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

Table of Contents

GitHub Username: https://www.github.com/joseandroidengineer

Home Eco

Description

App will be written solely in the Java Programming Language. This app is meant to get people around the house to become more productive in their household by doing their chores! People or kids if you're teaching them responsibility get to do their chores and complete them! Then the weekly or monthly Chore Manager checks to see if the task is complete to reward the person with some points! The higher the points the more likely they are able to win a prize from the designated Chore Manager! Prizes can be anything the chore manager want and the chore manager can be mom, dad, or lease holder of an apartment. Examples of prizes can range between a free lunch, snack, to a videogame or that special something chore participants can work towards!

Intended User

The intended users for the app are families, roommates, and other people who share living space. The application is intended to keep chores, errands, and any other home like tasks organized and assigned to others.

Features

List the main features of your app.

- Saves information
- Create lists of errands, prizes, and house mates
- Create profiles for house mates
- Take pictures for profile pictures
- App will support accessibility such as content descriptions and navigation using the D-Pad

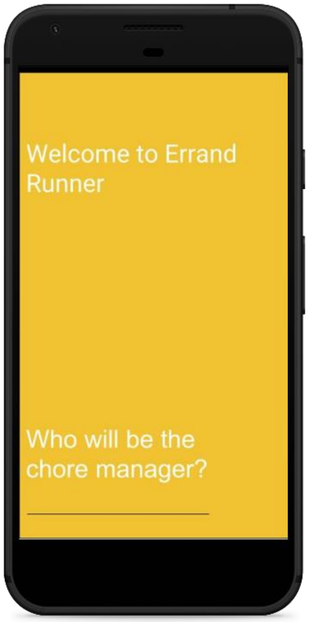
User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

Libraries to be used

Library	Version	Purpose
Picasso	2.7188	To load images
Firebase-core	16.0.8	View Analytics
Google Sign	16.0.1	Sign Users in to use the app
LiveData	2.1.0-alpha04	Build objects that notify views when database changes
Room	2.1.0-alpha06	Access App’s SQLite database with in-app objects and compile-time checks

Screen 1: Starter Screen

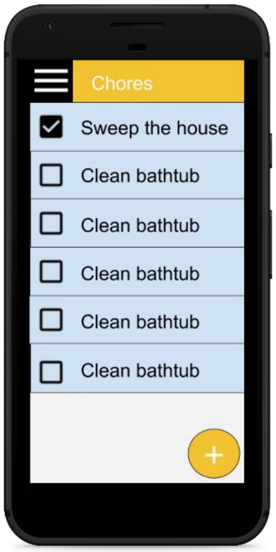


Welcome to Errand Runner

Who will be the chore manager?

This will be the first screen that first time users will see. All they would have to do is put the name of the chore manager or they will have the choice to sign into their google account using the Google Log in library. It will be saved into preferences so the name can be used later.

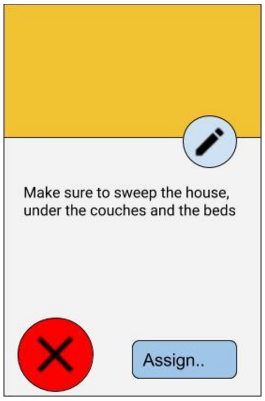
Screen 2: Main Activity/ Chore List



☑ Sweep the house
□ Clean bathtub
□ Clean bathtub
□ Clean bathtub
□ Clean bathtub
□ Clean bathtub

This screen represents the amount of chores that are either completed or need to be completed. The floating button will create an Alert Dialog Box and that will be used to create another chore and be added to the list. The Alert Box will prompt the user for the title of the chore and the description.

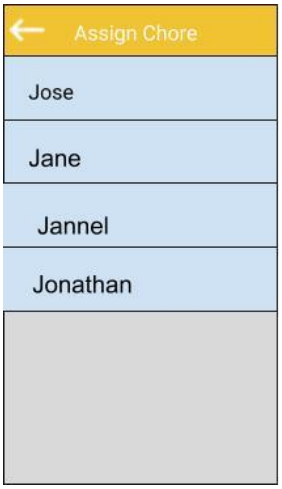
Screen 3: Chore Detail



Make sure to sweep the house, under the couches and the beds

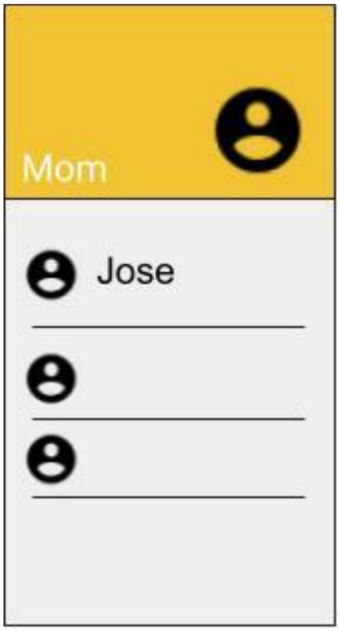
This screen represents the details of a chore when selected. The red ‘X’ represents that the chore/errand is not finished yet and the Assign.. button goes to the next screen which will show a list of house members.

Screen 4: House Members List



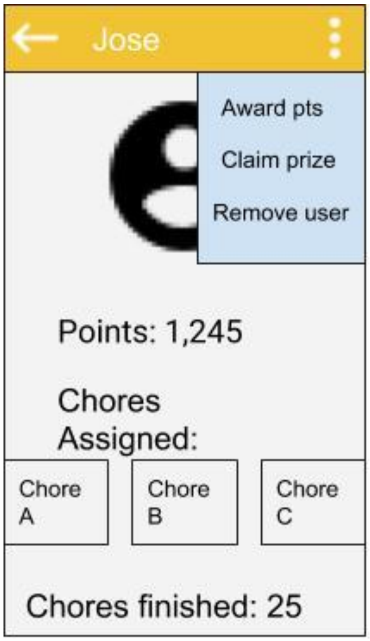
This screen represents the list of house members currently in the application. When a list item is selected, it will go back to screen 3 with a person assigned to the chore.

Navigation Drawer



The navigation drawer can be accessed from the first screen. It shows a list of house mates that are in the application. When selected it will take us to their profile detail Screen 5

Screen 5: Profile Detail



Points: 1,245

Chores Assigned:

Chore A

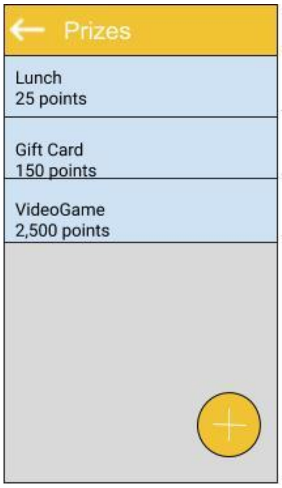
Chore B

Chore C

Chores finished: 25

The Profile detail screen shows the amount of points the housemate has, the number of chores assigned, and chores finished. Users have the option to use a different profile picture. Three options are given such as rewarding points to house mates, having housemates claim their prize and removing the user.

Screen 6: Prize List



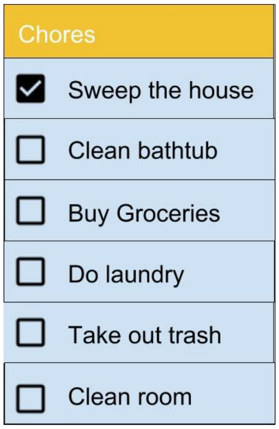
Lunch
25 points

Gift Card
150 points

VideoGame
2,500 points

The prize list screen will have a list of prizes and the amount of points needed to get those prizes. When tapped on the list item the prize will be awarded to the housemate in the previous screen and his points will be subtracted from the amount of points that was needed. When the floating action button is pressed an alert box will prompt the user what prize to add and how many points are necessary. Once the okay button is tapped another prize list item is added to the list.

Widget



Chores

☑ Sweep the house

□ Clean bathtub

□ Buy Groceries

□ Do laundry

□ Take out trash

□ Clean room

The widget will show a list of chores that are either completed or still not completed similar to the MainActivity/ Chore List. User will be able to check or uncheck the checkboxes through the widget and select list items to open ChoreDetail.

Key Considerations

How will your app handle data persistence?

Content Provider, Room and live data will be used to save errand data.

Describe any edge or corner cases in the UX.

When User rotates screen data should persist within the application without the app having any performance issues. When the weather is nice out, user will be reminded in a notification window that it is a beautiful day to get things done, this will be done using the **Accweather api** and an **IntentService** to retrieve that data.

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

Picasso to handle the loading and caching of images, Room and Live Data to handle data persistency within the application. Google Sign in will be used so users can log in using their google accounts. Firebase analytics will be used to track users and see how they interact with the application.

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

Firebase Analytics will be used to see which screen users interact with the most and Google Sign in will be used for users to log in to the application.

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

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

You may want to list the subtasks. For example:

- Configure libraries
- Add Room, Live Data, and Picasso to Gradle
- Also add the android support library

If it helps, imagine you are describing these tasks to a friend who wants to follow along and build this app with you.

Task 2: Implement UI for Each Activity and Fragment

List the subtasks. For example:

- Build UI for MainActivity
- Build UI for Navigation Drawer
- Build UI for First time Starter Screen
- Build UI for Chore Detail Screen
- Build UI for Assign Chore List
- Build UI for Profile Detail

Task 3: Set up Dao, Room, and Live Data

Describe the next task. For example, “Implement Google Play Services,” or “Handle Error Cases,” or “Create Build Variant.”

Describe the next task. List the subtasks. For example:

- Create Dao Interface
- Create ViewModels
- Create App Executors to move data between the internal database and the app

Task 4: Implement UI for Tablets

Describe the next task. List the subtasks.

- Create List Detail activity for MainActivity and ChoreDetail
- Build UI for Navigation Drawer
- Build UI for First time Starter Screen
- Create List Detail activity for ProfileDetail and PrizeList
- Build UI for Assign Chore List
- Build UI for Profile Detail

Task 5: Refactor code, debug

Describe the next task. List the subtasks. For example:

- Create test cases
- Make sure code works in all situations
- Make sure data persists within the app
- Keep all strings in strings.xml and make sure to have RTL layout switching on all layouts.

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

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**”