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: HarrisonCarmonaCastillo

My Event Agenda

Description

Tired of missing your events or rearranging your dates because you couldn´t remember them on time? Event Agenda will help you to manage your events and know how to go there. It will notify you when they are going to start.

Intended User

All public

Features

List the main features of your app. For example:

- Saves information (Events)
- Shows Maps and Routes
- Notifies by Notifications or Alarm
- Integration with wear
- Statistics

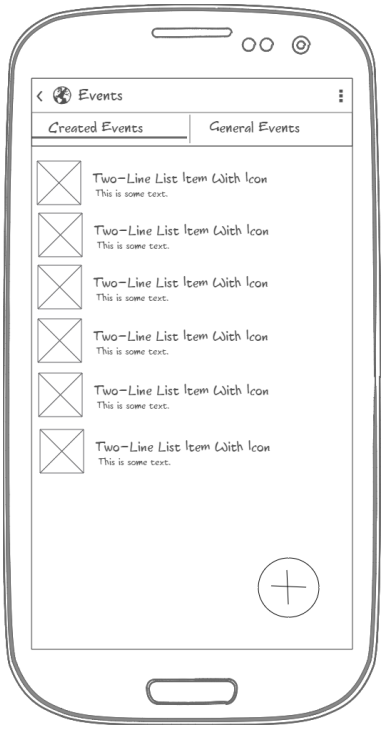
User Interface Mocks

Screen 1

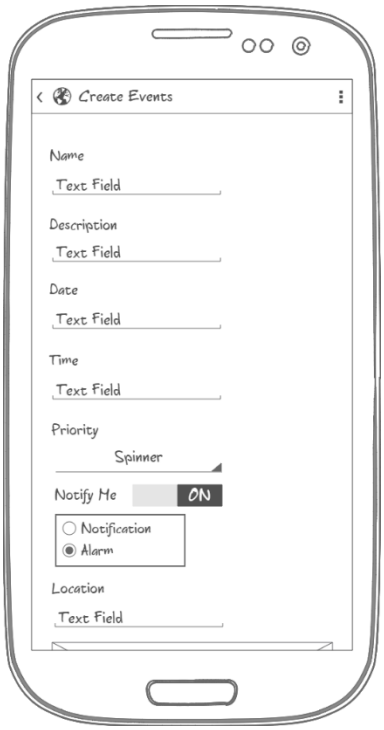
Screen 2

Screen 3

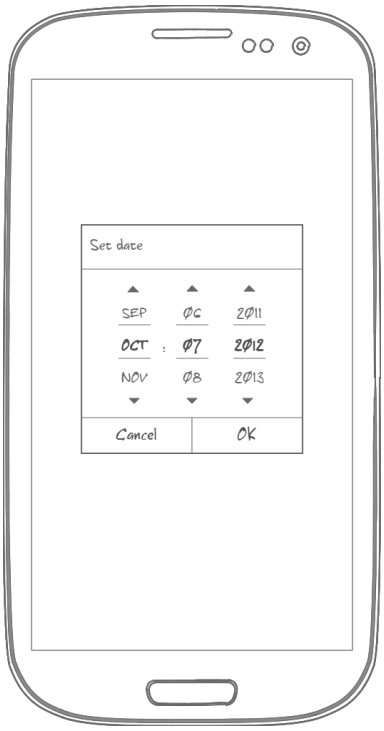
Screen 4



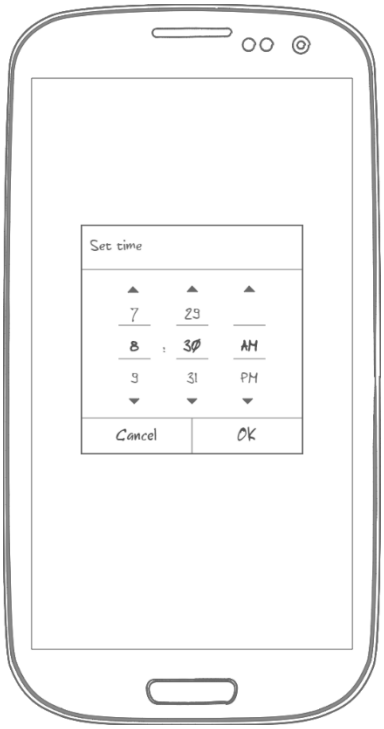
Screen 5



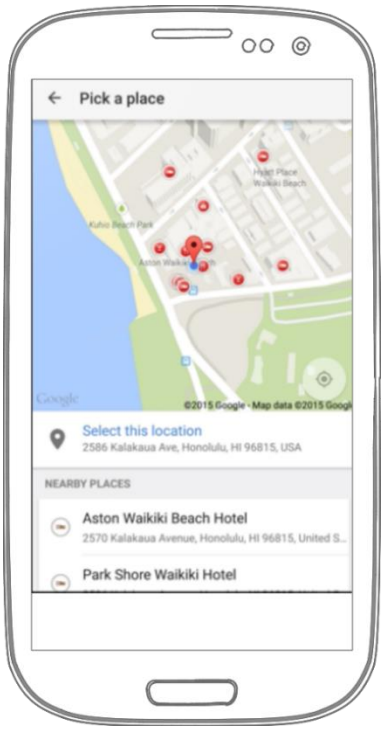
Screen 6



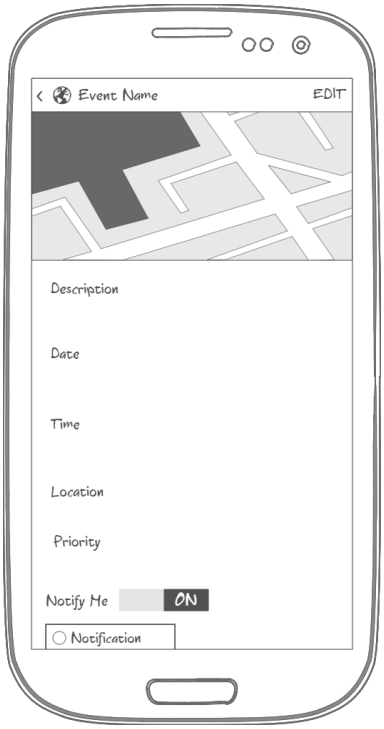
Screen 7



Screen 8

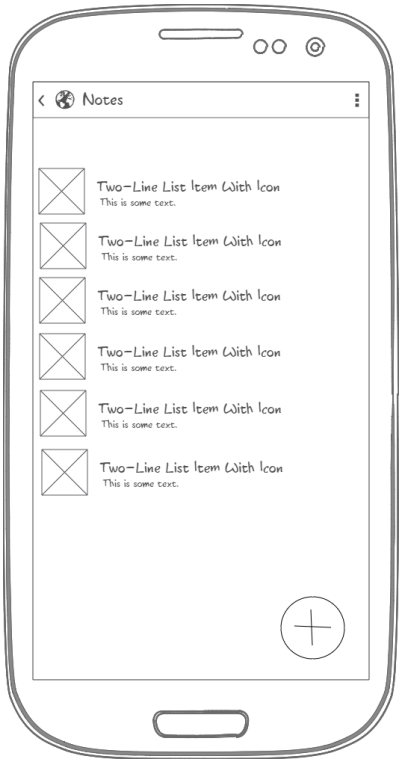


Screen 9

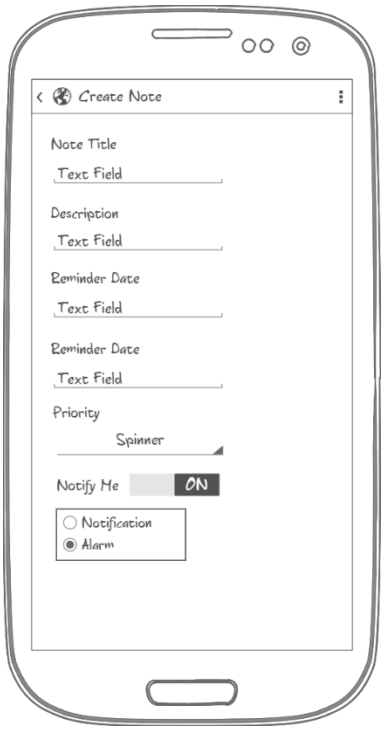


Screen 10

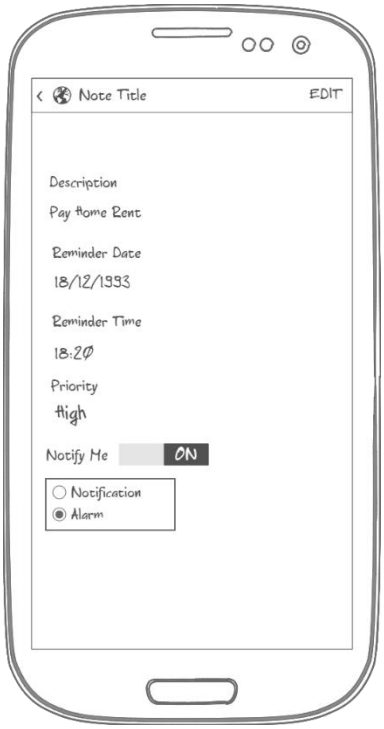
Screen 11



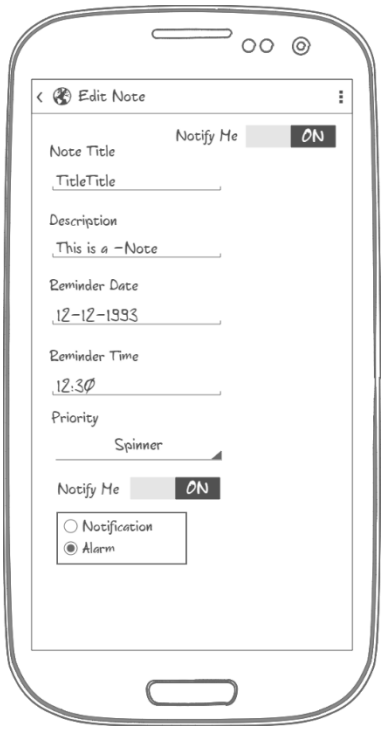
Screen 12



Screen 13

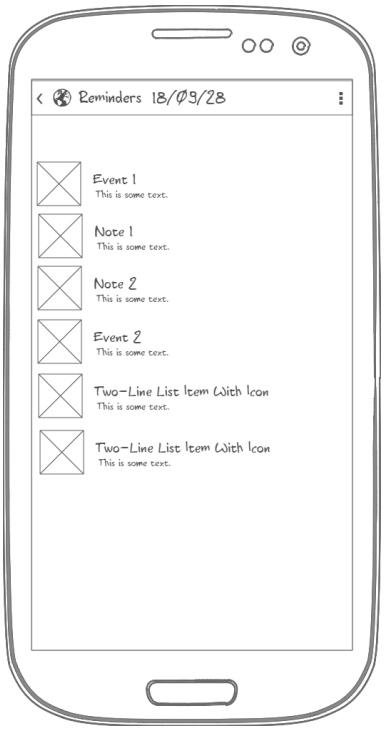


Screen 14

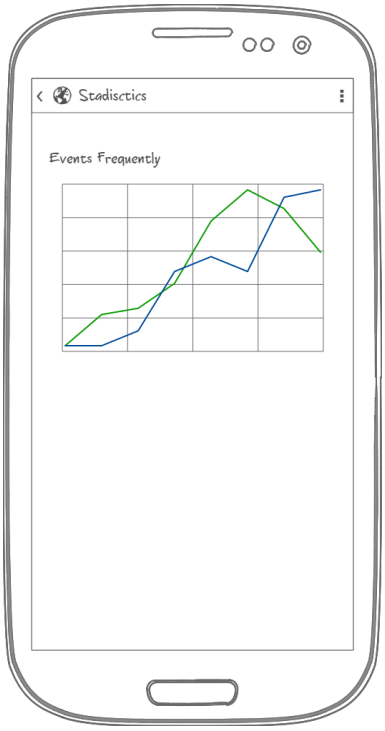


Screen 15

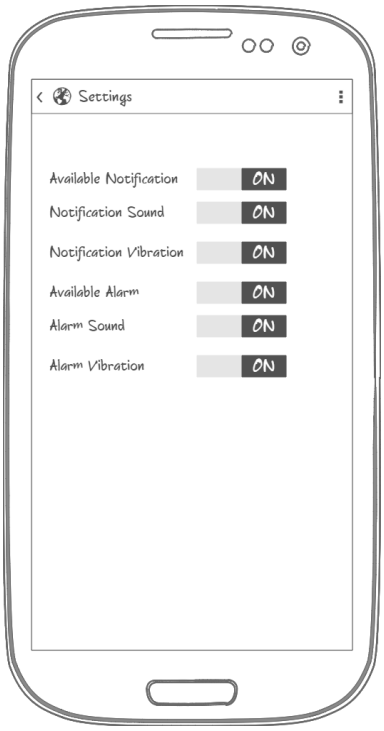
Screen 16



Screen 17

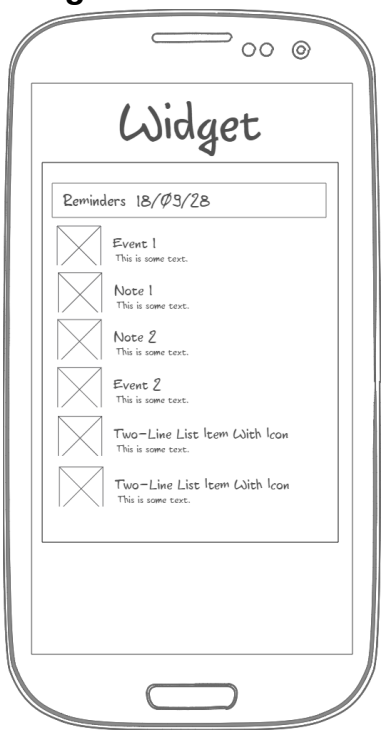


Screen 18



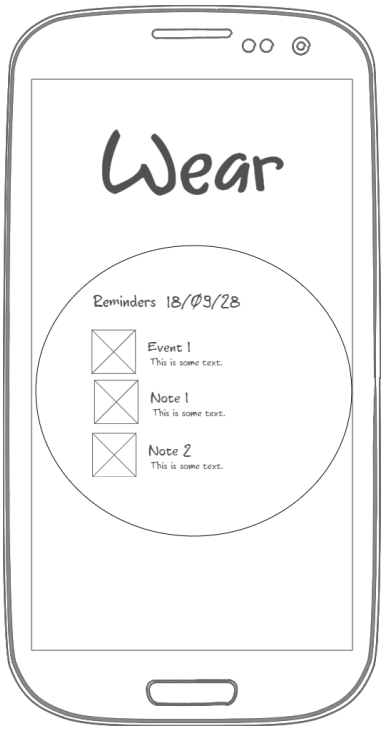
Screen 19

Widget



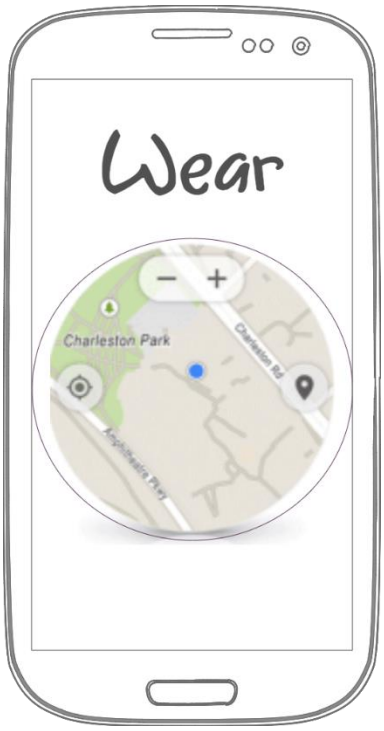
Wear WatchFace

Screen 1



Screen 2

Screen 3



Screen 4

Key Considerations

How will your app handle data persistence?

I'll persist my data via database using Sugar Orm. Also, I'll create a content provider to access to it.

Describe any corner cases in the UX.

I didn't found a corner cases in my application.

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

- **Sugar ORM**
  It is a database persistence library that provides a simple and concise way to integrate your application models into SQLite. In contrast to ActiveAndroid, which is mature, powerful, and flexible, Sugar ORM is:
  - Less verbose
  - Quicker to set up
  - More hands-free

- **MPAndroidChart .**
  It is a powerful & easy to use chart library for Android.

- **AndroidSwipeLayout**
  This will be the most powerful Android swipe UI component.

- **Material-calendarview**
  A Material design back port of Android's CalendarView.

- **Circular-progress-button**
  Android Button which can morph to Circular Progress.

- **Crashlytics**
  Provides deep and actionable insights, even the exact line of code your app crashed on.

- **Android Ripple Background**
  A beautiful ripple animation for your app. You can easily change its color, speed of wave, one ripple or multiple ripples. See demo below.

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

Configure libraries

- Sugar:

  1. Gradle:compile 'com.github.satyan:sugar:1.4'

  2. Configuration

  ```
  <application android:label="@string/app_name" android:icon="@drawable/icon"
  android:name="com.orm.SugarApp">

  <meta-data android:name="DATABASE" android:value="sugar_example.db" />
  <meta-data android:name="VERSION" android:value="2" />
  <meta-data android:name="QUERY_LOG" android:value="true" />
  <meta-data android:name="DOMAIN_PACKAGE_NAME" android:value="com.example"
  />

  </application>
  ```

  3. Create Entities.

- Crashlytics

  1. Download Fabric sdk for android studio.
  2. Open Fabric plugin.
  3. Integrate Crashlytics so click on install button of Crashlytics.

Task 2: Implement UI for Each Activity and Fragment

List of subtasks:

- Build UI for Tutorial.
- Build UI for MainActivity
    - Fragment for list of events
    - Fragment for list of notes
    - Fragment for calendar
    - Fragment for statistics
    - Fragment for settings
- Build UI for New Event
- Build UI for Event Detail
- Build UI for Edit Event
- Build UI for New Note
- Build UI for Edit Note
- Build UI for Event Note
- Build UI Alarm View
- Build UI Calendar Date Detail View

Task 3: Implement Google Play Services for Map

- Install the Google Play services SDK
- Create a Google Maps API key
- Create the layout of the main Google Maps v2
- Create the source code

Task 4: Create build Variant

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

- Configure Debug Mode
- Configure Release Mode

Task 5: Events

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

- Create Event Model with SugarOrm.
- Create, edit and Detail Event Functionality

Task 6: Notes

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

- Create Note Model with SugarOrm.
- Create, edit and Detail Note Functionality

Task 7: Calendar

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

- Create Decorators of the Calendar.
- Show days with events or notes.
- Render Calendar on click.
- Show Calendar date detail view.

Task 8: Statistics

- Show Statistics for events for each week and month.

Task 9: Settings

- Disable Notifications
    - Sound
    - Vibration

- Disable Alarm
    - Sound
    - Vibration

Task 10: Create Notifying Service

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

- Create Service and implement it.
- Handle when application is opened and closed.
- Show alarm or notification.

Task 11: Widget

- Declaring an App Widget in the Manifest.
- Create widget layout.
- Get data of Application to show in Widget.

Task 12: Wear Module

- Create wear Module.
- Create wear layouts.
- Get data of Application to show in wear.
- Add google Play Services for Map.
- Implement google maps.

.

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**"