Changing orientation should pause loop of flashcards.
- Navigation bar hides on full screen
- Back button on home screen prompts user to confirm exit, on full screen activities/fragments, brings up navigation bar
- On tap, toggle pause/play and display FAB
- Scroll vert, horiz, or flashcard style

Describe any libraries you'll be using and share your reasoning for including them.

- Glide to handle loading of images and related errors.
- Sugar ORM - for database creation and management

Next Steps: Required Tasks

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

Task 1: Data Collection

- Pull top 5,600 words from National Institute of Korean Language (via TOPIKGUIDE)
    - http://www.topikguide.com/korean-frequency-list-top-6000-words/
- Pull and format all possible korean syllables
    - http://koreangenealogy.org/book/korean-writing/hangul-syllables/
- Put together 50+ item lists of animals, nations, and idiomatic expressions
- Put together collection of free-use images for the above categories
    - www.pixabay.com or www.gettyimages.com
    - Map out images with corresponding items onto csv
- Using Sheets or Excel, format data so as to make it compatible with SQL tables
- Collect 3 each of handwritten and typed Korean fonts.
    - http://software.naver.com/software/fontList.nhn?categoryId=I0000000

Task 2: Project Setup

- New project with full screen activity.
- Add lists from Task 1 (key-value pairs)
- Add dependencies:
    - Sugar ORM (for SQLite management):

      ```
      compile 'com.github.satyan:sugar:1.4'
      ```

    - Glide (for image management):

      ```
      compile 'com.github.bumptech.glide:glide:3.7.0'
      compile 'com.android.support:support-v4:19.1.0'
      ```

- Create database using lists and image ids

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
    - Vertical, Landscape, Tablet support using Master Detail Template
    - Build Layout for Bottom Navigation Bar
- Build UI for all Top-Level Fragments
    - Flashcards - gridview of categories, select to start activity
    - Quiz - gridview of categories, select to start activity
    - Speed Reader - select number of syllables, click go to start
    - User Profile - list of achievements, settings button
- Create full screen activities for Flashcards, Quiz, Speed Reader, and Settings

Task 3: Implement Flashcards Activity

- Use ViewPager and FragmentStatePagerAdapter
- Create layout -- image, text, text, two buttons
- Randomly generate list of 5 items
- Use Glide to populate pics
- Tap buttons to change font, tap text to show romanization
- Implement instructive motion (slide in) on enter transition

Task 4: Implement Quiz Activity

- Use ViewPager and FragmentStatePagerAdapter
- Create layout -- image, text, text, four buttons
- Tap buttons to change font, tap text to show romanization

Task 5: Implement Speed Reader

- Take number of syllables set by user, retrieve random syllable(s) into list
- Reuse most of layout from flashcard
- Syllables set to disappear and text field for answer
- Accepts hangeul and romanization (RR Transliteration scheme)
- Compare. Score at the end (“x / 10 correct!”)

Task 6: Implement User Profile

- Create table for achievements

- Create table for settings
- Refactor each activity to write results of each exercise to table
- Summarize on User Profile Activity
- Implement Settings Activity
    - Gender, age, location, default font,

Task 7: Polish

- Handle errors
- Add transitions
- Add weekly notifications encouraging user to practice.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"