FINAL PROJECT ON

CRICKET FANTASY LEAGUE-MOBILE APPLICATION

RESEARCH PROJECT AND SEMINAR (CSCI 6838.03)

Team Members (Bleed Blue):

Sreeharsha Parankusham (parankushams8798@uhcl.edu)

Vishwas Gouru (gouruv8777@uhcl.edu)

Rakesh Gattu (gattur8756@uhcl.edu)

Prashant Kora (korap8332@uhcl.edu)

Instructor:

Dr.Pradeep Buddharaju

Mentor:

Mr. Raj Gupta

Acknowledgements

We would like to express our heartfelt gratitude to our parents without whom we would not have been privileged to achieve and fulfil our dreams.

We are extremely grateful to our project instructor Dr.Pradeep Buddharaju, Assistant Professor who directed us toward this project and helped us greatly by providing the necessary sources of information required for a deeper understanding without which we couldn't have completed IOS application successfully. His presence is invaluable.

We heartily thank our mentor Mr. Raj Gupta for his supportive guidelines and suggestions throughout the project.

The implementation could not have been possible without if we didn't have support of both of them.

We would like to extend our sincere gratitude to the people who had helped us indirectly through their websites, lectures and examples.

Contents

List of figures

1. Project description

Fantasy cricket is an online game where you can create a virtual team of real cricket players and score points depending on how your chosen players perform in the real matches. In the recent past, we have seen so many web applications that are created for playing fantasy leagues. But, a major problem with the web application is that we cannot play fantasy cricket while we are away from home, laptop or PC's. For that reason, a mobile application is considered a suitable way to connect to the users of cricket fantasy league who can access (on both iPhone and iPad) while they are on the go, and be able to view and change their fantasy team before the match begins and also check the leader board.

2. Existing applications

Firstly, there are a lot of mobile applications in the market like Draft kings for American football, IPL for Indian Premier League that are used on the go. The main problem being generalization that is, they can only be able to use for that particular game but not for the other ones. But here we are designing this application in a way that it can be generalized to any particular game. The advantage being that we need not design a different application for every different game played.

3. Research and development

I. Requirements

As soon as the project has started our research for requirements had begun. We have gone through a series of similar fantasy applications like NFL football league, IPL cricket fantasy league etc. to understand how they work. We have written the functionalities implemented in those applications. When we had meeting with our mentor and instructor, we have agreed upon some of the requirements.

II. Design phase

A week after, we have to design the kind of views that are to be present in the project. We have to draw UML diagrams to show how they work in the project. We have drawn class, use-case diagrams to know about the classes and the actors present in the project. As our project is mobile application, we have agreed not to implement functionalities of admin for this application.

III. Web services

The week after that, we are all set to work on web services. We have agreed to use web services in order to read and write data from the database. We have worked on learning about the kind of web services we have to use for this project. We came to know about XML, JSON formats to retrieve data from the database. We have agreed upon the using JSON format to retrieve data from the database.

IV. Implementation

When implementing the views we have been through a lot of applications of similar type. After going through them, we wrote down the important functionalities that are to be implemented in the application. Some of them being to display the names and scores of players in the previous games and to put a particular format to select for the game.

V. Future work

In order to continue this project and to improve it we have gone through several functionalities of an iOS device and noted them down, so that the one who continues to work on this project will have some functionalities from our side to work on.

4. Requirements gathering

I. Business / User requirements

- A login view is provided to all the players to login to their accounts and if they don't have one, they have to create one.
- Players will have access to choose their playing teams including fantasy players.
- Based on the performance of the fantasy players in the matches players will get the points.
- A leader board will be maintained to display.
- It should support IPhone/ IPad / IPod.

II. Functional / Technical requirements

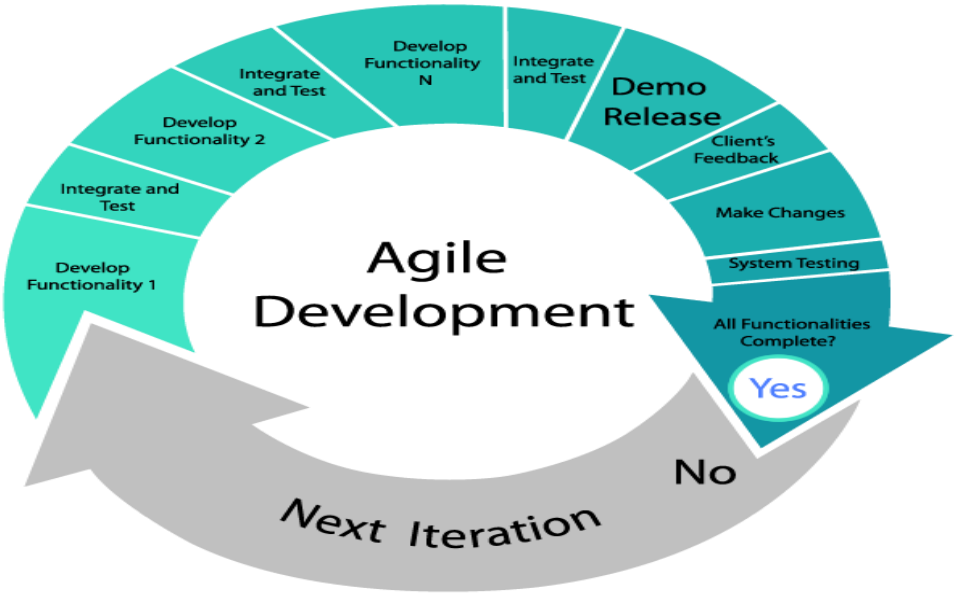
- A view with Sign-in and Register options will be shown.
- A view to register for an account.
- An initial amount is given to player to buy fantasy players.
- A list of fantasy players is kept in a pool.
- Static amount is dedicated to buy each and every fantasy player.
- Runs, wickets, strike rate, average, rank fetch points to the fantasy players as well as the players who chose them.
- Player's dashboard will be updated as soon as admins starts a league.
- Players can change their team members and teams once the league is completed.
- Player's performance is calculated based on the runs he scored, wickets he took, catches he took, run outs he did and stumps he had taken off.
- Sum of the points of all the fantasy players in the player's team are added to player's points table.
- Based on these points a leader board displays all the player's statistics and ranks in ascending order.

III. System requirements

- An IPhone/ IPod/ IPad with an IOS later version 7.0.
- 100MB of free Space.
- Internet connection is necessary.
- SQL Server 2012
- Xcode.

5. Agile software model

- It is also called as extreme process management.
- It simply refers to an iterative, incremental method of managing the design.
- It is a variant of iterative life cycle where deliverables are submitted in stages.
- Projects that develop in iterations can constantly gather feedback to help progressive prototypes.



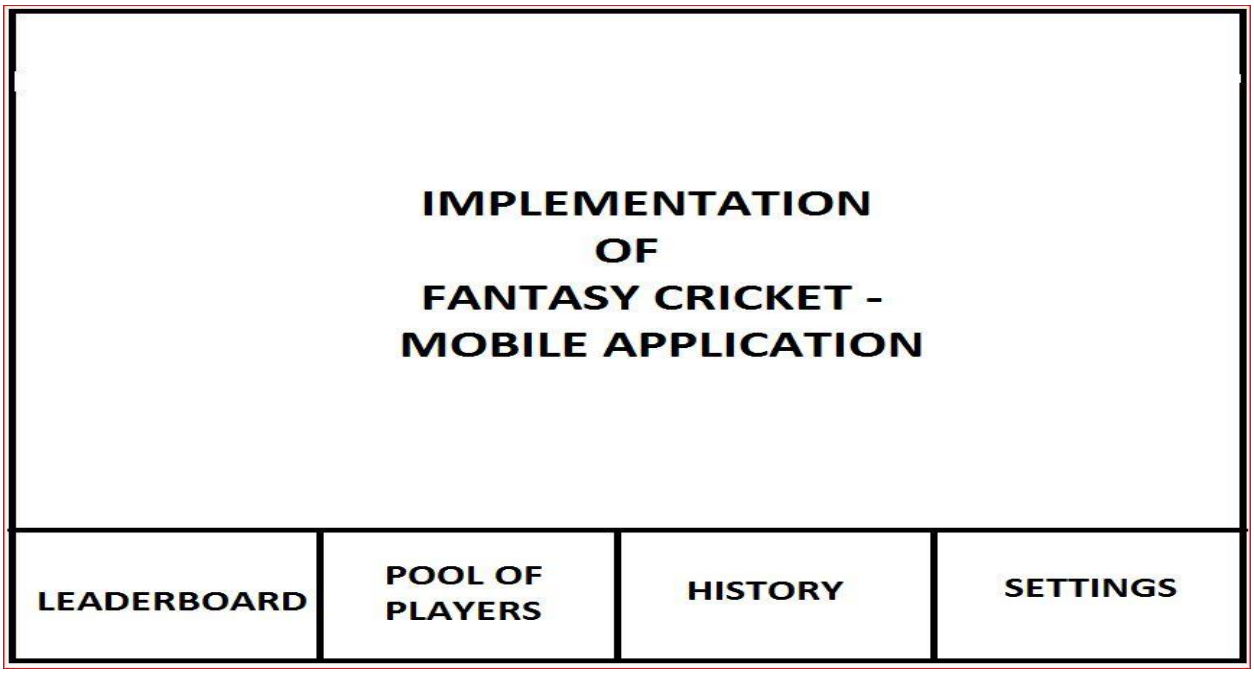
1. AGILE MODEL

6. Design

I. UI Design

The following application is a tabbed one and in order to navigate from one view to other, you have to click on a different tab.

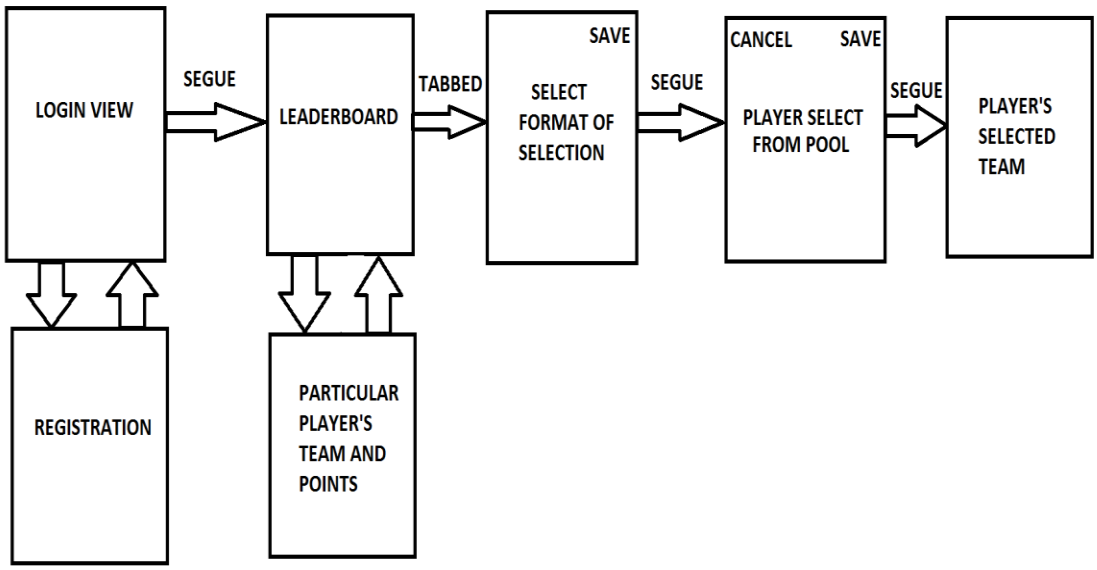
There are four tabs namely leader board, pool, history and settings. You have to click on this tab in order to navigate to that view. As simple as that!



2.MOBILE APPLICATION USING TABS TO NAVIGATE

II. Flow of the application

1. We have to login to the application if we have an account or we have to register an account.
2. After logging in, you are directed to leader board view with segue.
3. This is a tabbed application, in order to select team we have to select team format prior to the team selection.
4. You have to click the pool tab to select the players.
5. One you select the 11 players and click on save you are now directed to another view that contains your team.
6. If the player wants to know the score of the team through leader board he has to click on the player name or points.



3.DESIGN OF THE FLOW OF THE APPLICATION

7. Implementation

As our application is a mobile version of the web application, our first priority is to reflect the changes on the mobile version when a change is being made on web application. Our application doesn't have an admin module.

When a league is started in the web application, it has to be reflected on the mobile application. Not only updating the league, it has to update each and every view on the web application to the mobile application.

First of all, main functionalities of both the web and mobile applications have to be same but at the same time, the feel that they create should be different. Our application has to retrieve the data from the common database that is being created by the web application team.

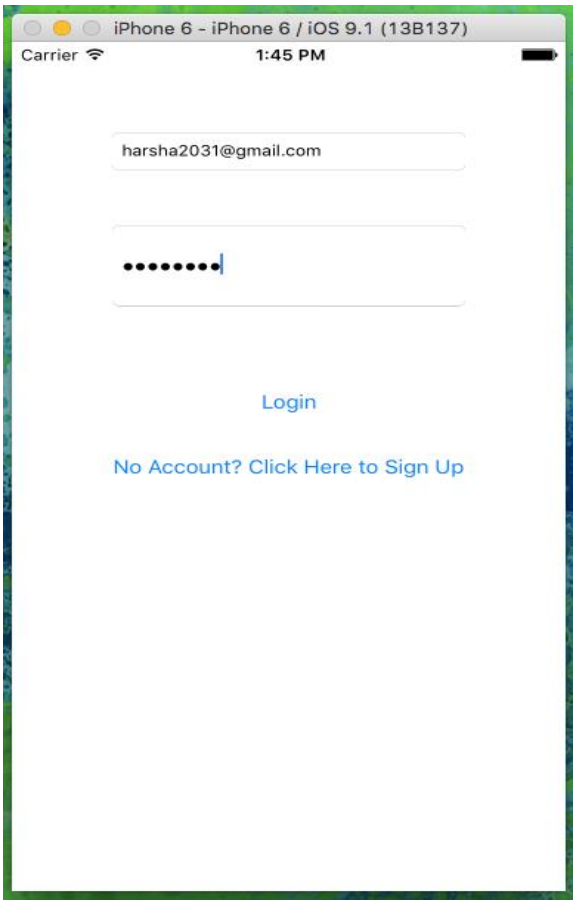
We chose "Swift" as the programming language on the Xcode platform. Programming is done by using MVC design pattern. We have used MVC design pattern because, if we want to change a view without changing the database, we can do it without any problem. By doing this, we need to make any changes to the database. Even if we want to add one more view we need to do everything from the start.

Web services has been implemented to write and read tables from database. The data is retrieved in JSON format. They are written with an extension of ".asmx" format. For version control, big bucket is used to commit a view, when its functionality is done.

A total of 9 views have been implemented as a part of the application. Each and every view represents a functionality to be implemented.

I. Login view

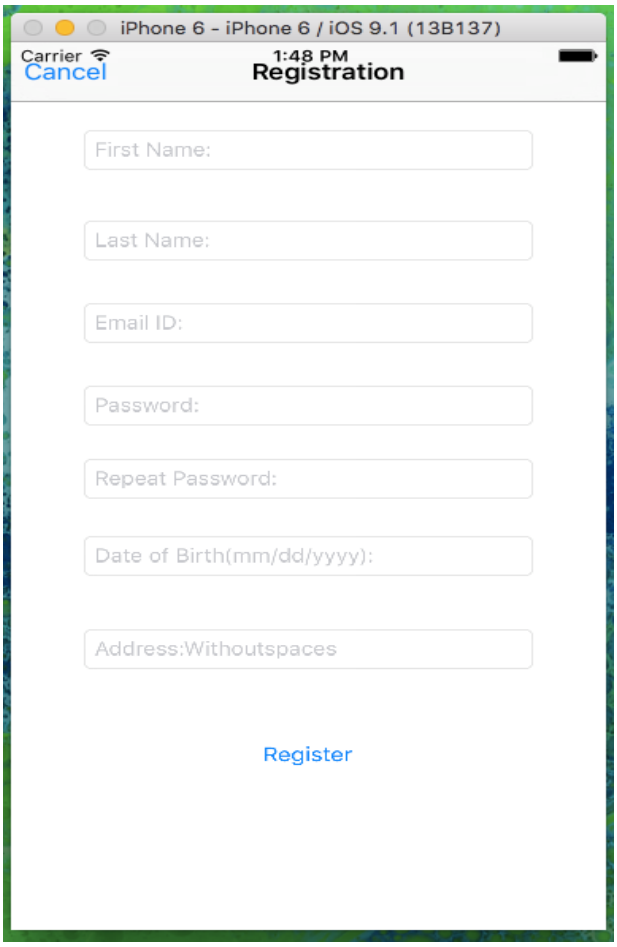
- This view asks for a login and password to login in to the app.
- When you click on submit, it will check for authentication and if you are authenticated it will navigate the player to leader board view.
- If you are not registered, you have to register by clicking on registration view.



4. LOGIN VIEW

II. Registration view

- This view asks for the details like first name, last name, g-mail address, password etc.
- When the player enters them and click on “register” button, the player is registered if there is no problem in registering.
- Then after, he is navigated to leader board view.



5.REGISTRATION VIEW

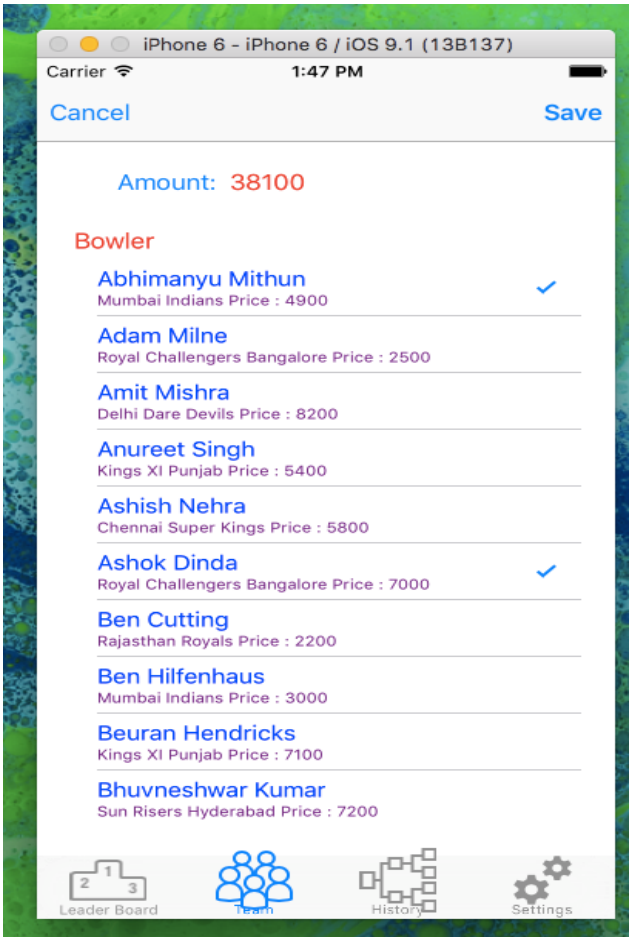
III. Leader board view

- It deals with the points of all players participating in the league.
- Names of the players are arranged in ascending order.
- A table view is used to display the player's names.
- Each cell is given to every player.
- A click on the cell navigates to a new view. This view displays his team member's names and points scored in that league.

6.LEADER BOARD VIEW

IV. Pool view

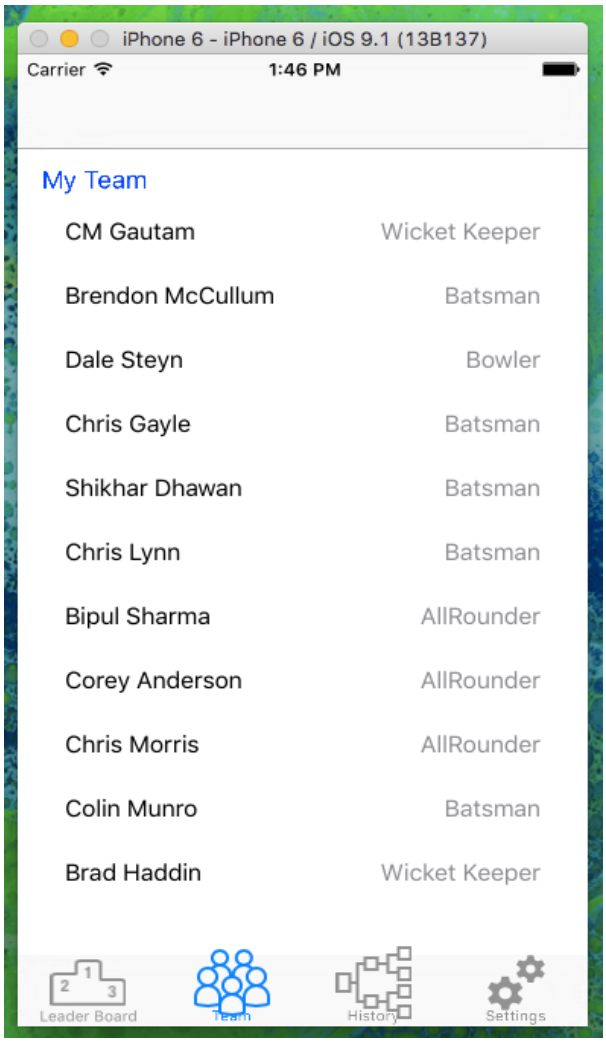
- One-third's part deals with total amount given and the amount left.
- Remaining two-third's part of the view displays all the fantasy player's names and their prices.
- Each cell is dedicated to every fantasy player.
- A click on the cell selects that particular fantasy player.
- After selecting the players, if you click on save button, it navigates to another view which shows your selected team.



7.POOL VIEW

V. My team view

- This view displays the fantasy players in the selected team.
- If the player wants to change any fantasy player from the team, he can do so by clicking on him.
- Once he click on the fantasy player he is navigated to pool view to select a player to replace the current player.



8.My team

VI. Another player team view

- This view displays the names of fantasy players in another player's team.
- We can navigate to this view, once we click on the name of the player name on the leader board.
- When you click on that name the player is navigated to this view.

9.ANOTHER PLAYER TEAM VIEW

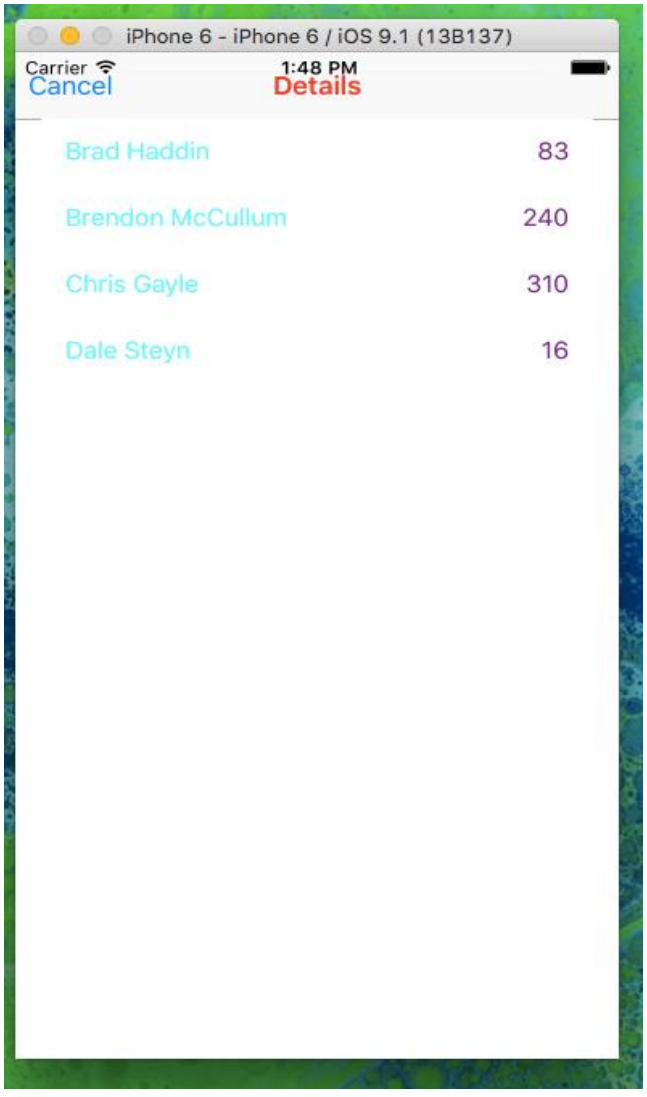
VII. History view

- This view displays the matches to be played in the league.
- We have used table view to each and every match in the league.
- When we click on each cell, we are navigated to points view.

10.MATCHES LIST VIEW

VIII. Fantasy players' points view

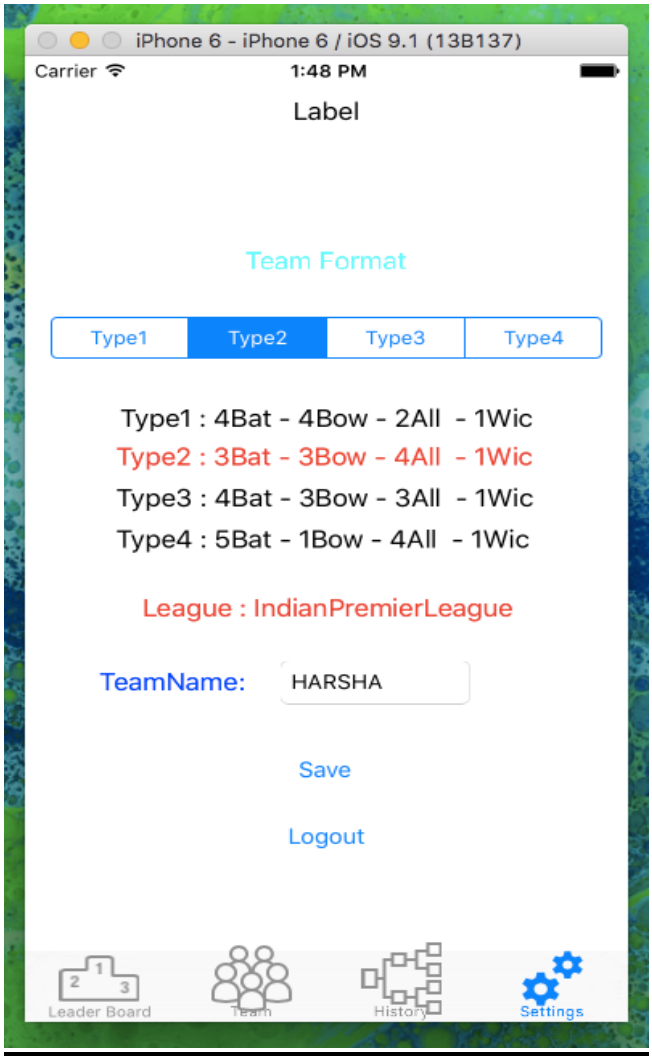
- This view displays the points of every fantasy player in players' team.
- Fantasy players have to play in that match and should be present in the player team.
- Each cell shows team names, match date and time.



11.FANTASY PLAYERS POINTS VIEW

IX. Settings view

- In the view, we ask the player to select the format of selection for the league.
- It will ask for the name of the team.
- Prior to the team selection, the player has to select the format.
- Formats to select from are:
  - 5 batsmen, 1 wicket-keeper, 3 all-rounders and 2 bowlers.
  - 4 batsmen, 1 wicket-keeper, 4 all-rounders and 2 bowlers.
  - 4 batsmen, 1 wicket-keeper, 2 all-rounders and 4 bowlers.
  - 4 batsmen, 1 wicket-keeper, 1 all-rounders and 5 bowlers.



12.SETTINGS VIEW

X. Unimplemented functionalities

There are two functionalities or views, which we have not implemented. One due to the lack of time and other due to a specific reason.

a. **Overview** :- *(decided not to implement this view)*
   - This view asks for the team name after you login in to the application.
   - After that it will navigate to another view that which contains all the views present in the application.
   - If we click one among them, we will be navigated to that particular view.

Why didn't we implement this view?

As we are doing an iOS application, it has to give you a feel that you are using an iOS device. If we implement this overview view, it will give you a feel that you are using a regular device other than iOS device.

We thought of using features of iOS devices but if we use this view it will not use the features of iOS devices, so we decided not use this view.

b. **Pass-code**:- (Due to lack of time we didn't implement it)
   - The application asks for a pass-code as soon as you open it.
   - This pass-code will not ask for login/password again and again if you open it from the same device.

Why didn't we implement this view?

We couldn't implement this view due to the lack of time. There is no other reason. We wish to have more time to do this functionality.

XI. Interesting aspects and challenges

1. While extracting the data from the database in JSON format.
    - We have used httpget (request) to retrieve in JSON format.
    - NSJSONSerialization to retrieve in JSON format, then after we have done parsing to get the data from JSON.
2. To post the data to the database.
    - We used different web methods to post the data to database.
3. To develop player selection view.
    - We have created a list array to store those fantasy player names before loading them on to the view.
    - Because of list array we are able to store them at one array and retrieve them easily.
4. While integrated views.
    - We have created a separate list array to store the selected fantasy players before loading on to the view.
5. While loading a view with different data in multiple situations.
    - We have used two different web methods to load different data on single view.

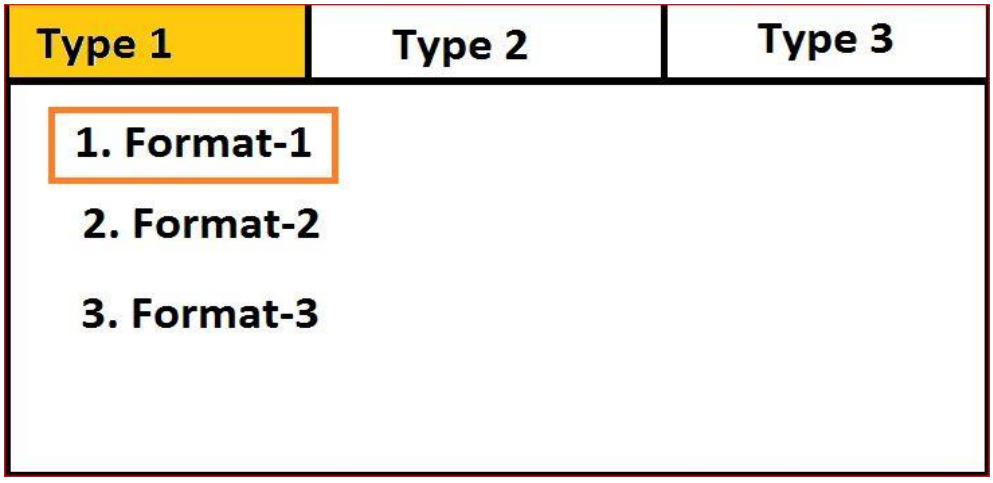
XII. Testing and verification

Testing has been done on each and every view in unit level testing. After integrating views, we have also found some more bugs in the functionality. Some of them are explained below.

Bug-1:-

Colour of the selected format has to be changed according to the functionality but it didn't.

- After clicking on a particular option on the segmentation control, colour of the format has to be changed indicating that particular format is selected.



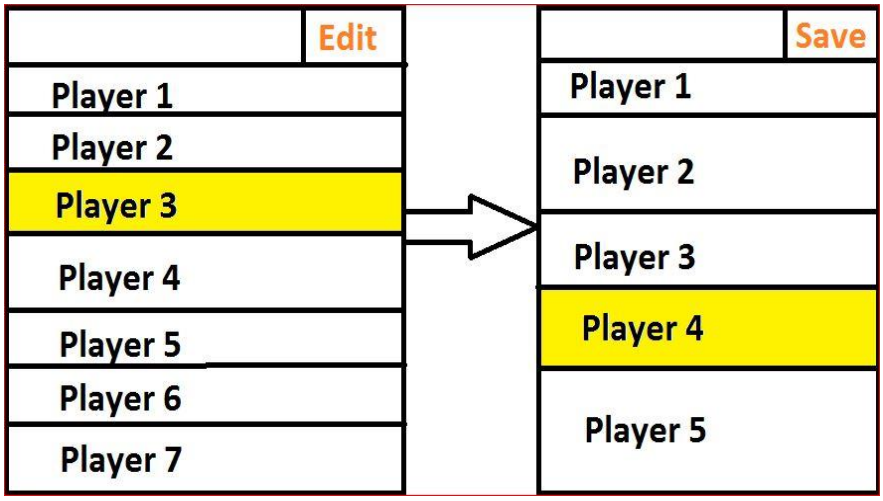
13.FORMAT SELECTION

This diagram indicates that after clicking on the type, format colour has been changed.

Bug-2:-

After integrating views, once we tried to change the team that is already been created, it is not allowing us to change only a single player.

- After clicking on the edit button, if the player clicks on a particular cell then he will be navigated to pool view to select his desired player.



14.EDITING VIEW

Here in the above diagram, player 3 is exchanged with player 4 but not the whole team.

Bug-3:-

If the player has already created a team with a certain team his name cannot be changed. We have changed it from editable to un-editable.

Reason:-

If a particular player starts the game with a team name and if he changes it after sometime, it leads to inconsistency. Because, with the new team name, it starts from zero points again which leads to mismatch of scores.

So we all have agreed to make it un-editable, so that we will not face the problem.

8. Web services

The term "Web services" describes a standardized way of integrating Web-based applications using the XML, SOAP, WSDL and UDDI open standards over an Internet protocol backbone.

Web services are more specially used to communicate between two machines in a readable formats such as XML and JSON. Web services usually provides an object-oriented web based interface to a database server, or by a mobile application to the end of the user.

In this application, we have used JSON format in order to retrieve the data from the database. In order to post the data, we have created list arrays to store and then to retrieve the same data.

9. Project management

I. Project timeline

	Software Development Plan	71 days	Wed 8/26/15	Wed 12/2/15
	Introduction	1 day	Wed 8/26/15	Wed 8/26/15
	Project selection	6 days	Wed 9/2/15	Wed 9/9/15
	Documentation	61 days	Wed 9/9/15	Wed 12/2/15
	user requirements	6 days	Wed 9/9/15	Wed 9/16/15
	Design, UI Diagrams, Plan	6 days	Wed 9/16/15	Wed 9/23/15
	Web services	31 days	Wed 9/23/15	Wed 11/4/15
	Database design	6 days	Wed 9/30/15	Wed 10/7/15
	Login, Register views	6 days	Wed 10/7/15	Wed 10/14/15
	Main view & presentation	6 days	Wed 10/14/15	Wed 10/21/15
	Pool & Individual views	6 days	Wed 10/21/15	Wed 10/28/15
	Leaderboard view	6 days	Wed 10/28/15	Wed 11/4/15
	Integration between views	26 days	Wed 10/7/15	Wed 11/11/15
	Testing &validation	6 days	Wed 11/11/15	Wed 11/18/15
	Deployment	6 days	Wed 11/18/15	Wed 11/25/15
	Final project presentation	6 days	Wed 11/25/15	Wed 12/2/15

II. Division of work

Prashant Kora:-

- Gather requirements.
- Communicate with mentor and instructor.
- Design and develop views for
    - Player pool view.
    - Points view.

Sreeharsha Parankusham:-

- Database administration.
- Design and develop views for
    - Log-in view.
    - Registration view.
- Application data read web-services.

Rakesh Gattu:-

- Update contents and maintain the website.
- Design and develop views for
    - History view.
    - Team view.
- Version Control.

Vishwas Gouru:-

- Update all the contents in to the document
- Design and develop views for
    - Settings view.
    - Leader board view.
- Application data write web-services.

10. Concepts learnt

How to deal with iOS interface.

- How to use web services for mobile application.
- How to use the database which is designed for web application in mobile application.
- How to retrieve data from the database.
- How to replace the data after editing it.

11. Conclusion and future work

- Fantasy cricket might not allow you to play in the league but it will give you a feel that your selected team is playing it.
- Your perception regarding team selection among other players can be ranked accordingly.
- We can notify the players about the league before it starts.
- We can ask for a passcode without asking for login/password if the application is opened on the same device repeatedly.
- We can also add a functionality like automatic selection of the team.

12. References

1) “An IOS Application”. Apple. Inc. (2012, Aug 2) [online]. Available: http://www.ieee.org/documents/ieeecitationref.pdf.
2) “Udemy website”. (2015, Oct 15th) [online]. Available: https://www.udemy.com/apple-mobile-app-development-with-swift-xcode-and-ios/.
3) “Stanford - developing IOS8”. (2015, Feb 8th) [online]. Available : https://www.youtube.com/watch?v=GOEPVM5OzJk&list=PLy7oRd3ashWodnpf8rjfYEkTgwbOEsKfU
4) “The swift programming language”. (2015, Sep 10th) [online]. Available : https://developer.apple.com/swift/

13. Appendices

I. System manual

1. If you just want to use the application, then you can do it on any iOS device.
2. If you want to run this application, then you can only do it on an iOS device.
3. A tool called “Xcode” needs to be installed on that iOS device.
4. To run this application just click on the “play” button on the top left.
5. It will compile itself once you click on play button.
6. If there are any errors, it will report the same.
7. Code can be seen when you click on the view which are on the left.

II. User manual

Steps to use the application are as follows:

1. As soon as the app loads, it will asks the user to enter the login and password details.
2. If the user is not yet registered, he can do so by clicking on the “Register now” button.
3. Then after you are directed to leader board view.
4. “Leader board” displays the details of all the players participating in the league.
5. When you click on any of the player name it will navigate you to “My Team” view.
6. In “My Team” view, you can see the fantasy player’s names of his team.
7. When you click on cancel, application is redirected to leader board.
8. Now, if you click on the players points, the application is navigated to “Points” view.
9. In this view, the points scored by the fantasy players is shown.
10. If you click on cancel button, application is redirected to leader board.
11. If the player wishes to create a team of his own in the league he has to click on the “Settings” tab at the bottom.
12. This view asks for the player’s team name along with the format of selection. After selecting options click on save.
13. Now the player has to click on the “pool” tab to select the fantasy players.

14. After selecting 11 players obeying the format he selected in settings tab, he has to click on save button.
15. Once the player clicks on save button, the application is redirected to “my team” view.
16. This view shows the names of his selected players.
17. After entering the team name and clicking on save button, the player can see his name on the leader board.
18. After a particular game is updated, the scores of the players is also updated according to the scores of their fantasy players.

III. Detail design

UML diagrams

Use case diagram

With the help of this diagram, we can understand that there are only two actors one of them being the player (user) and the other being a fantasy player.

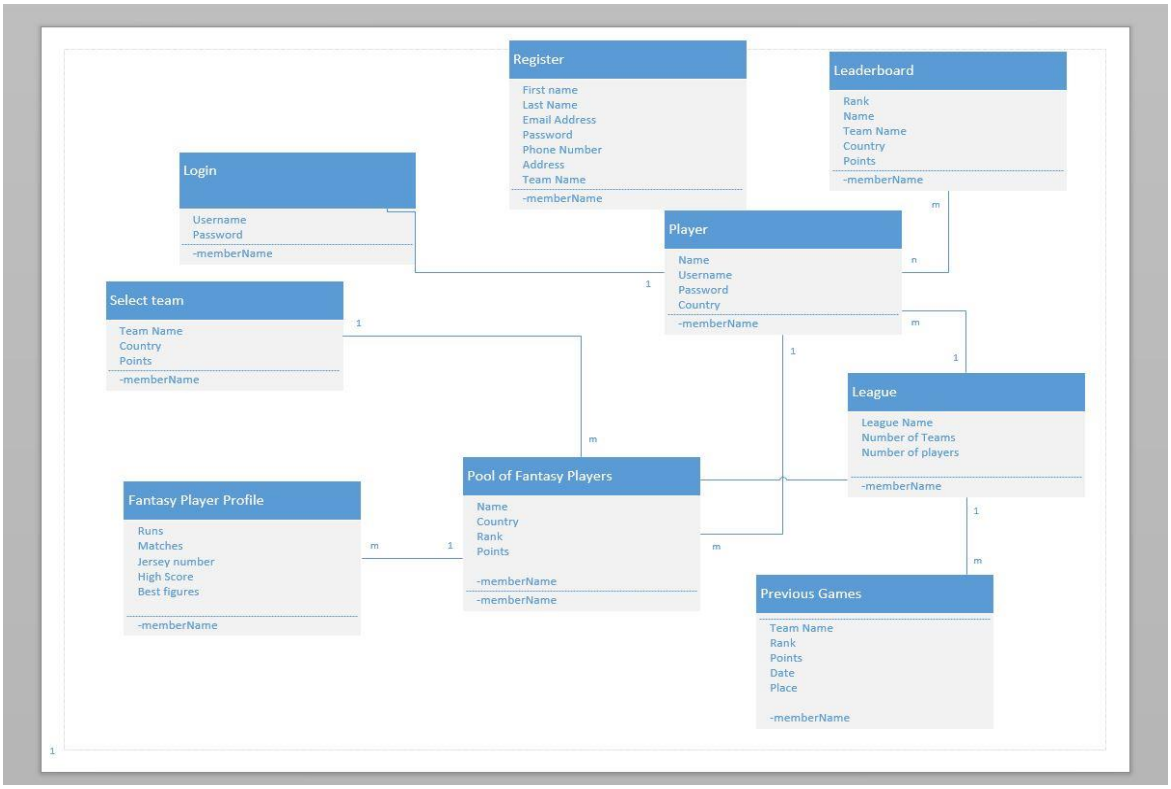
This diagram depicts that a player (user) can access every functionality in the application where as a fantasy player can only have access to pool, previous games and points views. We can clearly understand that the fantasy player is limited to certain functionalities.

15.USE CASE DIAGRAM

Class diagram

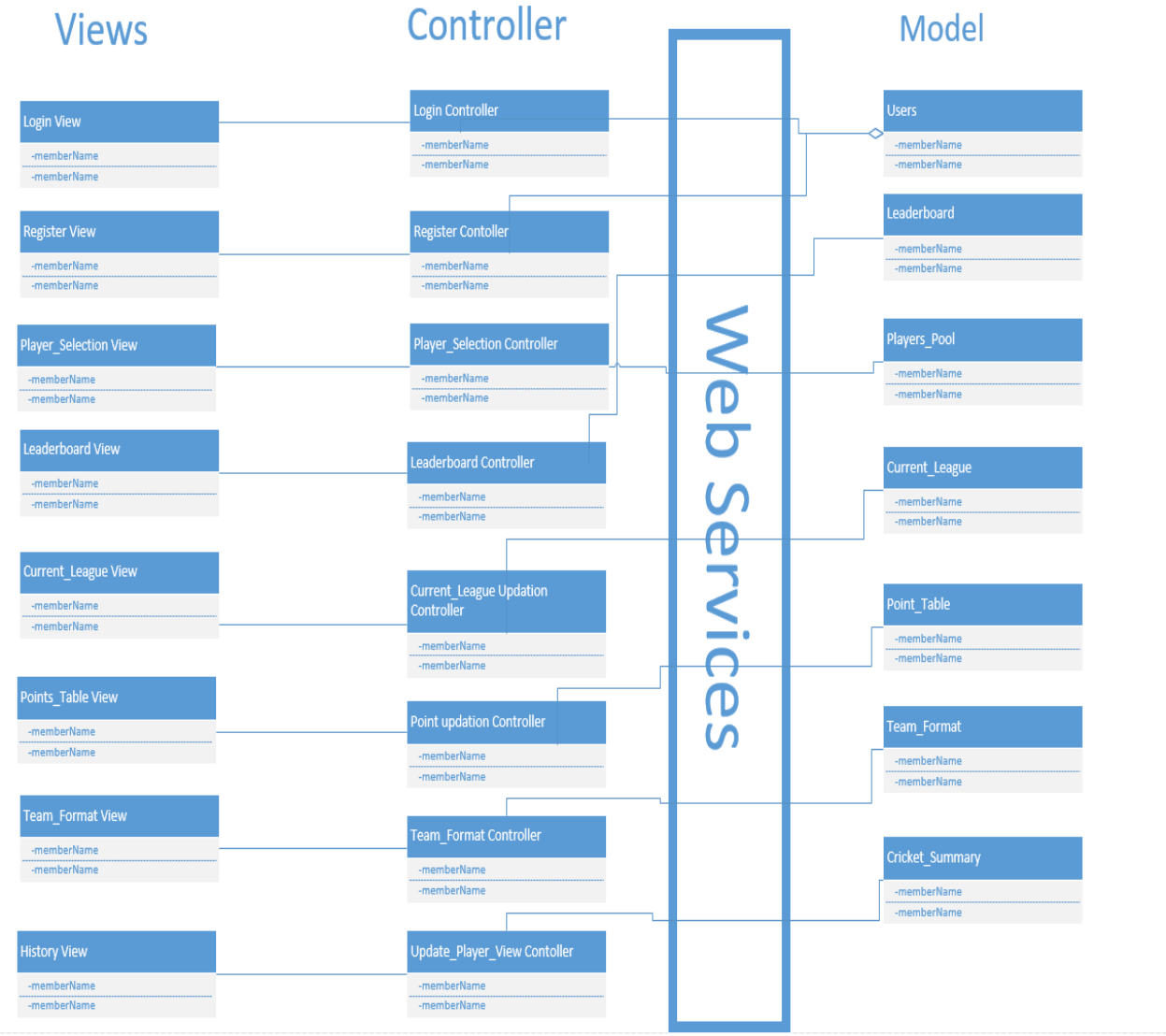
The following two class diagrams represents the classes that are present in this application**.**

Among those, one of them is drawn using MVC while the other without using it. We have got 9 classes while drawn without using the MVC. After using MVC, we have got 25 classes which come under modal, view and controller classes. There are several attributes in each class.



16.**CLASS DIAGRAM**

Class diagram using MVC design pattern



17.CLASS DIAGRAM WITH MVC DESIGN PATTERN