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: rahul09m

Smart Cards

Description

Smart Cards allows users to store loyalty cards electronically on their phone. Users can add cards by scanning the barcode portion of their cards. Their saved cards are displayed in an organized listview and upon selecting a card, a scannable barcode is generated and displayed for use.

Intended User

Anyone who uses loyalty cards, particularly individuals with many loyalty cards

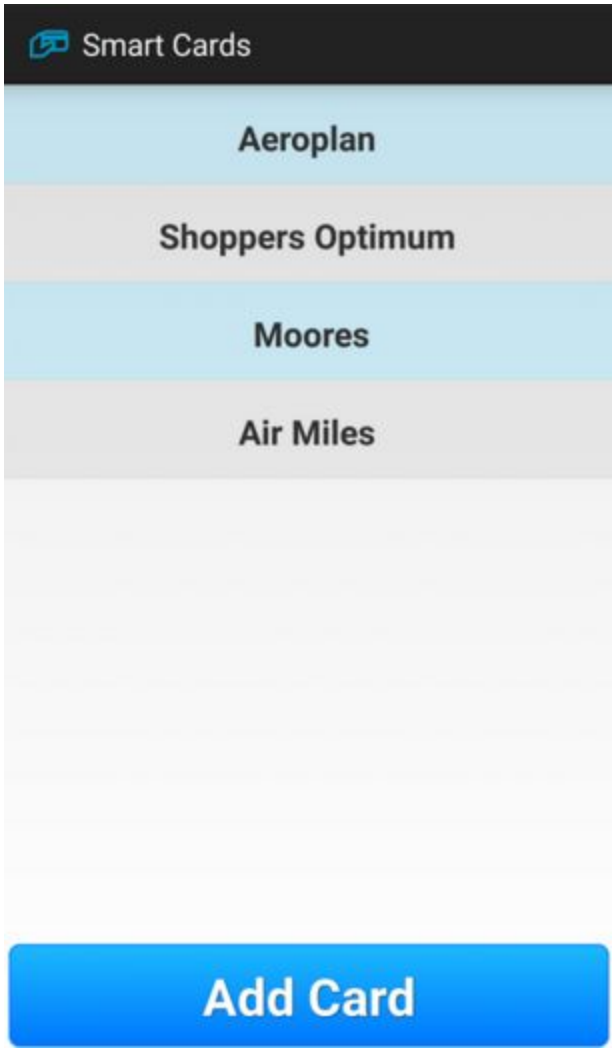
Features

- Stores Loyalty Cards via barcode scanning
- Display loyalty cards in a scannable manner
- Allows user to share their cards

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



A listview of the user's cards ready for selection and a button for adding a card

Screen 2

A selected card being displayed for scanning by the vendor. If the vendor is unable to scan, they can read the card number and enter it in manually.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

How will your app handle data persistence?

A Content Provider backed by SQL Lite will be used to store card data.

Describe any corner cases in the UX.

When on the card detail screen (screen 2), users can hit the back button to return to the main listview screen (screen 1)

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

Zxing to enable barcode scanning for adding cards. Picasso for loading the card barcode image in the detail view.

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 zxing - get card data by scanning the card
- Build Content Provider - determine data to save and respective columns
- Configure Picasso for card display.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity - listview of available cards
- Build UI for DetailActivity of selected card.
- Build UI for adding cards
- Build UI for editing and deleting a card
- Consider multiple screen size buckets

Task 3: Core Functionality

- Integrate adding cards - transition from scanning card to saving the data in SQL database
- Integrate detail view - transition from selecting a card to displaying it in detail view.
- Enable Editing and Deleting a card.

Task 4: Handle Error Cases and State Changes

- Handle error cases - scanning ,adding, editing, deleting, displaying
- Handle screen rotation and saving state.

Task 5: Enable Google Services and Build Widget

- Enable Admob for displaying ads in listview screen
- Enable Google Sign In for backing up cards to user's accounts
- Build a widget for easy card access to frequently used cards

Task 6: Accessibility and Cleanup

- Add accessibility support
- Code cleanup and refactoring.
- Test on different screen sizes and adjust layouts as needed

Add as many tasks as you need to complete your app.

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**"