

**GitHub Username:** misshannah

# Plan My Wedding

## Description

This is an android mobile application that helps a user organise their wedding planning details and stores this information through the entire process.

## Intended User

This is an app for couples planning their wedding or for a wedding planner to keep track of the entire wedding planning process.

## Features

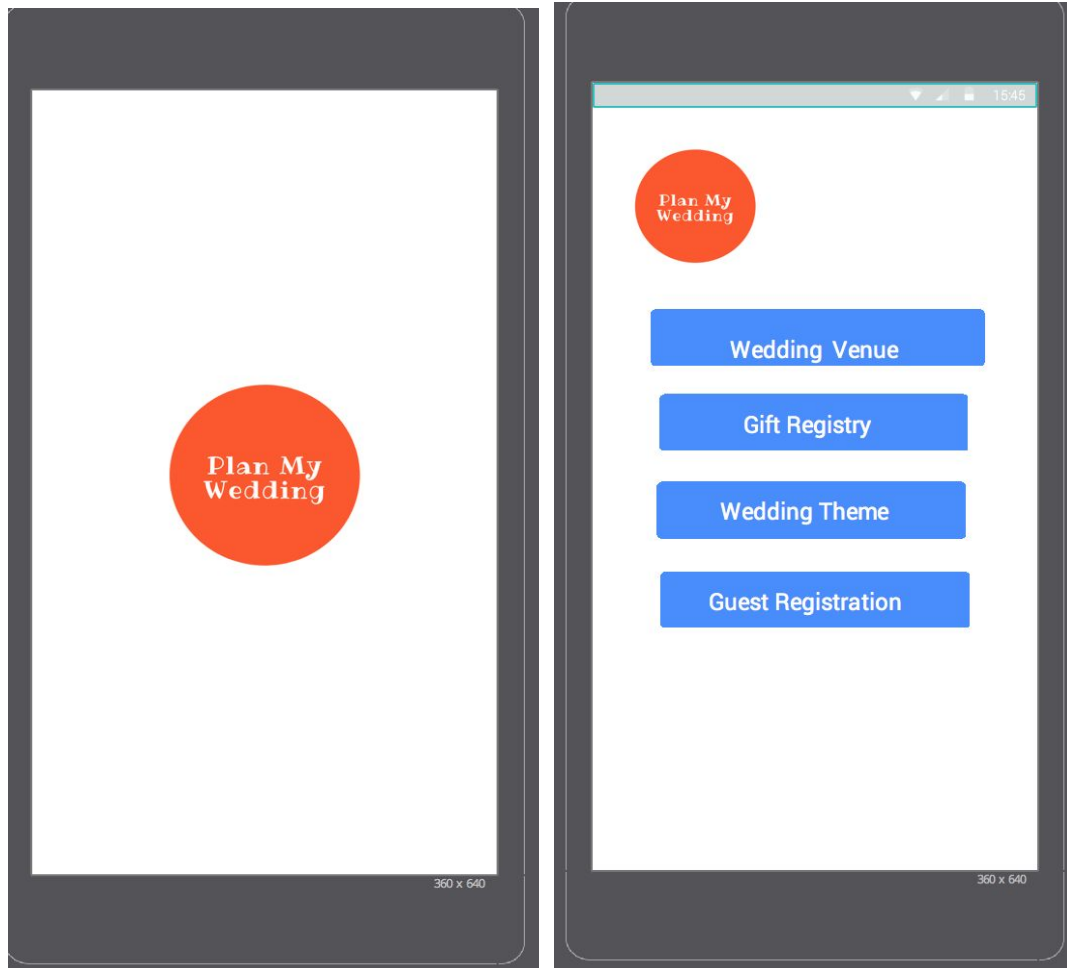
- Register guests
- Add gifts to a registry.
- Edit/Delete stored items on the various list.
- Uploads photos.
- Use google maps to show location.

## Development Tools

- Java programming Language - JDK Version 1.7
- Android SDK - Compile Version 27
- Android Studio - Version 3.0.1
- Gradle Build - Version 3.0.1
- Android Support Libraries - Version - 27.1.1

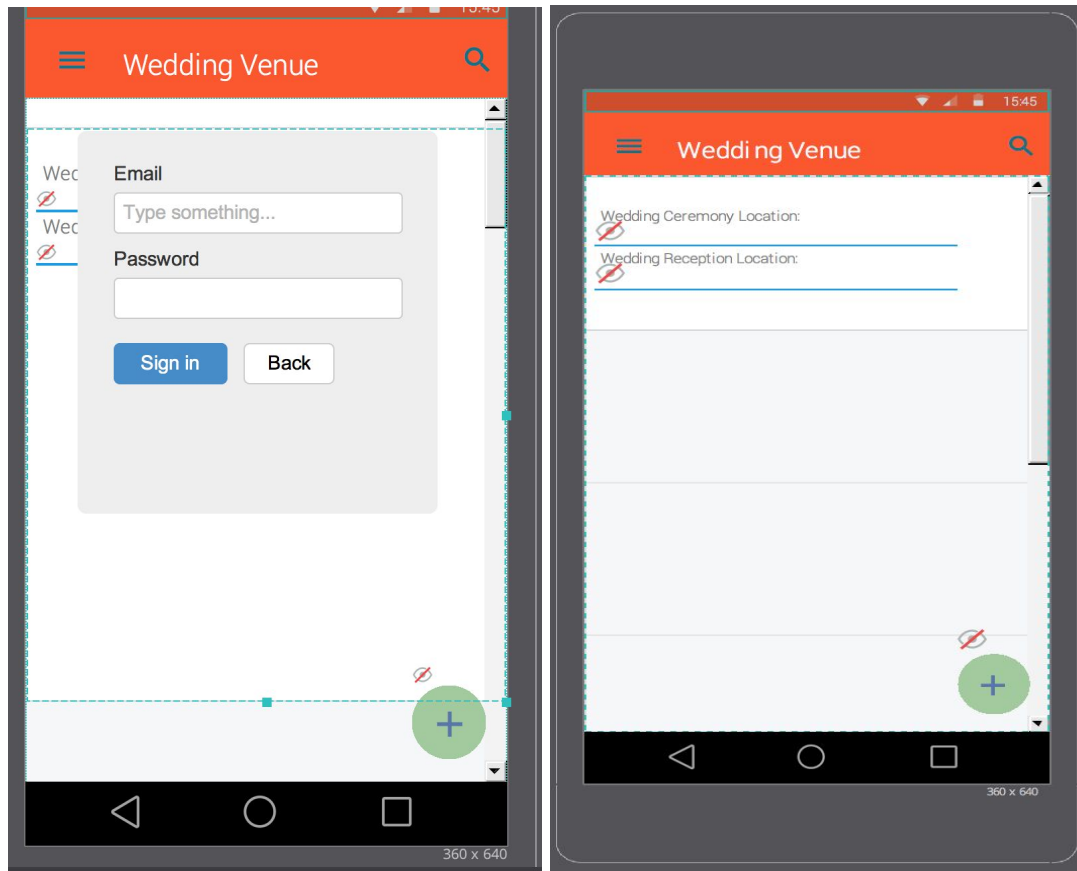
## User Interface Mocks

### Screen 1 & 2



The user views the splash screen then a dashboard with the categories of items available on the application.

## Screen 3 & 4



For example, when the user selects on the wedding venue option, they are required to login/signup then when successful he/she can see the details stored.

## Widget Screen



## Key Considerations

### How will your app handle data persistence?

Using the Room Persistence Library to store data in the mobile device.

### Describe any edge or corner cases in the UX.

- Using the back button to navigate away from the main screen.
- One can use the navigation drawer to easily access the dashboard categories.

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

- Using Picasso handle the loading and caching of images once captured by the user.
- Using RecyclerView and cardview

- Using Android Design
- Using Room

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

Using the Google Sign In Service to authenticate users in order to allow sharing/collaboration of the data stored.

## Next Steps: Required Tasks

### Task 1: Project Setup

- Create a new project on Android Studio.
- Add in the application images and logo.
- Set the theme and style using the material design palette.
- Set up the android widget for the application
- Configure all the necessary libraries i.e
  - Setup Room Libs
  - Setup Google Play Services for login with Firebase Auth
  - Setup logging using Firebase Analytics
  - Setup Google Maps for Location service
- Set all the strings and values in the correct xml file.

### Task 2: Implement UI for Each Activity and Fragment

- Build the various activities to show the screens as required.
- Include layouts for larger screens i.e Tablets
- Setup internal database storage using Room/LiveData and ensure no unnecessary calls are made to the DB.
- Use Async Task to make short duration requests on the API calls.
- Setup external APIs for data collaboration.
- Update the cache at regular intervals and enable syncing.
- The app enables RTL layout switching on all layouts.
- Handle configuration changes i.e rotation using the Android lifecycle events methods.

### **Task 3: UI tests**

- Implement the unit tests using sample code on Android studio
- Check functionality in different screen sizes.

### **Task 4: Handle Errors missed during implementation**

- From the tests in the previous tasks, make the necessary corrections to handle any errors faced.

### **Task 5: Clean up Code**

- Remove unnecessary logging.