GitHub Username: kientn123

Premier League

Description

Keeping yourself updated with Premier League is never easier. With Premier League app, you can check:
-   The most up to date ranking table
-   Teams' results and statistics
-   Teams' upcoming fixtures

Intended User

Soccer fans, especially people who are interested in Premier League.

Features

●   Get most up-to-date ranking table.
●   Check out results and teams' performances.
●   Look at upcoming matches of your favorite teams and when they're playing.

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.
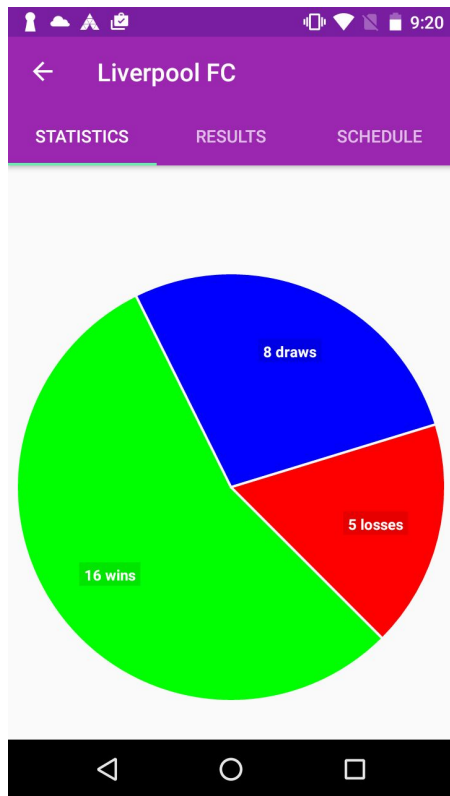
Screen 1

Ranking table.

Screen 2

Pie chart showing the team's performance.

Screen 3

Liverpool FC

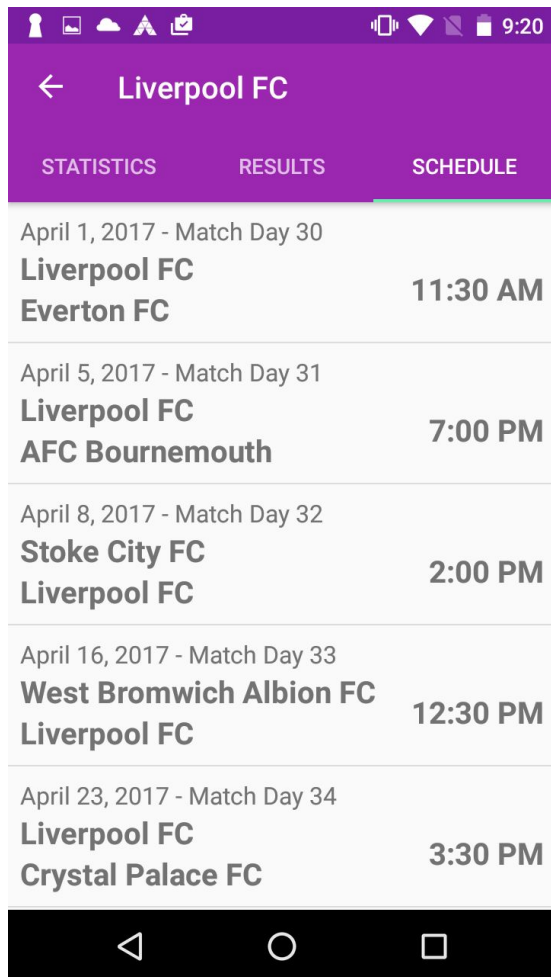STATISTICS   RESULTS   SCHEDULE

March 19, 2017 - Match Day 29
Manchester City FC   **1**
Liverpool FC   **1**

March 12, 2017 - Match Day 28
Liverpool FC   **2**
Burnley FC   **1**

March 4, 2017 - Match Day 27
Liverpool FC   **3**
Arsenal FC   **1**

February 27, 2017 - Match Day 26
Leicester City FC   **3**
Liverpool FC   **1**

February 11, 2017 - Match Day 25
**Liverpool FC**   **2**
Tottenham Hotspur FC   **0**

Results so far.

Screen 4

Liverpool FC	
STATISTICS	RESULTS	SCHEDULE

April 1, 2017 - Match Day 30 **Liverpool FC** **Everton FC**	11:30 AM
April 5, 2017 - Match Day 31 **Liverpool FC** **AFC Bournemouth**	7:00 PM
April 8, 2017 - Match Day 32 **Stoke City FC** **Liverpool FC**	2:00 PM
April 16, 2017 - Match Day 33 **West Bromwich Albion FC** **Liverpool FC**	12:30 PM
April 23, 2017 - Match Day 34 **Liverpool FC** **Crystal Palace FC**	3:30 PM

Upcoming matches.

Screen 5
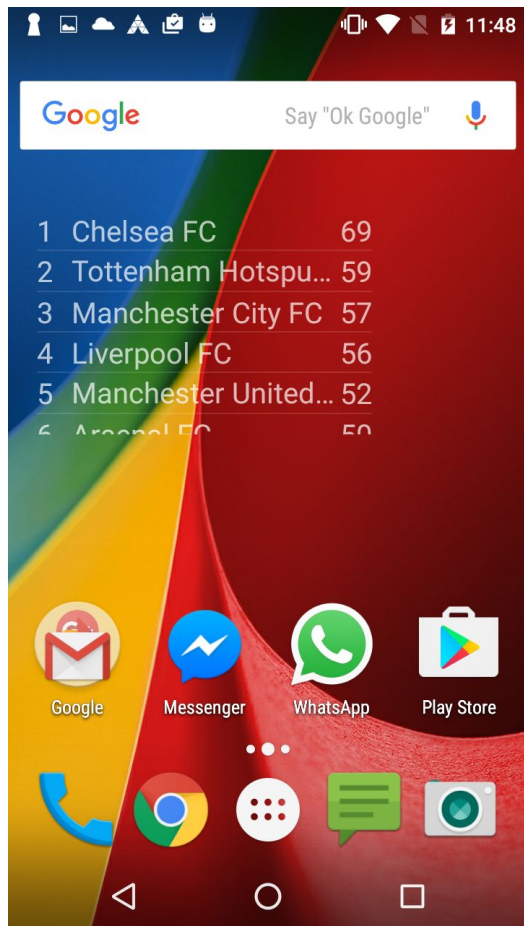
League table widget.

Key Considerations

How will your app handle data persistence?

Team's information (name, number of wins, scores, goals, etc.) and fixtures in two SQL tables: Teams & Fixtures. Then data will be accessed using ContentResolver.

Describe any corner cases in the UX.

In the TeamDetailsActivity, an api call with a specific teamId is used to get team's fixtures and statistics. The teamId is passed from the main activity, which is LeagueTableActivity via an intent. When we rotate TeamDetailsActivity, the intent that came from LeagueTableActivity is

gone, so if we try to get teamId from the intent again, the app will crash. Therefore, teamId should be saved in onSaveInstanceState(), so it is not lost on rotation.

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

- OkHttp 3 & Retrofit are used to request data remotely.
- ButterKnife is used to access views via ids faster.
- Schematic is used to generate ContentProviders.
- JodaTime is used to present time string.
- Hello Charts is used to make pie chart.

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

- Include all necessary libraries in Gradle build file.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for LeagueTableActivity.
- Build UI for TeamDetailsActivity. TeamDetailsActivity includes 3 tabs which correspond to 3 fragments:
    - Statistics - Winning statistics
    - Results - Results for matches that the team has played so far in the league
    - Schedule - Upcoming matches, including date and time

Task 3: Build Teams and Fixtures table in the database

- Describe table schemas for Teams and Fixtures.
- Use Schematic to generate Content Resolvers to access Teams and Fixtures tables.

Task 4: Build Intent Services to make api calls

- Use Retrofit to describe REST services or api calls used in the app.
- Build TeamIntentService to request team information and FixtureIntentService to request fixtures for each team.

Task 5: Hook UIs and database

- Hook LeagueTableActivity with Teams table to display ranking of all the team. Use LoaderManager to load data.
- Hook TeamDetailsActivity with Fixtures table to display the results so far for each team and their upcoming matches. Use LoaderManager to load data.

Task 6: Display winning statistics

- Use Hello Charts to draw pie chart to display the percentage of wins, draws and losses for each team.

Task 7: Implement league table widget

This includes implementing:
- LeagueTableWidgetFactory
- LeagueTableWidgetService
- LeagueTableWidgetProvider

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**"