

**NANYANG
TECHNOLOGICAL
UNIVERSITY**
SINGAPORE

**School of Computer Science & Engineering
CZ2006 Software Engineering**

Project Title: SupaSports

Tutorial Group: DSS2

Team Members:

Kee Kai Teng (U2020909E)
Zhong Shaojie (U2020758K)
Irwin Ding (U2022637K)
Imax Ooi (U2021094G)
Tan Wei Kit (U2020104H)
Darryl Chia Shen Yang (U2022120B)

1 Product Description	3
1.1 Purpose	3
1.2 Scope	3
1.3 Users and Stakeholders	3
1.4 Assumptions and Constraints	3
1.5 Constraints	3
1.6 UI Mockups	4
2. Functional Requirements	12
2.1 Use Case Models	17
2.2 Use Case Descriptions	20
2.3 Class Diagram	51
2.4 Sequence Diagram	52
2.5 Dialog Map	57
3 Non-Functional Requirements	58
4 Interface Requirement	59
4.1 User	59
4.2 Hardware	59
4.3 Software	59
4.4 Communication	59
5 Architecture Design	60
5.1 System Architecture Diagram	60
5.2 Design Pattern	61
6 Data Dictionary	63
7 Testing	64
7.1 Black box testing	64
7.2 White box testing	68
8 Appendix	71

1 Product Description

1.1 Purpose

The purpose of this mobile app is to provide sport enthusiasts with a platform to view news, results and upcoming fixtures of the sports they are interested in with the added benefit of being able to interact with one another through a forum. This app is both an information outlet app and a social media app.

1.2 Scope

Our app enables users to search for their favourite teams from various sports, favourite them and be updated with their news and upcoming fixtures. It also allows users to view matches based on the calendar if they want general updates. It provides users with the most updated news, segmented via different sports, and allows users to interact with one another through a forum function that can also be segmented via different sports.

1.3 Users and Stakeholders

We aim to target the general public, especially sports enthusiasts who follow closely with certain sports or support certain teams. Since the interface of the app is easy to navigate around, the general public could also use the app to follow news whenever they feel like it.

The stakeholders of the app are

1.4 Assumptions and Constraints

Users should have Internet access in order to use the application.

1.5 Constraints

Our SupaSports app currently supports Android OS.

As the sports API does not have comprehensive statistics for every sport it results in the application being limited to certain sports.

This application is only available in English.

1.6 UI Mockups

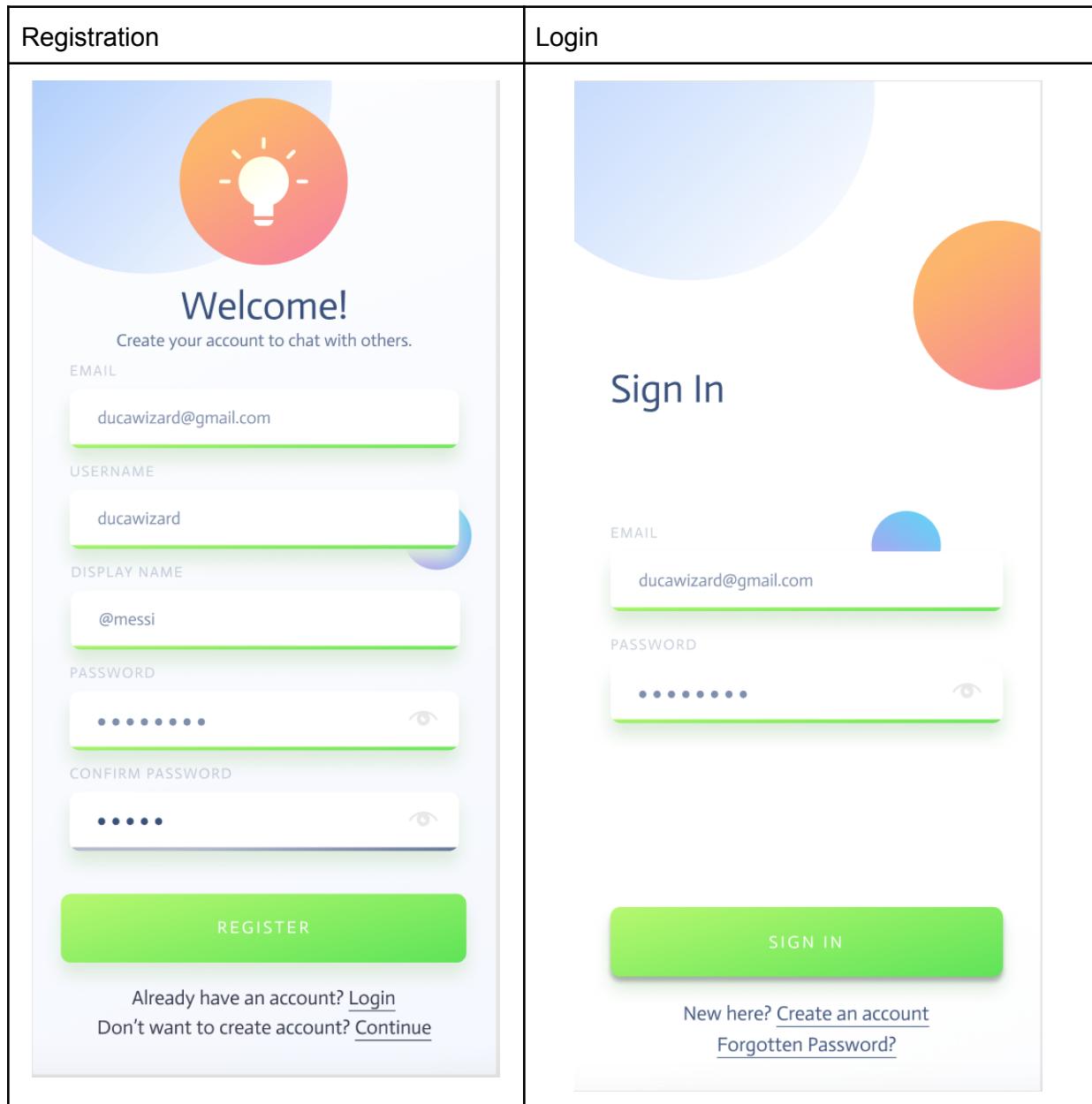


Figure 1: Registration & Login Screens

Figure 1 shows the user interface for login and registration. The user interface is designed similar to the registration and login screens used for other applications. This will enable the user to register for a new account and login to the application with ease as they will see familiar layouts and components.

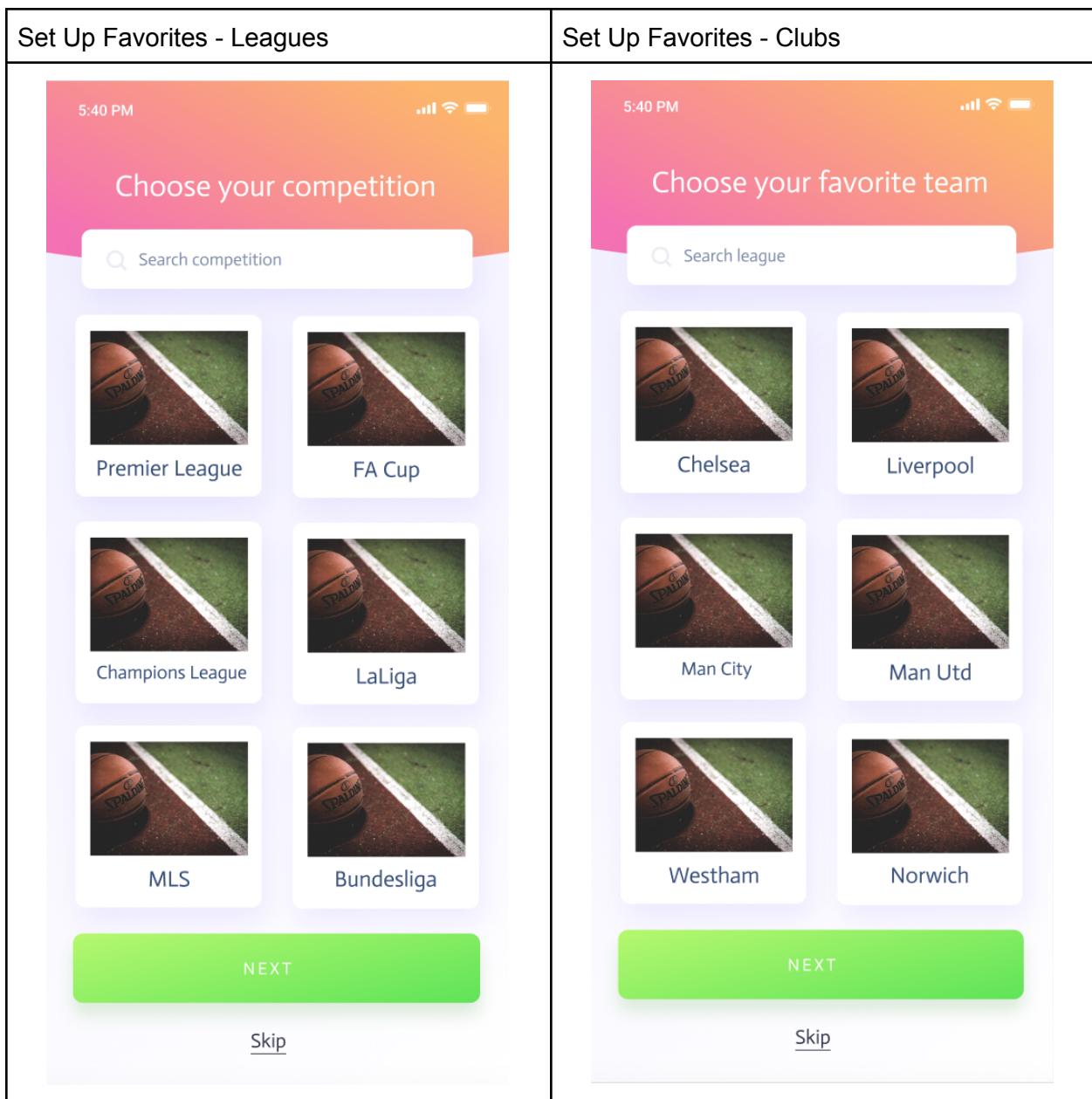


Figure 2: Set Up Favorites Screens

Figure 2 shows the user interface after the user registers for an account. The user will be asked to set up their favourites by selecting their favourite leagues and clubs. The user is also given the option to skip if they do not wish to add any.

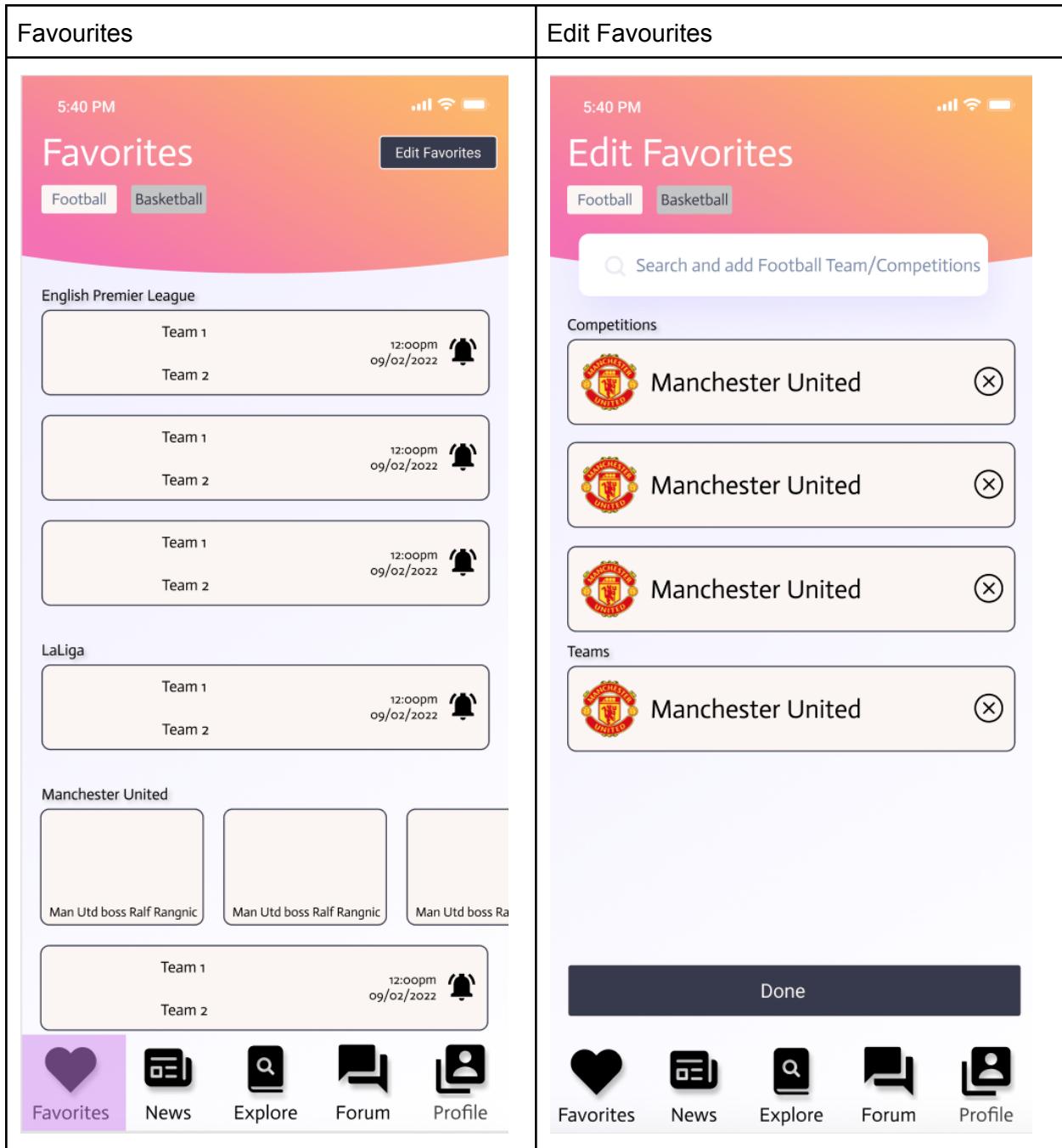


Figure 3: Favorites and Edit Favorites Screens

Figure 3 shows the user interface for the favorites page. The user is given the option to edit the list where they can add or remove certain things.

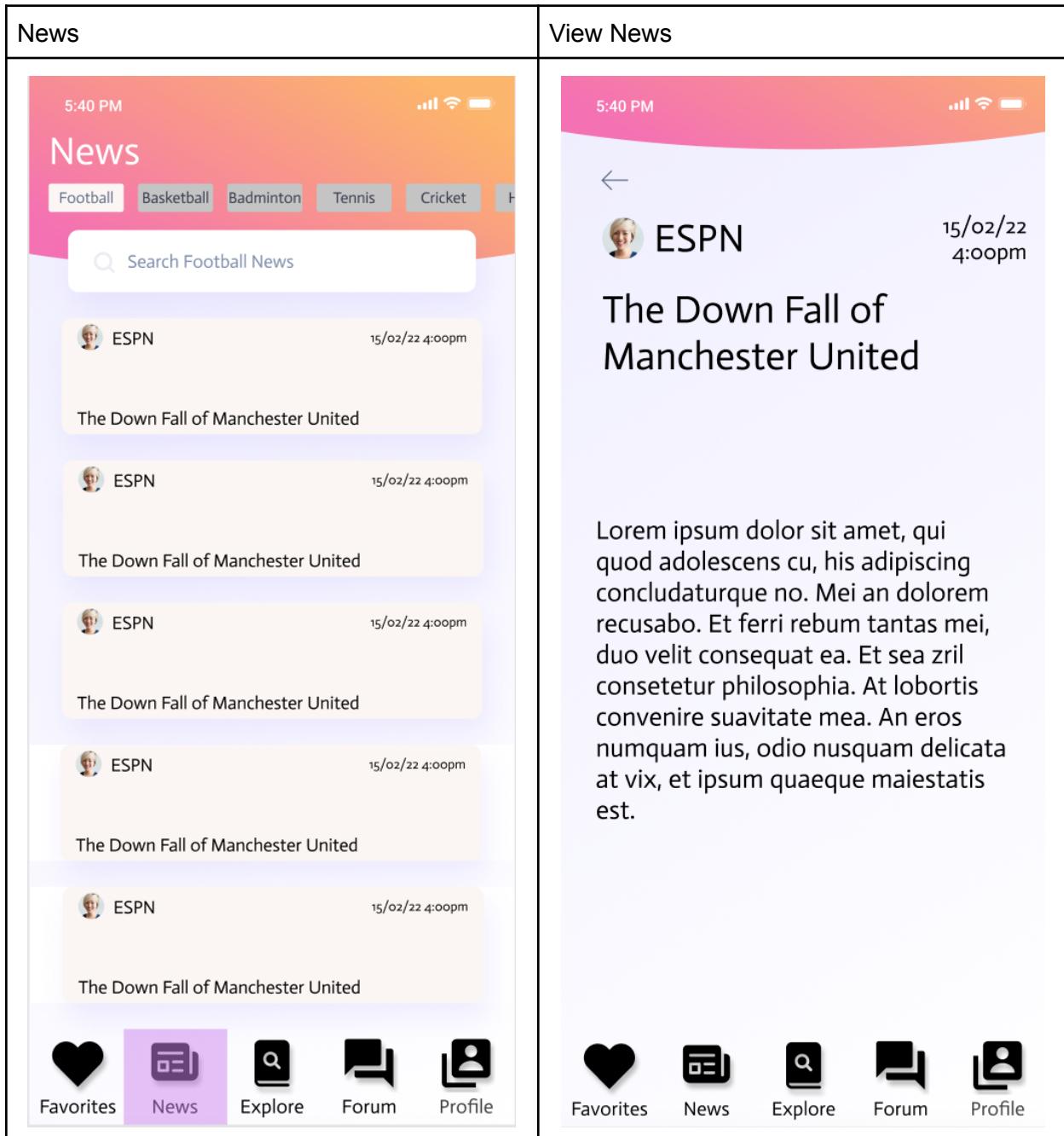


Figure 4: News Screens

Figure 4 shows the user interface for news. The user may click on a news banner with the headline that interests them and will be brought to the “View News” screen where they can read the news article in detail.

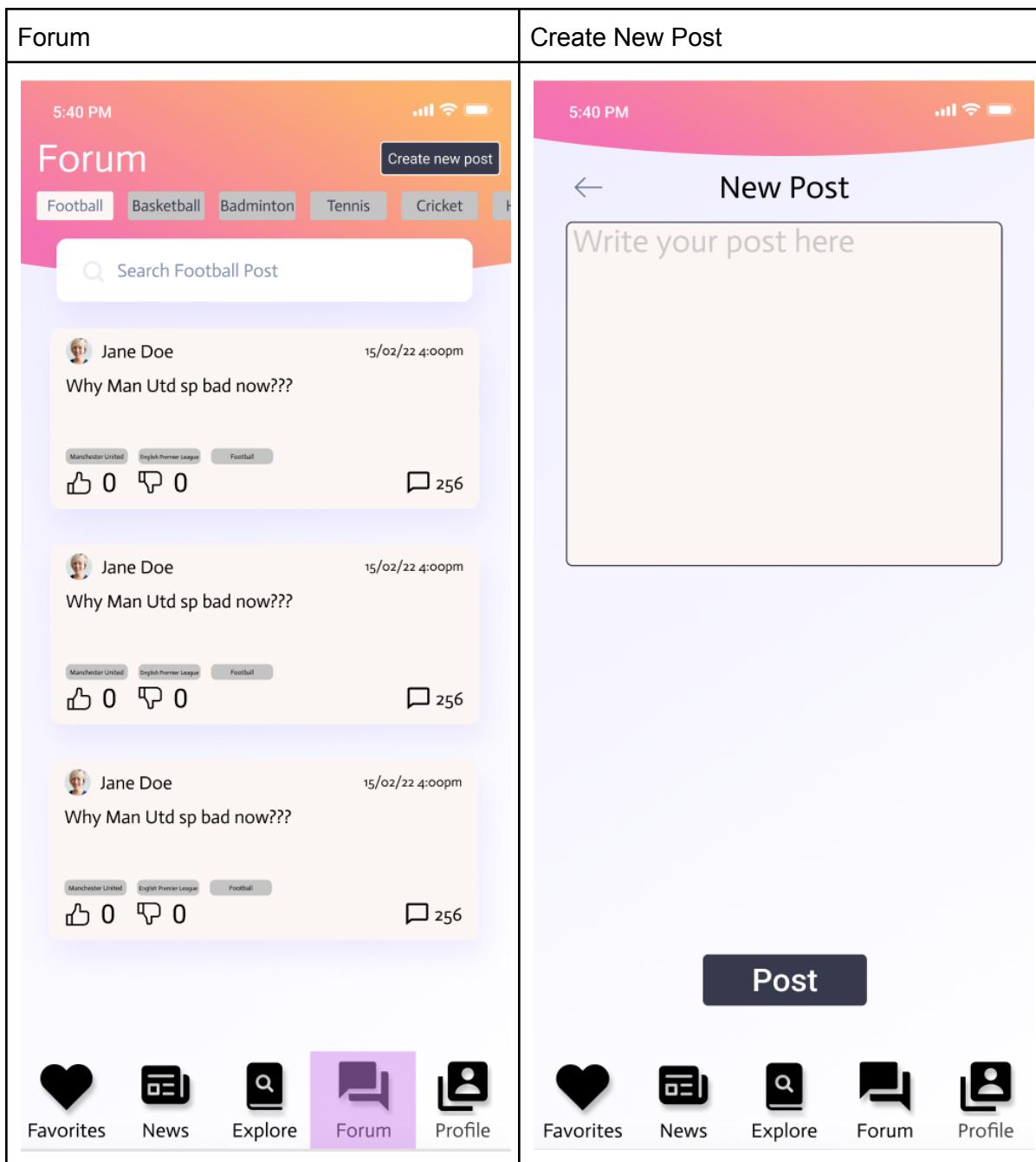


Figure 5: Forum and Create New Post Screen

Figure 5 shows the user interface for the forum page. The user will be able to view the latest post on this page. The user can tap into the post they are interested in to view the comments by other users . The user is given the option to create their own post on the top right corner. The user may type the contents of the post in the text box provided. They are also given the option to return to the forum page by tapping on the back button.

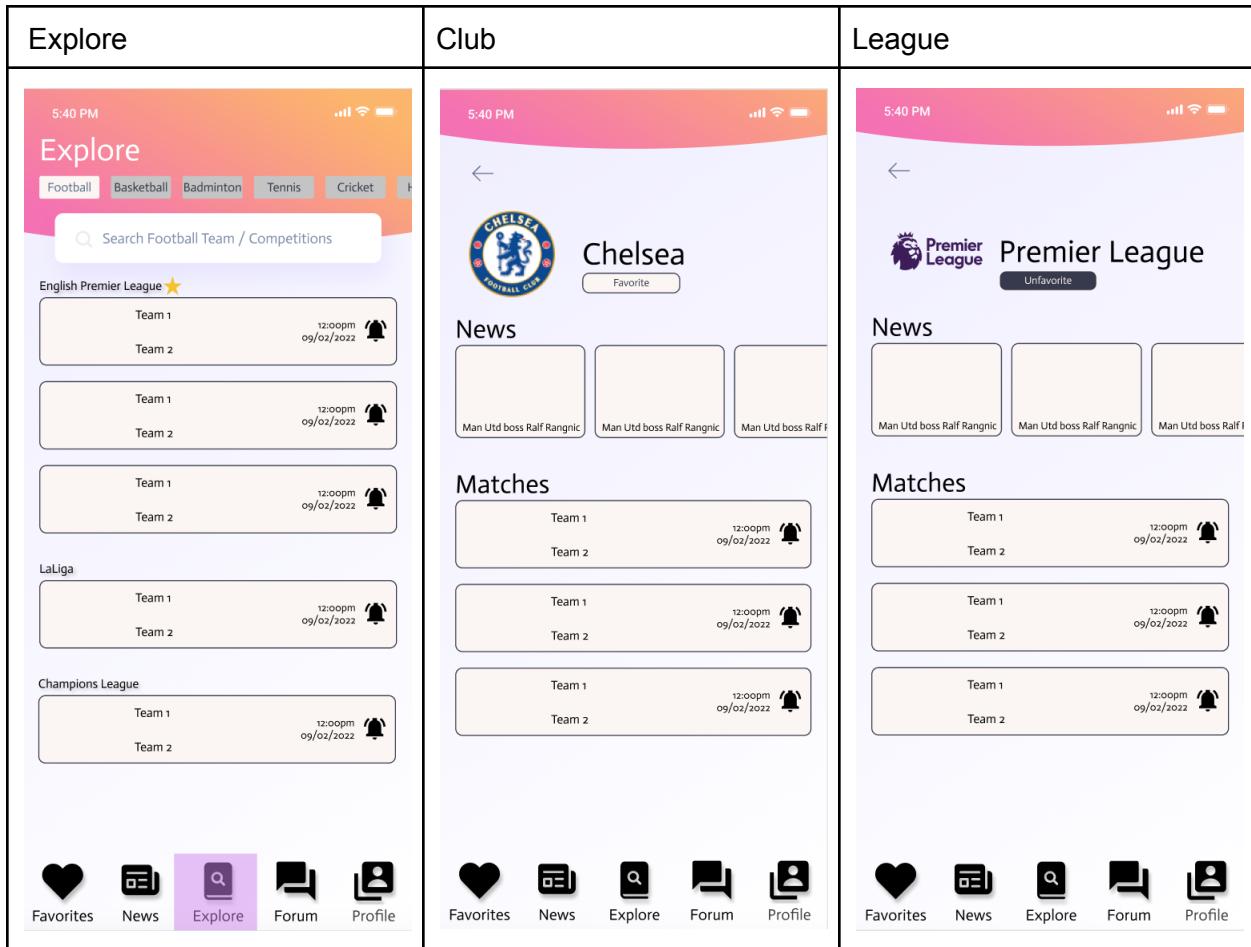


Figure 6: Explore, Team and Competition Screens

Figure 6 shows the user interface for the explore, club and league page. Under the explore page, the user will see the game schedules of their favourite clubs and league based on sports. Users can search for a particular team or competition using the search bar. The user may tap into the club or league to view more information as shown in “Club” and “League” screens.

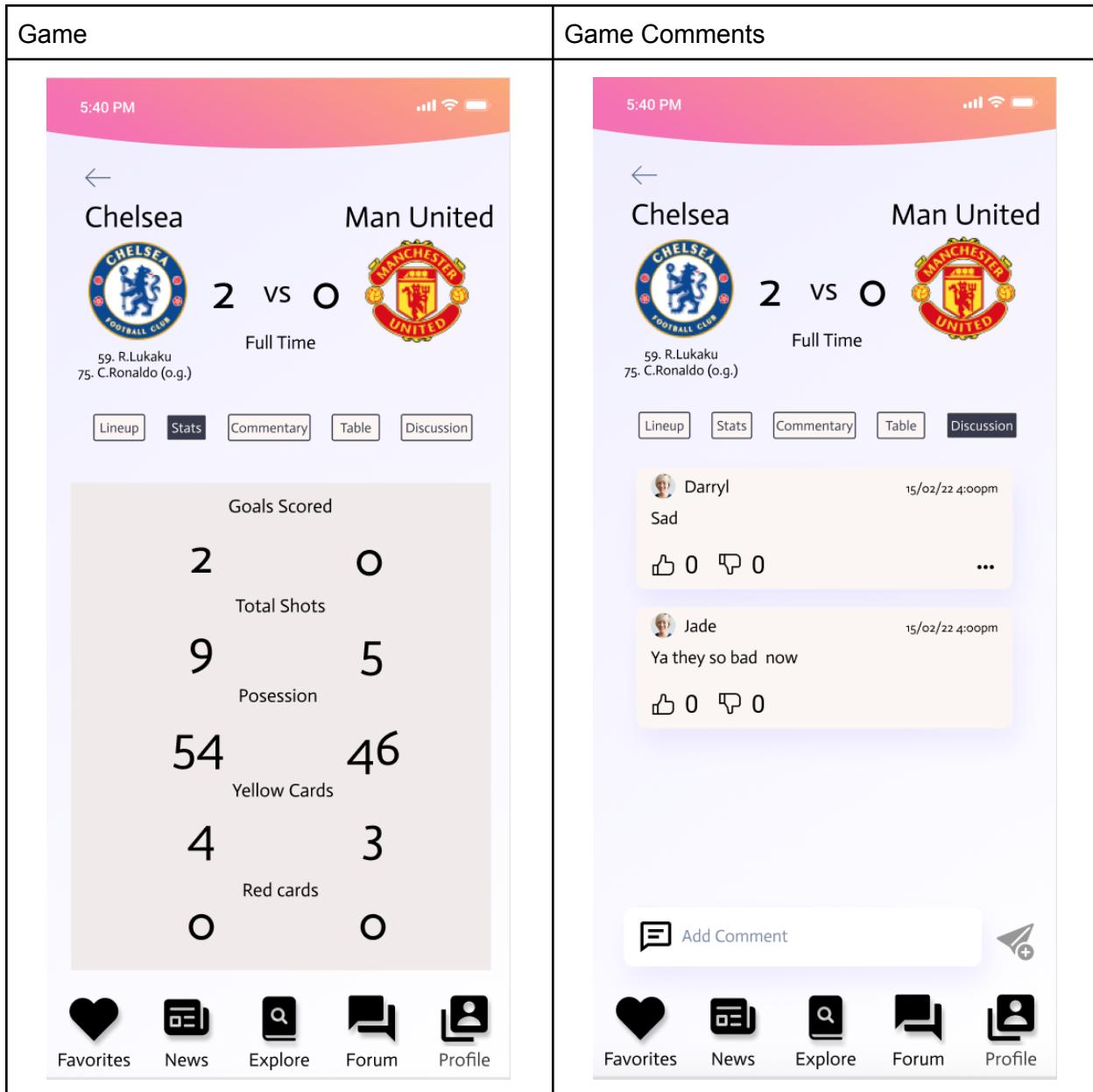


Figure 7: Game and Game Discussion Screens

Figure 7 shows the user interface for a particular game and the discussion section. Once the user clicks into the game, they will see the “Game” screen and if they wish to read the comments under it, they may click the “Discussion” tab.

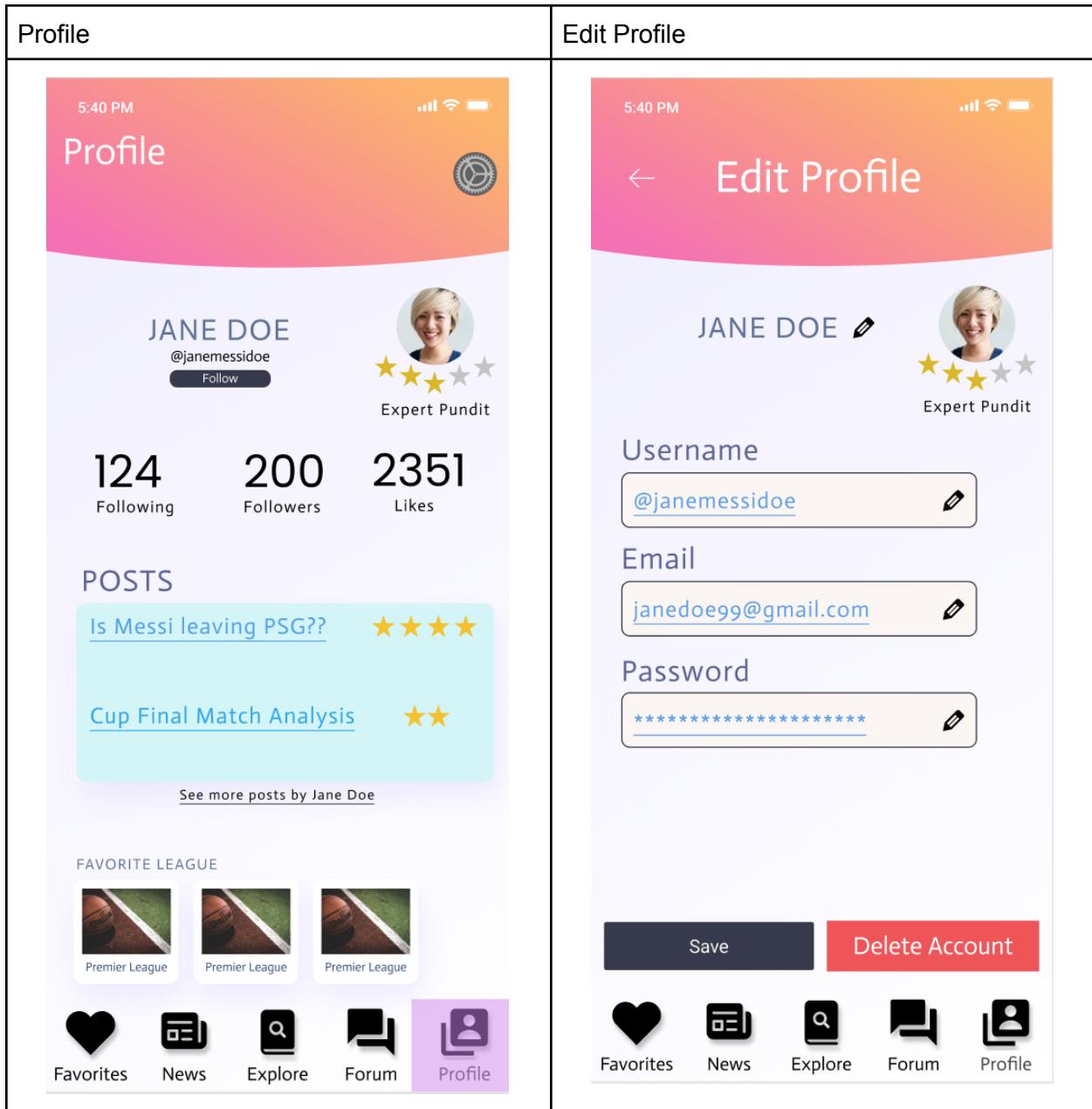


Figure 8: Profile & Edit Profile Screens

Figure 8 shows the user interface for profile and edit profile screens. The user may view or edit their own profile. The user can edit their profile by clicking on the icon on the top right corner.

2. Functional Requirements

1. Register for a New Account

- 1.1. The **user** must be able to **register** for a new account by entering the following details:
 - 1.1.1. Username
 - 1.1.2. Password & Confirm Password
 - 1.1.3. Email
 - 1.1.4. Display Name
- 1.2. The **system** must **validate** the password and the confirm password are the same.
- 1.3. The system must validate that the password is at least 6 characters.
- 1.4. The system must validate that the email is of a correct format and is not currently linked to another account.
- 1.5. The **system** must **validate** that the username is currently not taken by another account.

2. Login to an Existing Account

- 2.1. The **user** must be able to **login**, by entering their email and password.
- 2.2. The **system** must **validate** the input email and password.
- 2.3. If the login information is incorrect, the **system** will **display** an error message.
- 2.4. The **system** will **redirect** the user to the Favourites page after the user successfully logs in.

3. Updating User Details

- 3.1. The **user** must be able to **update** his user details which includes:
 - 3.1.1. Email
 - 3.1.2. Password
 - 3.1.3. Display Name
 - 3.1.4. Username
- 3.2. The **system** must **validate** that the new email is not currently linked to another account.
- 3.3. The **system** must **validate** that the password is at least 6 characters.
- 3.4. The **system** must **validate** that the username is currently not taken by another account.

4. Delete an Existing Account

- 4.1. The **user** must be able to **delete** his/her account.
- 4.2. The **system** must **remove** the user's email, display name and username.

5. Set up Favourites

- 5.1. Upon registration, the **user** must be able to **select** their favourite league and club.

5.2. The **system** must be able to **store** the user's favourite league and club.

6. View List of Favourites (On the favourite page)

6.1. The **system** must be able to **display** the upcoming games of the user's favourite leagues and/or clubs.

7. Edit User's List of Favourites

- 7.1. The **user** must be able to **add** and **remove** a favourite league/club from the existing List of Favourites.
- 7.2. The **user** must be able to **search** for their favourite team/competition.
- 7.3. The **user** must be able to **add** and **remove** a league/club from their List of Favourites through the search results.
- 7.4. The **system** must be able to **display** the updated List of Favourites after the user has made changes to their List of Favourites.

8. View Details of Game

8.1. The **user** must be able to **view** information on a game.

8.1.1. Information includes:

- 8.1.1.1. Game Statistics
- 8.1.1.2. General Information
- 8.1.1.3. Game Events
- 8.1.1.4. Game Comments

8.2. The **system** must be able to **display** information of the game retrieved from the API accurately.

9. View List of Popular Leagues and Clubs

9.1. The **user** must be able to **view** the games in a popular league.

9.2. The **user** must be able to **view** the games in a popular club.

9.3. The **system** must be able to **display** the upcoming games of popular leagues/clubs.

10. Search and View List of Leagues/Clubs

10.1. The **user** must be able to **search** for a specific league or club.

10.2. The **system** must be able to **display** a list of leagues/clubs based on the user's search.

11. View Details of League

11.1. The **user** must be able to **view** information on a league.

11.1.1. Information includes:

- 11.1.1.1. All past games
- 11.1.1.2. All future games
- 11.1.1.3. Current league standings

- 11.2. The **user** must be able to **add** and **remove** a league from their List of Favourites while viewing details of the league.
- 11.3. The **system** must be able to **display** the change in state of the ‘favourites’ button accordingly.
- 11.4. The **system** must be able to **update** the user’s List of Favourites accordingly.

12. View Details of Club

- 12.1. The **user** must be able to **view** information on a club.
 - 12.1.1. Information includes:
 - 12.1.1.1. All past games
 - 12.1.1.2. All future games
- 12.2. The **user** must be able to **add** and **remove** a club from their List of Favourites while viewing details of the league.
- 12.3. The **system** must be able to **display** the change in state of the ‘favourites’ button accordingly.
- 12.4. The **system** must be able to **update** the user’s List of Favourites accordingly.

13. View List of News

- 13.1. The **user** must be able to **view** a list of news headlines on a sport.
- 13.2. The **system** must be able to **display** the list of news retrieved from the API accurately.

14. Viewing a particular Sports News

- 14.1. The **user** must be able to **view** the news article from the list of news
- 14.2. The **system** must be able to **display** contents of the news article retrieved from the API accurately.

15. View List of Posts

- 15.1. The **user** must be able to **view** all the posts on the forum.
- 15.2. The **system** must be able to **display** the post in a reverse chronological order.

16. Create a Post

- 16.1. The **user** must be able to **create** a post under the forum.
- 16.2. The **system** must be able to **validate** that the post is not empty.
- 16.3. The **system** must be able to **display** the new post under the forum to all users.

17. View Post and Comments on the Post

- 17.1. The **user** must be able to **view** all comments under a post upon clicking on the post.
- 17.2. The **system** must be able to **display** all the comments under a post.

18. Edit a Post

- 18.1. The **user** must be able to **edit** their own post.
- 18.2. The **system** must be able to **validate** that the edited post is not empty
- 18.3. The **system** must be able to **display** the edited post to all users.

19. Remove a Post

- 19.1. The **user** must be able to **delete** their own post.
- 19.2. The **system** must be able to **display** a notification to confirm with the user that they are removing the post.
- 19.3. The **system** must be able to **display** the updated forum to all users.

20. Like a Post

- 20.1. The **user** must be able to **like** a post.
- 20.2. The **system** must be able to **display** that the post has been liked by the user and updates the total number of likes and dislikes accordingly.

21. Dislike a Post

- 21.1. The **user** must be able to **dislike** a post
- 21.2. The **system** must be able to **display** that the post has been disliked by the user and updates the total number of likes and dislikes accordingly.

22. Create a Comment

- 22.1. The **user** must be able to **comment** on a post or a game.
- 22.2. The **system** must be able to **validate** that the comment is not empty.
- 22.3. The **system** must be able to **display** the new comment under the respective post or game to all users.

23. Edit a Comment

- 23.1. The **user** must be able to **edit** their own comments.
- 23.2. The **system** must be able to **validate** that the edited comment is not empty.
- 23.3. The **system** must be able to **display** the edited comment to all users.

24. Remove a Comment

- 24.1. The **user** must be able to **delete** their own comments.
- 24.2. The **system** must be able to **display** a notification to confirm with the users that they are removing the comment.
- 24.3. The **system** must be able to **display** the updated comment thread to all users.

25. Like a Comment

- 25.1. The **user** must be able to **like** a comment.

25.2. The **system** must be able to **display** that the comment has been liked by the user and updates the total number of likes accordingly.

26. Dislike a Comment

26.1. The **user** must be able to **unlike** the comment.
26.2. The **system** must be able to **display** that the comment has been disliked by the user and updates the total number of likes and dislikes accordingly.

27. View a User Profile

27.1. The **user** must be able to **view** other users' profiles.
27.2. The **user** must be able to **view** their own profile.
27.3. The **system** must be able to **display** information about the user's profile.

27.3.1. Information includes:

- 27.3.1.1. Number of Followers
- 27.3.1.2. Number of Following
- 27.3.1.3. Display Name and Username of all users following them
- 27.3.1.4. Display Name and Username of all users they are following
- 27.3.1.5. Display Name
- 27.3.1.6. Username

28. Follow a User

28.1. The **user** must be able to **follow** another user while viewing their profile.
28.2. The **system** must be able to **display** that User A is now following User B and update the "Following" and "Followers" count on their User's profile correctly.

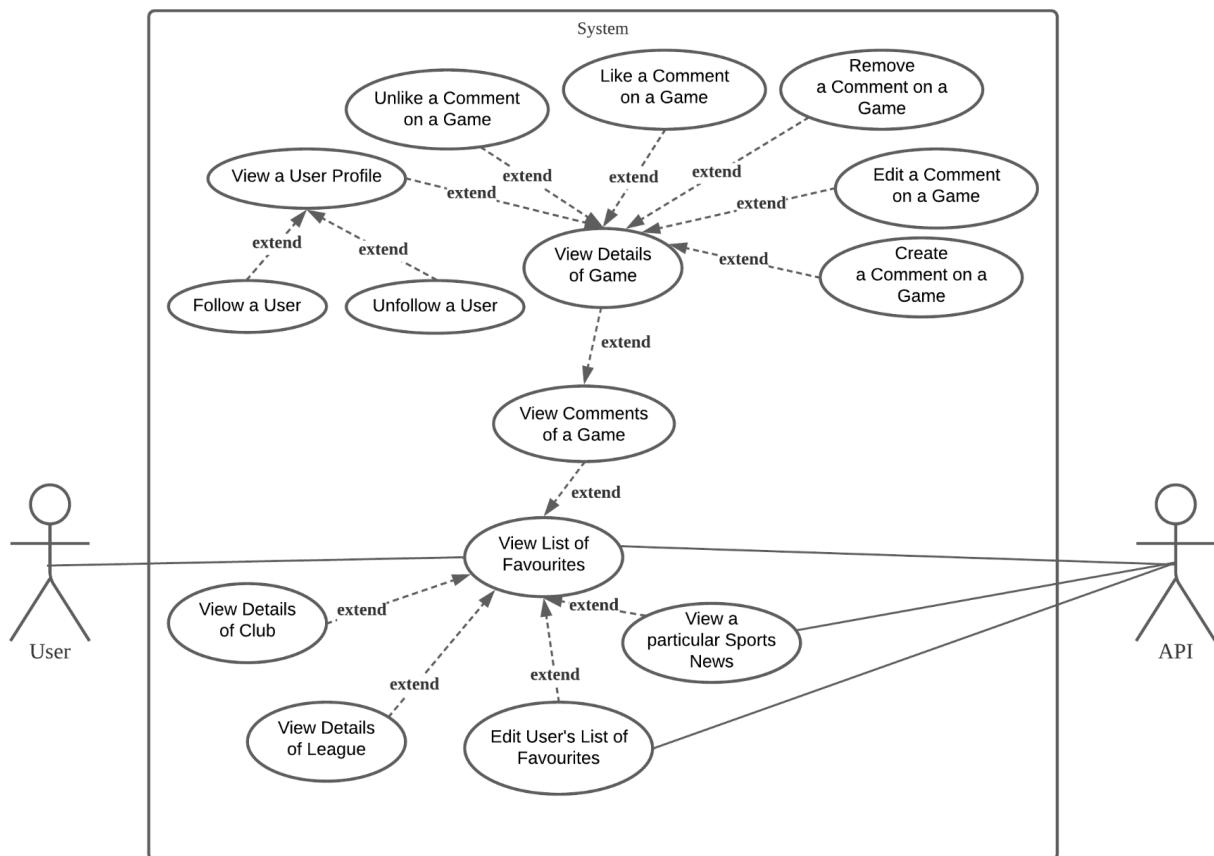
29. Unfollow a User

29.1. The **user** must be able to **unfollow** another user while viewing their profile.
29.2. The **system** must be able to **display** that User A is no longer following User B and update the "Following" and "Followers" count on their User's profile correctly.

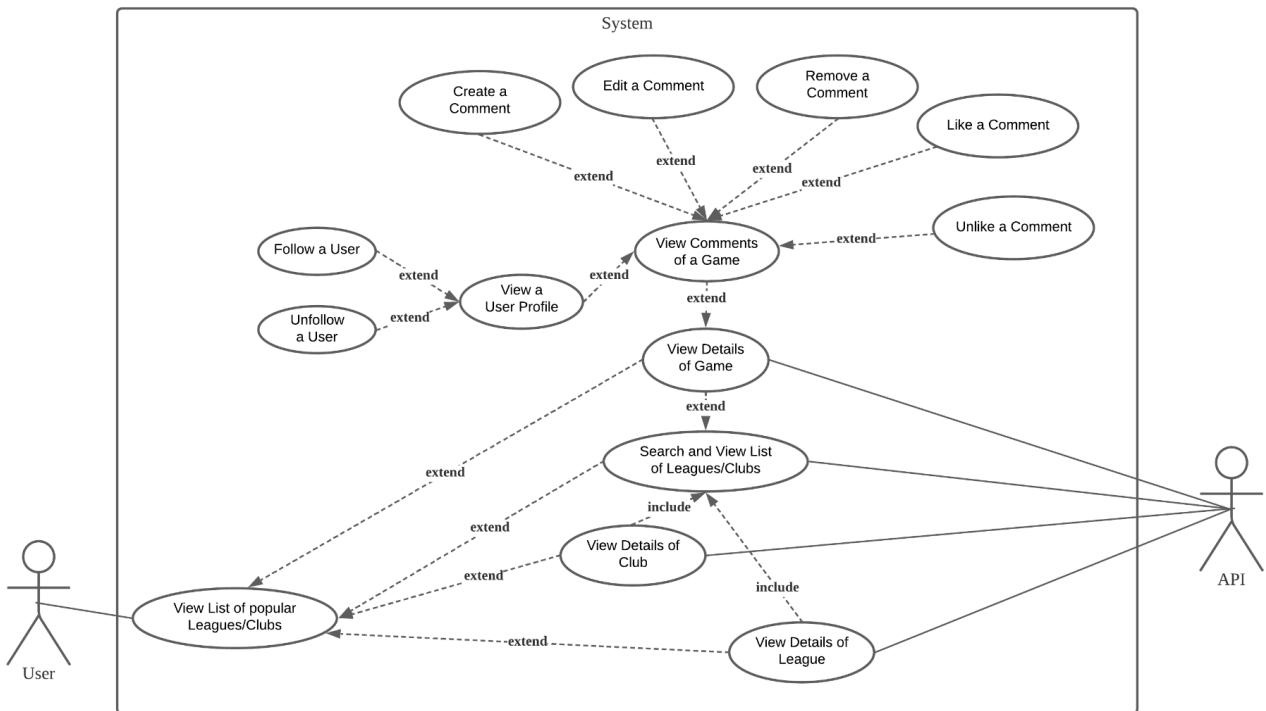
2.1 Use Case Models

As the application is fairly large, the use cases are split into 5 different groups, 1 for each tab of the application.

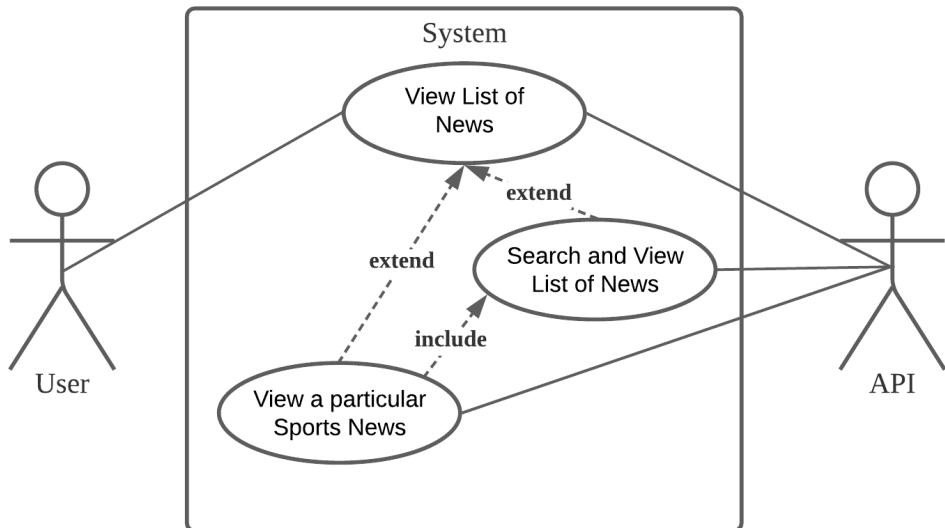
1. Explore



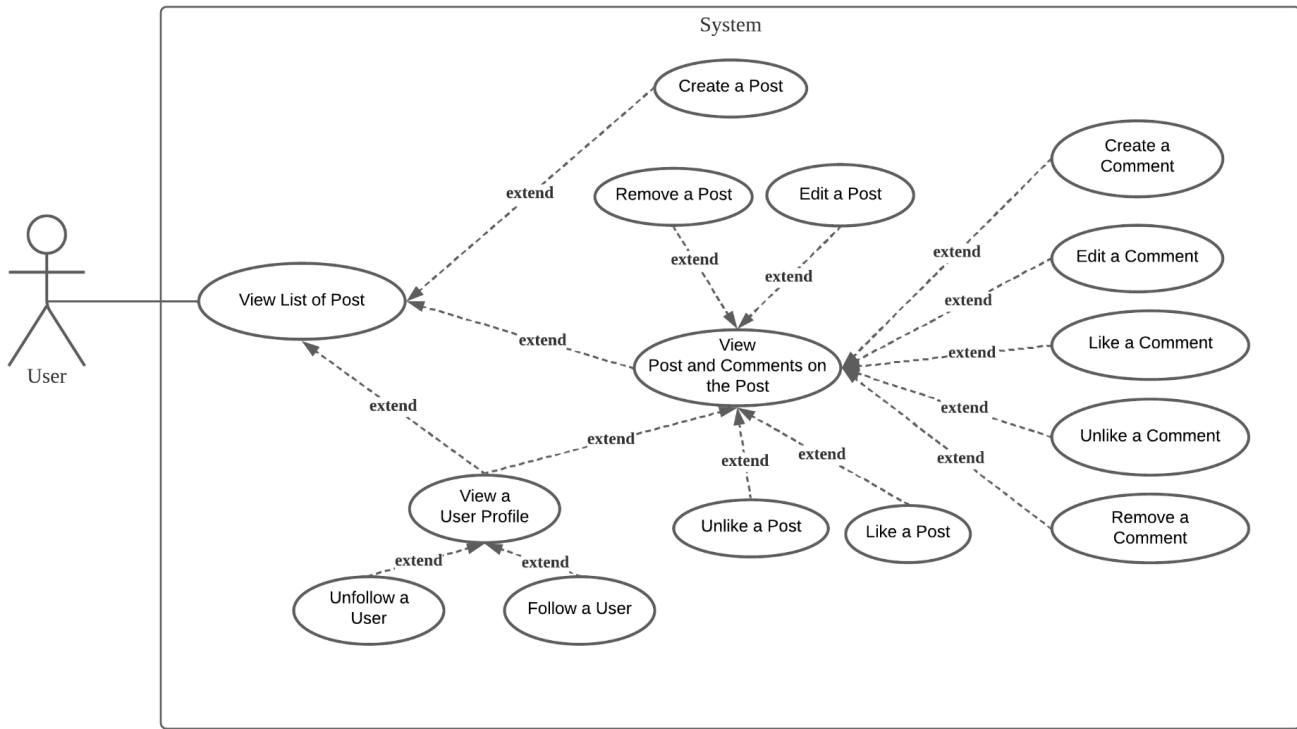
2. Favourites



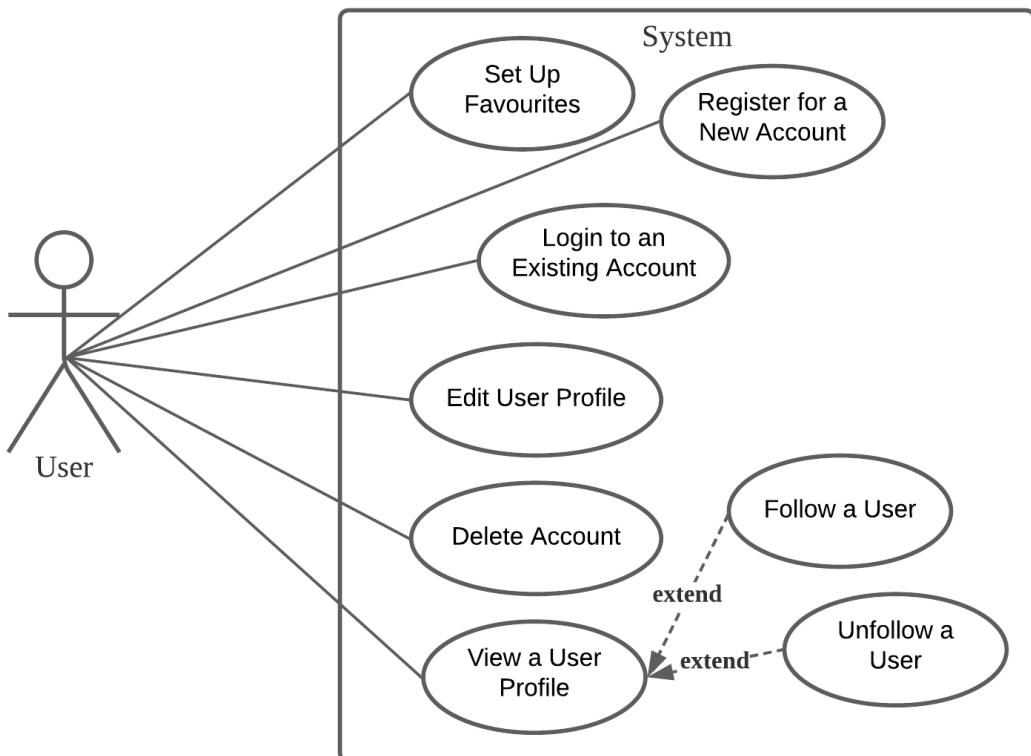
3. News



4. Forum



5. Profile



2.2 Use Case Descriptions

Use Case ID:	1		
Use Case Name:	Register for a New Account		
Created By:	Irwin Ding	Last Updated By:	Irwin Ding
Date Created:	5th February 2022	Date Last Updated:	5th February 2022

Actor:	User
Description:	The user registers for a new account.
Preconditions:	<ol style="list-style-type: none"> 1. The user has not logged in to the system. 2. The system displays the “Registration” page. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system has saved the new account details. 2. The system displays the “Set Up Favourites - League” page.
Priority:	High
Frequency of Use:	Once per user
Flow of Events:	<ol style="list-style-type: none"> 1. The user enters a username, email address and password, display name and enters his password again for confirmation. 2. The user selects the “Register” button. 3. The system ensures a match between the password and the re-entered password. 4. The system validates if the email address and username are available by checking with the database. 5. The system checks if the password is 6 characters or more. 6. The system saves the new account details and displays “Account Created”. 7. The system displays the “Set Up Favourites - League” page.
Alternative Flows:	<p>AF-S3: If the username is taken,</p> <ol style="list-style-type: none"> 1. The system displays the message “Username is taken. Please enter a different username.” 2. Return to step 1. <p>AF-S3: If the email address is taken,</p> <ol style="list-style-type: none"> 1. The system displays the message “There is an existing account registered with this email.” 2. Return to Step 1. <p>AF-S4: If there is a mismatch between password and the re-entered password,,</p> <ol style="list-style-type: none"> 1. The system displays the message “Passwords mismatch. Please re-enter your passwords.”

	2. Return to step 1.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	2		
Use Case Name:	Login to an Existing Account		
Created By:	Wei Kit	Last Updated By:	Wei Kit
Date Created:	5th February 2022	Date Last Updated:	5th February 2022

Actor:	User
Description:	The user signs in to the system using his existing account.
Preconditions:	<ol style="list-style-type: none"> 1. The user has not logged into the system. 2. The user is at the “Login” page. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system logs the user into their account. 2. The system displays the “Favourites” page.
Priority:	High
Frequency of Use:	1-5 times per year
Flow of Events:	<ol style="list-style-type: none"> 1. The user enters email and password. 2. The user selects the “Login” button. 3. The system validates the email and password by checking against the database.
Alternative Flows:	<p>AF-S2: If the email is not found</p> <ol style="list-style-type: none"> 1. The system displays the message “Invalid Email.” 2. Return to step 1. <p>AF-S2: If the password does not match the email</p> <ol style="list-style-type: none"> 1. The system displays the message “Wrong password.” 2. Return to step 1.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	3		
Use Case Name:	Updating User Details		
Created By:	Imax	Last Updated By:	Imax
Date Created:	20th February 2022	Date Last Updated:	20th February 2022

Actor:	User
Description:	The user updates his email, password, display name and username
Preconditions:	<ul style="list-style-type: none"> 1. The user has logged into the system. 2. The system displays the “Profile” page. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ul style="list-style-type: none"> 1. The system has updated the user’s new details. 2. The system displays the “Profile” page with the updated details.
Priority:	Medium
Frequency of Use:	1-3 times
Flow of Events:	<ul style="list-style-type: none"> 1. The user selects the settings button below their username. 2. The system displays the “Settings” page 3. The user inputs the fields that they want to change and the update button below the respective fields. 4. The system validates the changes by checking the database. 5. The system updates the account with the edited information. 6. The user selects the back button 7. The system displays the “Profile” page with the updated details.
Alternative Flows:	<p>AF-S5: If the user edits his username and the new username is taken</p> <ul style="list-style-type: none"> 1. The system displays the message “Username is taken. Please enter a different username.”. 2. Return to step 3. <p>AF-S5: If the user edits his email and the new email is taken</p> <ul style="list-style-type: none"> 1. The system displays the message “Invalid email”. 2. Return to step 3. <p>AF-S5: If the user did not input anything before click the update button</p> <ul style="list-style-type: none"> 1. The system displays the message “Invalid Input”. 2. Return to step 3 <p>AF-S5: If the user provides a new password of less than 6 characters</p> <ul style="list-style-type: none"> 1. The system displays the message “Invalid Password”. 2. Return to step 3
Exceptions:	-
Includes:	-

Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	4		
Use Case Name:	Delete an Existing Account		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user deletes his current account.
Preconditions:	<ol style="list-style-type: none"> 1. The user has logged in to the system. 2. The system displays the “Profile” page. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The user has been logged out. 2. The system has deleted the user’s account
Priority:	Medium
Frequency of Use:	Can only occur once per account
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects the “Delete Account” button. 2. The system removes the current account details from the database. 3. The system displays the “Login” page
Alternative Flows:	AF-S2: If the system is unable to delete the account <ol style="list-style-type: none"> 1. The system displays the message “Unable to delete account. Please try again later”. 2. The system displays the “Profile” page.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	5		
Use Case Name:	Set up Favourites		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user chooses which league or club to appear on their Favourites Page
Preconditions:	<ol style="list-style-type: none"> 1. The user has just finished registering his account. 2. The system displays the “Set Up Favourites - League” page. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system displays the games of leagues and clubs favoured by the user in the “Favourites” page.
Priority:	High
Frequency of Use:	Once after registering of account
Flow of Events:	<ol style="list-style-type: none"> 1. The system displays the leagues under the various sports. 2. The user chooses the leagues under the sport that they are interested in and selects “Next”. 3. The system displays the “Set Up Favourites - Club” page. 4. The user chooses the clubs under the sport that they are interested in and selects “Next”. 5. The system saves the user’s list of favourites. 6. The system displays the “Favourites” page with the games of their selected league and clubs and news related to the selected clubs.
Alternative Flows:	<p>AF-S2: If the user do not want to choose any league,</p> <ol style="list-style-type: none"> 1. The user selects “Next”. 2. Return to step 3. <p>AF-S4: If the user do not want to choose any club</p> <ol style="list-style-type: none"> 1. The user selects “Next”. 2. Return to step 5.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	6		
Use Case Name:	View List of Favourites		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user views the games of leagues and clubs and news related to clubs based on their list of favourites.
Preconditions:	<ol style="list-style-type: none"> 1. The user has logged in to the system. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system displays the “Favourites” page with the relevant details.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects the “Favourites” tab at the bottom of the page. 2. The system displays the “Favourites” page with the games of leagues and clubs and news related to clubs based on their list of favourites
Alternative Flows:	<p>AF-S2:If the user has no favourites yet,</p> <ol style="list-style-type: none"> 1. The system displays ‘You do not have any favourites.’ and a button with the text “Add some now!”.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	7		
Use Case Name:	Edit User's list of favourites		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user adds/removes leagues and/or clubs from their list of favourites.
Preconditions:	<ol style="list-style-type: none"> 1. The user has logged in to the system. 2. The system displays the 'Edit Favourites' page. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system saves the updated list.
Priority:	Medium
Frequency of Use:	1-3 times a month
Flow of Events:	<ol style="list-style-type: none"> 1. The user types the name of the league/club that he wants to add/remove into the search field. 2. The system displays a list with relevant suggestions based on the letters the user has typed. 3. The user clicks on the heart shaped icon league/club name. 4. The system updates the user's list of favourites by adding/removing the selected league/club. 5. The system updates the colour of the heart accordingly.
Alternative Flows:	<p>AF-S3: If there are no such league/club that begin with the letters the user has typed,</p> <ol style="list-style-type: none"> 1. The system does not display any suggestions. 2. The user clears the search field. 3. Return to Step 1.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	<ol style="list-style-type: none"> 1. The colour of the heart will be black if it is not in the user's list of favourites. 2. The colour of the heart will be pink if it is in the user's list of favourites.

Use Case ID:	8		
Use Case Name:	View Details of Game		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user views the details of a game, such as score, relevant statistics, game events and game comments.
Preconditions:	<ol style="list-style-type: none"> 1. The system displays the “Favourites” or “Explore” Page. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system displays the “Game” page with the details of the selected game.
Priority:	High
Frequency of Use:	5-6 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects the banner of the game he is interested in. 2. The system queries the API to fetch the required data. 3. The system displays the “Game” page with the information in an appropriate format.
Alternative Flows:	<p>AF-S2: If the system cannot find any data related to the game</p> <ol style="list-style-type: none"> 1. The system displays “Data currently unavailable. Please try again later”. 2. The system displays the “Favourites” or “Explore” page.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	9		
Use Case Name:	View List of Popular Leagues and Clubs		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user views a list of games in each popular league and club.
Preconditions:	<ol style="list-style-type: none"> 1. The user has logged into the system. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system displays “Explore” page
Priority:	High
Frequency of Use:	5-7 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects the “Explore” tab at the bottom of the page. 2. The system queries the API for games. 3. The system displays every game of each league and club.
Alternative Flows:	<p>AF-S2: If the system is not able to retrieve any information from the API,</p> <ol style="list-style-type: none"> 1. The system displays “Unable to retrieve information. Please try again later”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	10		
Use Case Name:	Search and View List of Leagues/Clubs		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user search for a specific league/club
Preconditions:	<ul style="list-style-type: none"> 1. The user is at the 'Explore' page. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ul style="list-style-type: none"> 1. The system displays the list of Leagues/Clubs that are relevant to the user's search.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ul style="list-style-type: none"> 1. The user clicks the search bar on the 'Explore' page. 2. The user types the name of the league/club they are interested in. 3. The system queries the API with the letters that the user has typed . 4. The system displays a dropdown list with relevant suggestions.
Alternative Flows:	<p>AF-S3: If there are no such teams/competitions that begin with the letters the user has typed,</p> <ul style="list-style-type: none"> 1. The system does not display any suggestions. 2. The user clears the search field 3. Return to Step 2.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	11		
Use Case Name:	View Details of League		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user can view the past and future games as well as the current league standings of a league.
Preconditions:	<ol style="list-style-type: none"> 1. The user is at the “Explore” or “Favourites” page. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system displays the “League” page with the league details.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks on the banner of the league that he wants to view information on. 2. The system queries the API to fetch the relevant data. 3. The system displays the “League” page with the information.
Alternative Flows:	<p>AF-S2: If the system cannot find any data related to the competition,</p> <ol style="list-style-type: none"> 1. The system displays “Data currently unavailable. Please try again later”.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	12		
Use Case Name:	View Details of Club		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user views details (eg. fixtures and results) of a club.
Preconditions:	<ul style="list-style-type: none"> 1. The user is at the “Explore” or “Favourites” page. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ul style="list-style-type: none"> 1. The system displays the “Club” page with the club details.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ul style="list-style-type: none"> 1. The user clicks on the banner of the club that he wants to view information on. 2. The system queries the API to fetch the relevant data. 3. The system displays the “Club” page with the information.
Alternative Flows:	<p>AF-S2: If the system is unable to retrieve the data,</p> <ul style="list-style-type: none"> 1. The system displays “Data currently unavailable. Please try again later.”.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	13		
Use Case Name:	View List of News		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user views the headline of various sports articles at the “News” page
Preconditions:	1. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	1. The system displays the “News” page
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects the “News” tab at the bottom of the page. 2. The system queries the API for sport news. 3. The system displays the “News” page with the headline of various sports articles.
Alternative Flows:	AF-S2: If the system is not able to retrieve any information from the API, 1. The system displays “Unable to retrieve information. Please try again later”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	14		
Use Case Name:	View a particular Sports News		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user views the full news article.
Preconditions:	<ul style="list-style-type: none"> 1. The user is at the “News” page. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ul style="list-style-type: none"> 1. The system displays the “New Article” page with the full news article.
Priority:	High
Frequency of Use:	5-6 times a day
Flow of Events:	<ul style="list-style-type: none"> 1. The user selects the news article that he wants to view from the “News” page. 2. The system query the API to fetch the news article. 3. The system displays the article in the “New Article” page.
Alternative Flows:	AF-S2: If the system cannot find the news article from the API <ul style="list-style-type: none"> 1. The system will display “Article currently unavailable. Please try again later”.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	15		
Use Case Name:	View List of Posts		
Created By:	Shaojie	Last Updated By:	Irwin Ding
Date Created:	10th February 2022	Date Last Updated:	17th February 2022

Actor:	User
Description:	The user views the list of posts on the forum.
Preconditions:	1. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	1. The system displays the list of posts under the “Forum” page.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects the “Forum” tab at the bottom of the page. 2. The system retrieves the list of posts and displays them at the “Forum” page.
Alternative Flows:	AF-S2: If the system is unable to retrieve the list of posts, <ol style="list-style-type: none"> 1. The system displays “Unable to retrieve information. Please try again later”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	16		
Use Case Name:	Create a Post		
Created By:	Shaojie	Last Updated By:	Irwin Ding
Date Created:	10th February 2022	Date Last Updated:	17th February 2022

Actor:	User
Description:	The user creates a new post on the forum.
Preconditions:	<ol style="list-style-type: none"> 1. The user is at the “Forum” page. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	1. The system displays the “Forum” page with the new post.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects on the “Create new post” button. 2. The system displays the “Create Post” page. 3. The user types their content into the text box provided. 4. When the user is done, the user clicks on the “Confirm” button. 5. The system validates the content of the post. 6. The system saves the user’s new post. 7. The system displays the “Forum” page with the new post.
Alternative Flows:	<p>AF-S4: If the user does not want to post on the forum,</p> <ol style="list-style-type: none"> 1. The user selects the back arrow on the top left of the page 2. The system displays the “Forum” page. <p>AF-S5: If the new post is blank,</p> <ol style="list-style-type: none"> 1. The system displays “Please enter a valid post” 2. Return to step 3 <p>AF-S6: If the system is unable to save the new post,</p> <ol style="list-style-type: none"> 1. The system displays “Unable to create post. Please try again later.” 2. The system displays the “Forum” page.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	17		
Use Case Name:	View Post and Comments on the Post		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user views a post and the comments under the post.
Preconditions:	<ul style="list-style-type: none"> 1. The user is at the “Forum” page. 2. There are existing posts. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ul style="list-style-type: none"> 1. The system displays the post and any comments on the “Post” page.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ul style="list-style-type: none"> 1. The user clicks on a post he wants to see. 2. The system retrieves the comments for the post. 3. The system displays the “Post” page with the selected post along with the comments below it.
Alternative Flows:	<p>AF-S2: If there are no comments,</p> <ul style="list-style-type: none"> 1. The system displays only the post on the “Post” page. <p>AF-S2: If the system is unable to retrieve the comments,</p> <ul style="list-style-type: none"> 1. The system displays “Unable to retrieve information. Please try again later”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	18		
Use Case Name:	Edit a Post		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user edits the contents of a post he posted.
Preconditions:	<ol style="list-style-type: none"> 1. The user has posted a post. 2. The user is at the “Post” page viewing his own post. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system saves the updated post. 2. The system displays the “Forum” page with the updated post.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks the “Edit Post” button on his post. 2. The system displays the “Create Post” page with the original contents of the post in the text field. 3. The user edits the post to his liking. 4. Once done, the user clicks the ‘Post’ button to complete editing the post. 5. The system validates the content of the post. 6. The system saves and displays the post page with the updated post .
Alternative Flows:	<p>AF-S4: If the user does not want to change his post,</p> <ol style="list-style-type: none"> 1. The user clicks on the back arrow on the top left of the page. 2. The system displays the “Post” page with the original post. <p>AF-S6: If the edited post is blank,</p> <ol style="list-style-type: none"> 1. The system displays “Please enter a valid post”. 2. Return to step 3 <p>AF-S6: If the system is unable to save the updated content,</p> <ol style="list-style-type: none"> 1. The system displays “Unable to update post. Please try again later
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	19		
Use Case Name:	Remove a Post		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user removes a post that he has posted.
Preconditions:	<ol style="list-style-type: none"> 1. The user has posted a post. 2. The user is at the “Post” page viewing his own post. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system removes the post and any associated comments from the forum. 2. The system displays the “Forum” page without the deleted post.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks on the dustbin icon. 2. The system shows a popup, asking if the user wants to delete the post. 3. The user clicks ‘Yes’ . 4. The system removes the post and any comments under the post. 5. The system displays the “Forum” page without the deleted post.
Alternative Flows:	<p>AF-S3: If the user does not want to delete his post,</p> <ol style="list-style-type: none"> 1. The user clicks “No”. 2. The system displays the “Post” page with his own post.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	20		
Use Case Name:	Like a Post		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user likes a post.
Preconditions:	<ol style="list-style-type: none"> 1. The user is on the 'Forum' or "Post" page. 2. There are existing posts. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system shows that the post is liked. 2. The system updates the number of likes for the post.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks the thumbs up icon of the post he wants to like. 2. The system makes the thumbs up turn green and the number of likes increase by 1. 3. The system saves the number of likes.
Alternative Flows:	<p>AF-S1: If the user has previously dislike the post,</p> <ol style="list-style-type: none"> 1. The thumbs down button will turn white and the number of unlikes decreases by 1. 2. The system saves the number of unlikes. <p>AF-S2: If the user has already liked the post,</p> <ol style="list-style-type: none"> 1. The system does not make any changes.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	21		
Use Case Name:	Dislike a Post		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user dislikes a post.
Preconditions:	<ol style="list-style-type: none"> 1. The user is at the “Forum” or “Post” page. 2. There are existing posts. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system shows that the post is disliked. 2. The system updates the number of dislikes for the post.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks the thumbs down icon of the post he wants to unlike. 2. The system makes the thumbs down turn red and the number of unlikes increase by 1. 3. The system saves the number of unlikes.
Alternative Flows:	<p>AF-S1: If the user has previously like the post,</p> <ol style="list-style-type: none"> 1. The thumbs ups button turns white and the number of likes decreases by 1. 2. The system saves the number of likes. <p>AF-S2: If the user has already disliked the post,</p> <ol style="list-style-type: none"> 1. The system does not make any changes.
Exceptions:	None
Includes:	None
Special Requirements:	None
Assumptions:	-
Notes and Issues:	-

Use Case ID:	22		
Use Case Name:	Create a Comment		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user creates a new comment on a post or a game.
Preconditions:	<ol style="list-style-type: none"> 1. The user is at the “Post” page or “Game Comment” page. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system displays the “Post” or “Game Comments” page with the new comment.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user selects the “Add Comment” text field at the bottom. 2. The user types their comment into the text field. 3. When the user is done, the user clicks on the send message icon on the right of the text field. 4. The system saves the new comment. 5. The system updates the page with the new comment.
Alternative Flows:	<p>AF-S4: If the text field is empty,</p> <ol style="list-style-type: none"> 1. The system displays “Please enter a valid comment.” <p>AF-S4: If the system is unable to save the comments,</p> <ol style="list-style-type: none"> 1. The system will display the message “Unable to post the comment.” Please try again later”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	23		
Use Case Name:	Edit a Comment		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user edits the contents of a comment made by him.
Preconditions:	<ul style="list-style-type: none"> 1. The user has commented on a post or a game. 2. The user is viewing his own comment. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ul style="list-style-type: none"> 1. The system saves the updated comment 2. The system displays the edited comment.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ul style="list-style-type: none"> 1. The user clicks the ellipsis icon on his comment. 2. The system displays a dropdown list below the ellipsis. 3. The user selects the “Edit Comment” option. 4. The system converts the comment text box into an editable text field with the current comment. 5. The user clicks the “Save” button after editing their comment. 6. The system saves the updated comment. 7. The system converts the text field back into a text box with the updated comment.
Alternative Flows:	<p>AF-S6: If the edited comment is blank,</p> <ul style="list-style-type: none"> 1. The system displays “Please enter a valid comment.” <p>AF-S6: If the system is unable to save the edited comment,</p> <ul style="list-style-type: none"> 1. The system displays “Unable to save comment. Please try again later.”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	24		
Use Case Name:	Remove a Comment		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user removes a post posted by him.
Preconditions:	<ul style="list-style-type: none"> 1. The user has posted a comment. 2. The user is viewing his own comment. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ul style="list-style-type: none"> 1. The system has removed the user's comment. 2. The system displays the list of comments without the deleted comment.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ul style="list-style-type: none"> 1. The user clicks the ellipsis icon on his comment. 2. The system displays a dropdown list below the ellipsis. 3. The user selects the "Delete" option. 4. The system removes the comment. 5. The system displays the list of comments without the deleted comment.
Alternative Flows:	<p>AF-S4: If the system is unable to remove the comment,</p> <ul style="list-style-type: none"> 1. The system displays "Unable to remove comment. Please try again later."
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	25		
Use Case Name:	Like a Comment		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022		Date Last Updated: 10th February 2022

Actor:	User
Description:	The user likes a comment.
Preconditions:	<ul style="list-style-type: none"> 1. The user is viewing a comment. 2. There are existing comments. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ul style="list-style-type: none"> 1. The system shows that the comment is liked. 2. The system has updated the number of likes for the comment.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ul style="list-style-type: none"> 1. The user clicks the thumbs up icon of the comment he wants to like. 2. The system makes the thumbs up turn green and the number of likes increase by 1. 3. The system saves the number of likes.
Alternative Flows:	<p>AF-S1: If the user has previously unlike the comment,</p> <ul style="list-style-type: none"> 1. The thumbs down button turns white and the number of unlikes decreases by 1. 2. The system saves the number of unlikes. <p>AF-S2: If the user has already liked the comment,</p> <ul style="list-style-type: none"> 1. The system does not make any changes.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	26		
Use Case Name:	Dislike a Comment		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user dislikes a comment.
Preconditions:	<ol style="list-style-type: none"> 1. The user is viewing a comment. 2. There are existing comments. 3. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system shows that the comment is disliked. 2. The system has updated the number of dislikes for the post.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks the thumbs down icon of the comment he wants to dislike. 2. The system makes the thumbs down turn red and the number of dislikes increases by 1. 3. The system saves the number of dislikes.
Alternative Flows:	<p>AF-S1: If the user has previously like the comment,</p> <ol style="list-style-type: none"> 1. The thumbs ups button turns white and the number of likes decreases by 1. 2. The system saves the number of likes. <p>AF-S2: If the user has already disliked the comment,</p> <ol style="list-style-type: none"> 1. The system does not make any changes.
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	27		
Use Case Name:	View a User Profile		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

Actor:	User
Description:	The user views the details of other users
Preconditions:	<ul style="list-style-type: none"> 1. The user is viewing the comment or post made by the other user. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ul style="list-style-type: none"> 1. The system displays the “Profile” page with the details of the other user.
Priority:	High
Frequency of Use:	5-7 times
Flow of Events:	<ul style="list-style-type: none"> 1. The user clicks on the username of the other user. 2. The system retrieves the information of the other user 3. The system displays the “Profile” page with the details of the other user.
Alternative Flows:	<p>AF-S2: If the account has been deleted,</p> <ul style="list-style-type: none"> 1. The system displays “Account does not exist anymore” <p>AF-S2: If the system is unable to retrieve the user's data,</p> <ul style="list-style-type: none"> 1. The system displays “Unable to retrieve data. Please try again later.”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

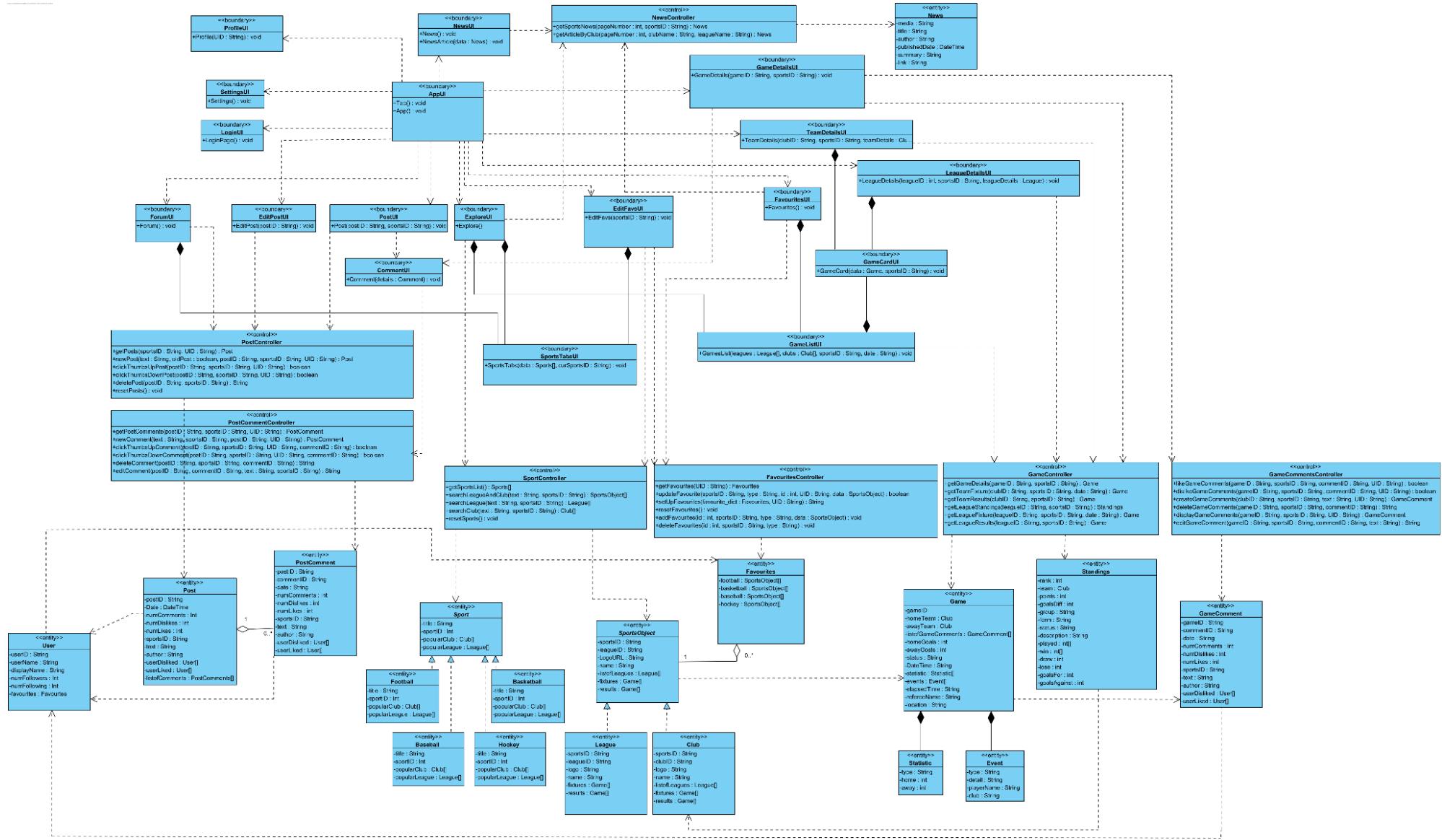
Use Case ID:	28		
Use Case Name:	Follow a User		
Created By:	Kai Teng	Last Updated By:	Kai Teng
Date Created:	11th February 2022	Date Last Updated:	11th February 2022

Actor:	User
Description:	The user follows other users to receive notification when they make a post.
Preconditions:	<ul style="list-style-type: none"> 1. The user is at the “Profile” page of another user that he has yet to follow. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ul style="list-style-type: none"> 1. The system has updated the following list of the follower and the follower list of the followee.
Priority:	High
Frequency of Use:	1-3 times
Flow of Events:	<ul style="list-style-type: none"> 1. The user clicks on the “Follow” button. 2. The system updates the following list of the follower and the follower list of the followee. 3. The “Follow” button now displays “Following”, showing that the user has followed the other user.
Alternative Flows:	<p>AF-S2: If the system is unable to update the following list,</p> <ul style="list-style-type: none"> 1. The system displays “Unable to follow user. Please try again later.”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

Use Case ID:	29		
Use Case Name:	Unfollow a User		
Created By:	Shaojie	Last Updated By:	Shaojie
Date Created:	10th February 2022	Date Last Updated:	10th February 2022

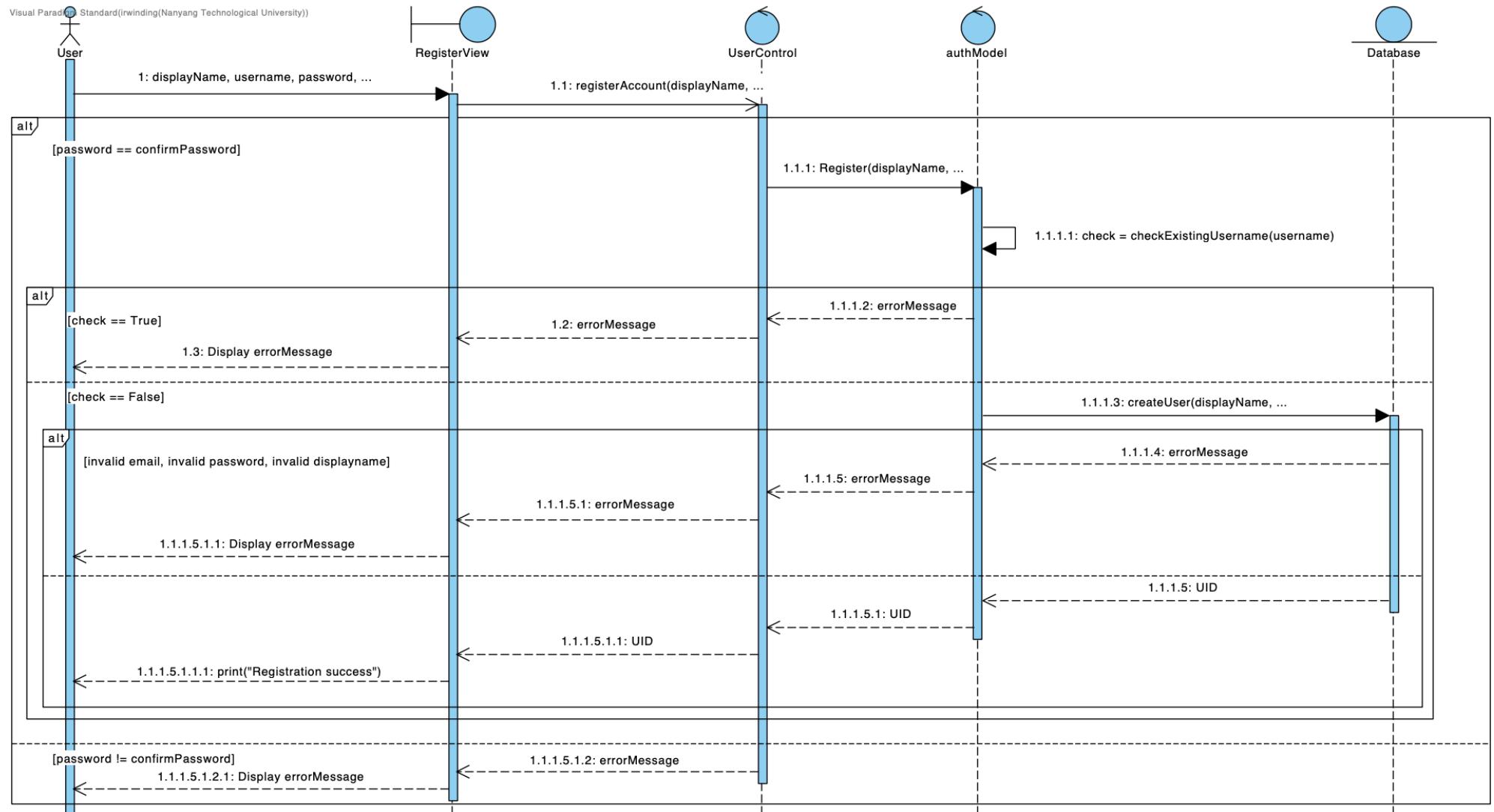
Actor:	User
Description:	The user unfollows another user that he is currently following.
Preconditions:	<ol style="list-style-type: none"> 1. The user is at the “Profile” Page of the user he wants to unfollow. 2. Mobile must be connected to WiFi/Mobile Data.
Postconditions:	<ol style="list-style-type: none"> 1. The system has updated the following list of the follower and the follower list of the followee.
Priority:	Medium
Frequency of Use:	1-3 times a day
Flow of Events:	<ol style="list-style-type: none"> 1. The user clicks on the “Following” button. 2. The system updates the following list of the follower and the follower list of the followee. 3. The “Following” button turns from white to black and displays “Follow”, showing that the user is currently not following the other user.
Alternative Flows:	<p>AF-S2: If the system is unable to update the following list,</p> <ol style="list-style-type: none"> 1. The system displays “Unable to unfollow user. Please try again later”
Exceptions:	-
Includes:	-
Special Requirements:	-
Assumptions:	-
Notes and Issues:	-

2.3 Class Diagram

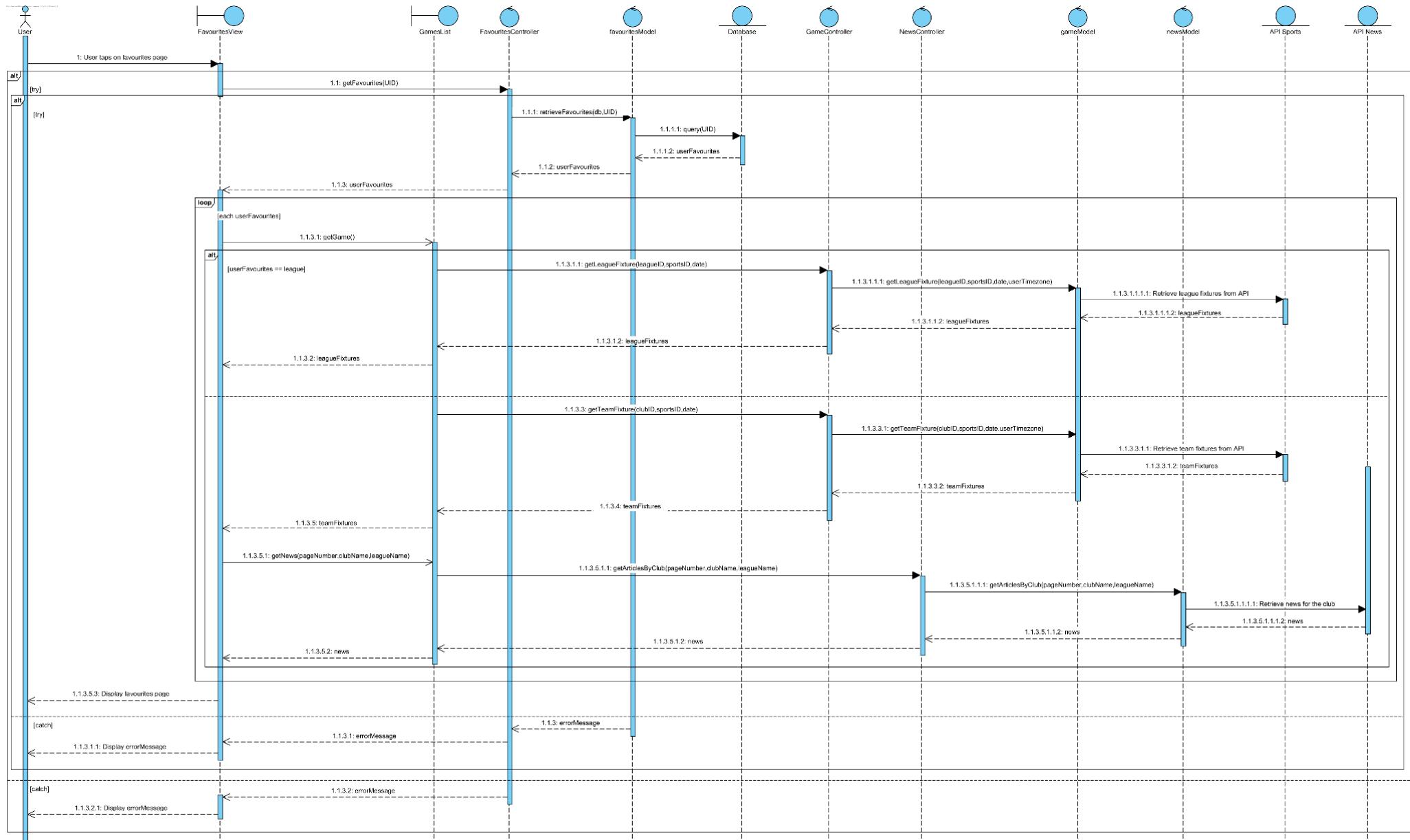


2.4 Sequence Diagram

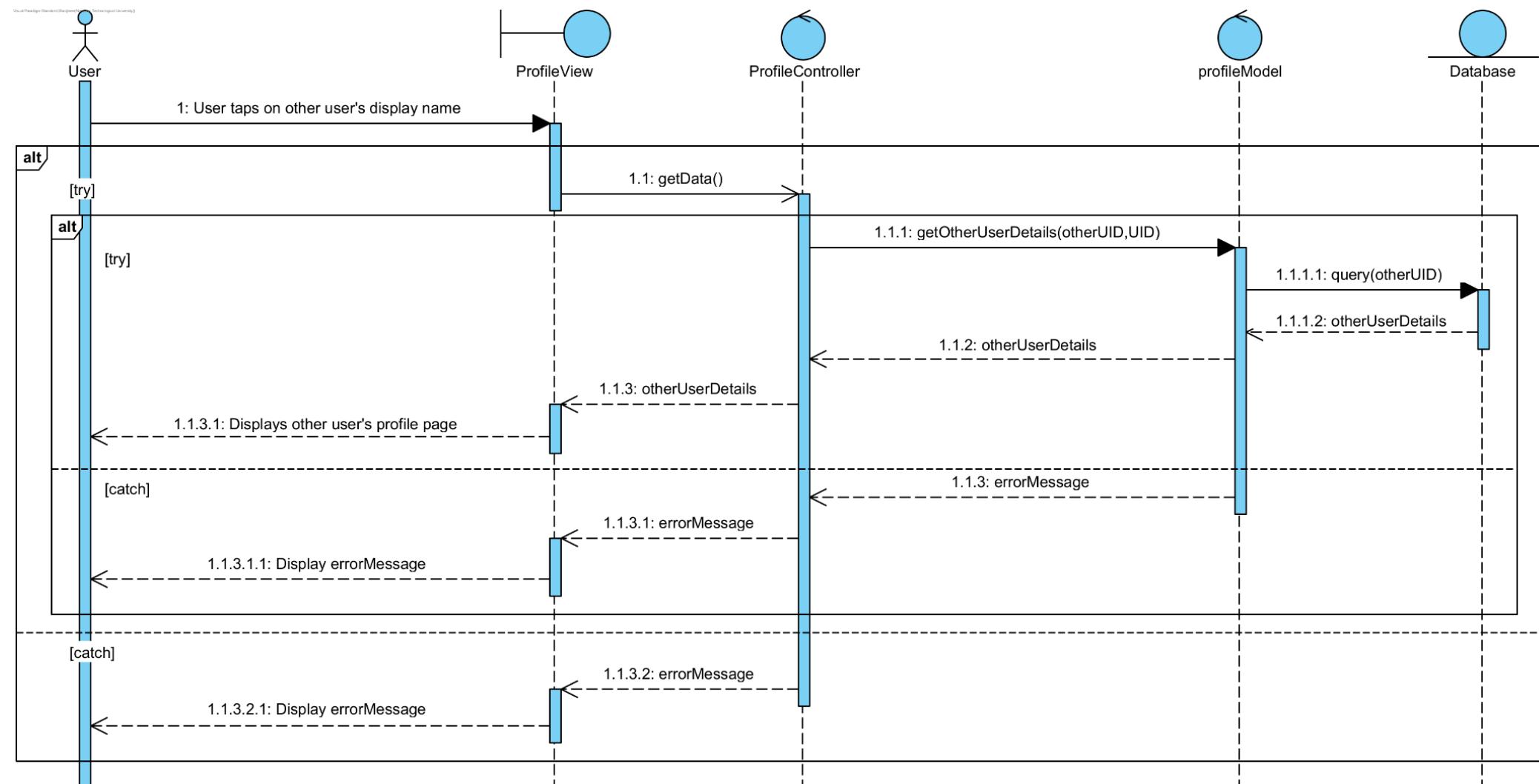
1. Register New Account



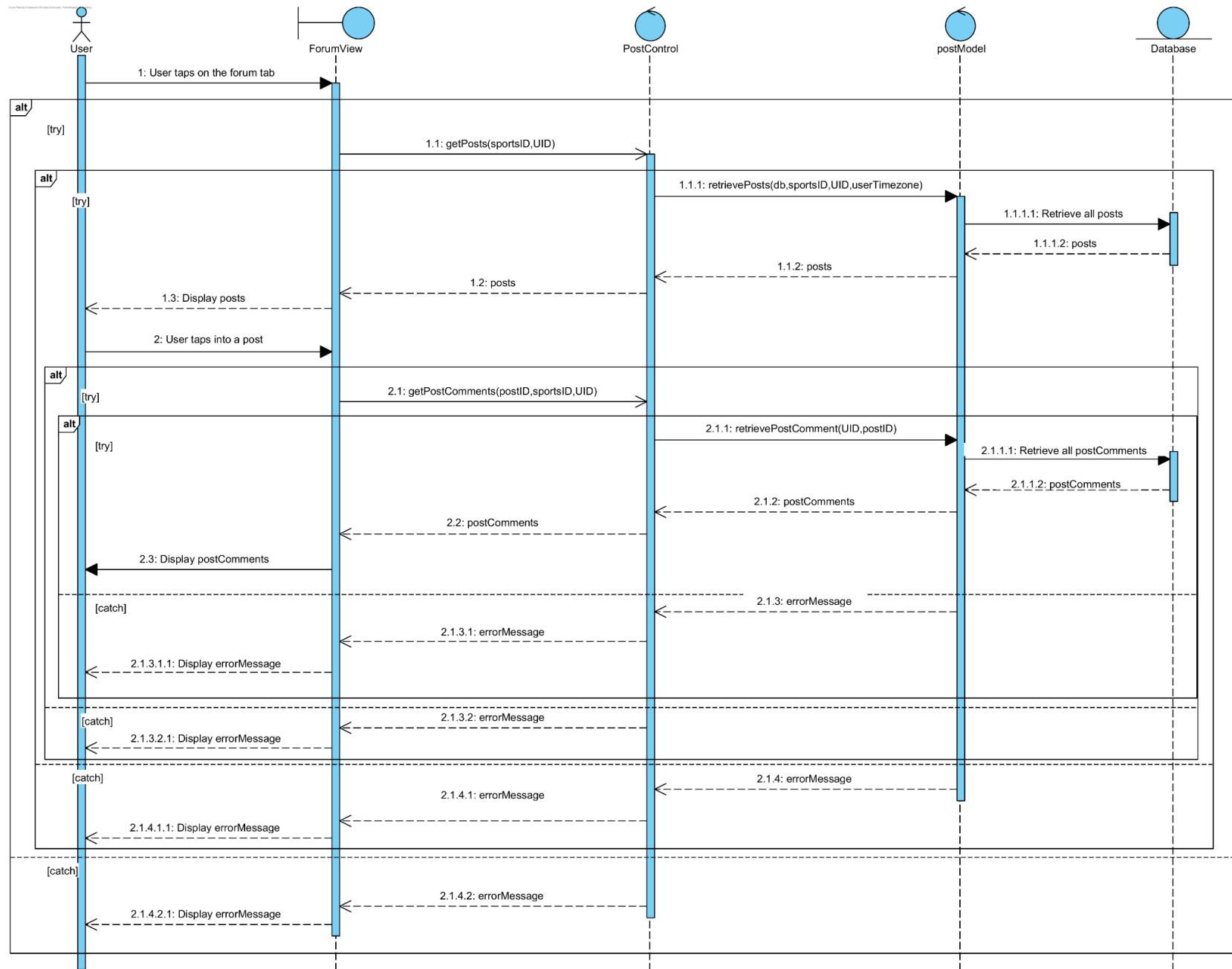
2. Display Favourites



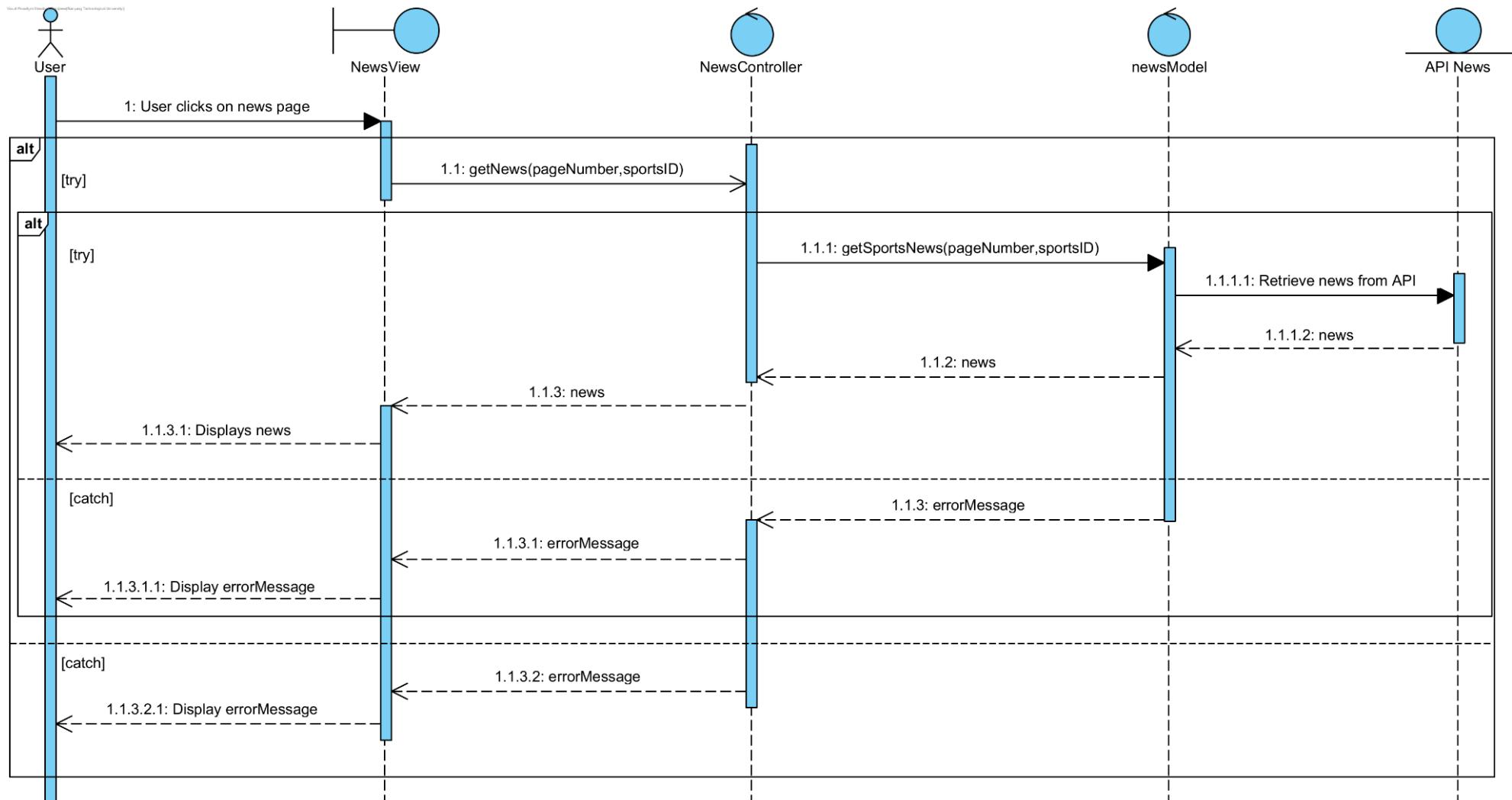
3. Display Profile



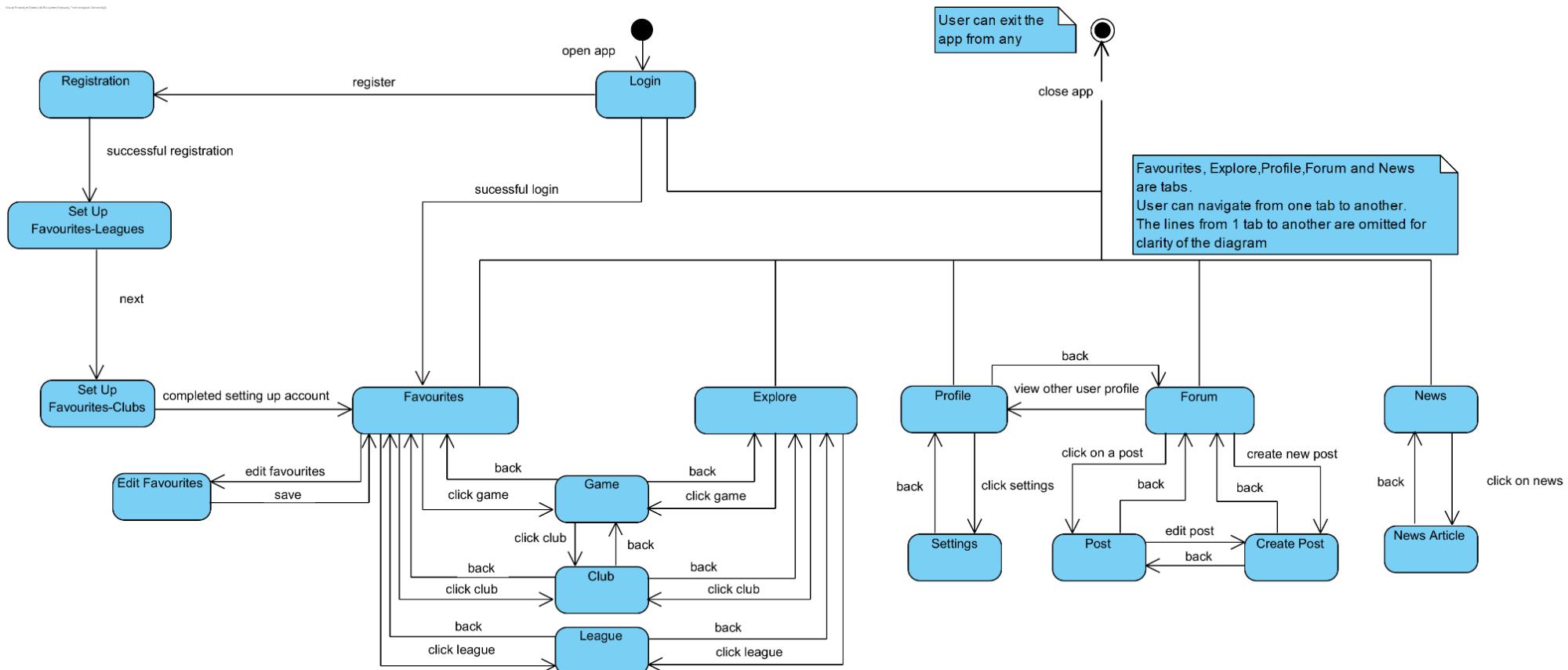
4. Display Forum



5. Display News



2.5 Dialog Map



3 Non-Functional Requirements

1. Usability Requirements

- 1.1. The system must provide informative feedback to the user.
 - 1.1.1. Provides necessary feedback to the user when invalid inputs are detected.
 - 1.1.2. Displays an appropriate error message when certain processes fail.
- 1.2. The system must reduce short term memory load on the user.
 - 1.2.1. User Interfaces must be intuitive and easy for the user to use.
- 1.3. The system must permit easy reversal of actions.
 - 1.3.1. Users can easily add and remove leagues and clubs from their list of favourites.
- 1.4. The system must strive for consistency.

2. Performance Requirements

- 2.1. The system must not crash while the user is using the application.
- 2.2. The system must take less than 10 seconds to display information (eg.news article)

3. Security Requirements

- 3.1. The system masks input password to prevent any potential shoulder surfing.
- 3.2. The system hashes the password when it is stored into the database.

4. Extensibility Requirements

- 4.1. Additional sports can be added into the app easily without changing much of the source code under the View.

5. Recoverability Requirements

- 5.1. The user's account details will be stored in the app's database, such that in the case that the user deletes the app or changes their device, they can get their account back easily.

4 Interface Requirement

4.1 User

SupaSports works for all users who want to read information about sports including but not limited to match details, transfer news, live score updates etc. It is also a place for users to share their opinion on all sports related topics and engage in a discussion with other users.

4.2 Hardware

SupaSports requires a smartphone.

4.3 Software

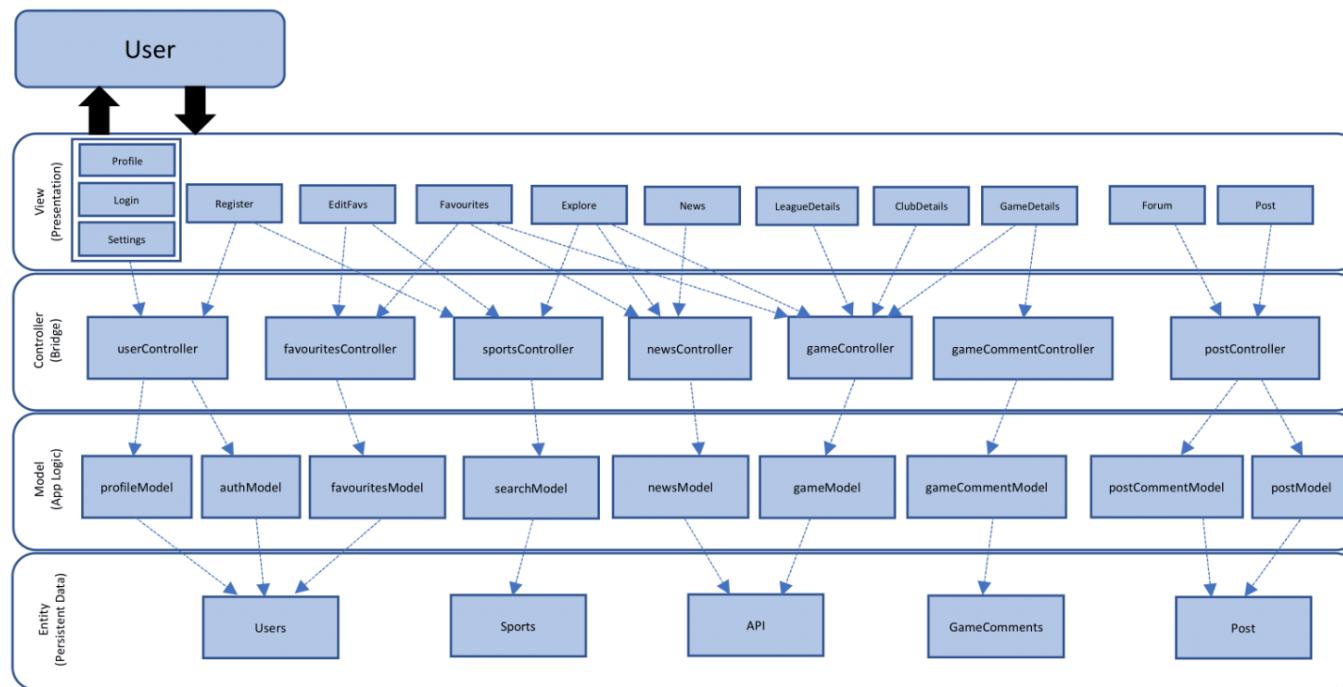
SupaSports is designed to work on Android Devices.

4.4 Communication

SupaSports will require access to the internet to retrieve account related information from Google Firestore and game information and news articles from an API.

5 System Design

5.1 System Architecture Diagram



Our system architecture follows a 4-layered architecture with a top to bottom component communication. This allows us to properly understand the layer to layer communication.

Our Model(app logic) is hosted on Google Cloud Functions as an API. The Controllers act as a bridge between the model and the view by calling the appropriate endpoints of the Model to run the appropriate functions in order to retrieve or manipulate our Entities. Our Entities(persistent data) are hosted on Google Firestore, with the exception of the API being a third party data source.

5.2 Design Pattern

1. Object Design (Reusable Components)

Problem

Many of the components in a view are similar across the different views and coding them out individually in each view can cause the code to look unnecessarily long and complicated. As a result, debugging required a longer time.

Solution

The property of reusable components in React Native helped to improve the readability of our code. By importing the functional components that are reused frequently, chunks of repeated code were reduced into only a few lines. Furthermore, in certain contexts, the imported components were extended in order to make the code more reusable.

2. Strategy Pattern

Problem

One of the most important features of the app is that users must be able to view details of both leagues and clubs. Initially, we separated the functions which were responsible for retrieving and displaying information of both leagues and clubs. This made the code harder to maintain as the debugging process is longer. In addition, any change to one of the functions might affect the whole module which increases the chance of creating an error in already-working code.

Solution

By implementing the strategy pattern, we are able to ensure that the correct algorithm runs based on the context given by the User's input. This allowed us to merge many functions together.

For example in the Favorites Page, there is no need to separate the functions for displaying information of a league and a club. Instead, when the user chooses to view information of a club or league, the corresponding parameters will be passed into the function which decides the appropriate algorithm to run to provide the correct information to the user.

As a result, it provides encapsulation and allows us to easily add similar functionality without changing much of the source code in the future.

With reference to the previous example, if a feature that allows the User to view player details was added into the app, an additional algorithm just needs to be added to extract player details.

3. Model-View-Component Design

Problem

The UI components handle interaction with the database as well which makes the code extra long and difficult to follow the flow.

Solution

We split up the complicated functions into their respective roles - Model, View or Component.

This makes the code have a better flow and each function has its own responsibility which satisfies the Single Responsibility Principle as well.

For example, when a user wants to view a list of teams under “Soccer”. The user will enter their input by clicking the Soccer tab in the Explore page. The ExploreController class reads in the click input by the user and calls the “getListOfTeams” function which is in the ExploreModel class. Then the ExploreModel class will interact with the Sports API to gather a list of teams in Soccer and return the list back to ExploreController.

ExploreController then sends that list of teams to ExploreView to present to the user the list of soccer teams.

6 Data Dictionary

Terms	Definition/Examples
User	A person who is using this application to read the news, scores and schedules of any sport.
System	A system is the application.
List of Favourites	List of Favourites refers to leagues and/or clubs that the user is interested in.
Favourites	Favourites refers to the page that allows the user to view games of leagues/clubs in their List of Favourites
League	League refers to a competition that a club competes in.
Club	Club refers to an individual team.
Post	Post refers to a block of text created by a user starting a discussion or stating their opinion.
Comments	Comments refers to remarks made by users under a post or games.
Forum	Forum refers to the page where users can create and view posts also where the users comment on other posts as well.
Profile	Profile of a user refers to a page where someone else can see the particulars of the user
Game	Game refers to a match played between two clubs.

7 Testing

7.1 Black box testing

1. Account Creation

1.1. Generic Cases

Test Id	Test	Expected outcome	Actual outcome
1	User registers successfully with a unique username and a unique email address and password.	The system displays the main menu for user to continue the operation	The system displays the main menu for user to continue the operation
2	User attempts to register with an existing username or email.	The system shows a prompt that says the username or email has been taken	The system shows a prompt that says the username or email has been taken
3	User attempts to register with an invalid email	The system shows a prompt that says the email is invalid	The system shows a prompt that says the email is invalid
4	User attempts to register with incomplete fields	The system prompts the user to fill up the required fields for registration	The system prompts the user to fill up the required fields for registration
5	User attempts to register with password mismatch	The system prompts the user re-enter the password	The system prompts the user re-enter the password
6	User attempts to register with a short password (<6 characters)	The system will show a prompt that says the password is too short	The system will show a prompt that says the password is too short

1.2. Specific Cases

Username	Email	Password	Re-enter password	Expected Outcome	Actual outcome
uniqueusername	unique@gmail.com	validpass	validpass	Account Created!	Account Created!
existingname	unique@gmail.com	validpass	validpass	Username is already taken.	Username is already taken.
uniqueusername	existing@gmail.com	validpass	validpass	There already exists an account with the given email address.	There already exists an account with the given email address.
uniqueusername	invalid	validpass	validpass	Invalid email.	Invalid email.
uniqueusername	invalid@invalid	validpass	validpass	Invalid email.	Invalid email.
uniqueusername	unique@gmail.com	invalidpass	invalidpass	Password requires at least 6 characters.	Password requires at least 6 characters.
uniqueusername	unique@gmail.com	“password”	diffpassword	Please ensure the passwords entered are the same!	Please ensure the passwords entered are the same!
(Empty)	unique@gmail.com	validpass	validpass	Please fill in the empty field.	Please fill in the empty field.
uniqueusername	(Empty)	validpass	validpass	Please fill in the empty field.	Please fill in the empty field.
uniqueusername	unique@gmail.com	(Empty)	validpass	Please fill in the empty field.	Please fill in the empty field.
uniqueusername	unique@gmail.com	validpass	(Empty)	Please fill in the empty field.	Please fill in the empty field.

2. User logs in

2.1. Generic Cases

Test ID	Test	Expected Outcome	Actual outcome
1	User attempts to log in with valid email and correct password	The system displays the main menu for user to continue the operation	The system displays the main menu for user to continue the operation
2	User attempts to log in with empty fields	System will prompt user to fill in empty fields	System will prompt user to fill in empty fields
3	User attempts to log in with an invalid email	The system shows a prompt that says the email is invalid	The system shows a prompt that says the email is invalid
4	User attempts to log in with a valid email but incorrect password	The system shows a prompt that says the password is wrong	The system shows a prompt that says the password is wrong
5	User attempts to log into a deleted account	The system shows a prompt that says the account was deleted previously	The system shows a prompt that says the account was deleted previously
6	User attempts to log in with a new email.	The system shows a prompt that says the user is not found.	The system shows a prompt that says the user is not found.

2.2. Specific Cases

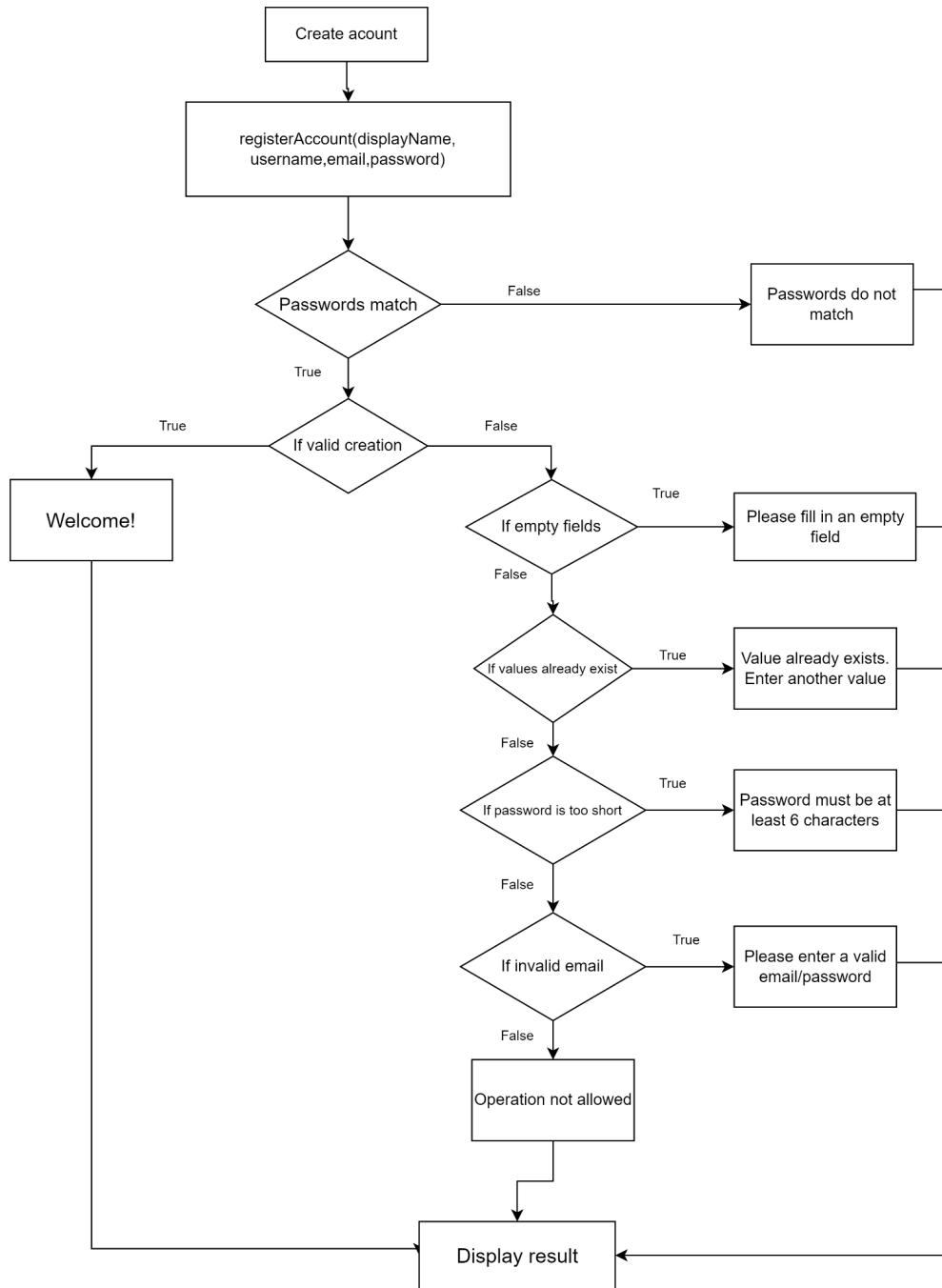
Email	Password	Expected outcome	Actual outcome
valid@gmail.com	correctpass	Logged in	Logged in
invalid	correctpass	Invalid email.	Invalid email.
deleted@gmail.com	correctpass	Account Deleted.	Account Deleted.
new@gmail.com	newpass	User not found.	User not found.
valid@gmail.com	incorrectpass	Wrong Password.	Wrong Password.
"" (Empty)	correctpass	Please enter an email.	Please enter an email.
valid@gmail.com	"" (Empty)	Please enter a password.	Please enter a password
"" (Empty)	"" (Empty)	Please fill in both email and password.	Please fill in both email and password.

3. Favourites

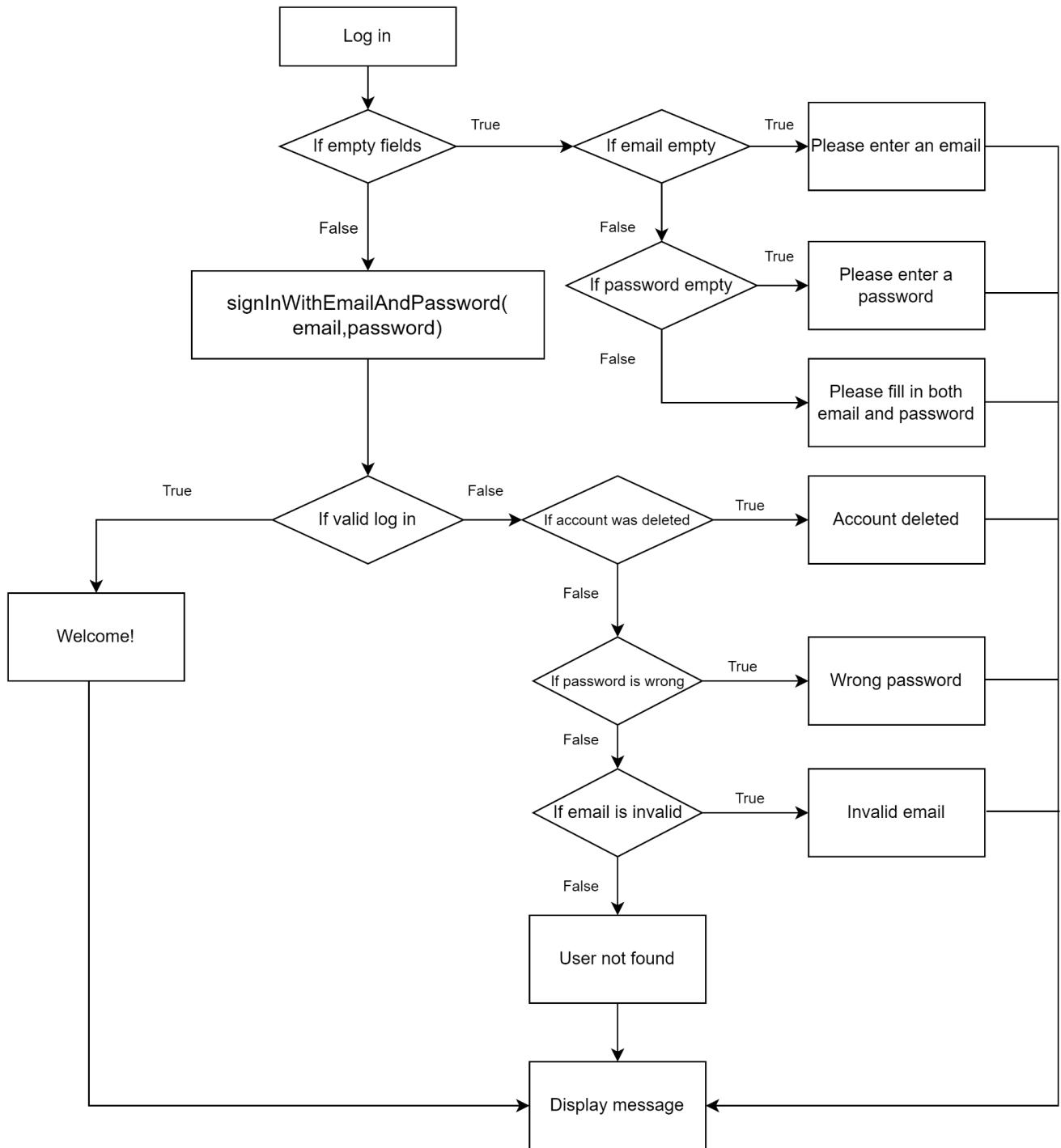
Test Id	Test	Expected Outcome	Actual Outcome
1	Favourite a club/league	The system saves the team/league into the list of favourites	The system saves the team/league into the list of favourites
2	Remove a club/league from the list of favourites	The system removes the club/league from the list of favourites	The system removes the team/league from the list of favourites

7.2 White box testing

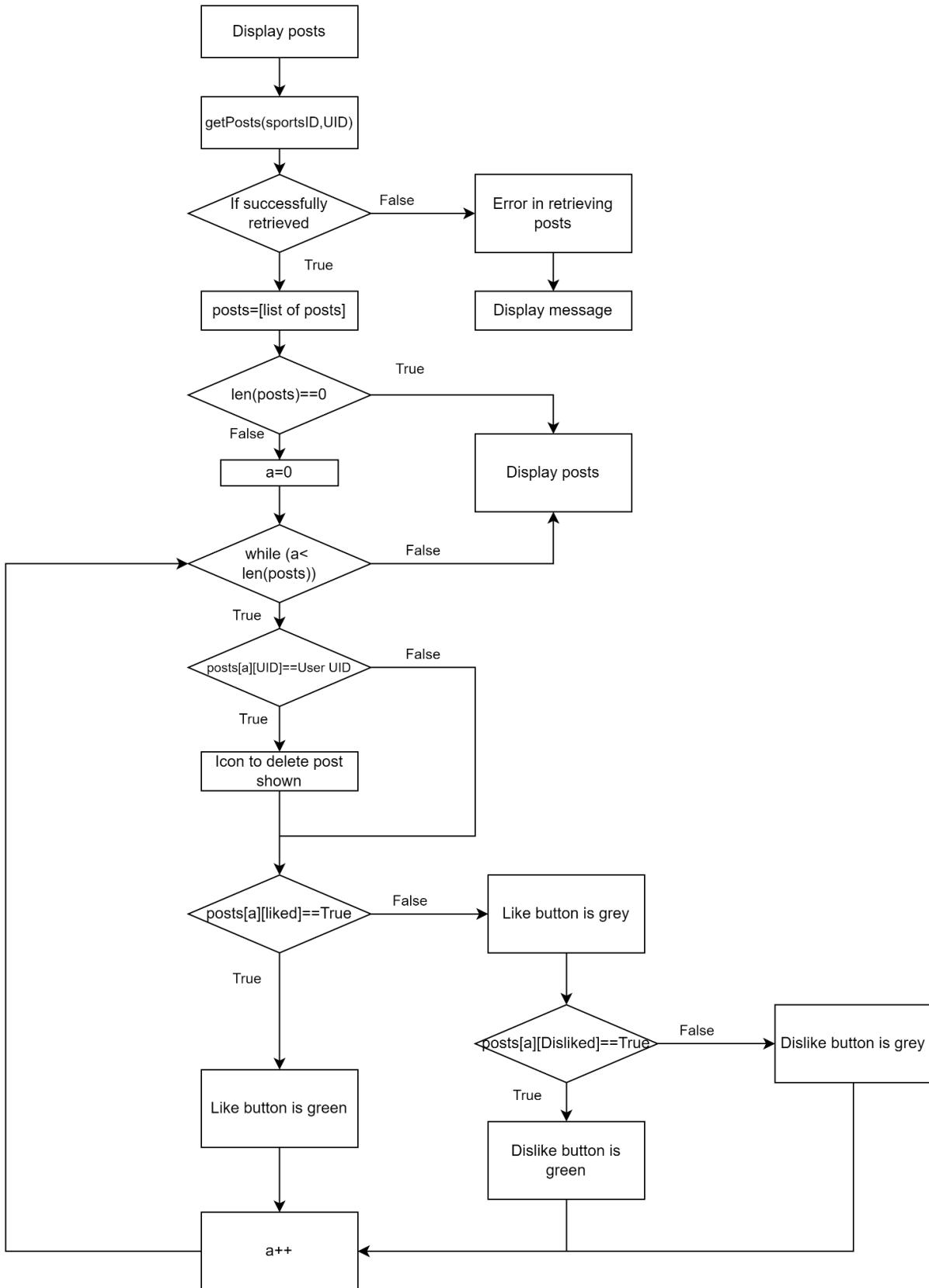
1. Account Creation



2. User logs in



3. Display posts



8 Appendix

Please refer to the Youtube link below for more information and detailed demonstration of our SupaSports app:

[SupaSports Demonstration](#)