

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

Description

Intended User

Features

User Interface Mocks

Screen 1

Screen 2

Key Considerations

How will your app handle data persistence?

Describe any corner cases in the UX.

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

Describe how you will implement Google Play Services.

Next Steps: Required Tasks

Task 1: Project Setup

Task 2: Implement UI for Each Activity and Fragment

Task 3: Your Next Task

Task 4: Your Next Task

Task 5: Your Next Task

GitHub Username: Your GitHub username here

Travel Baseball App

Description

This is a travel ball baseball app to help players and parents receive updated information on game/practice times and locations.

Intended User

The app has two intended users and thus will be available in two different flavors. The first intended user is the players and parents or more generally “non-coaches”. Non-coaches should be able to access game time information but should not be allowed to edit or add game time

information. The second intended user is the coaches. Coaches need the ability to update or add game times and game locations.

Features

The primary features for the players app will allow both parents and players to receive notifications on updated and added game/practice times and locations. Players will also be allowed to set a reminder to receive notifications (e.g. 1 day or 1 hour before game time). All practice and game locations will be integrated with Google Maps so players and parents can easily get directions to the proper location. A secondary function will allow players and parents to email the coach directly from the app. This will give a convenient way for players to notify the coach that they cannot make practice, etc.

The coaches app will have the added feature of adding, editing, and deleting practice/game times and locations.

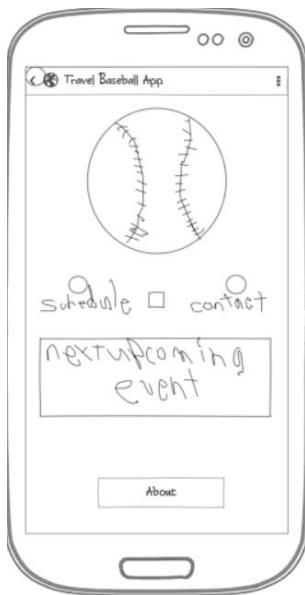
User Interface Mocks

Screen 1



The LoginActivity allows players to sign-in through Google. Once registered, user information is saved to the database.

Screen 2



The MainActivity will show the next upcoming events time and location. The Schedule FAB will open the ScheduleListActivity which will show the entire list of calendar events.

Screen 3



The ScheduleListActivity will show the current schedule (e.g. the events time, date, and location). When the user clicks on the location it will open Google Maps for directions.

Screen 4 (Widget)



The widget will provide information on the next upcoming event.

Key Considerations

The app will be written solely in the Java programming language. App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts. App will include content descriptions for all applicable UI elements. App provides a widget to provide relevant information to the user on the home screen. App will use IntentService to fetch data for the widget. The app will use Gradle 4.9 and Android Studio 3.1.4.

How will your app handle data persistence?

Firebase Realtime Database will be used to save user information (e.g. name, email) and location information. I am also looking into using the Calendar Content Provider to store events and attendees.

Describe any edge or corner cases in the UX.

I can't think of any specific edge cases at the moment.

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

I am testing a few calendar libraries. Ideally, the library should display a calendar and add, edit and delete events. The first library is Horizontal Calendar View: <https://android-arsenal.com/details/1/7061#!description> (0.1.0 (Jul 24, 2018)). The second library is LightCalendarView: <https://android-arsenal.com/details/1/4696#!description> (1.0.1 (Jan 13, 2017)). The final library is MyDynamicCalendarLibrary: <https://android-arsenal.com/details/1/5562>. The last library is the oldest of them (last updated April 2017), but provides the clearest examples of adding events to display on a calendar. I am also familiarizing myself with Androids built in calendar app and the corresponding Calendar Content Provider. The idea is to have one of these libraries display the calendar information.

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

When a user clicks on the location of an event Google Maps will launch to give accurate directions.

Next Steps: Required Tasks

Task 1: Project Setup

The first task is to create the LoginActivity and corresponding LoginActivityLayout. Open Android Studio and create a new project called Travel Baseball App. Create a new LoginActivity then follow the steps for Firebase Authentication found here:

<https://firebase.google.com/docs/android/setup>

Task 2: Implement UI for Each Activity and Fragment

Build UI for MainActivity and ScheduleListActivity. The Schedule FAB on the MainActivity should launch the ScheduleListActivity. In the ScheduleListActivity events can be added, deleted, and edited.

MainActivity subtask:

- FAB button to email coach

ScheduleListActivity subtask:

- Save the event. Read <https://firebase.google.com/docs/database/>
- Launch Google Maps

Task 3: Google Maps

Implement Google Play Services for Google Maps.

Task 4: Calendar Content Provider

When an event is created and displayed in the ScheduleListActivity an event should be created using the Calendar Content Provider with the appropriate attendees. When the event is created, Google Calendar will handle the reminders and notifications.

Task 5: Widget

Create a widget to display the next upcoming event.

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**"