

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen: Subjects](#)

[Screen: Card Decks](#)

[Screen: Study Mode](#)

[Screen: Tablet Subjects/Card Decks](#)

[Screen: Quiz Mode](#)

[Screen: Statistics](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share you're reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task: Create initial project and repository](#)

[Task: Design database structure](#)

[Task: Implement 'Subjects' activity and fragment](#)

[Task: Implement 'Subjects' CRUD](#)

[Task: Implement 'Card Deck' activity and fragment](#)

[Task: Implement 'Card Deck' CRUD](#)

[Task: Implement 'Study' activity and fragment](#)

[Task: Implement 'Quiz' activity and fragment](#)

[Task: Implement tablet optimized Activity](#)

[Task: Create widget to provide relevant information](#)

[Task: Integrate at least two Google Play services](#)

[Task: Implement ShowcaseView in app](#)

[Task: Verify views have content descriptions and supports RTL layout](#)

[Task: Review Rubric criteria](#)

[Task: Populate shipping database with example data](#)

[Task: Test on various devices](#)

[Task: Submit project for review](#)

GitHub Username: www.github.com/glennmiller36

Studiosity

The quality of being studious.

Description

Studiosity is an effective flashcard memory-aid tool to help students learn and remember new material quickly. Condense and simplify information so that you only learn the important facts and ideas.

Supports two modes: Study mode displays the list of all questions and answers and Quiz mode randomly shuffles the cards to prevent students from simply memorizing the order of answers. Quiz mode can reverse the flash cards answers can be seen first and students must surmise what the original questions were.

Highlights:

- Create/Edit your own flashcards within the app
- Supports two modes: Study and Quiz mode
- Custom text and background colors
- Simple and easy-to-use
- Charts for progress tracking
- Makes studying fun, easy, and effective

Intended User

This app targets students who need help to learn and remember new material – from elementary through college levels.

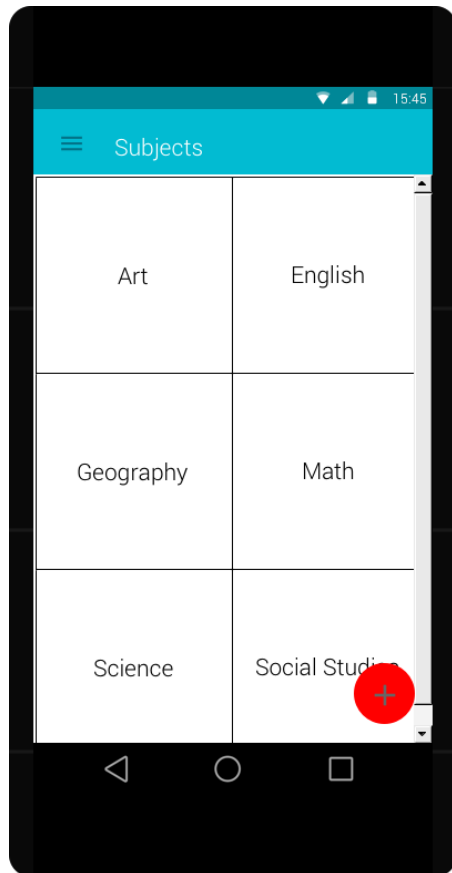
Features

Main features:

- Create / Read / Update / Delete flashcards to a local SQLite database
- Display a flashcard question to the user and allow the user to indicate if they gave a correct or incorrect answer.

User Interface Mocks

Screen: Subjects

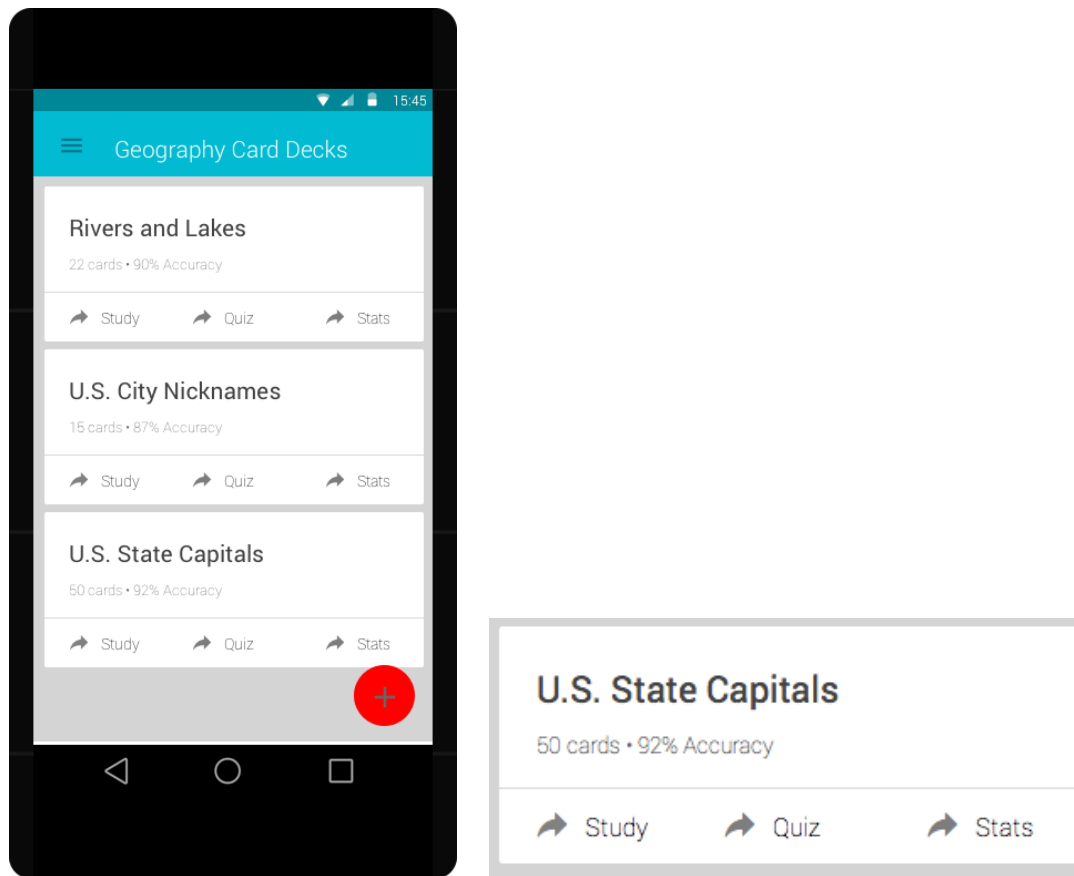


Display a list of school Subjects the user has created. Flashcards are grouped into Subjects.

Navigation:

- The user can click on an individual Subject to drill-down to a list of Card Decks defined for that Subject.
- The user can click the '+' to add a new Subject

Screen: Card Decks

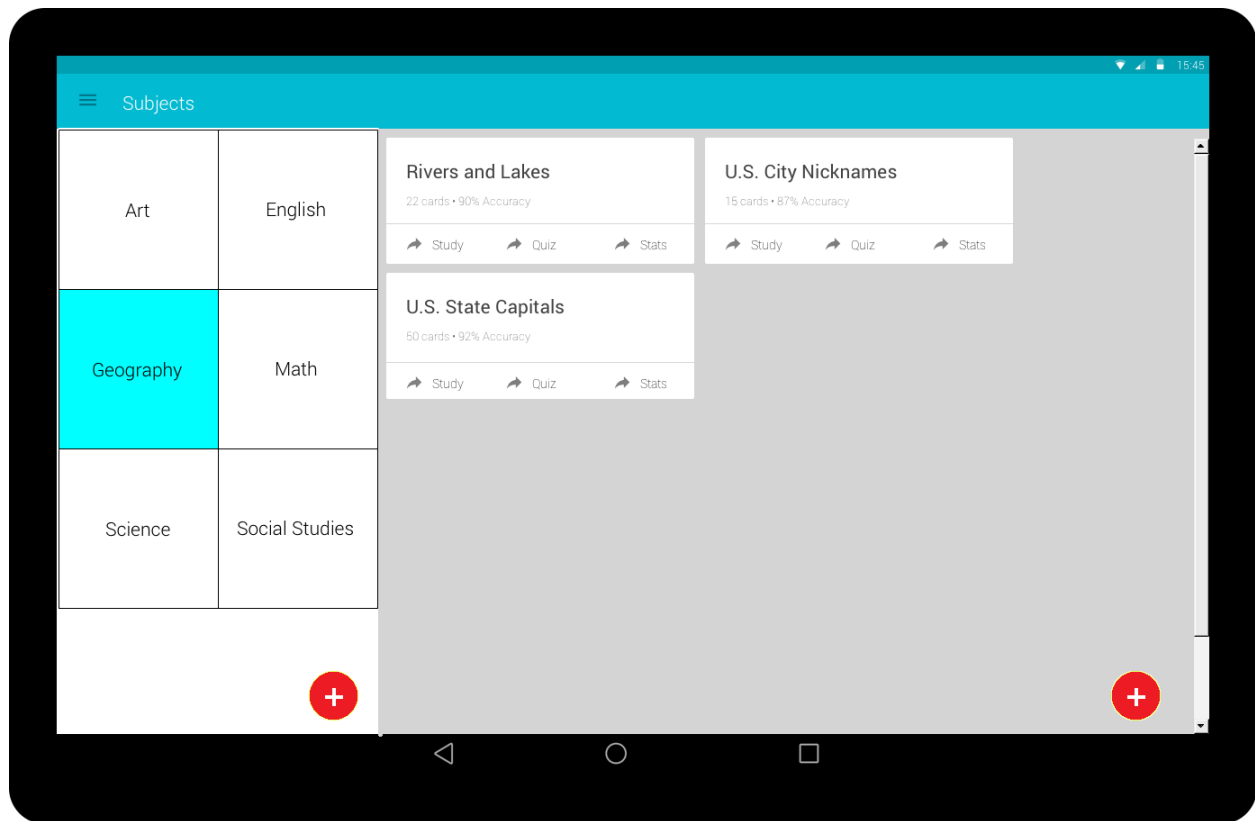


OnClick of a Subject will navigate to the list of all Card Decks the user has created for the selected Subject.

Navigation:

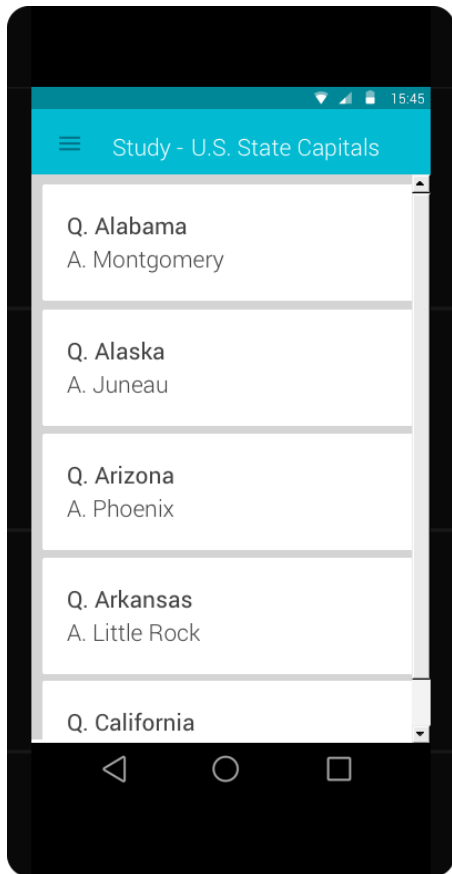
- The user can click on an individual Card Deck to drill-down to the details where they can Edit or Delete the Card.
- The user can click on the Card Deck links: Study | Quiz | Stats.
- The user can click the '+' to add a new Card Deck

Screen: Tablet Subjects/Card Decks



Subject and Card Decks will be Master/Detail on the tablet.

Screen: Study Mode



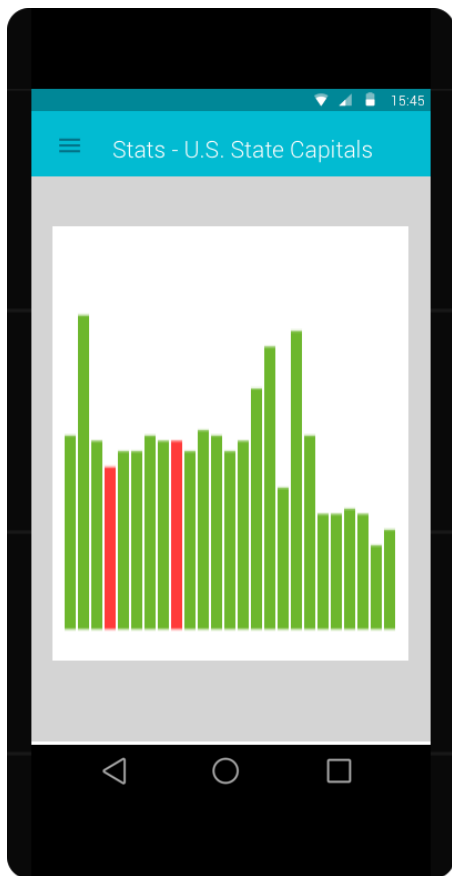
Study Mode allows the user to scroll through all questions with the corresponding answer so they can learn and remember the material.

Screen: Quiz Mode



Quiz Mode displays all Card Deck questions in a random order so the user can vocally or mentally say the answer. The user can swipe the sliding cover to reveal the answer then record if they got the question right or wrong.

Screen: Statistics



Displays statistics about how many questions the user answered correctly each time they took a quiz.

Key Considerations

How will your app handle data persistence?

Data will be stored in a local SQLite database. A CursorLoader will run asynchronous queries in the background against a ContentProvider to return the results to the Activity or Fragment. Even though the app will *not* be sharing data with other applications, still decided to go with a ContentProvider because it will provide an extra level of abstraction over the database.

Describe any corner cases in the UX.

If a user hits the back button while in the middle of taking a Quiz, the Card Decks activity will display a prompt or menu giving the user the option to resume taking the Quiz.

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

ShowcaseView – highlight and showcase how to use the app during the users initial app launch.

Ott or EventBus – de-couple components by having them communicate with each other without the need to hold a direct reference of each other.

MPAndroidChart – create nice looking charts for progress tracking.

Next Steps: Required Tasks

Task: Create initial project and repository

- Configure libraries
- Framework for EventBus

Task: Design database structure

- Determine what database tables and columns will be needed
- Create initial ContentProviders

Task: Implement 'Subjects' activity and fragment

- Implement CursorLoader to fetch Subjects from database
- Initially designed for phone – later task to implement Master – Detail for tablet

Task: Implement 'Subjects' CRUD

- Implement activity and fragment to allow Create and Update of Subject name
- Implement ability to Delete a Subject

Task: Implement 'Card Deck' activity and fragment

- Implement CursorLoader to fetch Card Decks from database
- Initially designed for phone – later task to implement Master/Detail for tablet

Task: Implement 'Card Deck' CRUD

- Implement activity and fragment to allow Create and Update of Card Decks name
- Implement ability to Delete a Card Deck

Task: Implement 'Study' activity and fragment

- Implement CursorLoader to fetch Cards from database

Task: Implement 'Quiz' activity and fragment

- Implement CursorLoader to fetch Cards from database
- Randomly generate what order to display the questions
- Implement slide cover functionality (thirdparty library??)
- Implement 'Resume Quiz' in Card Decks activity incase Back button pressed

Task: Implement tablet optimized Activity

- Master/Detail screen for Subjects/Card Decks

Task: Create widget to provide relevant information

- Home screen widget

Task: Integrate at least two Google Play services

- Probably use mobile ads and analytics

Task: Implement ShowcaseView in app

- Teach first-time users how to use the app

Task: Verify views have content descriptions and supports RTL layout

- Test all views to verify they satisfy requirements

Task: Review Rubric criteria

- Verify all requirements have been met

Task: Populate shipping database with example data

- Ship with Sample data so user can play with app without having to set it up first

Task: Test on various devices

- Create signed .apk
- Send .apk to friends with various Android devices and let them play with the app

Task: Submit project for review

- Yeah!