How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: “**Capstone_Stage1**”
3. Replace the text in green

GitHub Username: baileytye

Decisive

Description

Decisions made easy

Decisive lets you monitor and weigh options when making a significant purchase. Create custom requirements, input their importance, and let Decisive do the hard work behind the scenes to give you an overview of what options suit you best. Still can’t make up your mind? Share the project with friends and get their opinion!

Intended User

This app is for anyone that cannot make up their mind when making a purchase. It is designed for more expensive purchases, but can be used for anything, even items without a price.

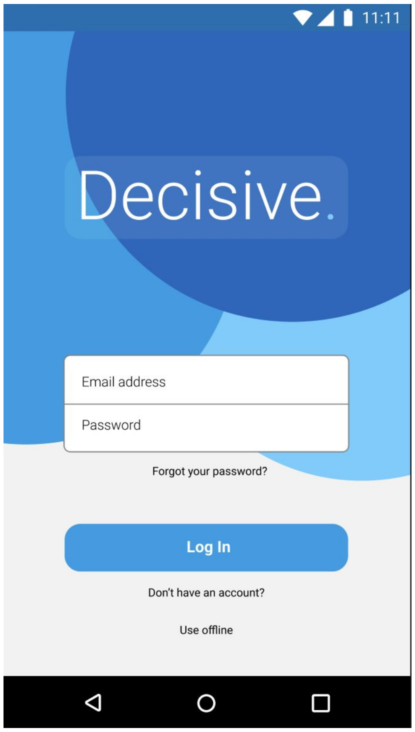
Features

Main features:

- Stores user projects to help make decisions when making an expensive purchase.
- Using requirements provided by the user, the app calculates and displays an overall rating of each option.
- See detailed analysis of how overall ratings are calculated.
- Share projects with friends to get their opinion on which option to buy - also add notifications to inform user when a friend has voted on an item.

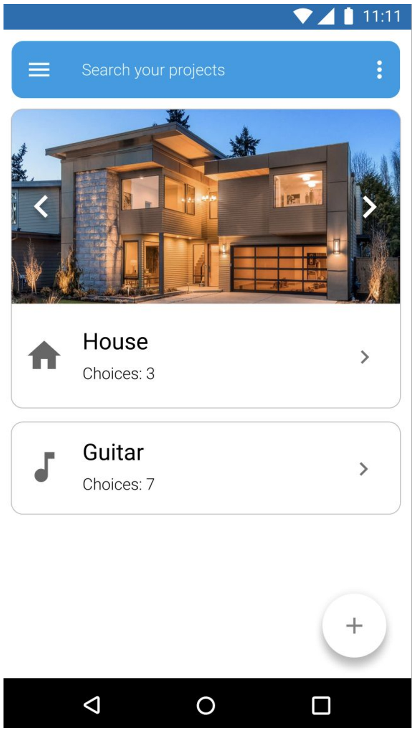
User Interface Mocks

Screen 1: Sign in for Paid Variant



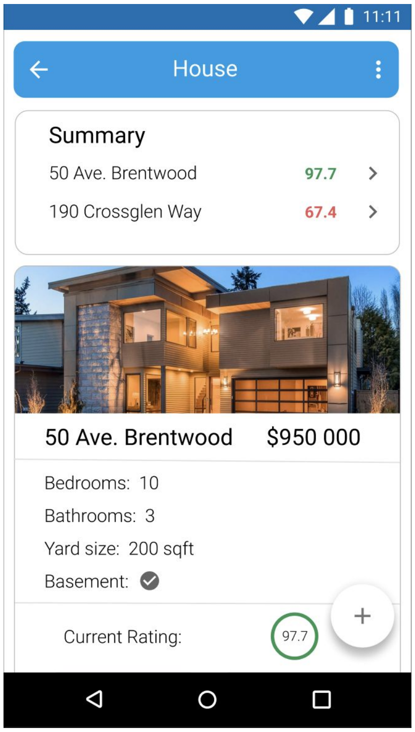
This is the sign in screen for the paid variant (The free would go straight to home). The option to use an offline mode is still available for the paid variant though.

Screen 2: Home Screen



This is the home screen. It shows user projects, a scrollable list of images if the user has added them to the project, and a floating action button to add new projects. The navigation drawer contains, sign in/out option, and a link to settings/help and feedback.

Screen 3: Project Details Screen



Summary

50 Ave. Brentwood	97.7
190 Crossglen Way	67.4

50 Ave. Brentwood $950 000

Bedrooms: 10

Bathrooms: 3

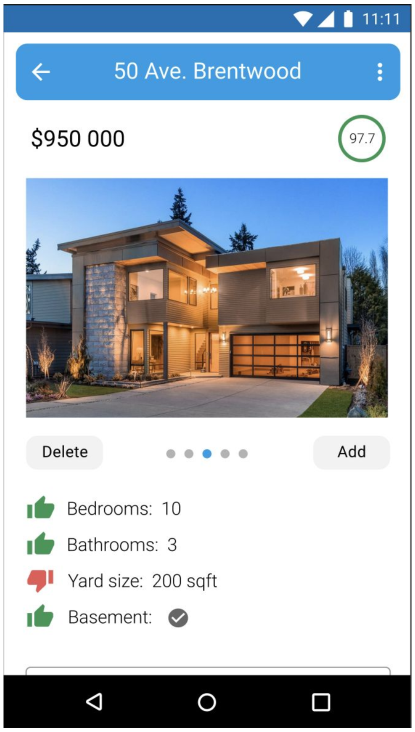
Yard size: 200 sqft

Basement: ✓

Current Rating: 97.7

The project details displays cards of items in the project. You can click details to see more information on each item, or to add/remove photos. You can also rule out an option. See github readme for GIF.

Screen 4: Item Details Screen



$950 000

👍 Bedrooms: 10

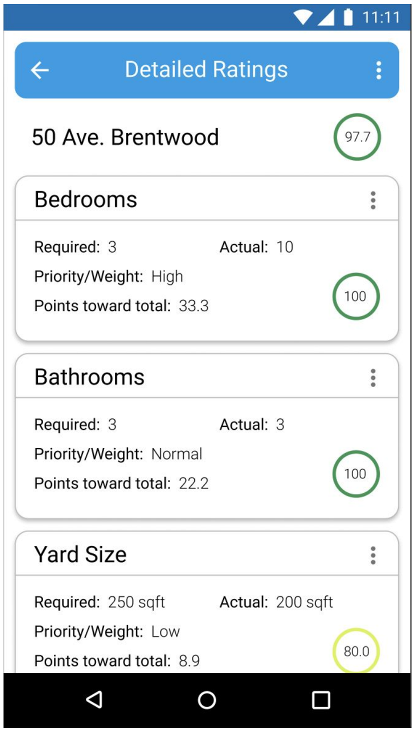
👍 Bathrooms: 3

👎 Yard size: 200 sqft

👍 Basement: ✓

Details of an individual item, option to edit the entry, add/remove photos.

Screen 5: Detailed Ratings Screen



50 Ave. Brentwood 97.7

Bedrooms

Required: 3 Actual: 10
Priority/Weight: High
Points toward total: 33.3
100

Bathrooms

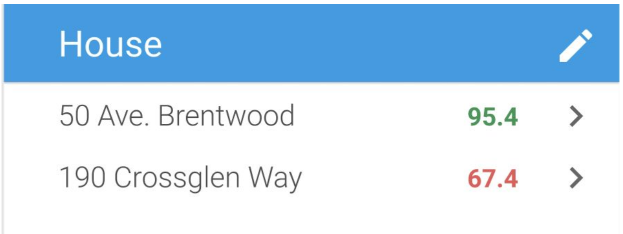
Required: 3 Actual: 3
Priority/Weight: Normal
Points toward total: 22.2
100

Yard Size

Required: 250 sqft Actual: 200 sqft
Priority/Weight: Low
Points toward total: 8.9
80.0

Detailed Ratings screen showing how the overall score was calculated and where it comes from. You can remove requirements here to see how it affects the overall score.

Widget



50 Ave. Brentwood	95.4
190 Crossglen Way	67.4

Widget of project picked by the user shows the highest and lowest rated option they currently have.

Overall Flow

See github readme for GIF.

Key Considerations

How will your app handle data persistence?

The app will use a firebase database to store user projects with the paid variant and an offline Room database with the free version. Also settings will be held in sharedPreferences.

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

- Room - for offline database
- Butterknife - for view binding
- Potentially Dagger - for dependency injection
- Timber - for logging

Describe how you will implement Google Play Services or other external services.

I will be using the following google play services:

1. Firebase Database - for online storage of user projects
2. Google sign in - to allow for firebase database

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

1. Setup github repo/android studio project
2. Import libraries making sure no errors/deprecated warnings
3. Create POJOs for project/requirement classes

Task 2: Implement UI for Each Activity and Fragment

Create a general UI for the home screen, add project, and display project.

- Build UI for home

- Build UI for add project
- Build UI for project details

Task 3: Create Database

Create and test Room database for projects:

- Create Room files
- Test database functions correctly
- Incorporate saving of projects into UI from Task 2
- Make sure a repo is used so that when the paid variant is added, it is easy to switch between offline and online database

Task 4: Create UI Adding Items to a Project

- Build UI for add item
- Build UI for item details

Task 5: Add Item Storage to Database

- Add necessary methods for storing items in database
- Will need to use Mediastore to store user images of items and to retrieve images from the camera or filesystem
- Test new database methods
- Incorporate saving/displaying into UI from Task 4

Task 6: Implement Rating Helper Class/UI

Create a class that does all the calculations and processing for displaying average ratings of items in a project.

- Create rating helper class
- Build detailed ratings UI
- Combine the two

Task 7: Implement Widget

- Create widget which allows user to choose a project to display.

- Widget takes user to the project details when clicked

Task 8: Add Paid Variant with Option of User Sign

- Create free and paid flavors (potentially in app purchase?)
- Add sign in option for paid which allows for firebase storage/sharing projects with friends

Task 9: Add Firebase Storage

- Implement firebase storage
- Have an option in the home screen to switch from online to offline storage to make sure both work and can migrate between the two

Task 10: Add Sharing Capabilities (Potential Feature)

As a final task, implement project sharing to allow friends to vote for their favorite items with a project. This could be very challenging so leave it for last as it may take too long.

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"