**GitHub Username**: Dablaze-ufc

App Name

# My Adventures

## Description

One App for all your travel destinations.

The adventures you have had are all packed together to give you an extensive Journal for all your great memories and to give you a chance to relive all the adventures in every travel destination you have ever had.

Whether it's you, you and your spouse or your whole family it's a great place to store memories in an orderly place.

## Intended User

Travelers

## Features

List the main features of your app. For example:

- Saves information
- Takes pictures

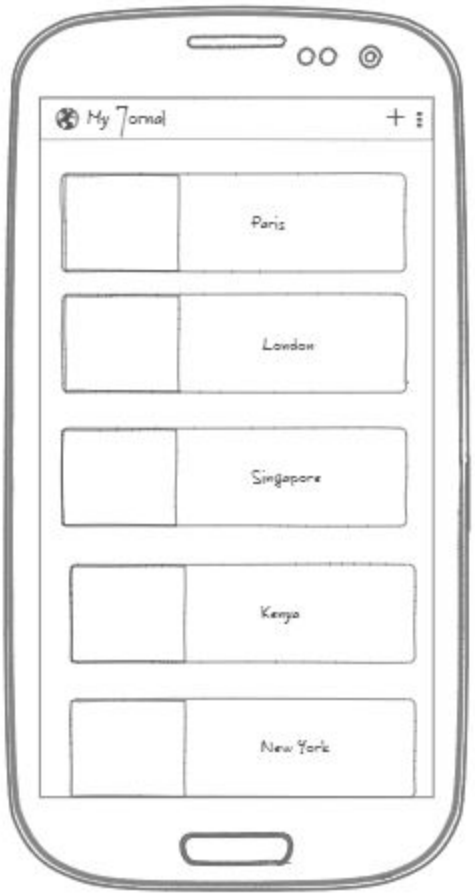
## Language Used

Java

## 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.

## Screen 1

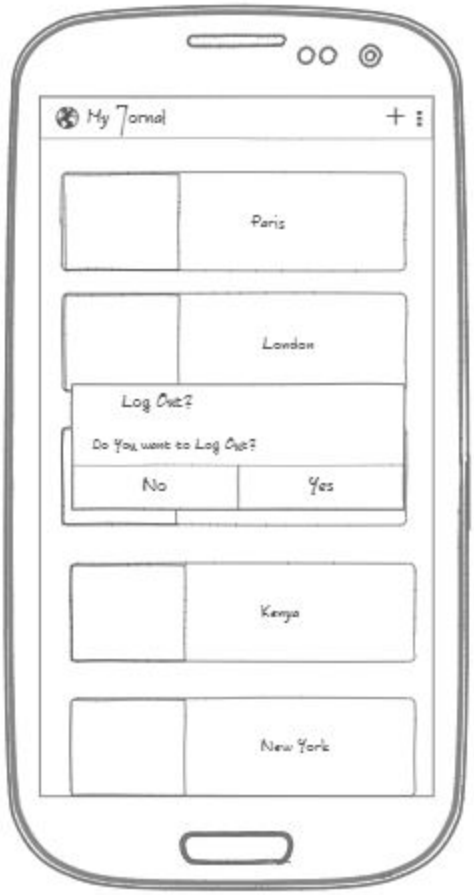



This is the journal list screen that keeps a journal based on country, state or provinces visited.

## Screen 1a

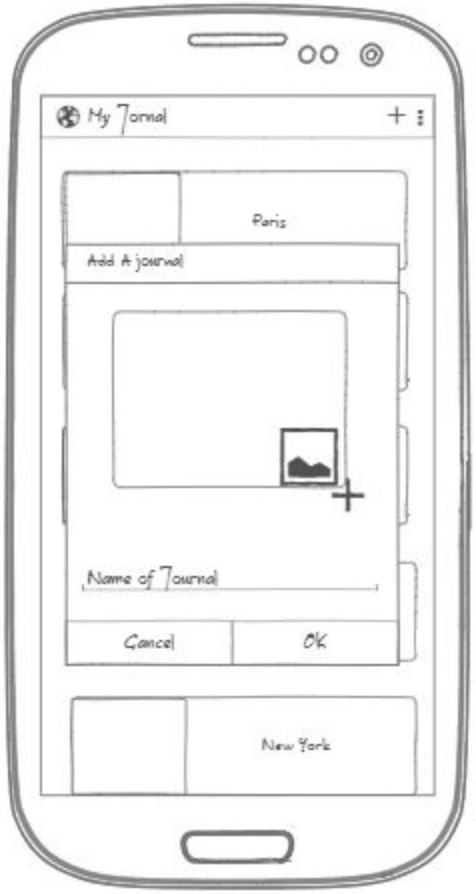

This screen shows the overflow menu

## Screen 1b



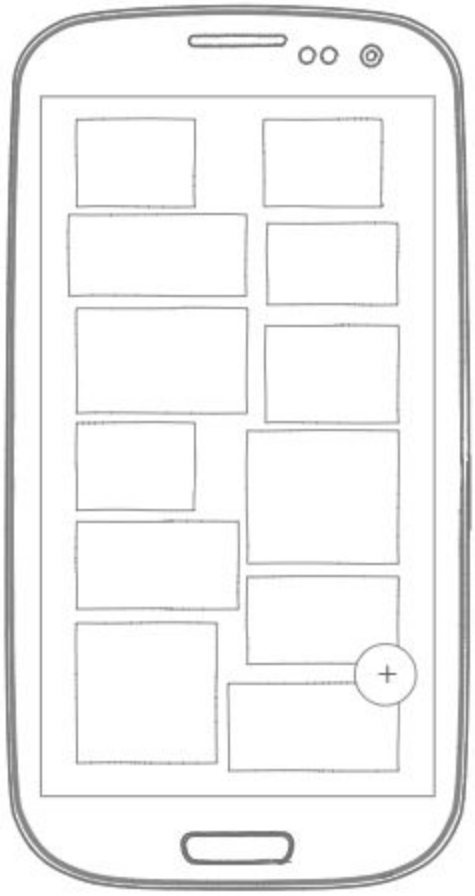

This Screen Shows the dialog to confirm user logout selection

## Screen 1c




This screen displays the dialog to add a new journal.

## Screen 2



This screen arranges the pictures based on destinations within a country, state or provinces

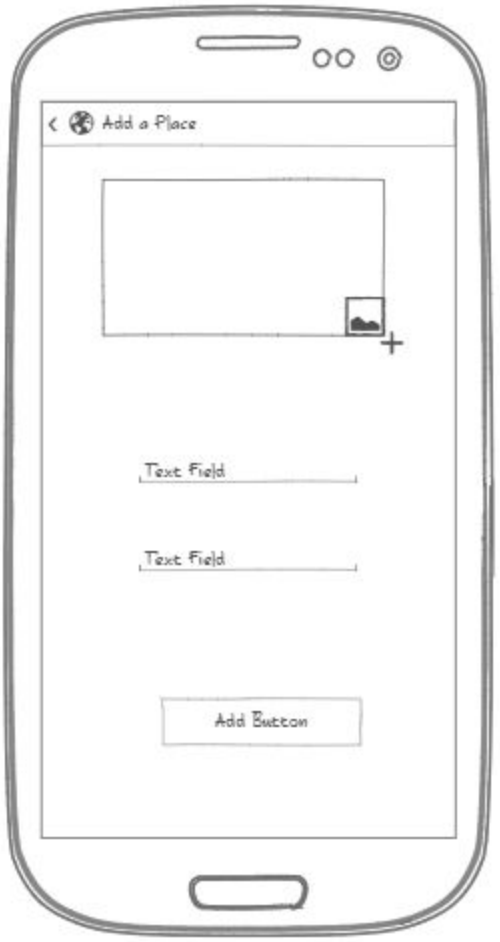
## Screen 3


This is the place detail screen for every entry

## Screen 3b

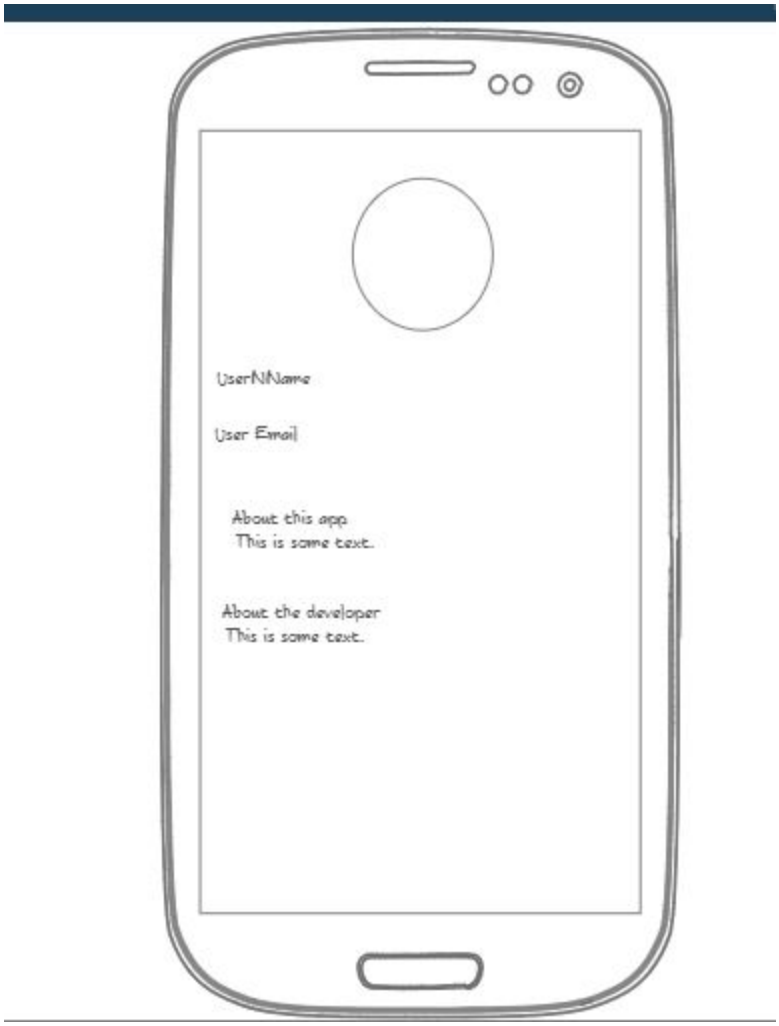

This screen shows the dialog confirming the user’s deletion action.

## Screen 4




Screen to add a new place entry

## Screen 5




Settings Screen

## App Widget


This frame shows the app Widget On the Homescreen

# Key Considerations

### How will your app handle data persistence?

Firebase Realtime Database for the data and the Firebase Storage for the pictures

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

User opens the app for the first time and logs in.

User sees a dialog that tells the user how to add a journal.

User clicks the add button on the app bar and a dialog is shown to add a journal.

User clicks on a dialog and is taken to the places List screen

User clicks the FAB and is taking the add a place screen.

User fills in the fields and clicks the add button and a place is added.

User clicks a place and sees the details.

Hitting the back button or the arrow icon on the app bar will take the user to the previous screen.

User clicks the delete icon on the place list screen and a dialog appears to confirm the user's deletion command.

If the screen is the last screen on the queue the user exits the app on clicking the back button.

An app widget would be available to display the list of places in a journal that was previously selected.

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

Glide to handle the loading and caching of images.
Firebase to handle data persistence and Authentication.
Circle imageView for easy displaying of images

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

I will use google sign in to handle user authentication.

# 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

Creation of github repo

- Configure libraries
- Arrange the project parts in folders example all UI elements inside a ui folder
- Run the project to see if everything compiles.
- Commit and share the project to github.

Connect the project to Firebase

- Log into firebase and create a firebase project.
- Follow the documentation on how to connect an android project to a firebase project.
- Setup firebase real time database and corresponding read and write rules
- Enable google authentication.

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

SetUp Main Activity

- Create layout for MainActivity

SetUp PlaceDetailActivity

- Create layout for PlaceDetailActivity.

SetUp DetailActivity

- Create layout  for DetailActivity

SetUp AddPlace Activity

- Create layout for the AddPlace Activity

Commit and push to github.

## Task 3: Implement User-flow and Logic

SetUp logic for MainActivity

- SetUp RecyclerView Adapter for the journals in MainActivity
- Hook up the Menu Items example add journal and logout
- Setup dialogs to handle logout selection and add a journal selection.
- Fetch dummy data from firebase database in MainActivityViewModel
- Wire adapter to observe the fetched data from firebase.
- Wire up click handler to respond to user user clicks.
- Add transition animation.
- Hook up back navigation on the back arrow icon on the app bar.
- Run the project to see if everything compiles.

Commit and push to github.

SetUp logic for PlaceDetailActivity

- SetUp RecyclerView Adapter for the places in PlaceDetailActivity.
- Hook up back navigation on the back arrow icon on the app bar.
- Fetch dummy data from firebase database in PlaceDetailActivityViewModel
- Wire adapter to observe the fetched data from firebase.
- Wire up click handler to respond to user user clicks.
- Enable navigation to add a place screen
- Add transition animation.
- Run the project to see if everything compiles.

Commit and push to github.

SetUp DetailActivity

- Display the selected data on the screen
- Hook up back navigation on the back arrow icon on the app bar.
- Wire up the delete action on the icon on the app bar
- setUp dialog to confirm user deletion selection.
- Run the project to see if everything compiles.

Commit and push to github.

SetUp AddPlaceActivity

- Validate user input to avoid empty fields
- Enable user to open either camera or photos from gallery to add photos
- Upload the photo to firebase storage and save the link to the image.
- Display loading bar as the upload process takes place.
- Navigate the user to the Place list screen
- Run the project to see if everything compiles.

Commit and push to github.

## Task 4: Implement TabLayout and Landscape Layout

TabLayout creation

- Create tabLayout layout folder with the sw600dp qualifier
- Create layouts for tabViews
- Create fragments for place list screen and place detail screen to implement master-detail style.

Run on a tablet to verify behaviour.
Commit and push to github.

Landscape Creation

- Create a landscape view  for the place detail screen

Run to verify behaviour.
Commit and push to github.

## Task 5: Make the app material

Implement material design principles

- Create app theme
- Add elevations to cardviews, fabs and appbars
- Ensure legible fonts
- Ensure pictures are scaled properly for all screens.
- Add an app icon.

## Task 6:Implement App Widget

- Create widgets from android studio wizard
- Customize the layout files to display a list
- Customize the appWidget Service to display list using shared preference
- Add a widget icon.