

SENG 303 - 2024 Mobile App Design and Development

Assignment 1 - Flash Card App

Overview

Your responsibility in this project is to get acquainted with the Android user interface system and its event-driven programming model. You will do this by writing an app that makes use of the features that we discuss in Term 1 of the semester. The app you will be building is a Flash Card App for creating, editing, and playing multiple choice type flash cards.

Requirements

On Learn we provide a **reference implementation video** showing the features that your app must implement. You should recreate the functionality of the app shown in the video as faithfully as you can (though not necessarily look and feel, which we will discuss more below).

To prepare your submission follow these requirements:

- **Use Kotlin** for program logic, *not* Java.
- **Create your user interface declaratively** with Jetpack Compose, *not* imperatively through Kotlin (non-Compose) code or using XML Views.
- **Maintain and submit your project** in a Git repository on the departmental GitLab server, and add Ben (bta47) as a member.
- **Document your app**, using at most 1 page, and upload to Learn. In your document, briefly discuss your development process. Enumerate which of the reference implementation features you believe you have met and how. Be brief but specific in your enumeration. Include screenshots and the link to your Git repository in your report.

This is an **individual project**. You may not collaborate or share code while working on this assignment.

Submission due date: Thursday, August 22nd @ 5pm.

Drop dead date (no penalty): Tuesday, August 27th @ 5pm. Please note, however, no tutor or lecturer support will be available after the due date above.

App features

The following is a (possibly non-exhaustive) list of features that are shown in the reference implementation video:

- App Icon
- Splash screen with animation
- Home page containing three button options:

- View flash cards – navigates to screen to see existing flash cards
 - Create flash card – navigates to screen to create a new flash card
 - Play flash cards – navigates to play screen to test your knowledge
- Create flash card screen
 - Input fields for question and answer texts
 - Selection for the correct answer
 - Button to add additional answer fields
 - Save and return button that adds the flash card
 - Input validation with Toast messages
- View flash cards
 - Message if there are no flash cards created
 - List of flash cards
 - Uniquely labeled by question
 - Search icon button which starts a common Intent to open a browser to perform a web search on the card's question
 - Edit icon button to open “edit flash card” screen (with similar functionality to “Create flash card”)
 - Delete icon button to remove a flash card from the list
 - Delete action has confirmation dialog box
- Play flash cards
 - Iterates through each flash card, showing question and possible answers.
 - Selection allows the player to select one answer.
 - Input validation with Toast messages requiring the player to select at least one answer.
 - Submit button
 - Progress counter showing which question the user is on
 - Toast message saying right or wrong answer after each submission.
 - After submitting final question shows results with score and which questions the player got right or wrong
- Managing Lifecycle
 - App must persist flash card data, so if the app is closed and restarted they will remain.
 - App must work seamlessly in portrait and landscape modes, and field input data must be retained after rotation.

Additional optional features not shown in video:

- Deleting a question on the create/edit flash card page
- Randomized order of answers each time a flash card is shown
- Randomized order of flash cards
- Entry of player name at beginning of play and high score screen
- Other features you think of (please mention in your 1-page documentation)

App look and feel

The reference implementation shows one possible design for your app, but you should feel free to choose your own style as long as it maintains the functionality described above.

Marking guideline

Marking will be based on a holistic assessment of features + UI design of your app. If you recreate the features shown in the video with a basic UI design and clear documentation, then you can expect to get a B-range grade. If you do not implement all the features or your app does not fully function, you will get a lower grade. An A-range grade requires you demonstrate substantial effort on creating a smooth UX design (including look and feel mechanisms such as animation) and/or implementation of additional optional features.

v1.1 2024/07/26