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: shubham1g5

Six Four Fantasy

Description

Six Four Fantasy is an cricket app that allows you to see scores of past and live matches.

Intended User

Cricket lovers all round the world who wants to be up to date with cricket matches.

Features

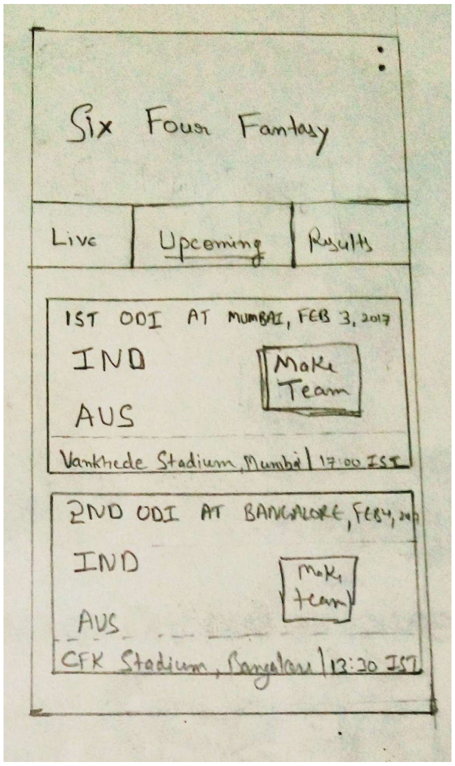
Six Four Fantasy has following features:

- Users can see ongoing and upcoming matches.
- Users can see scores for the ongoing matches.
- Users can see results of matches concluded in past.

User Interface Mocks

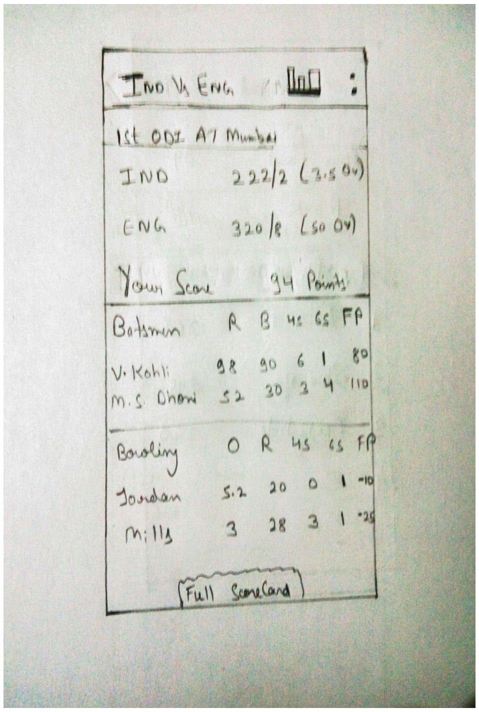
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1: Match List / Home Screen


Six Four Fantasy
Live
Upcoming
Results
1ST ODI AT MUMBAI, FEB 3, 2017
IND
AUS
Make Team
Vankhede Stadium, Mumbai | 17:00 IST
2ND ODI AT BANGALORE, FEB 4, 2017
IND
AUS
Make team
CFK Stadium, Bangalore | 13:30 IST

This is the home screen that user sees first on login. It has 3 tabs for the live, upcoming and concluded matches. Every Match in the list shows the 2 team who are playing the match along with date, time and venue of the match. If the match is live or concluded then the list item also shows the score of the corresponding match.

Screen 2: Match Detail/ Scorecard



Ind Vs Eng

1st ODI AT Mumbai

IND	222/2 (3.5 Ov)
ENG	320/8 (50 Ov)
Your Score	94 Points

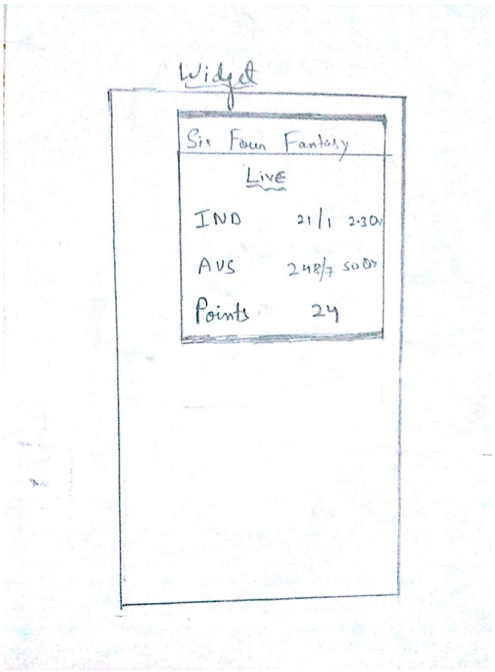
Batsmen	R	B	4s	6s	FP
V. Kohli	98	90	6	1	80
M.S. Dhoni	52	30	3	4	110

Bowling	O	R	4s	6s	FP
Jordan	5.2	20	0	1	-10
Mills	3	28	3	1	-25

Full ScoreCard

This screen shows the scorecard for the match.

Screen 3: Widget



Six Four Fantasy

Live

IND	21/1 2.3Ov
AUS	248/7 50Ov
Points	24

This is the app widget that shows the scores and fantasy points of all the live matches going on currently. User can click on the match to go directly to the match detail screen. Clicking on the title opens up the home page of the app.

Key Considerations

How will your app handle data persistence?

App will use content providers for user independent data like match list and scorecard.

Describe any corner cases in the UX.

- Clicking on a upcoming match won't open the detail score page as there is no score for the upcoming match and hence there is no point in showing the detail screen.

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

- RxJava and RxAndroid: To use reactive Java and android APIs.
- Retrolambda: To use Java 8 lambda syntax.
- Retrofit: To make http requests with RXAndroid Observables.
- Firebase JobDispatcher: For syncing match and scores data locally from 3rd party APIs.
- Dagger: Dependency Injection
- Android Data Binding: For binding UI elements to model.
- Android Design Support: For taking advantage of material design components like Coordinator Layout.

Describe how you will implement Google Play Services.

- Crash Reporting: To get information about app crashes and catched exceptions.
- Analytics: To understand user behaviour and demographics.

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 in build.gradle
- Configure min and max sdk
- Configure Java 8 compile options.

Task 2: Implement UI for Each Activity and Fragment

- Implement MatchListActivity
    - Coordinator layout with tabs for different match categories.
    - UI for each type of match Item using cardView
    - UI for tab fragments using RecyclerView with different match items layout.
    - Use Cursor Loader to load match data and live scores.

- MatchDetailActivity
    - Scorecard for live and finished matches.

Task 3: Model Classes

Design and add model classes for all data elements.

- User Model
- Player Model
- Match Model
- Team Model

Task 4: Data Layer

- Implement content provider for user independent data

Task 5: App Widget

- List UI to show live Matches
- Intents to integrate app with widget

Task 6: Google Play Services

- Integrate Crash reporting
- Integrate Analytics

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

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**"