

**GitHub Username:** maddom73

# Diretta calcio

## Description

Diretta Calcio gets information about the most important European football Leagues and shows them in a very gorgeous style. You can get fixtures (with live updating), standings, squads and players. You can see not only the matches ordered by the matchdays but also display them ordered by the teams (for example all the matches for Juventus in the Italian Serie A). Another nice feature is the possibility to get information about players (Nationality, Date of Birth, Contract Title and Market Value).  
Enjoy your football!

## Intended User

I think that this app is for a large range of people. Of course in a Country like Italy where people is very interested in football the range is very wide: it's for all the fans.

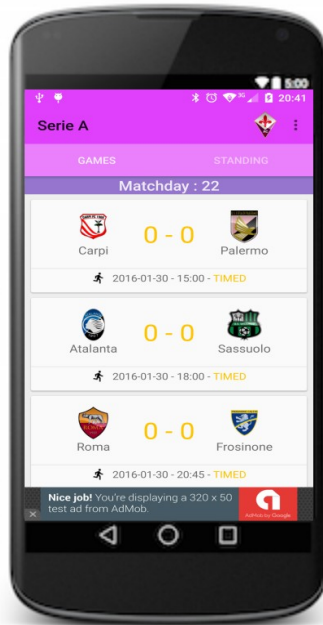
## Features

List the main features:

- Displays fixtures and standings about: Italian serie A – English Premier League – Spanish Liga – Germany Bundesliga – French League 1 – Portugal Primera Liga – Dutch Eredivisie – Champions League – European Championships 2016
- Displays Squads with informations about players: Nationality, Date of birth – Contract title -Market value

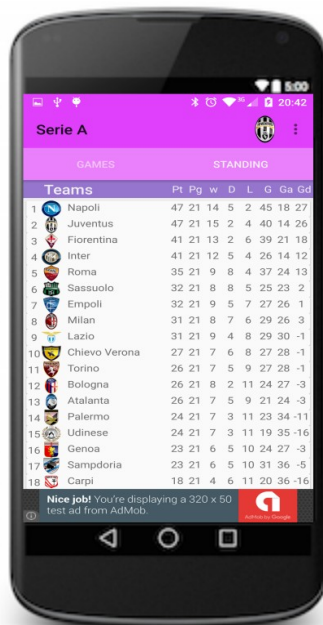
## User Interface Mocks

### Screen 1



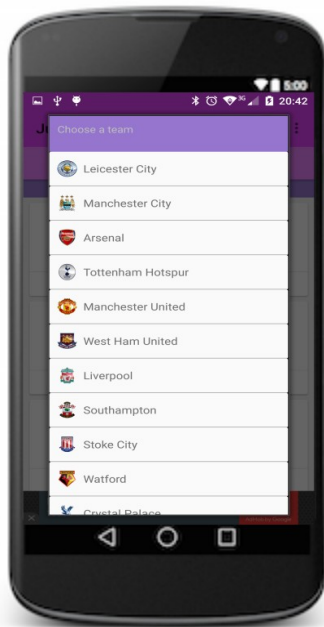
All the fixtures of the Italian Serie A, Match day 22. The icon badge at the right top is about Fiorentina, the football team of Florence, and when you click on it you can get all the fixtures of Fiorentina since the beginning of this season.

### Screen 2



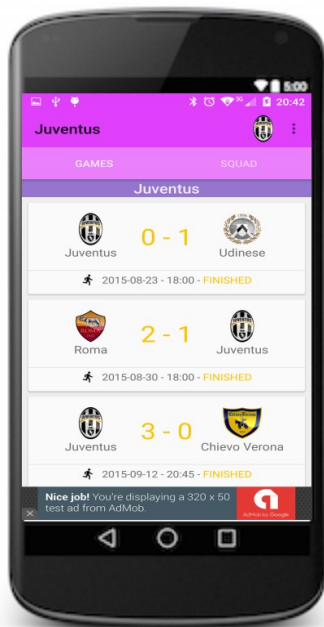
The standing of the league

### Screen 3



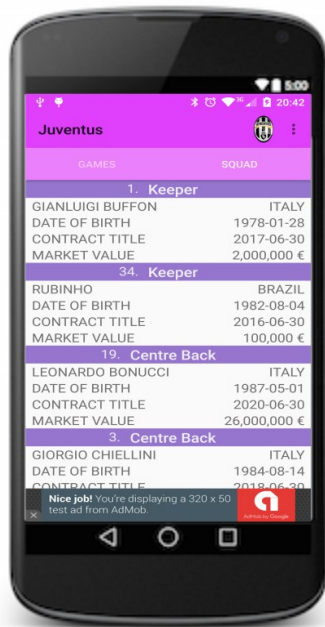
One of the lists to choose the team you want to get all the fixtures. These lists are made in relation to the league previously selected.

### Screen 4



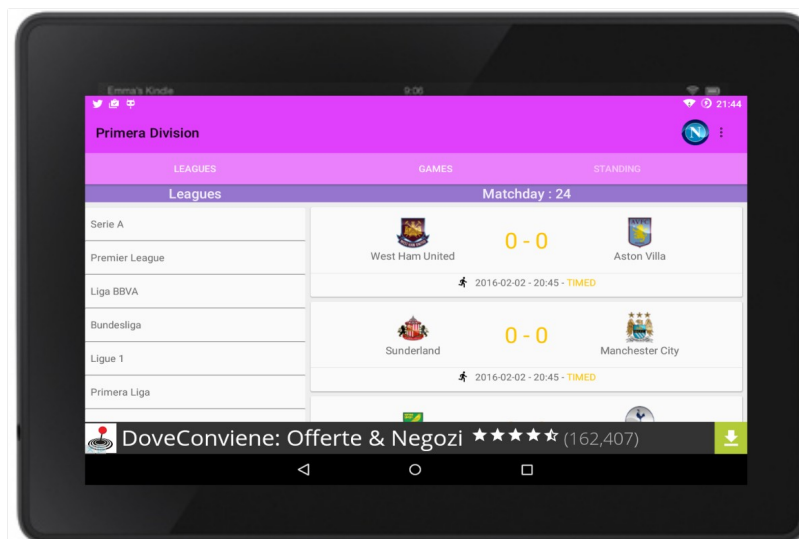
The fixtures about Juventus.

## Screen 5



Some informations about the players.

## Screen 5



Version for tablet

## Key Considerations

### Handle data persistence

Diretta calcio uses syncadapter to fetch data from an online server. It stores the data in a sqlite database through content provider built especially for it. Using a Sync Adapter has many benefits because the system automatically handles the scheduling of Sync Adapters to optimize battery life. Also only the content provider's implementation has direct access to the SQLite database. All other parts of the app can only access data through the Content Resolver. This is a very flexible architecture.

### Corner cases in the UX.

The UX is quite simple and it's a little bit different between phone and tablet. For the first one when you want to change the league you can access to the settings activity through the top menu. Settings activity contains a list of the available leagues with radio button. You can select one, on back pressed the activity finishes and a progress dialog starts until the syncadapter stops syncing and the league is changed. For tablet in landscape mode the list of leagues is in a fragment on the main screen. The same design is set for the list of matches about a single team. There are two online servers to fetch data: the first one is for the stats (the author hasn't set live data yet except for Premier League and Champions League) and the second one to get live results. So the choice to fetch live data has been to use swipe refresh layout checking when the data for the league displayed is available; if it's not a toast displays a message: "No live matches for this league at the moment". Finally for Champions league and European championships after the group stages you can see only the first knockout rounds until you know the winners. A message says to wait for the final part of the competition and another one warns that there isn't any standing.

### Libraries I'll be using.

Diretta Calcio uses android support design, recyclerview and cardview to set: CoordinatorLayout, AppBarLayout, Toolbar and the list of score; Glide to handle the loading and caching of images; androidsvg for the svg images and google play services for Admob and Analytics.

## Next Steps: Required Tasks

### Task 1: Project Setup

The first step is creating projects and configuring libraries in the gradle file.

- Create utilities class to retrieve the variables and to set icons, numbers of match days and so on
- Create team activity, dialog and fragment with the respective adapters

### Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity, TeamActivity and SeyyingActivity
- Set different layouts for tablet in landscape mode
- Build UI for PagerFragment, MainScreenFragment, StandingsFragment, SquadsFragment
- Set a viewpager in main activity to host two fragments for games and standings;
- Set another viewpager in main fragment to swipe across the match days

### Task 3: Implement sync architecture

- Create sync adapter
- Sync service
- Sync authenticator service
- Sync authenticator

### Task 4: Create database

Create:

- Database contract
- Database helper
- Content provider

### Task 5: Implement adapters

Create adapters for loading data into the following fragment:

- Main
- Standings
- Team
- Dialog
- Squad

### **Task 6: Implement Ads and Analytics**

- Implement Google Play Services
- Set ads from Admob
- Set Analytics

### **Task 7: Handle error cases**

Handle:

- ParseException
  - NullPointerException
  - IllegalStateException
  - SQLiteException
-