

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1](#)

[Screen 2](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any edge or 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 or other external 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: cebuckmaster

Coaches Corner

Description

Coaches Corner provides help to all the volunteer youth soccer coaches out there. This easy to use app comes in handy on game days. Allowing you to put more focus on the game and less time worrying about keeping that scorecard updated or what substitutions you will be making next.

Intended User

Volunteer youth Soccer Coaches of America.

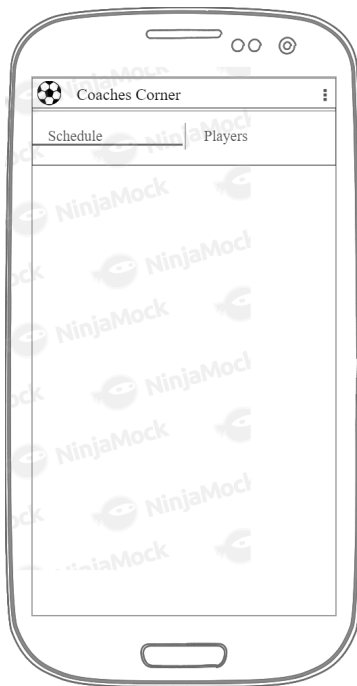
Features

The main features of the Coaches Corner app:

- Keep track of players
- Keep track of game schedule
- Create game lineups for each game and track player game time
- Easy to use Scorecard during game days
- Allows the coach to share of the final scorecard with player, parents, or league.

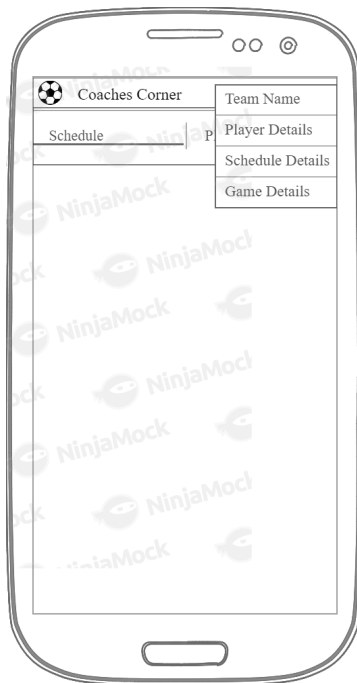
User Interface Mocks

Screen 1- Main Activity



This is the main activity screen that starts off. This represents the first time the app is run with no data added.

Screen 2 - Menu off MainActivity



This is the menu you will use to enter in details for the app.

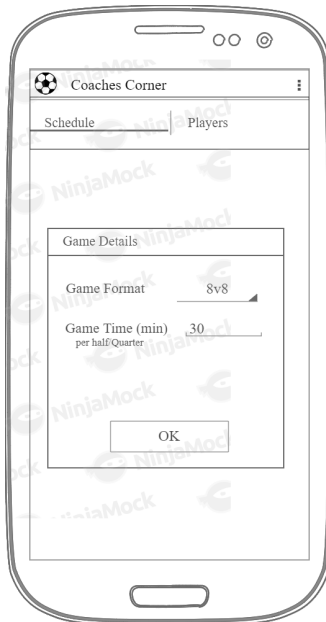
- Team Name -> dialog popup
- Player Details -> links to new Activity
- Schedule Details -> links to new Activity
- Game Details -> dialog popup

Screen 3 - Team Name Dialog



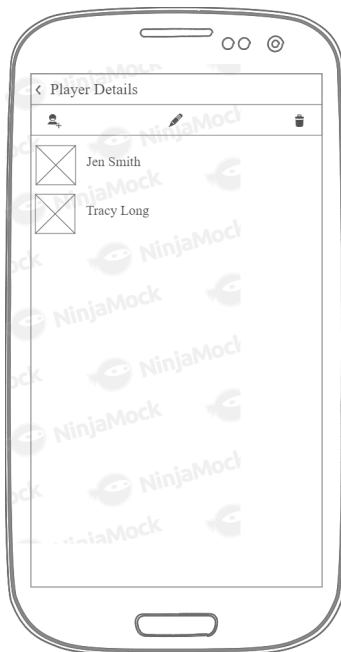
This is the team Name PopUp

Screen 4 - Dialog box for Game Details



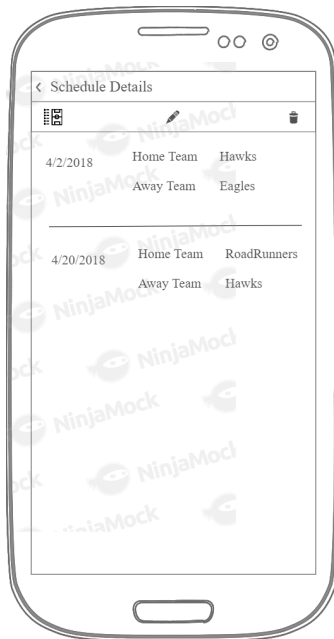
This is the Game Details Dialog box

Screen 5 - Player Detail Activity



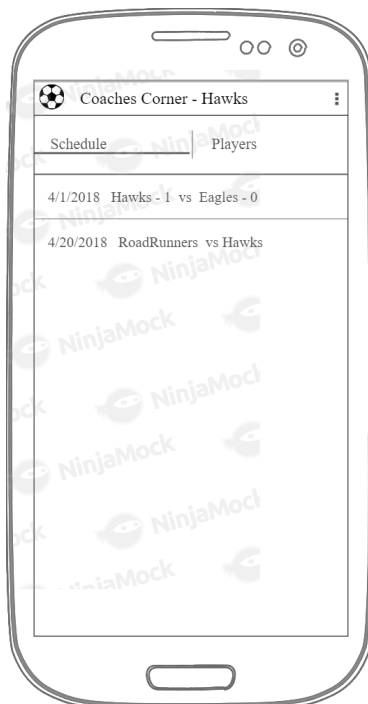
This is a new activity that will allow you to edit the players. Either manually or link over to Contacts.

Screen 6 - Schedule Detail Activity

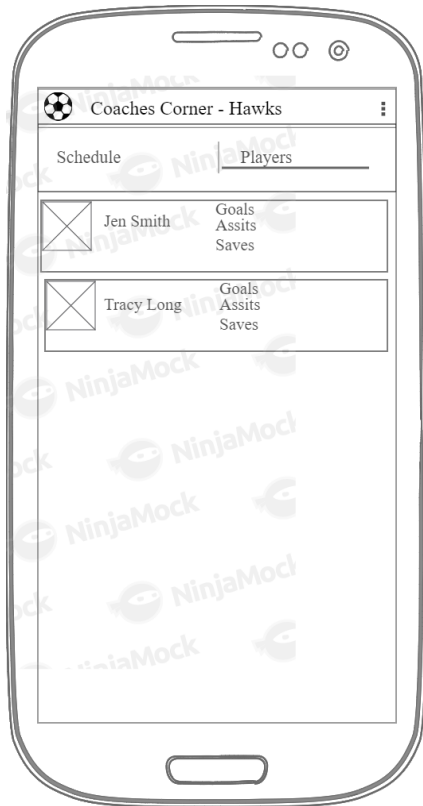


This activity will allow you to edit the schedule of games for the season.

Screen 7 - MainActivity showing Schedule filled out



This is back to the main screen after some games have been entered. Selecting a game will take you to the Game Activity Screen for that game.

Screen 8 - MainActivity showing players filled out

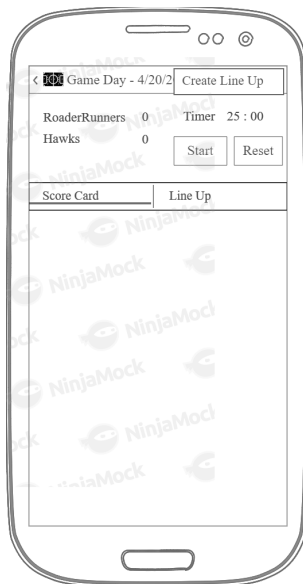
This is also back at the main activity showing the players info if you select the Players tab at the top of the screen.

Screen 9 - Game Day Detail activity



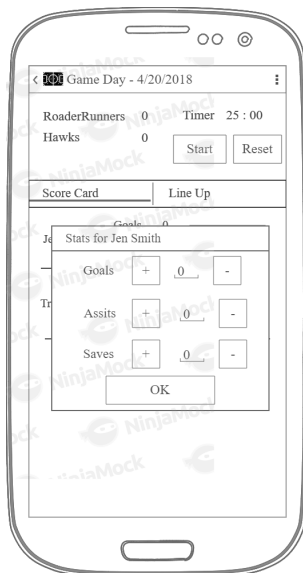
This is the game activity. The top bar will have information about the score and a Timer so that the coach can keep track of when to sub in players. The Scorecard tab will allow the coach to easily track stats of each play by pressing the Stat button and incrementing up/down the Goals/Assits/Saves a player has throughout the game. The Lineup tab will show the pre-defined Lineup for the game.

Screen 10 Game Day Detail Activity Menu to create lineup



This shows the inflated menu to allow the coach to create line ups for the game. This will link over to the LineUp Activity.

Screen 11 Game Day Activity showing Stat Dialog



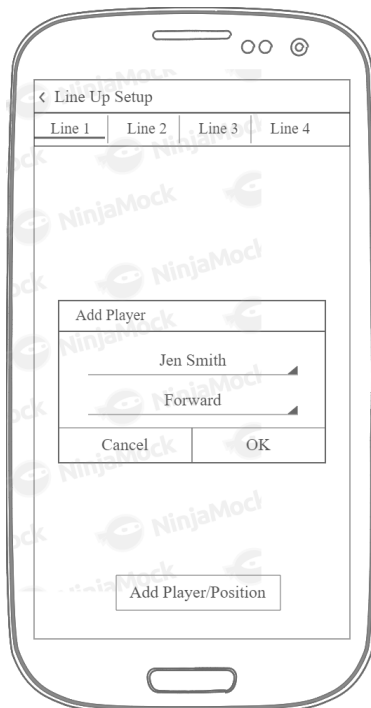
This shows the dialog popup for the stats window when the players stat button is pressed.

Screen 12 Game Day Activity showing Line Up



This shows what it might look like when you select the lineup Tab. It will display the pre-defined lineup and allow the coach to know who and what position each player will be subbed in for.

Screen 13 Creating Line Up activity



This screen shows the LineUp Activity and adding players to a line up.

Key Considerations

How will your app handle data persistence?

Data persistence will be done as follows:

- Sharedpreferences to store team name and game details
- Content Provider to store players, schedule, lineups
-

Describe any edge or corner cases in the UX.

The back buttons will all flow back to the MainActivity screen (Screen 1 Above). The corner menus are defined above that will flow to each individual detail screens.

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

- Anticipating using Picasso for displaying pictures. Might have away to add pictures to a game and for individual players.
- Will probably ButterKnife for easy access layout binding.
- Checking into using a Content Provider Library.

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

Since the concept of this app is that a coach will be using it during game day while the team is on the pitch, this app will be self contained with no network access required. The assumption is that most soccer fields are in remote locations with little to no wireless network access.

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

Create the project in Android Studio

- Setup the required libraries in build.gradle files
- Setup the java folder structure in a logical manner
- Load png files into drawable folders including app ICON.

Task 2: Implement UI for Each Activity and Fragment

- Create MainActivity with PageViewer
- Create menu for MainActivity
- Create Player Detail Activity
- Create Schedule Detail Activity
- Create Game Day activity
- Create ScoreCard Fragment
- Create Lineup Fragment
- Create Create Lineup detail Activity

Task 3: Create Content Providers

- Create Content Providers
- Define Databases
 - Player DB
 - Game DB
 - Lineup DB

Task 4: Link Databases with Layouts

- Setup code to integrate the DBs with the Layouts
 - Setup code to allow the coach to add/update/delete players
 - Setup code to allow the coach to add/update/delete game schedule
 - Setup code to allow the coach to add/update/delete game lineups.

Task 5: Enhance the layouts

- Enhance the Layouts to make it fabulous with FABs and interactive AppBars

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**"