GitHub Username: pallucandy

CricBuddy

Description

CricBuddy allows once to know the current matches i.e. matches scheduled for that day. They can check the live scores and inning requirements.

Intended User

This app is intended for Cricket Lovers.

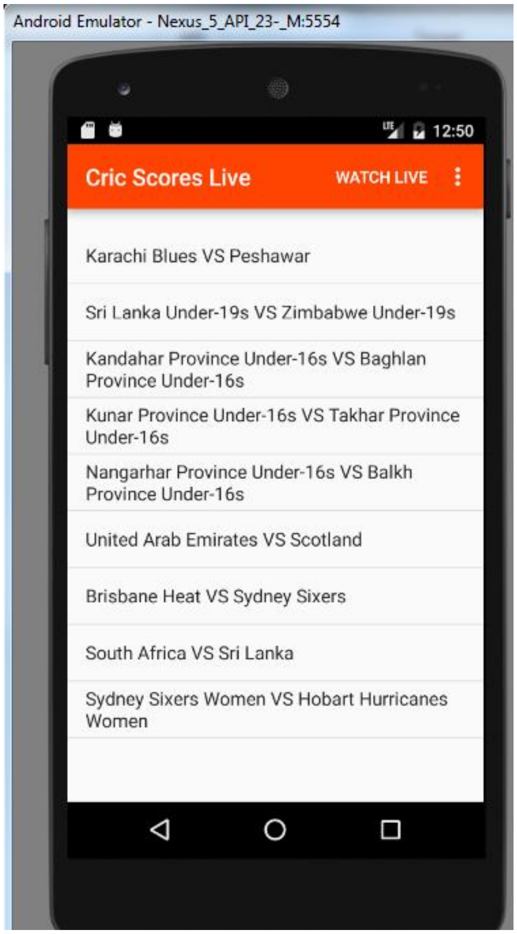
Features

- Know which matches are scheduled for the day
- Get more details on the matches like scores, schedule

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



Karachi Blues VS Peshawar

Sri Lanka Under-19s VS Zimbabwe Under-19s

Kandahar Province Under-16s VS Baghlan Province Under-16s

Kunar Province Under-16s VS Takhar Province Under-16s

Nangarhar Province Under-16s VS Balkh Province Under-16s

United Arab Emirates VS Scotland

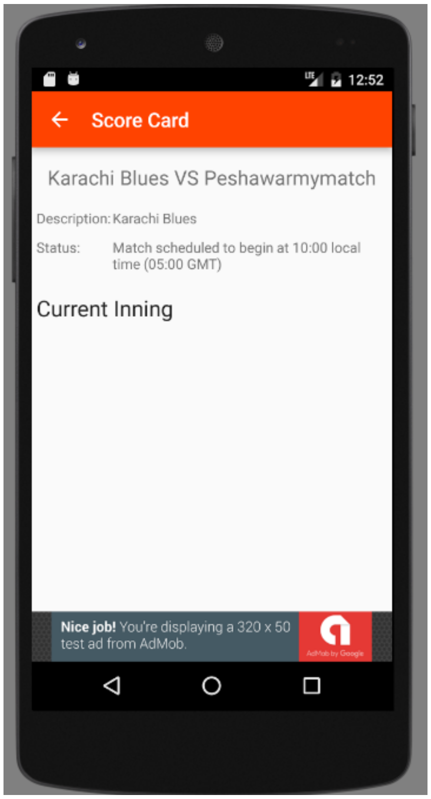
Brisbane Heat VS Sydney Sixers

South Africa VS Sri Lanka

Sydney Sixers Women VS Hobart Hurricanes Women

Main activity showing the list of matches scheduled for that day

Screen 2



Karachi Blues VS Peshawarmymatch

Description: Karachi Blues

Status: Match scheduled to begin at 10:00 local time (05:00 GMT)

Current Inning

Nice job! You're displaying a 320 x 50 test ad from AdMob.

Screen showing the details of the selected match.

Key Considerations

How will your app handle data persistence?

I would be using the API’s provided by cricapi to get the required data.

Describe any corner cases in the UX.

- User can use back button from the details view to get to the Main screen
- The details view is refreshed automatically

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

- Google Support Library (AppCompat, Design, CardView, RecyclerView) for using Material Design UI components, font styles etc.
- Google Play Services library for integrating AdMob for showing some.
- Butterknife to bind views to fields and methods

Describe how you will implement Google Play Services.

- Set up Google AdMob ads. We will be using one interstitial ad when we launch the details View for the match.

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

- Add library dependencies to build.gradle file

Task 2: Implement UI for Each Activity and Fragment

- Build SplashScreen Activity
- Build UI for MainActivity
- Build UI for ScoreCardView

Task 3: Your Next Task

- Create the Model, MatchModel and MatchDetailsModel
- Use these models to add data parsed from the json response
- Display the data stored