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:

https://github.com/printart/Capstone-Project/tree/master/Stage1

Payment Reminder

Description

Make your credit card, mortgage, car, and all other payments on time! Don't miss due dates, don't pay extra fees for late payments, don't ruin your credit! This app will help you track your payments, remind you of the due dates, even let you know of the payment card expiration dates.

Intended User

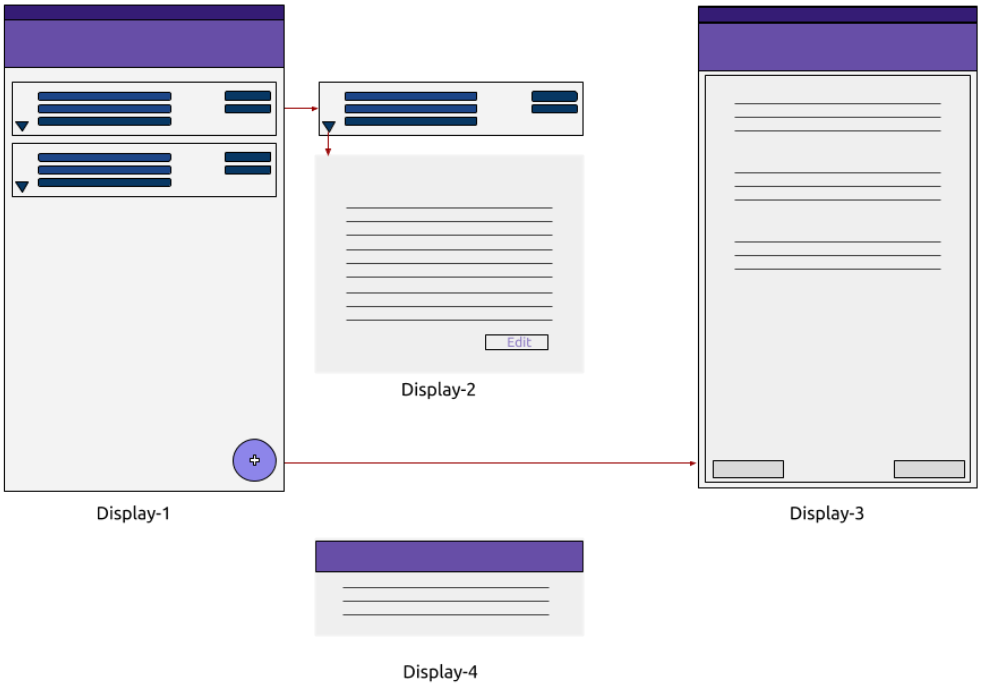
Everyone who has bills to pay

Features

- User sets up one time payments or recurring payments notifications
- User gets a notification of payments due
- No need to save sensitive information
- Data can be backed up on Google Drive

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.



Display-1 is MainActivity that contains RecyclerView and shows the main information. By clicking on down arrow (Display-2) view is expanding to display more information with capability to edit current data. Display-3 opens new window to add payment data with create/cancel buttons. Display-4 app widget.

Key Considerations

How will your app handle data persistence?

Data will be handled on phone in SQLite which will populate RecyclerView through data binding. App will use content provider for CRUD and Loaders to pull data from DB. App will implement IntentService for Google Drive backup at user's request.

Describe any corner cases in the UX.

After filling up payment details (Display-3), user clicks on Create button. This will create DB entry, finish current activity and return to MainActivity.

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

App is using RxJava and RxAndroid to simplify network connection and data stream control. It also uses Google Play services library to use Google API.

Describe how you will implement Google Play Services.

App will use Google Drive Android API to allow user to backup data on Google Drive by including in gradle dependency
compile 'com.google.android.gms:play-services-drive:^10.2.1'
And also by creating credentials in Google console.
^ - most recent

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

Add to gradle:

1. Project is using Data Binding Library
    - dataBinding {
      enabled true
      }
    - all UI xml should appear between <layout></layout>
2. Project is using RxJava and RxAndroid
    - compile 'io.reactivex.rxjava2:rxjava:^2.0.8'
    - compile 'io.reactivex.rxjava2:rxandroid:^2.0.1'
3. Project is using RecyclerView
    - compile 'com.android.support:recyclerview-v7:^25.3.1'
4. Project is using Google API
    - compile 'com.google.android.gms:play-services-drive:^10.2.1'
    - Create credential in Google console

^ - most recent

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for PaymentActivity
- Bui

Task 3: Create Classes

- Create MainActivity+Fragment
- Model class for data binding
- Create RecyclerView with data binding
- Create PaymentActivity+Fragment
- Create ContentProvider
- Create a class that implements LoaderCallbacks
- Create IntentService for backup user data to Google Drive

- Create DB
- Test DB

Task 4: Google console

- Set credentials for app in google console
- Test setup

Task 5: Finals

- Test app
- Push to github
- Submit
- Remove credentials from console

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**"