ASSISTANT COACH

Assistant Coach provides a unique collaborative environment for baseball teams. It allows the coach the ability to communicate with players, share the team lineup in real time, and players to interact with one another.

Created By Team 23:
Mark Schultz
Josh Ford
Bilal Ahmed

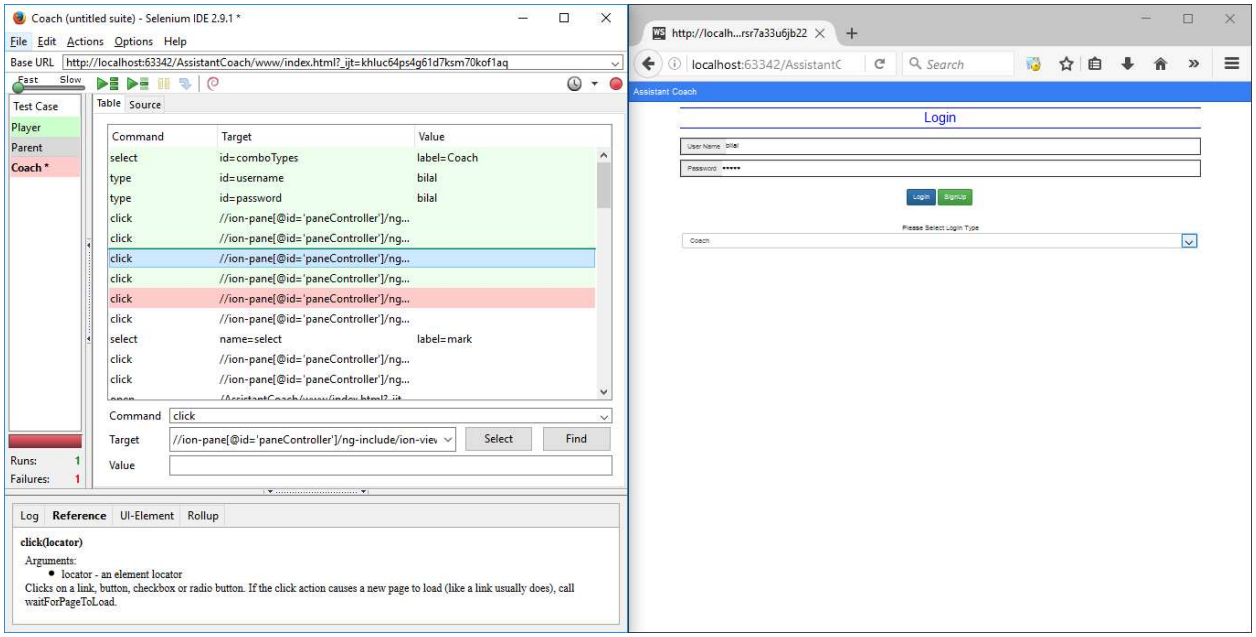
Project Deployment & Management

Introduction

Our project titled “Assistant Coach” is basically a team management and collaboration mobile application that we built using the hybrid application development framework called “Ionic Framework”. It basically provides a collaboration and team management bridge between the team players, their parents, and the coaches, in our case there is one coach. The application is fully dynamic and uses the realtime NoSQL database (Firebase) for it’s storage tier. The players can able to view the current lineup of their team. Also the players can signup and login to their own accounts using the Login interface. Players are able to update their basic information which is updated in real time in the database. Players can chat in the public team chat group. The chat is completely synchronized among all the interfaces. The parents of the players can see the current team setup. Also parents signup using their child’s user id but their own password. The parents can update their information. There is one another interface for coach in which he can setup and modify the team lineup. The coach can signup for their own account with the key provided by the application administrator. The team view has one other feature of the ground map which provides the real time placement of the players on the field and gets updated dynamically as the changes in the team lineup are done. There is another interface connected to the Google Conversation API through which the coach can query basic information about the team.

The coach can issue voice commands to modify the team lineup. The coach speaks to the Google API agent and issue some of the data modification commands to the application. For example coach can swap the field placements of the two players. Coach is able to query the information of the player. Coach can query the field placement of the player.

1. Testing

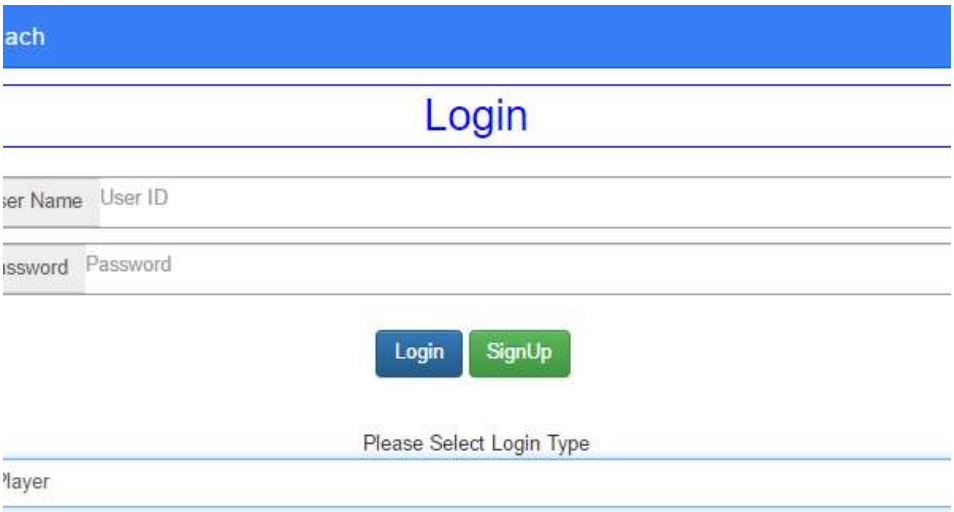


2. Implementation

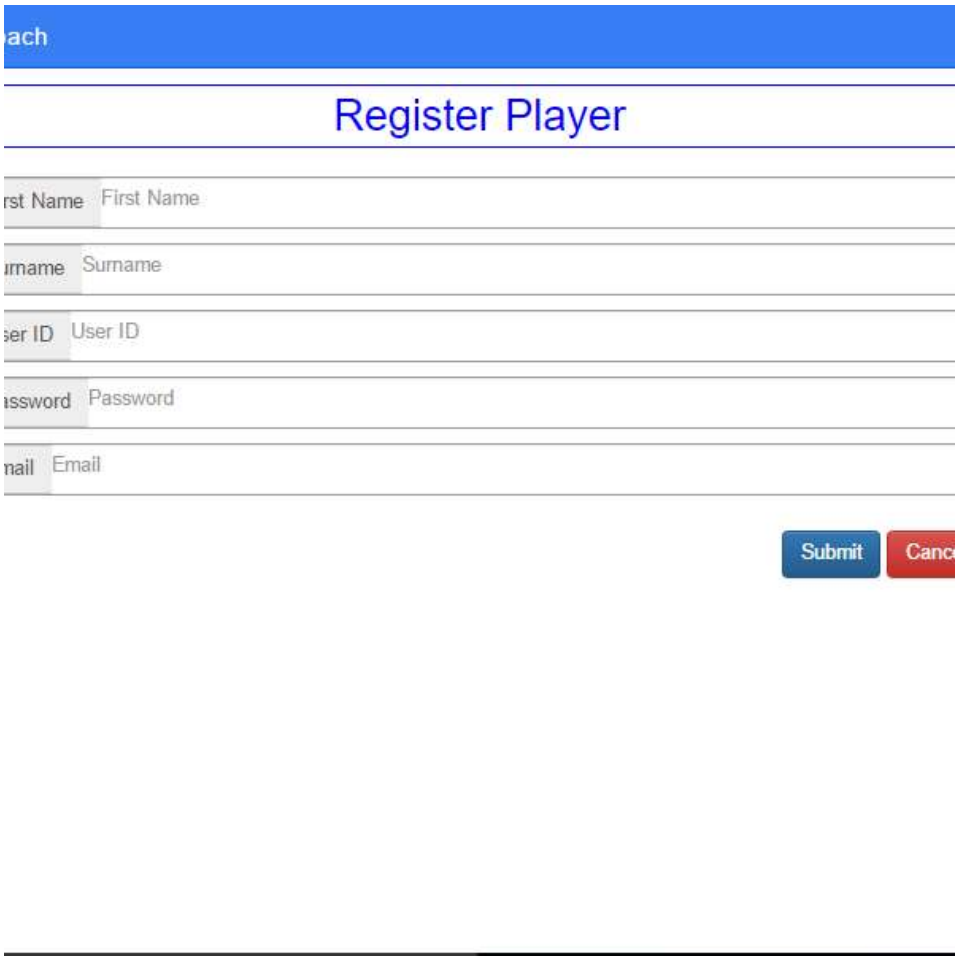
The implementation for the application is simple. There are basically three interfaces for the application. Player, Parent, Coach. The three interfaces are described below.

I. Player

Login Screen

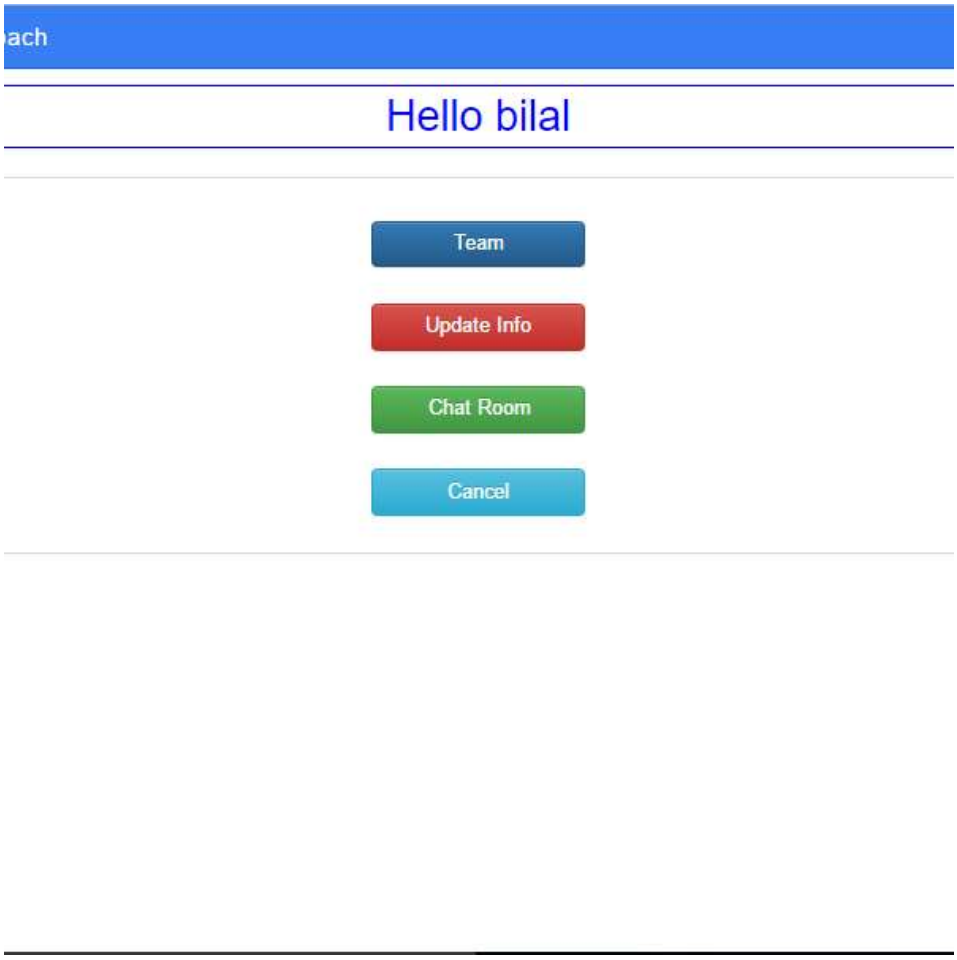


Player Signup

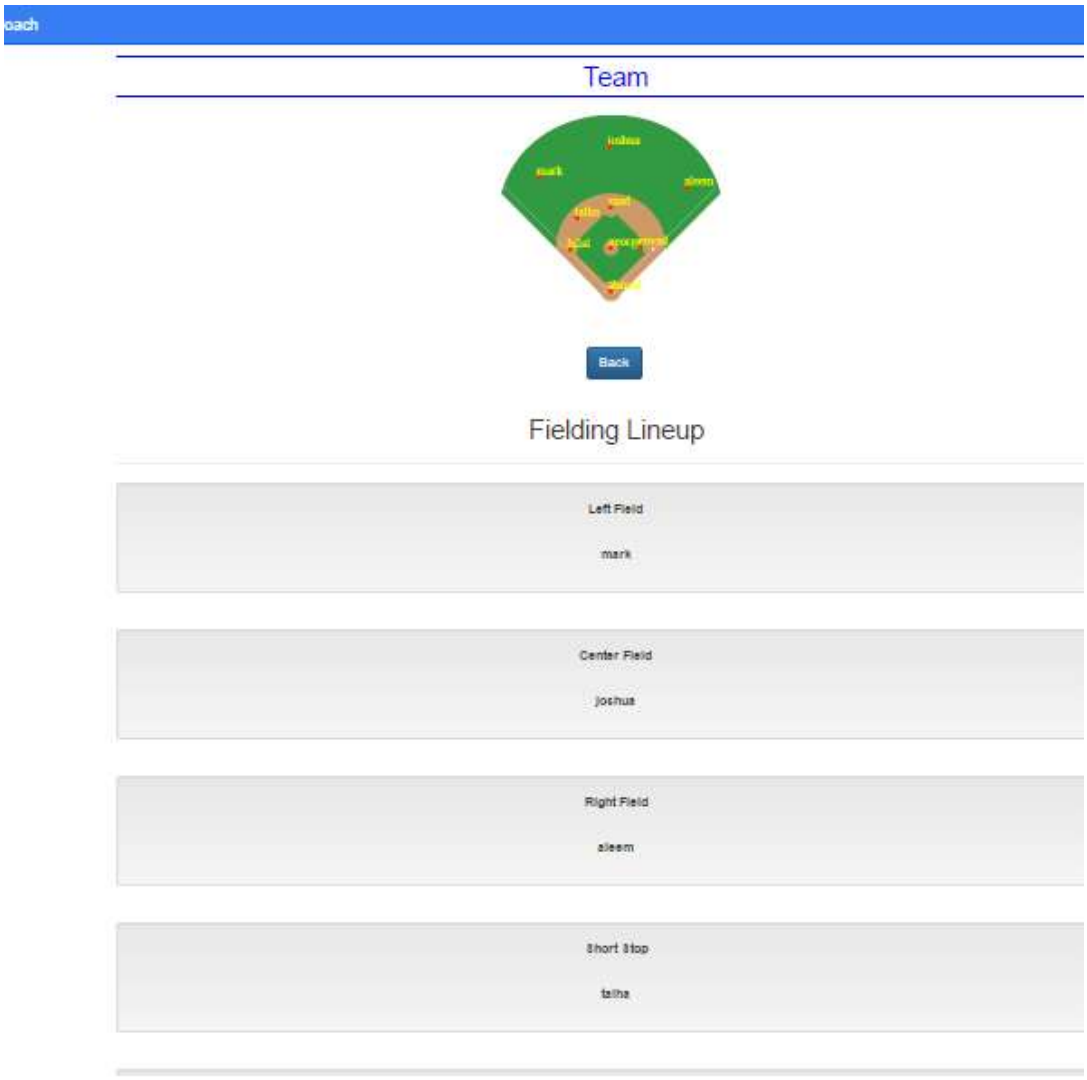


The starting point of the interface is the login screen which is common in all the interfaces. In login screen, the player can signup as the new player in the system. Next the player can sign in using the data he used at the sign up time. The main screen appears for the player in which can view the current read-only details of the team with the fancy ground map that shows the players' placement on the field. All the batting and fielding lineups are shown in that interface.

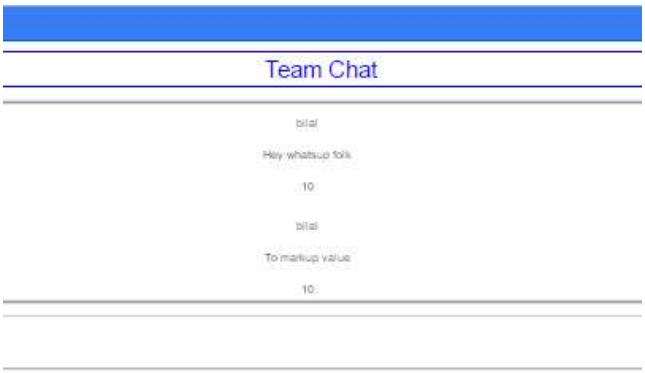
Player MainView



Player TeamView



Player ChatView



The next interface player is going to interact with is the chat interface. It is the real-time chat box that is completely synchronized among all the players session including the coach. The players can collaborate with each other and chat with the coach as well. The coach can call for the team meetings on the chatbox.

Player UpdateView

The Application part is implemented using “Ionic Framework” which is the hybrid application framework for mobile applications. The client UI is implemented using standard web technologies like HTML, , CSS and AngularJS. The overall application development and testing is performed using local http-server from NodeJS.

Firebase engine is used as a storage engine for the application engine. All the data pertaining to the players, parents, coaches, locations, team is stored in the form of JSON objects in the firebase database. Publish – Subscribe based approach is used to query the data from firebase. The data modification event handlers are described in the AngularJS controllers which executes once the data in the certain database reference gets updated. This feature allows us to have a dashboard like functionality.

For UI appearance, bootstrap framework has been used to design different application views.

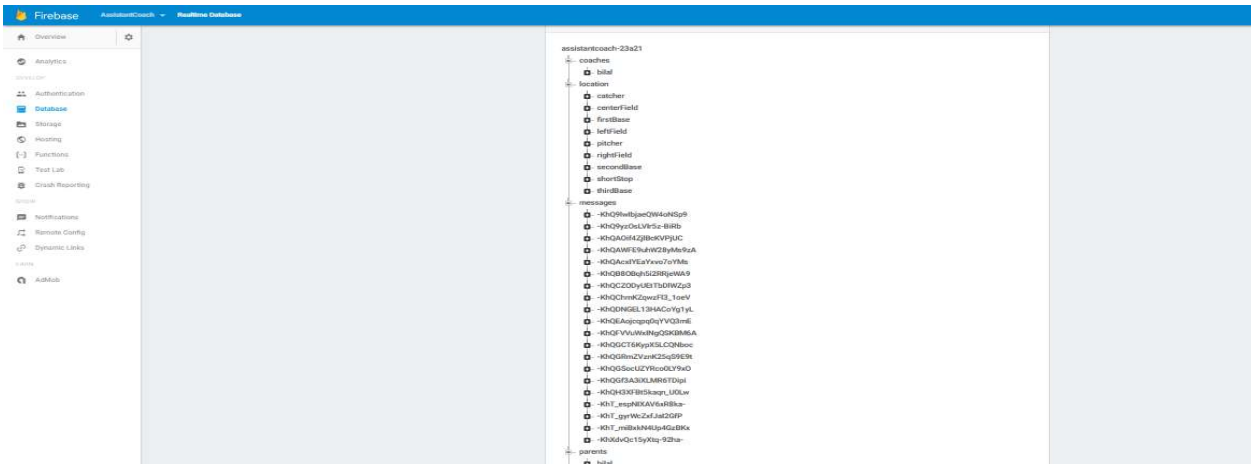
II. Coach

The coach interface is much like players plus he has some of the extra features that the players don’t have. The coach can modify the team lineup anytime which is reflected among all the players’ sessions in no time. The coach can chat with other players in the same manner as the other players can. Same chatbox is shared among players and coach. For simplicity, each player lineup option has a select field that gets real-time data from the database. And the cool feature is that it updates the available player options instantly by validating the options with the current state of the database. For example if player A is not assigned any part in the team, he’ll be shown in all of the possible team positions selection menu until assigned to some. Coach can clear up any certain position to make it available for other players to occupy. Each field position can be updated independently.

Coach TeamView

3. Deployment

Firebase Model

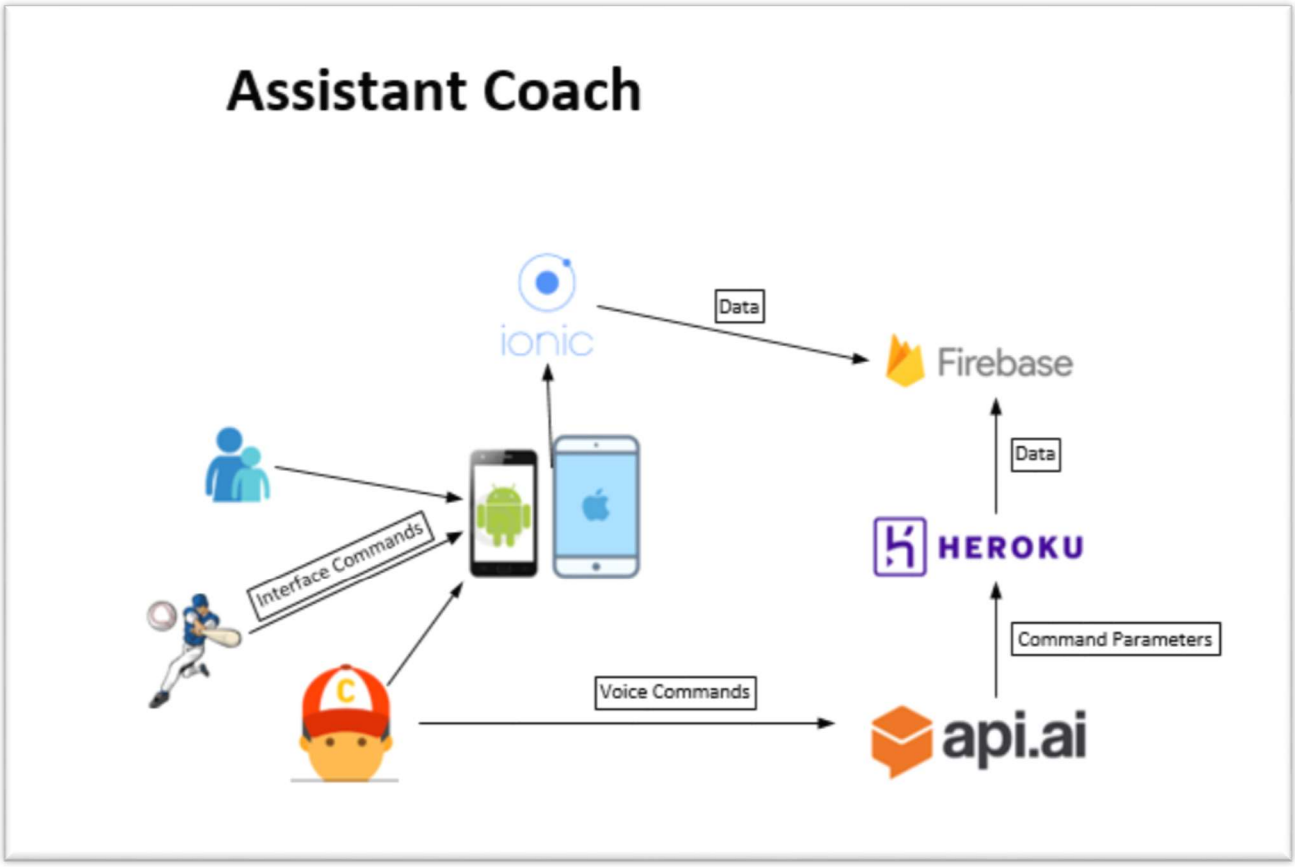


4. Conversational Interface

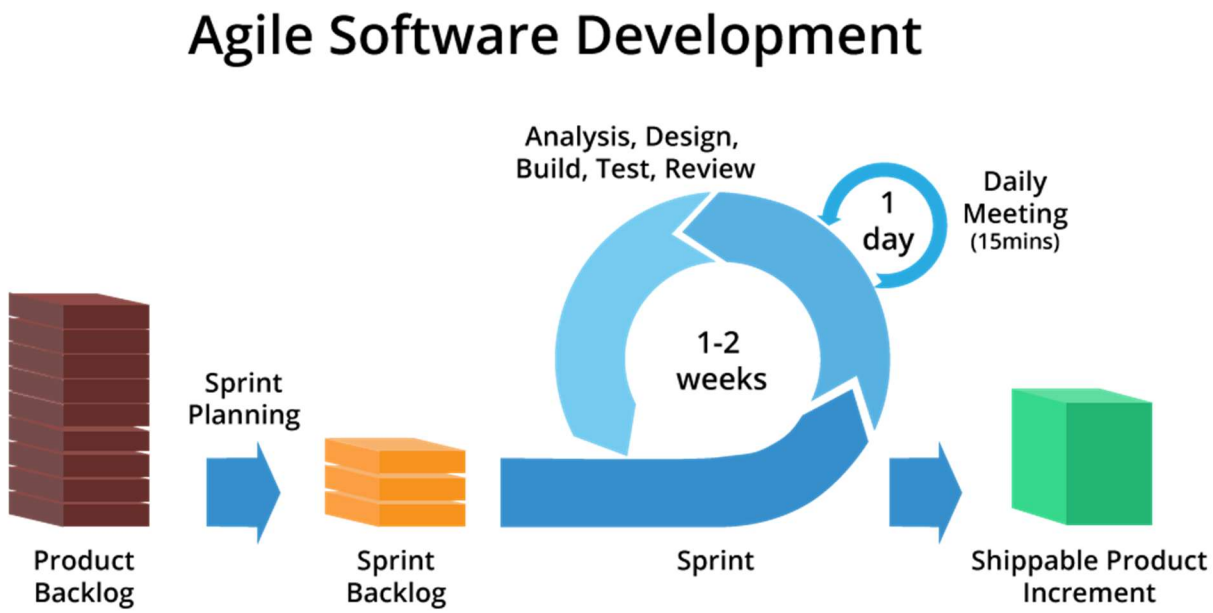
The coach can send voice messages to the team's chatbox using Google Conversation API interface. The coach can issue voice messages to the team and the text is going to be send to the chatbox for that voice command. The API.AI agent is trained for all the possible entities of the chat that the coach might be needing for example : if the coach says "send I am not coming tomorrow". So the bot will recognize the "send" as an operation and "I am not coming tomorrow" as the message body and hook an external web service which will call the interface that modifies the chat.

Another functionality that the coach can do is to manage team lineup using voice commands. The coach is able to query any field position with the updated player name. Also the coach can issue command to swap the players on two different field position.

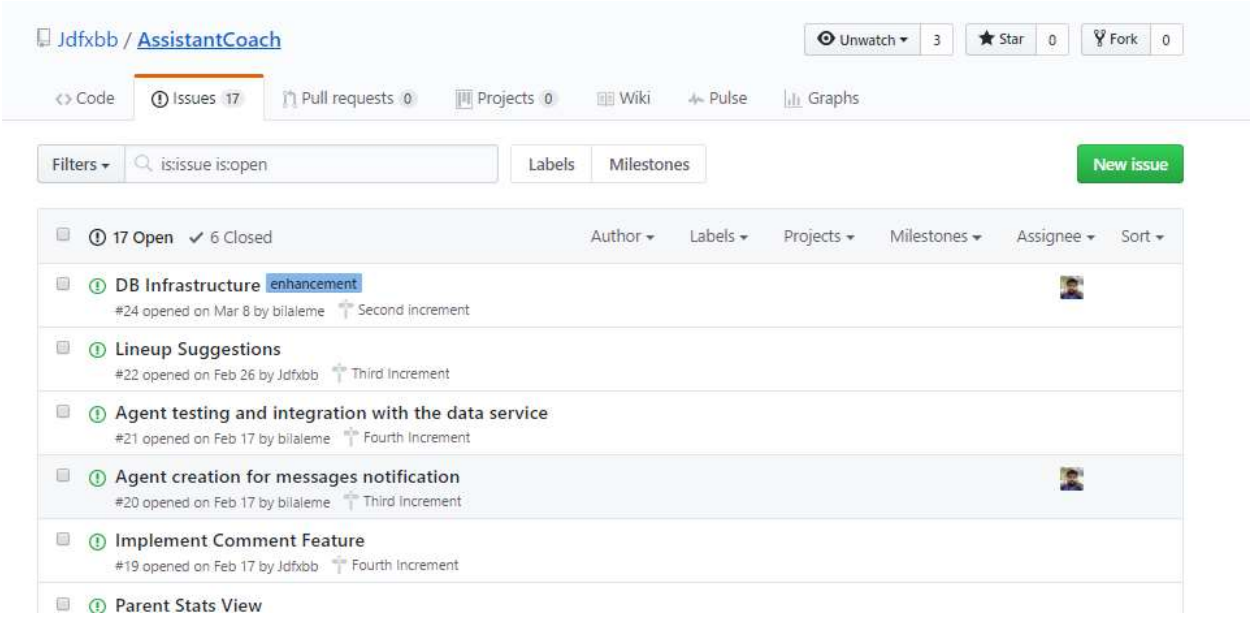
5. Architecture



6. Project Management



Our project was completed using Agile development principles. Each increment, we focused on creating working software, and building on top of the previous increment. Because most of our development progress was from remote locations, we were able to work together by choosing tasks from our issue board and completing them. Although we were all assigned general task, we granted one another leeway to choose issues and complete them on our own initiative.



7. Project Evaluation

In the end, we were quite pleased with our results. Our team is short one member, yet, we were able to produce a project that competed well with the rest of the projects. In the end, there were three aspects of our software that gave our product an edge, which makes our product a pragmatic choice for any little league team to use. First, our product is individually tailored to the needs of a baseball team. Second, our project allows updates that a coach makes to be visualized by the players on a picture of a baseball diamond in real time. Third, we used voice command so that the coach can dictate changes in the lineup and announcements to the team.

Why shouldn't a baseball team just create a Google Group, or a Facebook Group to communicate with one another. After all, there is nothing inherently unique about a chat room, or a list. The reason our software is more practical is that it is designed specifically with the needs of a baseball team in mind. The positions on the field are set, and the coach just needs to put names on the field. Sure, a coach could type out a lineup for the upcoming game every week, but sometimes saving clicks is the difference between a coach using the product and a coach falling behind on updating the lineup.

This product also provides a visualization for the lineup. As the coach updates each position, a baseball diamond on top of the page is updated for all players to see in real time. This adds a little bit of flair to the task of assigning positions to each player. While this is certainly practical, we also think this feature will be enjoyable to the players.

Let's be honest, a little league coach can be a little bit lazy. Nothing against the coach, but usually it is one of the fathers who has a full time job. The dad is coaching the team as a favor to the rest of the parents. Our goal in this project is to make the tasks of running a baseball team as simple as possible. That is why we introduced voice commands using AI.API. This way, the coach can dictate announcements and lineup changes to the software. The purpose of this software is to reduce the amount of work for the coach, not create another piece of software to learn.

Of course, as described in the rest of the paper, there are other key features of this project that make it a great product. However, I wanted to use the project evaluation to focus on the features that make this product not only usable, but unique to other products available.

Assistant Coach

Project Proposal

I. **Introduction**

a. Assistant Coach is a mobile application that will aid little league coaches and other youth leaders with organization and communication. Assistant coach will initially focus on little league baseball coaches but the application will be scalable to work with any sport and even non-sports activities.

II. **Project Goal and Objectives**

a. Overall Goal

i. The goal of Assistant Coach is to provide a simple and quick mode of communication between coaches, players, and parents. This application will give coaches an organizational tool where they can store player contact information and track statistics and player performance.

b. Specific Objectives

i. Little league baseball coaches have the heavy task of organizing nine or more young athletes into a manageable baseball team for the summer. This involves setting practice times and making sure parents know when to take their kids to practices and games. Little league baseball coaches also have to keep track of important statistics like playing time of players, pitch count for pitchers, and unique abilities and strengths of certain players. This application makes the difficult task of coaching a little league team more manageable.

c. Specific Features

i. Player Profile

1. Name
2. Age
3. Contact Information(Player and Parent optional)
4. Positions
5. Strengths
    a. Speed
    b. Power hitter

c. Contact hitter
d. Good glove
e. Good arm

6. Statistics
    a. Batting Average
    b. Innings Played
    c. Innings Pitched

ii. Send message
    1. Sends message to all players or parents

iii. Set lineup
    1. Set the starting lineup for a game

iv. Enter schedule
    1. Enter practice schedule
    2. Enter game schedule

v. Parent/fan view
    1. Parents can sign in to a specific view of the app
        a. Cannot make changes to team
        b. View schedule
        c. View player statistics (if coach makes them visible)
        d. View lineup

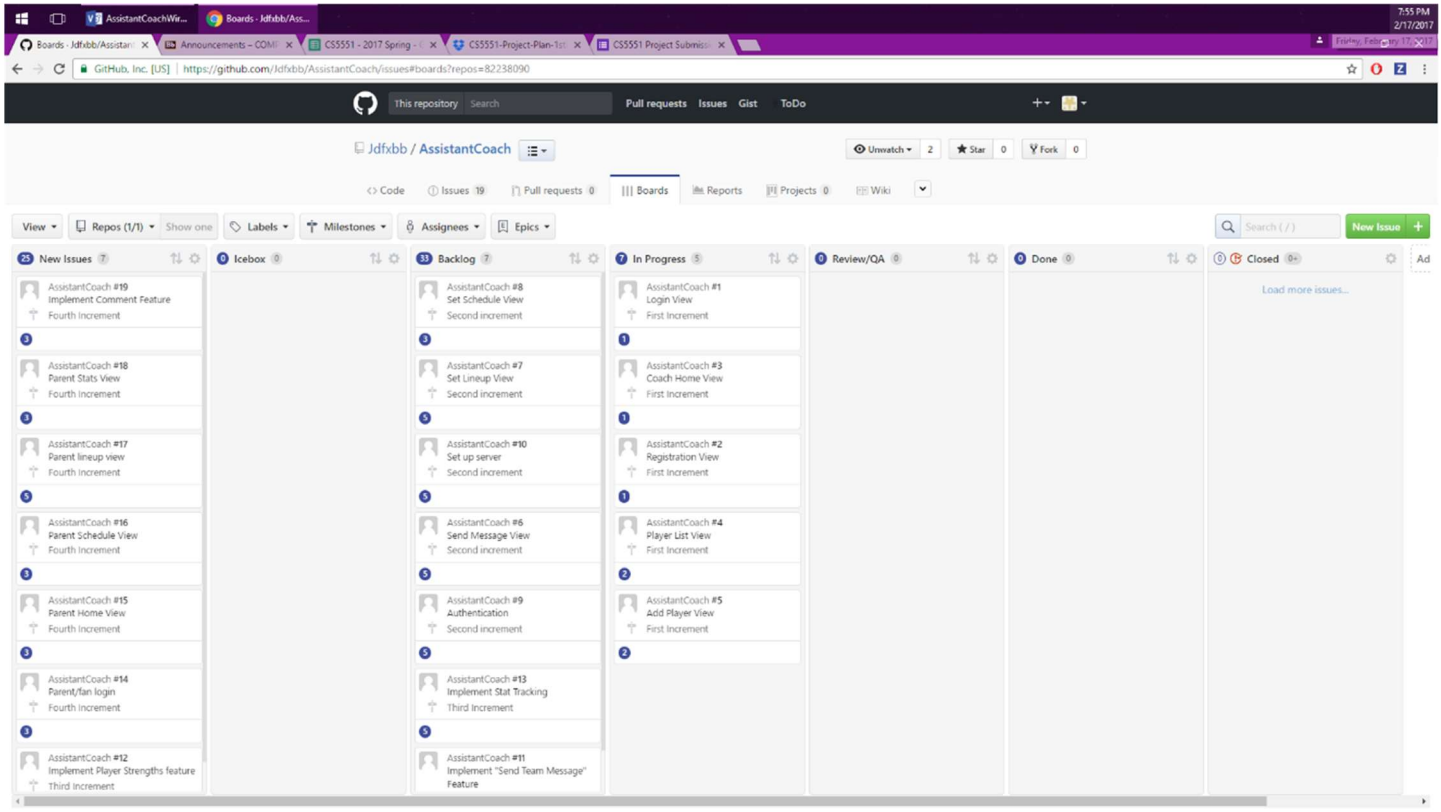
d. Significance

i. There are many apps that provide group chat or mass communication to users. There are also apps that can track statistics for a sport. This app is unique, however, because it provides all of these features in one place. This app will also provide a unique experience to parents and fans because they will be able to see the changes that the coach is making to the team.

III. Project Plan

a. Schedule

i. **Stories**



b. Project Timelines, Members, Task Responsibility

i. Increment I 17 February 2017

1. Login View: Joshua
    a. The coach can login to the app
2. Coach Home View: Mark
    a. The coach can see a home view with buttons to the features
3. Registration View: Bilal
    a. The coach can register if he/she is a new user
4. Player List View: Joshua
    a. The coach can view a list of players
5. Add Player View: Mark
    a. The coach can add players to the team

ii. Increment II 10 March 2017

1. Set Schedule View: Bilal

a. The coach can set practice times and game times

2. Set Lineup View: Joshua
    a. The coach can set the lineup for a game
3. Server: Joshua
    a. The coach can make changes to the team and they are saved on a server
4. Send Message View: Mark
    a. The coach can send a message to his team via email API
5. Authentication: Bilal
    a. The coach can sign in with email, Google+, or Facebook

iii. Increment III 7 April 2017

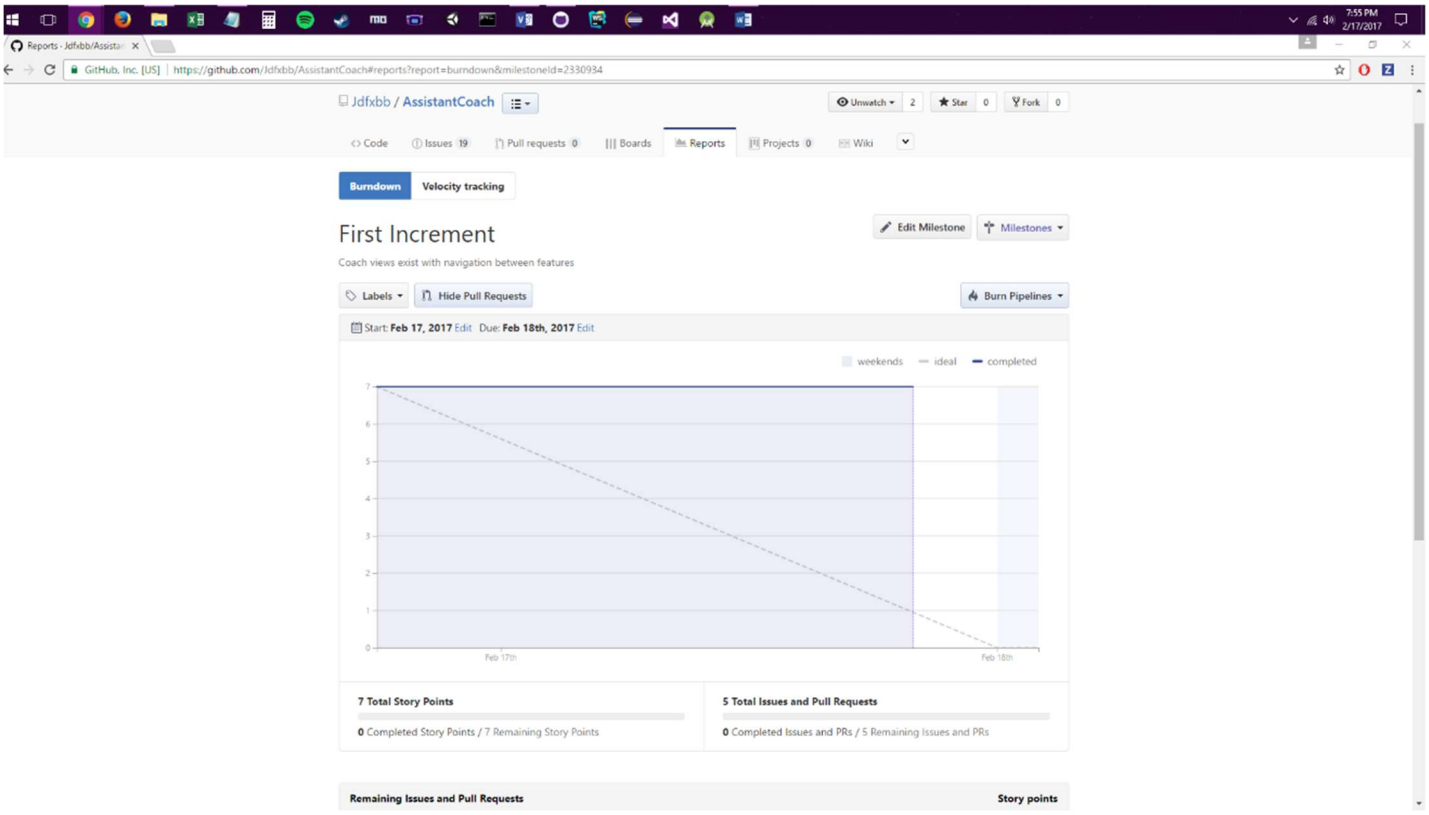
1. Strengths Feature: Mark
    a. The coach can select strengths of certain players based on their ability
2. Stat Tracking: Bilal
    a. The coach can enter the stats for a game and view aggregate stats for a season or multiple seasons
3. Lineup Suggestions Mark
    a. The coach can see a suggested lineup based on stat history
4. Google Conversation API Joshua, Bilal
    a. User can listen to the new chat messages from the team using conversation API and can know about the players in the team and their respected roles.
    b. User is able to ask questions regarding the team using google conversation API.

iv. Increment IV 26 April 2017

1. Parent login: Joshua
    a. The parent can login to the app
2. Parent Home View: Mark
    a. The parent can access a home view with buttons to different features

3. Parent schedule view: Bilal
    a. The parent can view the team schedule
4. Parent lineup view: Joshua
    a. The parent can view the lineup for a game
5. Parent stats view: Mark
    a. The parent can view statistics for players
6. Comment feature: Bilal
    a. The coach can make comments on the progress of players
    b. The parent can view comments on their child
7. Run the trained agent on Google Home. Mark, Joshua
    a. User is able to send a message to the team via Google Home.
    b. (Optional) Coach is able to set the team lineup via google home.

c. Burndown Chart

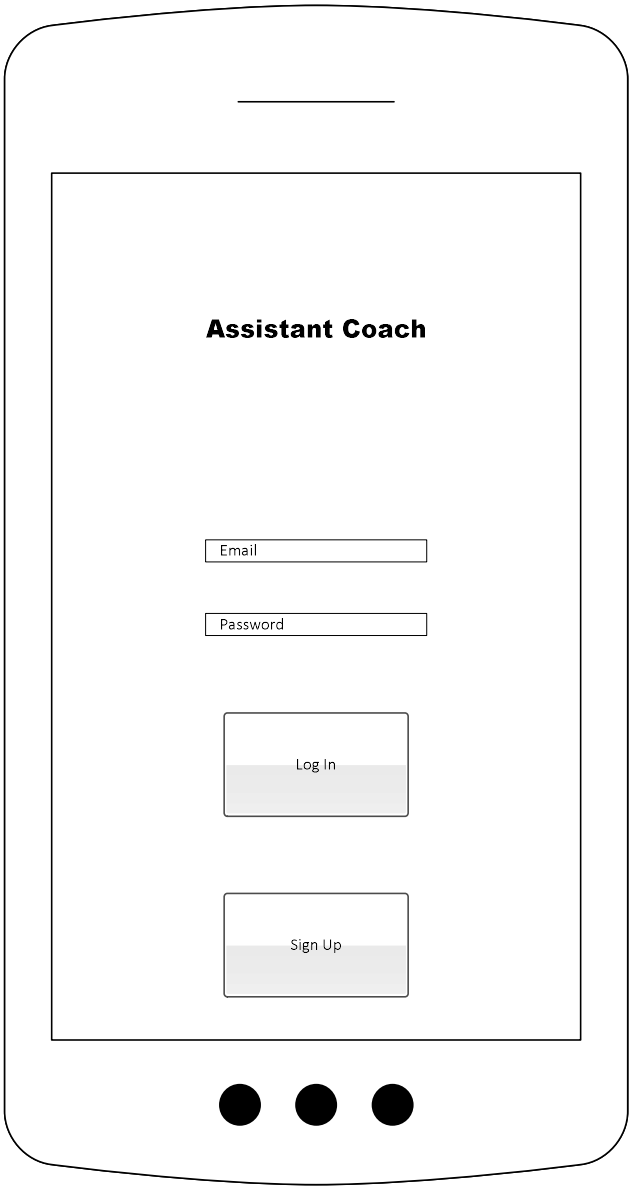


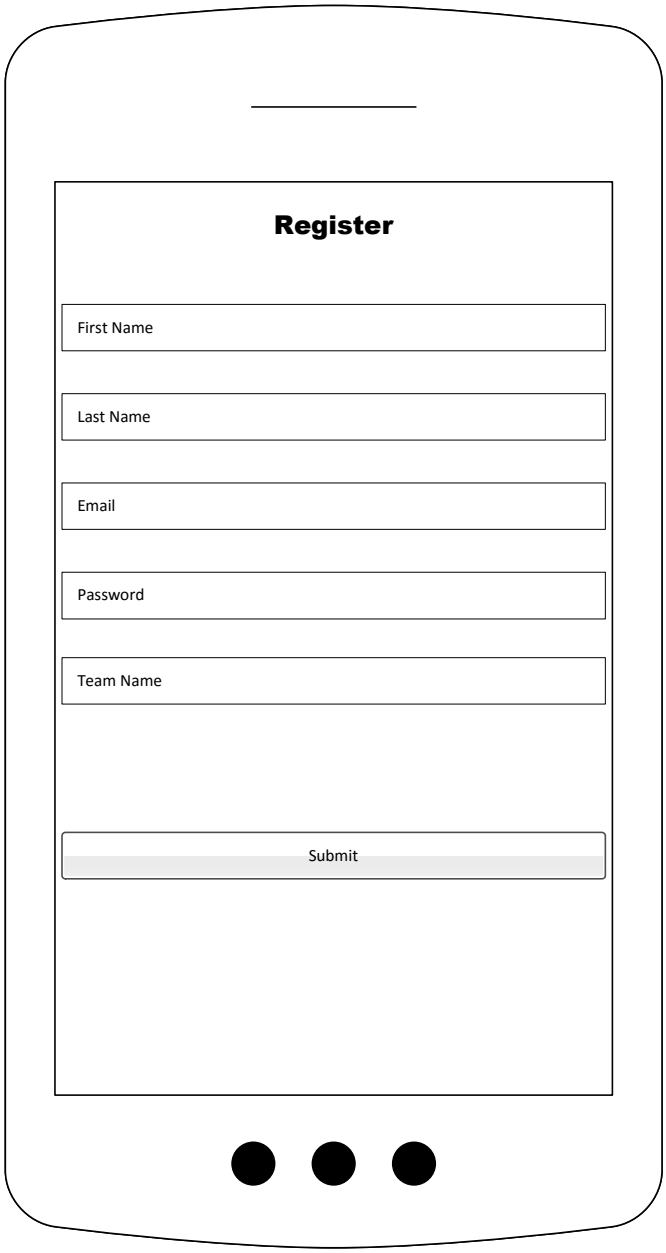
IV. First Increment Report

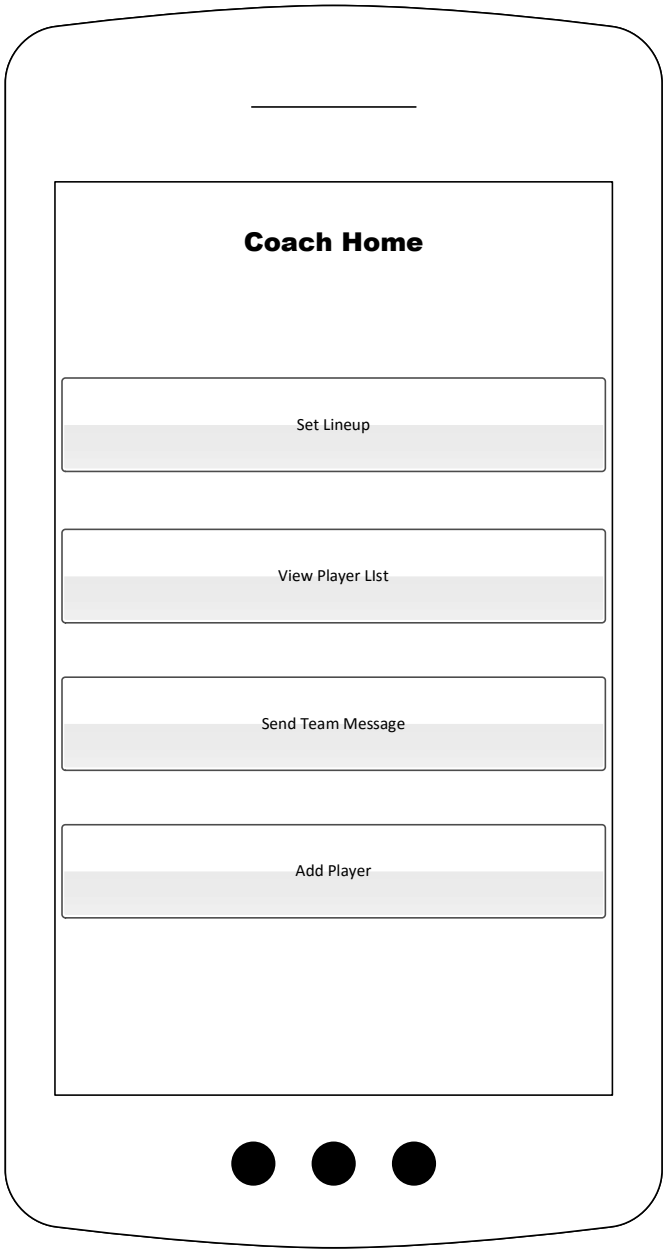
a. Existing Services

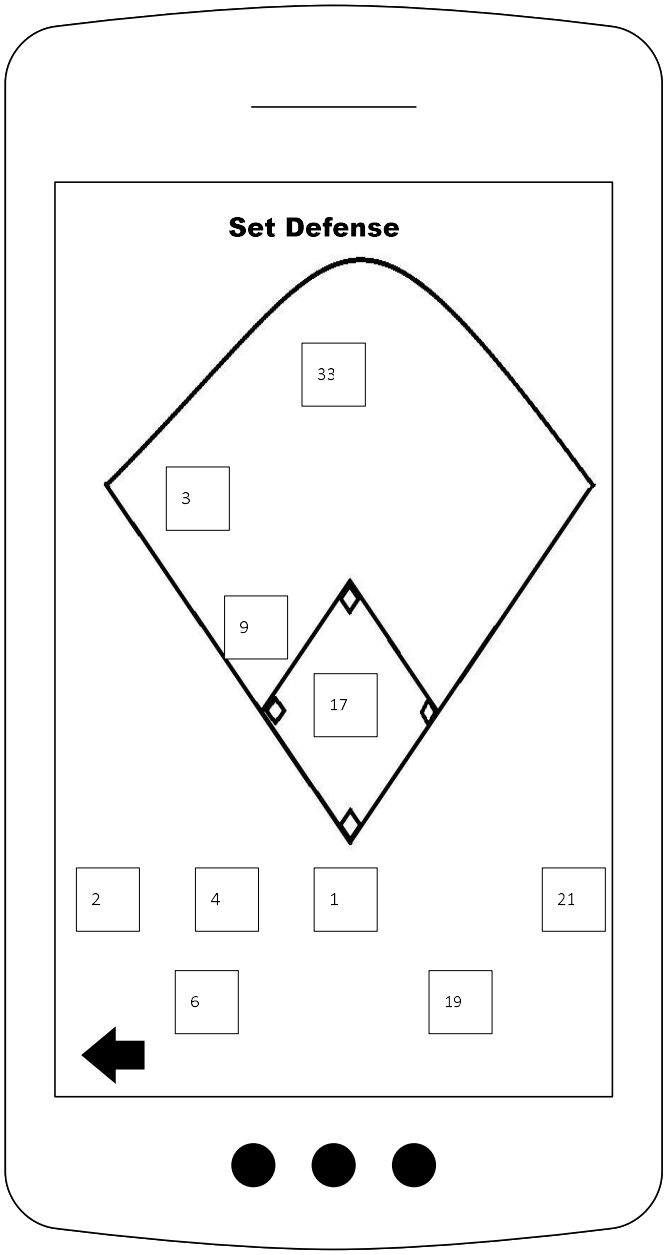
i. Google Gmail API
ii. Android Studio
iii. Visio

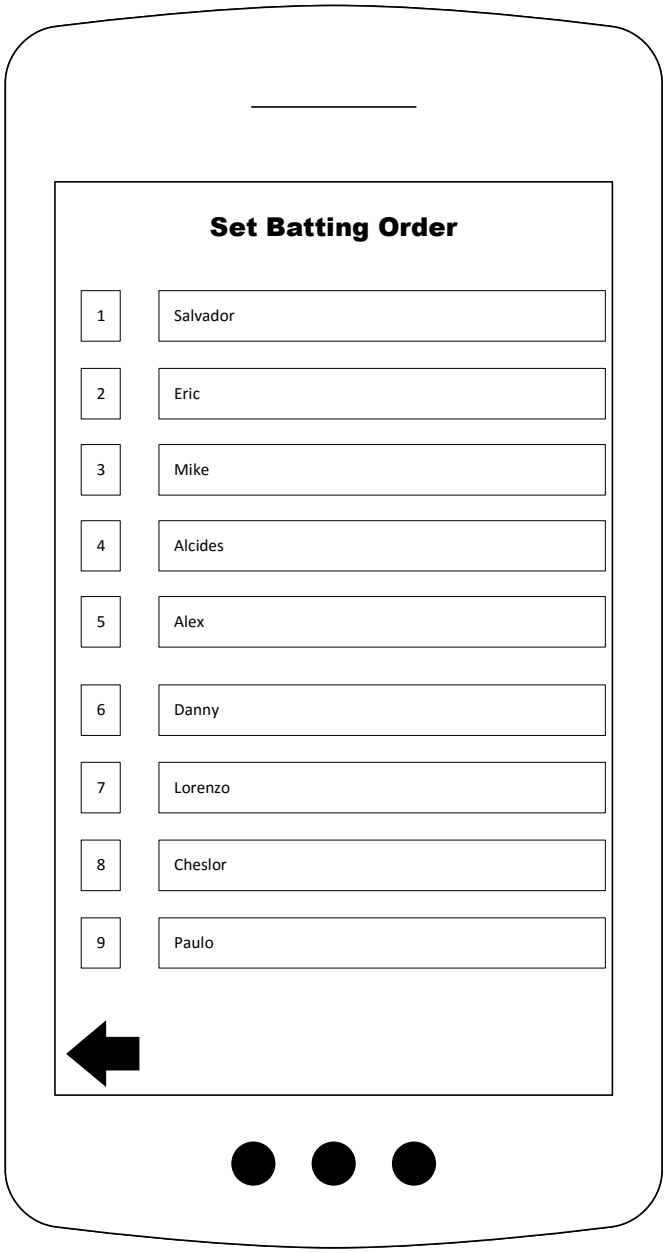
b. Detail Design of Features

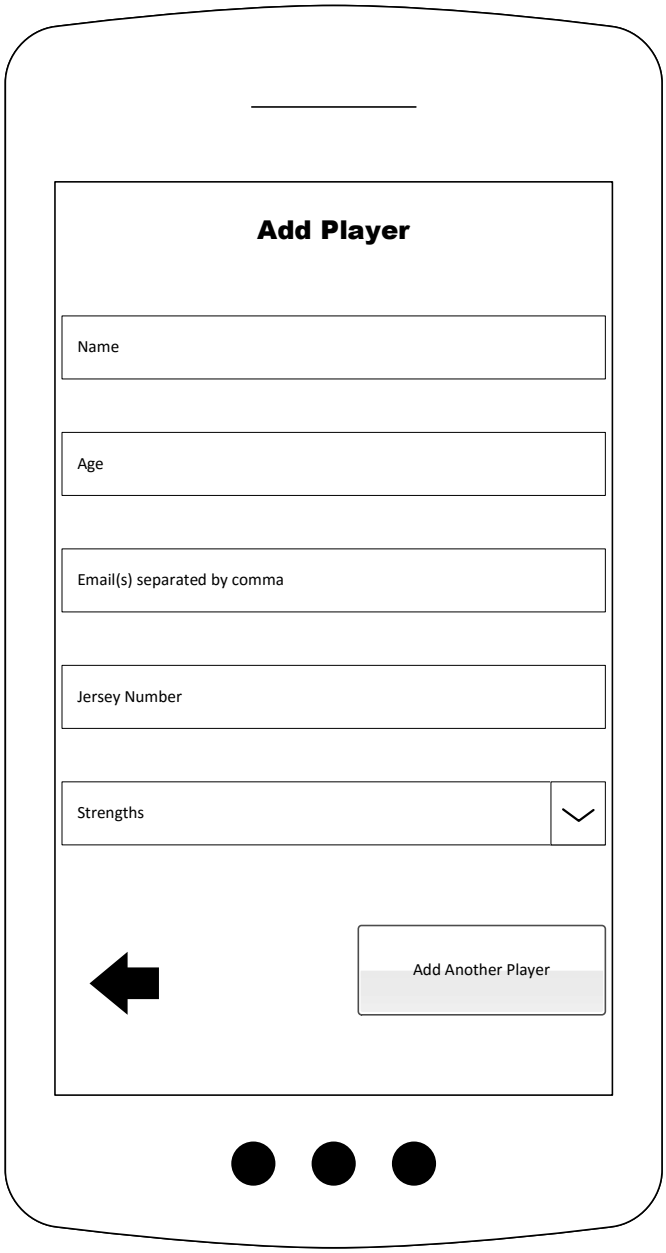


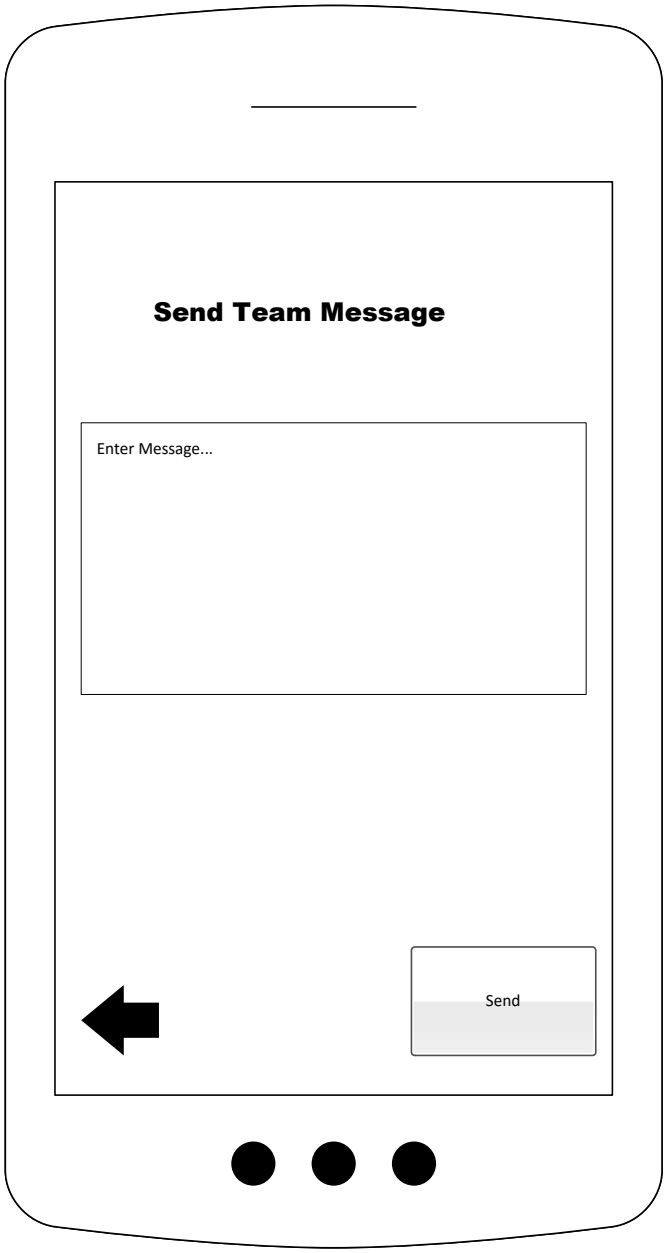

Register
First Name
Last Name
Email
Password
Team Name
Submit


Coach Home
Set Lineup
View Player LIst
Send Team Message
Add Player

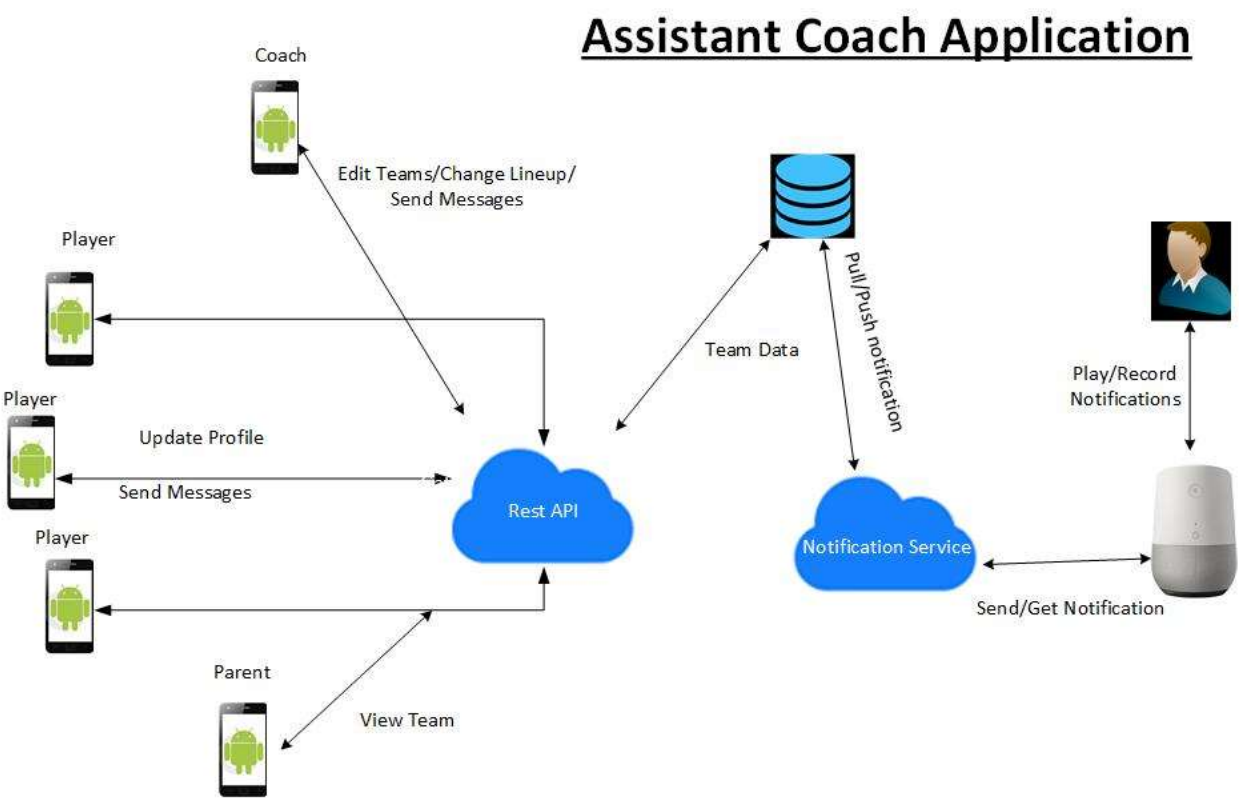

Set Defense
33
3
9
17
2
4
1
21
6
19


Set Batting Order
1
Salvador
2
Eric
3
Mike
4
Alcides
5
Alex
6
Danny
7
Lorenzo
8
Cheslor
9
Paulo


Add Player
Name
Age
Email(s) separated by comma
Jersey Number
Strengths
Add Another Player


Send Team Message
Enter Message...
Send

d. Architecture Diagram



PRESENTATION SLIDES

AssistantCoach

By: Mark Schultz
Bilal Ahmed
Josh Ford

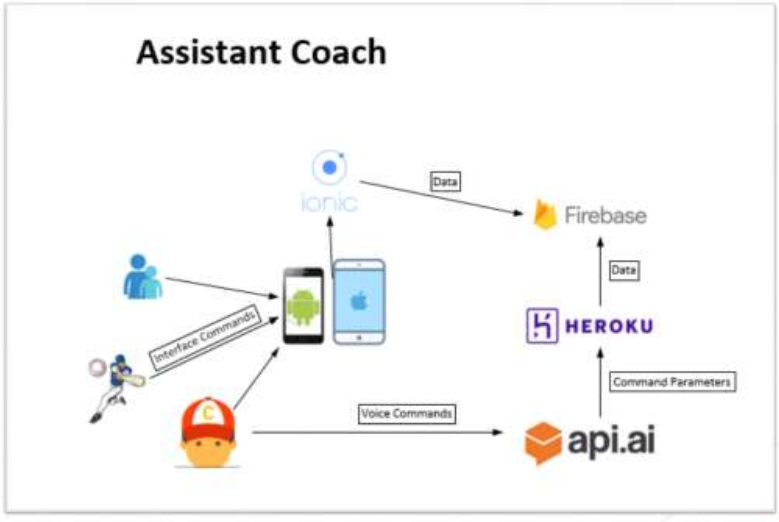
Function

- Application intended for easier interaction for a baseball team
- Provides a means of communication between a Coach and his players
- Individually tailored to the needs of a baseball team (Coach can adjust batting order and field position)

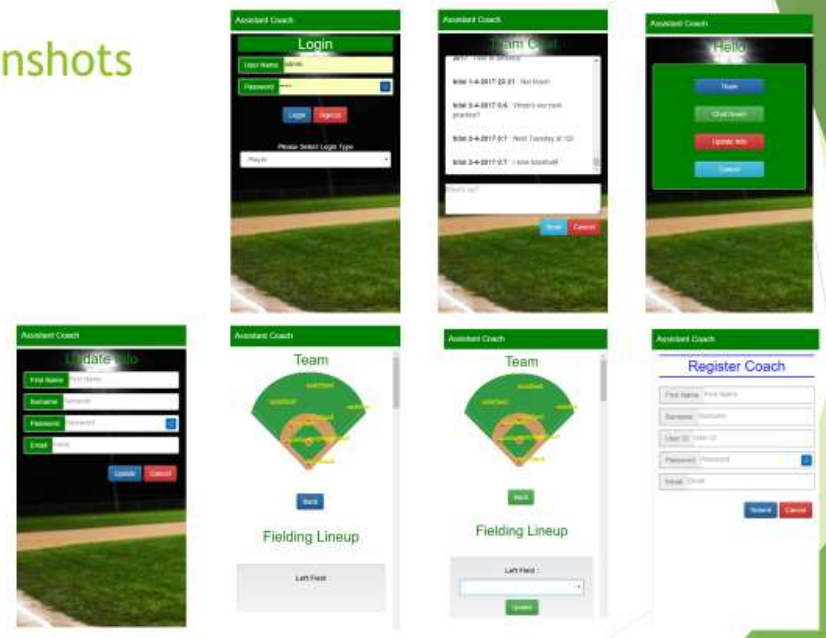
Features

- Coach can login
  - Coach can set the team's lineup, and send messages to the team
- Players can login
  - Players can view the team's lineup, and chat with one another
- Parents can login
  - Parents can view information about their child
- View
  - Coach can update positions, which can be viewed on an interactive field

Architecture



Screenshots



Live Demo

- Youtube Video: https://youtu.be/eHnbUNcKfW4
- Github URL: https://github.com/Jdfxbb/AssistantCoach