GitHub Username: anothrNick

Scorch

Description

Scorch is a scorekeeping app that allows users to create and track tournaments and games for teams of players. This is a generic score keeper that is useful for any sport!

Intended User

Any user who wishes to a keep score for a game like ping pong, football, baseball, etc would use Scorch.

Features

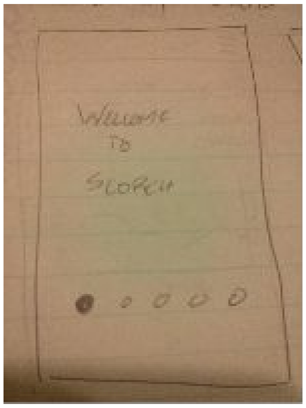
- Saves player, teams, game, tournament information on local device

- Displays tournament bracket

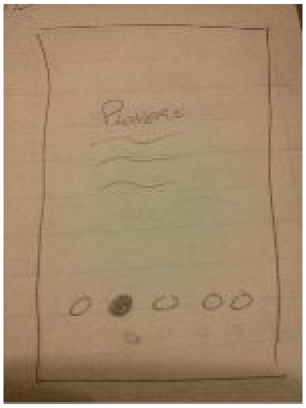
User Interface Mocks

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.

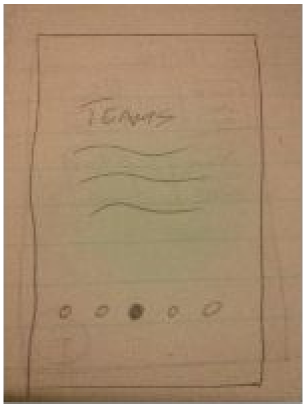
Welcome Screens



Welcome Screen
This is the first screen the user is presented with on initial app load



Players Screen
Part of the 'welcome' view pager, this fragment provides a description of the players functionality



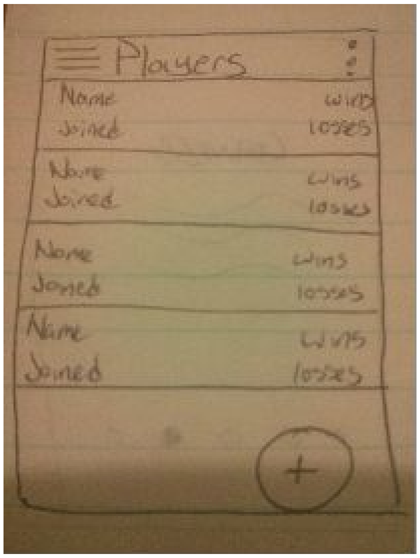
Teams Screen
This fragment provides a description of the team functionality

Games Screen
This fragment provides a description of the game functionality

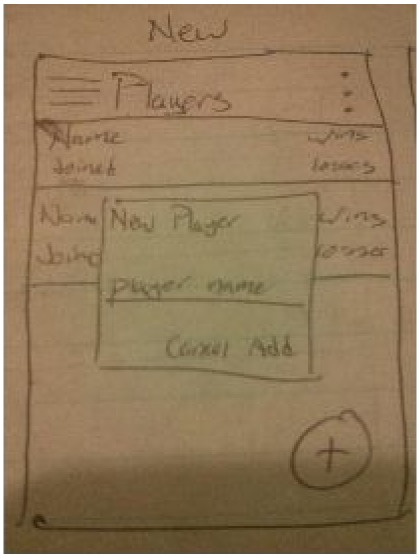
TOURNAMENTS

Tournaments Screen
This fragment provides a description of the tournament functionality

Player Screens

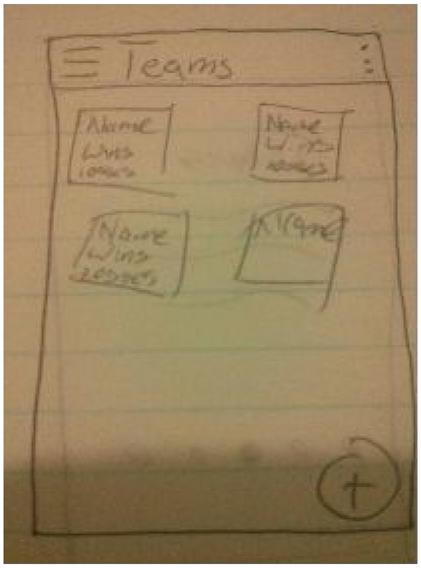


Player List
List of created players.

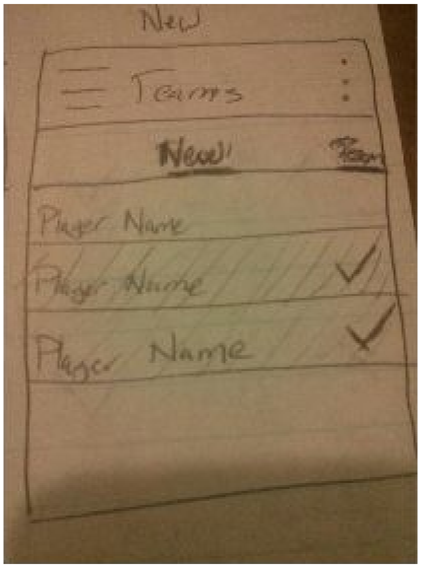


New Player
Enter name of new player

Team Screens

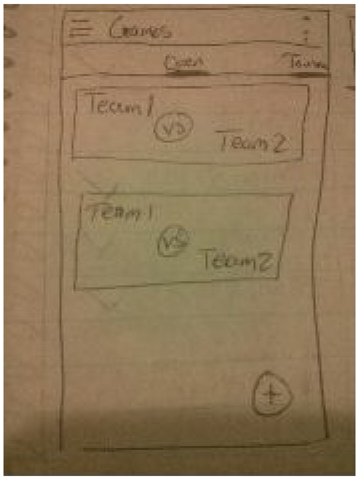


Team Grid
List of all created teams

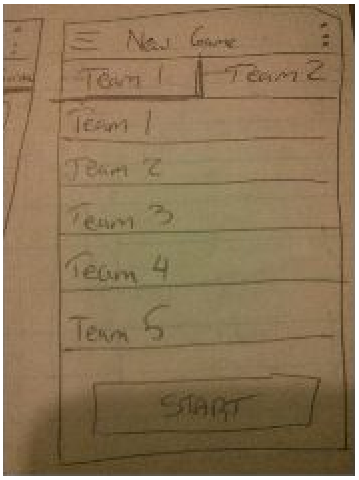


New Team
Select players to place on new team

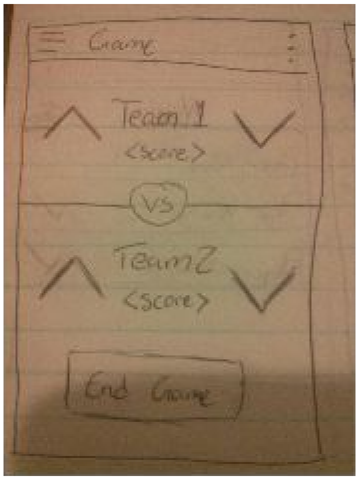
Game Screens



Game Lists
ViewPager of open, tournament, and completed games

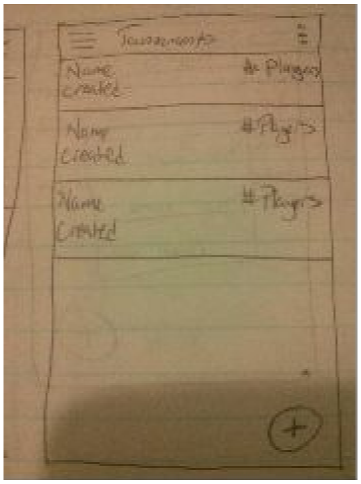


New Game
Choose which players/teams for new game

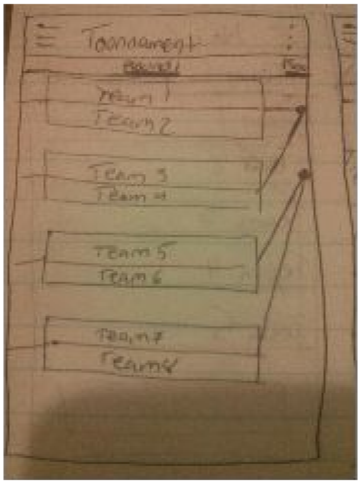


Game
Increment or decrement teams' scores. End game when complete

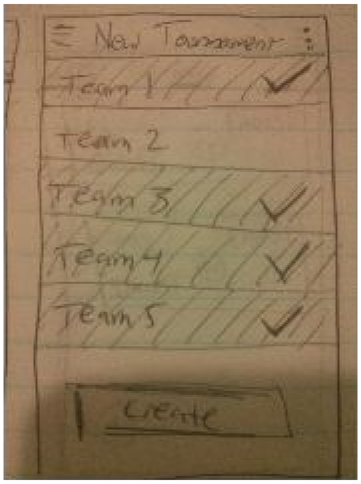
Tournament Screens



Tournament List
List of open tournaments

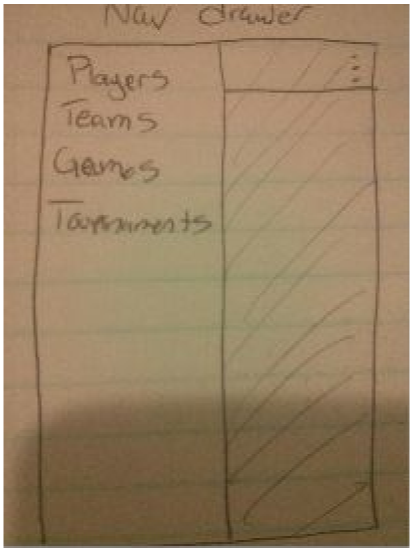


Tournament View
View pager of tournament rounds



New Tournament
Select players/teams to participate in the tournament. This will determine how many rounds will exist

Navigation Screens



Navigation
A navigation drawer will be used for Scorch's navigation

Key Considerations

How will your app handle data persistence?

A custom database contract will be created to store data on the local device. Data storage may be moved to a server, accessible via REST API, in the future.

Describe any corner cases in the UX.

Based on how the tournament screen will be implemented, how will we display the lines connecting the different rounds? Assuming a canvas, how do we ensure when scrolling through the viewpager, the rounds still look like a single screen?

When creating a game, should users be able to select players AND/OR teams?

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

No libraries will be needed for this iteration of Scorch

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

- Create git repository for Scorch
- Find icon for app
- Look into color schemes

Task 2: Implement UI for Each Activity and Fragment

- Main Activity
- Navigation Fragment
- Tournament Fragment
    - List
    - Individual tournament
        - ViewPager
            - Each round is a view
    - New tournament (select teams or players)
    - Chromecast view
- Player Fragment
    - List
    - New player
    - Individual player
- Team Fragment
    - List
    - New team (select players)
    - Individual team
- Game Fragment
    - List
        - ViewPager
            - Current Games
            - Tournament Games
            - Finished Games
    - New game (select teams or players)
    - Individual game
    - Chromecast view

Task 3: Create Database Contract

- Schemas
  - Players
  - Teams
  - TeamPlayers
  - Games
  - GamePlayers (or teams)
  - Tournaments
  - TournamentGames

Task 4: Create Models to Contracts

Model wrapper to create level of abstraction for each content provider. Scorch's data storage may be moved to a REST API in the future, this abstraction would reduce the amount of work needed to support that change on the client.

- Players
- Teams
- TeamPlayers
- Games
- GamePlayers (or teams)
- Tournaments
- TournamentGames

Task 5: Attach UI To Data Models

For each UI element, connect the appropriate date model to perform the necessary CRUD operation