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: Kagen47

Ayudanki

Description

Memorize words you want to remember by quizzing yourself.
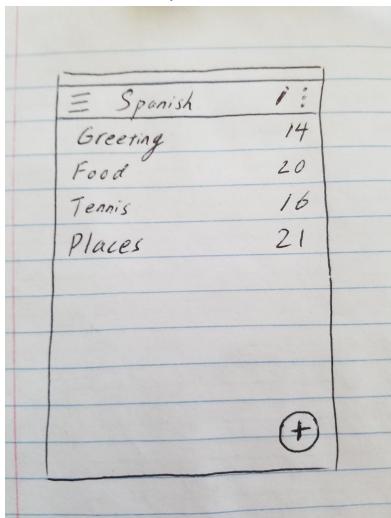
Intended User

Students

Features

- Saves pairs of term and definition the user enters
- Creates quizzes from the words

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

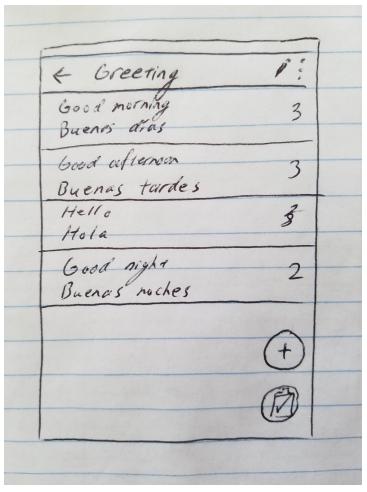## Screen 1 MainActivity

Launch screen.
The hamburger menu will be used to select the category (in this case Spanish).
The last item in the hamburger menu is an add button to add a new category.
Users will be able to add a subcategory with the Floating Action Button with the plus sign.

For each subcategory the number of terms will be shown in the right.

Screen 2 QuizInfoActivity
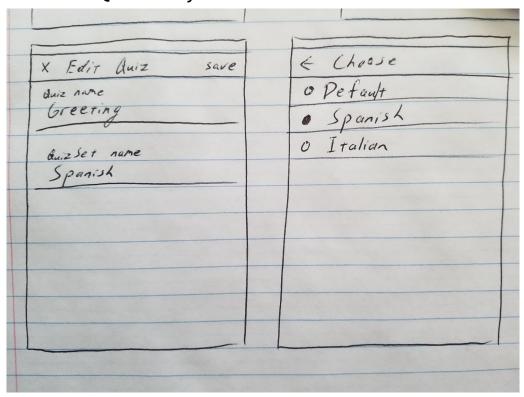

This screen shows up when you tap the subcategory.
Each list item shows the term on top and definition on the bottom.
The number on the right shows points on the term, which goes up by answering correct in the quiz.
The Floating Action Button with the plus sign displays a dialog to add a new term.
The Floating Action Button with the check sign starts the quiz.
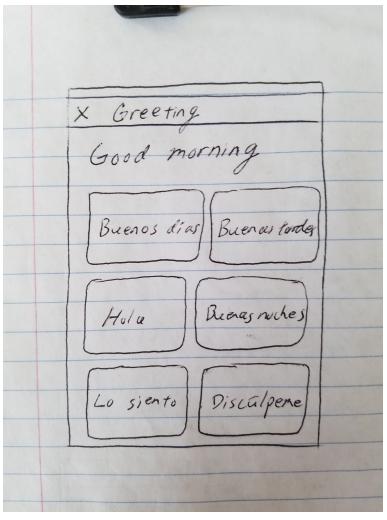
Screen EditQuizActivity

This screen on the left shows up when you tap Floating Action Button in screen 1 or the pencil icon in screen 2.
Users can edit the name of the subcategory and also move the subcategory into a different category.
Tapping the textinput of QuizSet name opens the screen on the right.
Choosing one item on the right screen updates the QuizSet name input.

Screen 4 QuizActivity

This screen on the left shows up when you tap the check Floating Action Button in screen 2.
The user is quizzed on the words in the subcategory, in this case "Greeting".
When the user chooses the correct answer the CardView will turn green and a Floating Action Button with the icon "→" will appear to move to the next question.
When the user chooses the wrong answer the CardView will turn red and a Floating Action Button with the icon "→" will appear to move to the next view which will show the correct term and definition.

Key Considerations

How will your app handle data persistence?

Content Provider with 3 tables. QuizSet, Quiz, and Card.
QuizSet columns: id, name
Quiz columns: id, name, quizSetId
Card columns: id, term, definition, points, quizId

Describe any corner cases in the UX.

If the user deletes all QuizSets, create a default QuizSet.
Show error on EditText if user does not enter anything for required fields.

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

Firebase Realtime Database for reduce storage space on device.
Admob for advertisement

Describe how you will implement Google Play Services.

Ads will be displayed at the bottom of the screens.
Firebase Realtime database will be the same structure as the content provider, so the terms can be downloaded for offline use.

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: Project Setup

- Configure libraries
- Create BuildConfig constants for release and debug

Task 3: Create Content Provider

- Create contract
- Create OpenDbHelper
- Create Content Provider

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Implement Navigation Drawer for MainActivity
- Build UI for QuizInfoActivity
- Build UI for EditQuizActivity

Task 4: Implement Dialogs for adding quizzes

- Create AlertDialog for creating QuizSet
- Create AlertDialog for creating Card with term and definition

Task 4: Implement quiz

- Build UI for QuizActivity
- Create Quiz class to generate quiz using the terms and definitions

---

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**"