SW Engineering CSC 648/848 Summer 2022

YourSports

Dynamic Debugger (Team 02)

Team & Backend Lead

Kshitiz Sareen: ksareen@sfsu.edu

Frontend

Lead Kevin Islas

Developer Shamar Ireland

Functional

Lead Sabur Saigani

Assistant Jonathan Ip

Database Administrator

Wenye Guo

Github Master

Mathew O Abiola

Milestone 2

Milestone	Date
M2V2	July 14th, 2022
M2V1	July 7th, 2022
M1V2	June 30th, 2022
M1V1	June 9th, 2022

Table of Contents

Data Definition

Users

Users will be referring to the group of people who use our application. Users will be given wide access to features like searching news, access to live Twitter feeds, following live games, posting opinions in the discussion forum of a game, and viewing player statistics.

There will be two types of users in our application:

Regular User

These are the users who have not logged in. Regular users will only be able to watch scores of live games and search for news and tweets.

Regular Users will have the following attributes:

Tracking ID

> This refers to a unique id given to the unregistered user.

IP address

> This refers to the IP address of the device that the user uses to access the website.

Registered User

These are users who have an account. Registered users will have access to all features our application provides. Registered users inherit from regular users.

Registered users will have the following attributes:

User

> This refers to the tracking ID of the user.

Account

> This refers to the account of the user.

Account

Each registered user will have an account that contains all the details and keys that are required to identify a user.

There will be three types of accounts in our application:

Basic Account

A Basic account will have access to features like searching news, access to live Twitter feeds, following live games, posting opinions in the discussion forum of a game, and viewing player statistics.

The Basic Account has the following attributes:

Name

>This refers to the name that the user used to sign up, and it is usually the legal name of the user.

Username

> This refers to the username that the user used to sign up for, and it will be used to distinguish a user from other users.

Email

> This refers to the email that the user used to sign up. The email will be used to associate their account with their email, and also serve as a communication method between the user and the application.

Password

> This refers to the set of characters that a user used to sign up, and it will be used to authenticate a user along with their email.

Creator Account

A Creator Account will have access to all the privileges and properties that a basic account has with the additional privilege of publishing articles to the website. Every user that has a creator account will have a basic account. A creator account will inherit from a basic account.

The creator account will have the following properties:

Account

> This refers to the account of the user.

Admin Account

An Admin account is used to regulate and ensure proper usage of the application by other accounts. An Admin account has all privileges in the application along with access to the database and backend service to manipulate, change, add and remove data. There will only be one admin account. The admin account will also inherit from the basic account.

The admin account has the following properties:

Account

> This refers to the account of the user.

News

News will refer to the information that users can access from our application.

Our application will provide different types of news:

Tweets

These are the tweets collected from Twitter according to users' preferences and interests.

Tweets will have the following properties:

Username

> This is the username of the user who posted the tweet. The username will follow twitter data conventions.

Content

> This is the content of the tweet.

Articles

These are the articles posted by users who have a creator account. Articles can be searched and filtered by date posted and which sport is it used to talk about.

Articles have the following properties:

Post Time

> This is the date and time the article was posted.

Image

> This is the image that will be used as the thumbnail for the article. The acceptable image formats are png and jpeg, with a maximum size of 5 MB.

Heading

> This is the title of the article.

Subheading

> This is the subheading of the article.

Introduction

> This is the introduction part of the article.

Content

> This is the main content that contains all the detailed information of the article.

Conclusion

> This is the conclusion of the article.

Author

> This is the Author of the article.

Sport

> This is the type of sport which the article targets. E.g. Basketball and Baseball.

Comment

Registered users will be able to post comments on articles to express their views and opinions on the information given in the article.

Comment will have the following properties:

Article

> This is the Article for which the comment was posted.

Author

> This is the user who has posted the comment.

Post Time

> This is the date and time where the comment was posted.

Content

> This contains all the information that a user wants to convey.

Games

This refers to the past, current, and future sports games that users can follow. Users can follow the scores, and post any opinions they have about the game.

Games will have the following properties:

Player One Score

> This is the score of the home team.

Player Two Score

> This is the score of the opponent team.

Game Location

> This is the location of the game.

Team One

> This is the home team.

Team Two

> This is the away team.

Group

This refers to all the groups like group chats and discussion forums that allow users to interact with other users.

There will be different types of groups:

Discussion Forum

The discussion forum will refer to the forums of every sports game where users can post their opinions about the game. This discussion forum will have no owner and every registered user who follows a game will be able to join the discussion forum.

Discussion forums will have the following properties:

Forum Name

> This is the name of the forum.

Game

> This is the sports game for which the forum has been created.

Account List

> This is the list of all the accounts of the users that are part of the forum.

Group Chat

The group chat will refer to private groups of sports fans who share similar interests. The private group chats will have an owner and every registered user will only be able to join a private group chat upon the owner's approval.

Group chats will have the following properties:

Group Chat Name

> This is the name of the group chat.

Group Chat Owner

> This is the user who owns the group.

Group Account

> This is the list of all the accounts of the users that are part of the group chat.

Message

This refers to the message a user posts in a group. All communication between users of the application will happen between the users.

Message has the following properties:

Account

> This is the account of the user who has posted the message.

Content

> This is the content of the message.

Group

> This is the group in which the user has posted a message.

Dashboard

This refers to the Dashboard of every user that has a creator account.

Dashboard has the following properties:

Article List

> This is the list of all the articles that the creator has uploaded.

Content

> This is the number of views for each article.

Creator

> This is the creator account of the user who owns the dashboard.

Backend Service

The backend service will be used frequently to refer to all the operations that will happen on the server-side like filtering and searching for articles, filtering and retrieval of tweets, fetching live updates from games, and handling messages from users, updating and retrieving player statistics.

Website

The Website will be used frequently to refer to all the operations that will happen on the client-side like signing up and logging in, and verification and validation of data.

Database

The database will be used frequently to refer to all the operations that involve the storage of data. It will be used to store the news, account information of users, information on sports games, information on players, and all the messages that a user posts in a group.

Prioritize Functional Requirements

Priority 1

Regular user

1. Regular users shall be able to create an account.
2. Regular users shall be able to sign up as a creator account.
3. Regular users shall be able to access the homepage.
4. Regular users shall be able to view the live scores of a game.
5. Regular users shall be able to view the duration of a game.
6. Regular users shall be able to view updated statistics of players.
7. Regular users shall be able to search for articles through a search bar.
8. Regular users shall be able to filter articles they would like to read by keywords.
9. Regular users shall be able to filter articles they would like to read by different types of sports.
10. Regular users shall be able to search articles by heading of the article.
11. Regular users shall be able to view the article heading in the search results for articles.
12. Regular users shall be able to view the article author in the search results for articles.
13. Regular users shall be able to view the article postdate in the search results for articles.
14. Regular users shall be able to view the article image as a thumbnail in the search results for articles.
15. Regular users shall be able to search games by the teams playing in the search bar.
16. Regular users shall be able to filter games by the sport type.
17. Regular users shall be able to filter games by their date.
18. Regular users shall be able to view the game score in the search results for games.
19. Regular users shall be able to view the game date in the search results for games.
20. Regular users shall be able to view the game location in the search results for games.

21. Regular users shall be able to search for player statistics according to the player name in the search bar.
22. Regular users shall be able to filter player statistics by the sport type.

Registered user

23. Registered users shall have the same privileges as regular users.
24. Registered users shall be able to log in.
25. Registered users shall be able to log out.
26. Registered users shall be able to reset their password.
27. Registered users shall be able to delete their account.
28. Registered users shall be able to post in the discussion forum of the game.
29. Registered users shall be able to delete their posts in the discussion forum.

Basic Account

30. Accounts shall be able to be deleted by users.
31. Accounts shall contain the name of the users.
32. Accounts shall contain the username of the users.
33. Accounts shall contain the email of the users.
34. Accounts shall contain the password of the users.

Creator Account

35. Creators shall have the same privileges as registered users.
36. Creators shall be able to upload articles to the website.
37. Creators shall be able to view Dashboard.

Admin Account

38. Admin shall have all the privileges.
39. Admin shall be able to create discussion forums for upcoming games.

Articles

40. Articles shall be posted by creator users.
41. Articles shall be able to be filtered by date.
42. Articles shall be able to be filtered by sports type.
43. Articles shall be able to be filtered by keywords specified by the user.
44. Articles shall contain a heading.
45. Articles shall contain a subheading.
46. Articles shall contain an introduction.
47. Articles shall contain contents.
48. Articles shall contain a conclusion.

49. Articles shall contain an image.

Games

50. Games shall be filtered by date.
51. Games shall be filtered by sport type.
52. Games shall be filtered by keywords specified by users in the search bar.
53. Games shall have an individual discussion forum.
54. Games shall be able to show live scores of corresponding teams.
55. Games shall be able to show the date of the game.
56. Games shall be able to show the location of the game.

Discussion Forum

57. Discussion forum shall be able to be viewed by all users.
58. Discussion forum shall be able to show posts from users.

Dashboard

59. Dashboard shall be able to show the number of views of the article.
60. Dashboard shall be able to show the heading of the article.
61. Dashboard shall be able to show the posted date of the article.

Backend Service

62. The backend service shall be able to validate the email for duplicates when users try to sign up.
63. The backend service shall be able to check for duplicate accounts when users try to sign up.
64. The backend service shall be able to create an account in the database.
65. The backend service shall be able to store posts by a user in the discussion forums.
66. The backend service shall be able to update player statistics in the database.
67. The backend service shall be able to filter articles specified by the user.
68. The backend service shall be able to filter games specified by the user.
69. The backend service shall be able to filter player statistics specified by the user.
70. The backend service shall be able to fetch updates from a game every 5 seconds.

Website

71. The website shall be able to check if the first name is not empty users try to sign up.

72. The website shall be able to check if the last name is not empty when user try to sign up.
73. The website shall be able to check if the email is not empty when users try to sign up.
74. The website shall be able to check if the password is at least 8 characters before the users try to sign up.
75. The website shall be able to check if date of birth is at least 13 years old when users try to sign up.
76. The website shall be able to check if the terms of service has been accepted when users try to sign up.

Priority 2

Regular user

1. Regular users shall be able to be redirected to a live broadcast of the game.
2. Regular users shall be able to search for specific comments in the discussion forum.

Registered user

3. Registered users shall be able to change their profile picture.
4. Registered users shall be able to view profiles of other users.
5. Registered users shall be able to invite their friends to use the YourSports website.
6. Registered users shall be able to red flag other registered users for any posts in the discussion forum.
7. Registered users shall be able to follow sports games.
8. Registered users shall be able to unfollow games they are currently following.
9. Registered users shall be able to search for specific messages in the discussion forum.
10. Registered users shall be able to mark posts as read so that they don't receive notifications.
11. Registered users shall be able to comment on articles.
12. Registered users shall be able to mark as read on articles.
13. Registered users shall be able to receive responses to their posts in the discussion forum.
14. Registered users shall be able to create a group chat.
15. Registered users shall be able to send messages in the group chat.
16. Registered users shall be able to receive messages in the group chat.

17. Registered users shall be able to view messages in the group chat.
18. Registered users shall be able to see other online users in the group chat.
19. Registered users shall be able to delete messages they previously sent in the group chat.
20. Registered users shall be able to edit messages they previously sent in the group chat.

Creator Account

21. Creators shall be able to select their target audience.

Group chat

22. Group chat shall contain multiple group members.
23. Group chat shall have one group owner.
24. Joining a specific group chat shall require permission of the group owner.
25. Group members of a group chat shall be able to post messages.

Message

26. Messages in the group chat shall be posted by registered users.
27. Messages in the group chat shall be able to be deleted by poster.
28. Messages in the group chat shall be able to be marked using a red flag.
29. Messages in the group chat shall be responded to by other registered users.

Dashboard

30. Dashboard shall be able to show the comments on the article.
31. Dashboard shall be able to show the redflags of an article.

Backend Service

32. The backend service shall be able to send a verification email for user to verify their account.

Comment

33. Comments shall be able to be viewed by regular users.
34. Comments shall be posted by registered users.
35. Comments shall be able to be deleted by poster.

Discussion forum

36. Discussion forum shall be sorted with specific words.

Priority 3

Regular user

1. Regular users shall be able to read live twitter feeds.
2. Regular users shall be able to switch between light and dark mode.

Registered user

3. Registered users shall be able to bet on live and upcoming games.
4. Registered users shall receive 100 YourSports coins when signing up.
5. Registered users shall get cash by selling their YourSports coins.
6. Registered users shall win YourSports coins if they win a bet.
7. Registered users shall lose YourSports coins if they lose a bet.
8. Registered users shall be able to buy YourSports coin using actual money.
9. Registered users shall be able to view their current amount of YourSports coins in their account.
10. Registered users shall be able to enable/disable receiving emails about sports news.
11. Registered users shall be able to choose how frequently they receive emails about sports news.
12. Registered users shall be able to read live Twitter feeds.
13. Registered users shall be able to type the topics they would like to follow on Twitter.
14. Registered users shall be able to unfollow topics they are currently following.
15. Registered users shall be able to receive emails of any new tweets.
16. Registered users shall be able to enable receiving notifications by email.
17. Registered users shall be able to choose if they want to enable receiving sms that contain sports news.
18. Registered users shall be able to enable receiving notifications by messages on phone.

Group Owner

19. The Group's Owner shall be able to change the group chat name.
20. The Group's Owner shall be able to invite other users to the group chat.
21. The Group's Owner shall be able to block any users in the group chat.

Message

22. Messages shall be able to be edited by poster.

Tweets

23. Tweets shall be pushed to users.

24. Different topics of tweets shall be pushed to users.

Notification

25. Notification shall be sent by email.
26. Notification shall include new tweets or/and sport news.
27. Notifications shall be sent by messages.

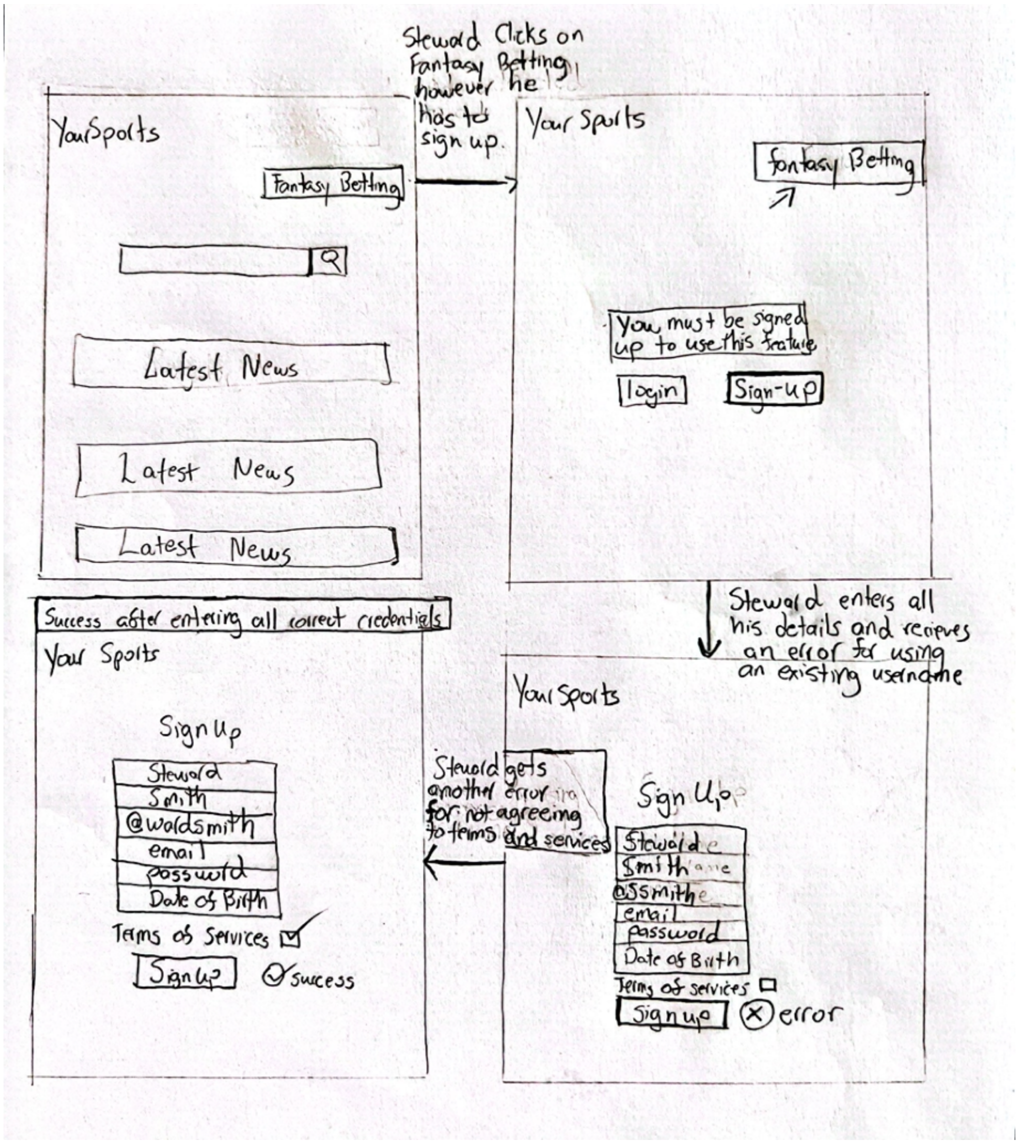
Backend service

28. The backend service shall be able to filter tweets by the topics a user follows.

UI Mockup and Storyboards

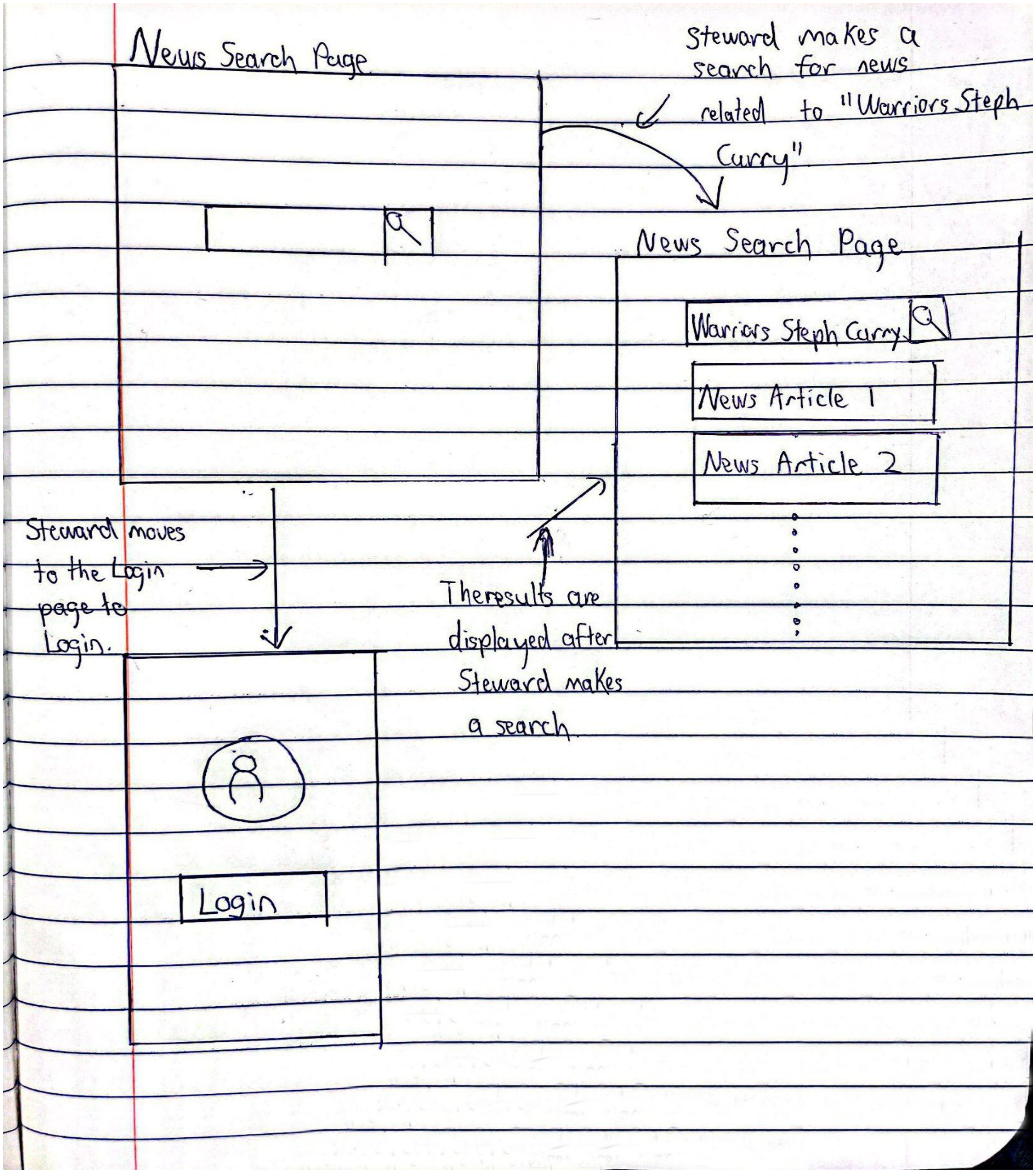
Signing Up

Steward wants to participate in fantasy betting however he learns that he has to be signed up. He proceeds to sign up by entering all his credentials. He thinks he's in the clear but when he clicks the sign up button at the bottom of the form he gets an error. This is so because he forgot to agree to the terms and conditions and used a username that already existed. He attempts to do this again and he is successful once he enters a username that's not already taken and agrees to the terms and conditions.


Steward Clicks on Fantasy Betting however he has to sign up.
YourSports
Fantasy Betting
Latest News
Latest News
Latest News
Your Sports
Fantasy Betting
You must be signed up to use this feature
login
Sign-up
Steward enters all his details and recieves an error for using an existing username
Your Sports
Sign Up
Steward
Smith
@SSmithe
email
password
Date of Birth
Terms of services
Sign up
error
Steward gets another error for not agreeing to terms and services
Success after entering all correct credentials
Your Sports
Sign Up
Steward
Smith
@worldsmith
email
password
Date of Birth
Terms of Services
Sign Up
Success

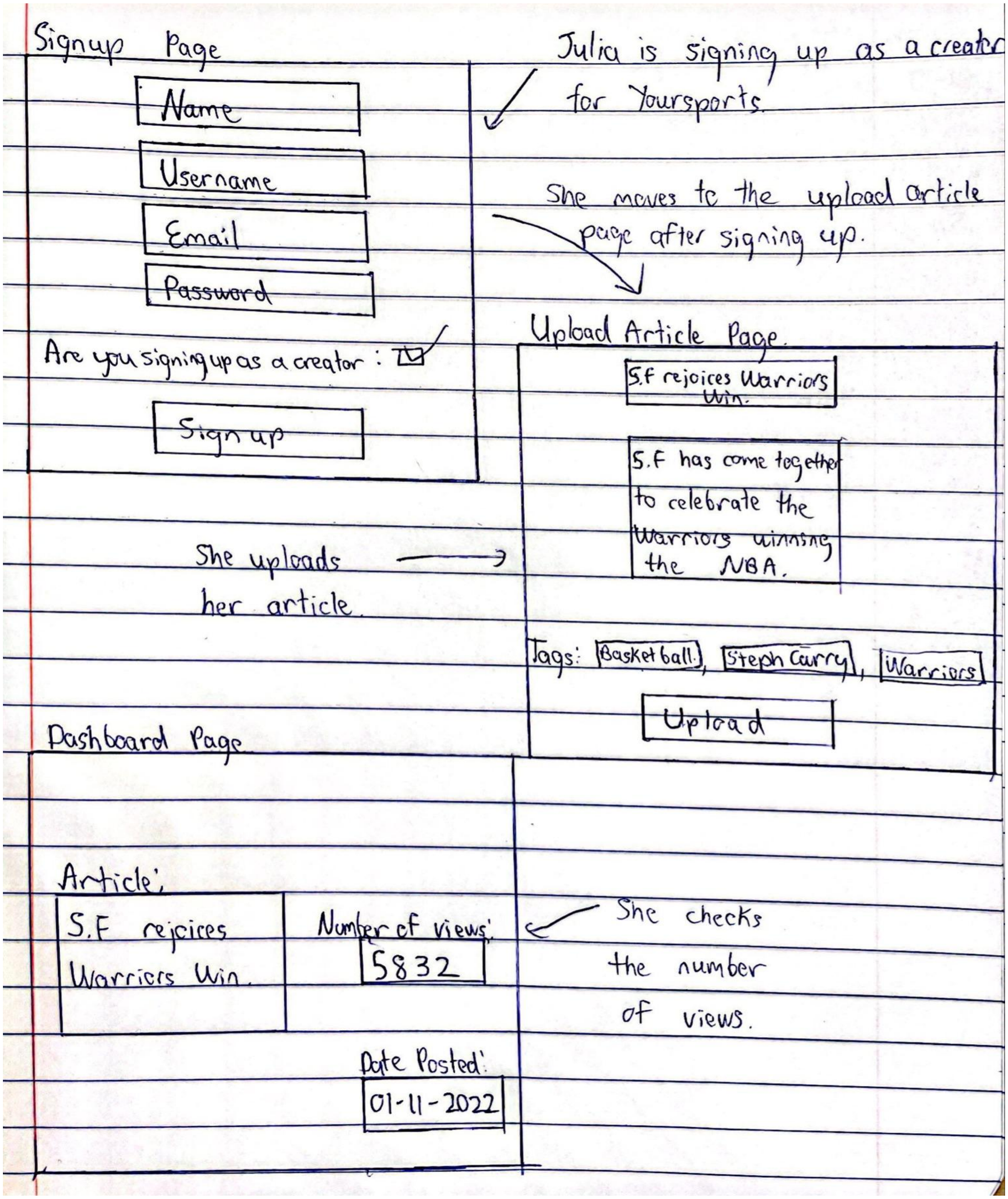
Checking Sports news

Steward makes a search for news related to “Warriors Steph Curry”. The results for all the news related to “Warriors Steph Curry” are displayed on the news search page.


News Search Page
Steward makes a search for news related to "Warriors Steph Curry"
News Search Page
Warriors Steph Curry
News Article 1
News Article 2
Steward moves to the Login page to Login.
Theresults are displayed after Steward makes a search.
Login

Posting news articles

Julia is signed up as a creator for YourSports. After signing up she uploads her article. While uploading her article, she sets her audience tags as Basketball, Steph Curry, Warriors. She uploads her article. After a few days, she checks the number of views on her article.



Julia is signing up as a creator for Yoursports.

She moves to the upload article page after signing up.

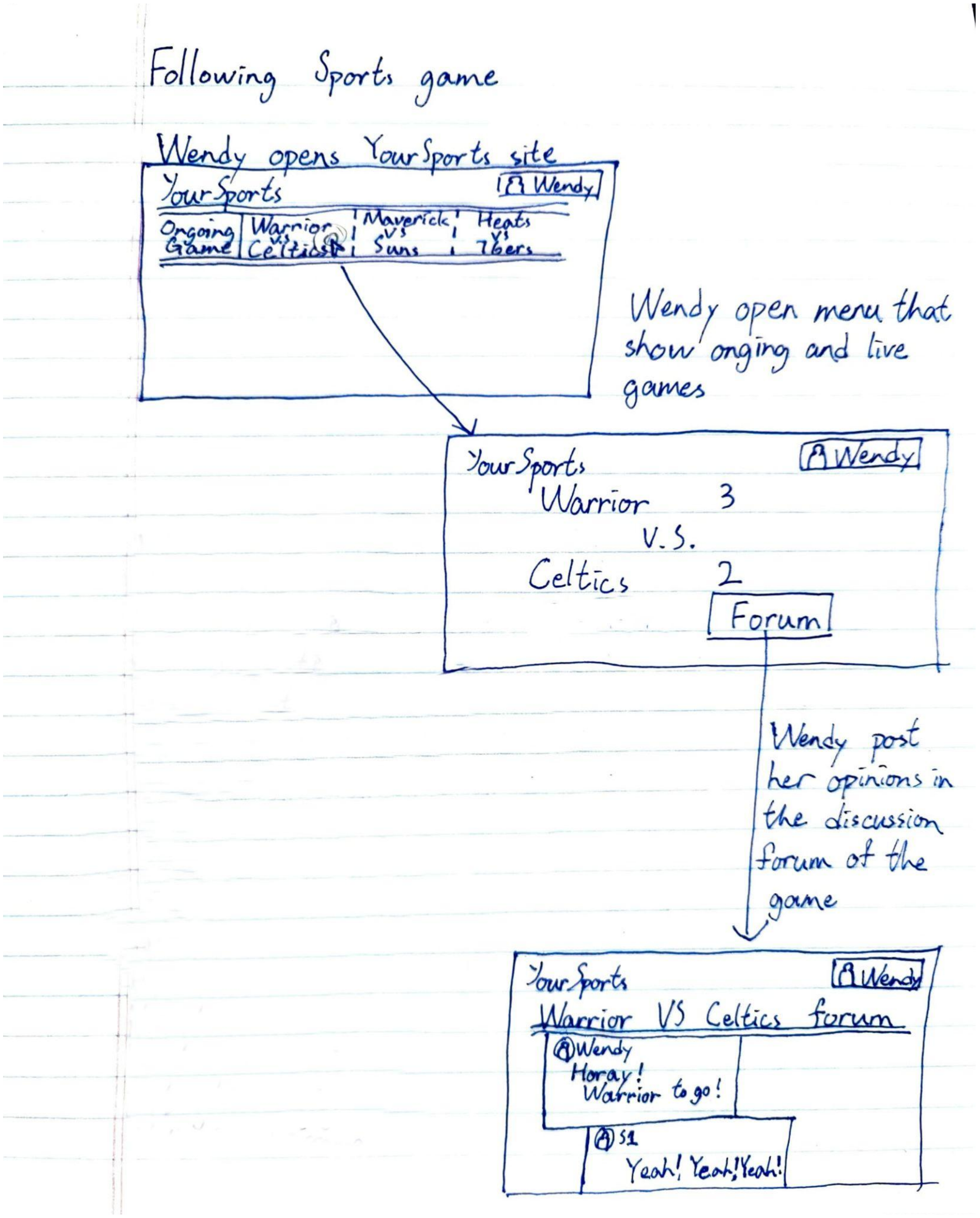
She uploads her article.

She checks the number of views.

Signup Page
Name
Username
Email
Password
Are you signing up as a creator : ☑
Sign up
Upload Article Page.
S.F rejoices Warriors Win.
S.F has come together to celebrate the Warriors winning the NBA.
Tags: Basket ball., Steph Curry, Warriors
Upload
Dashboard Page
Article:
S.F rejoices Warriors Win.
Number of views:
5832
Date Posted:
01-11-2022

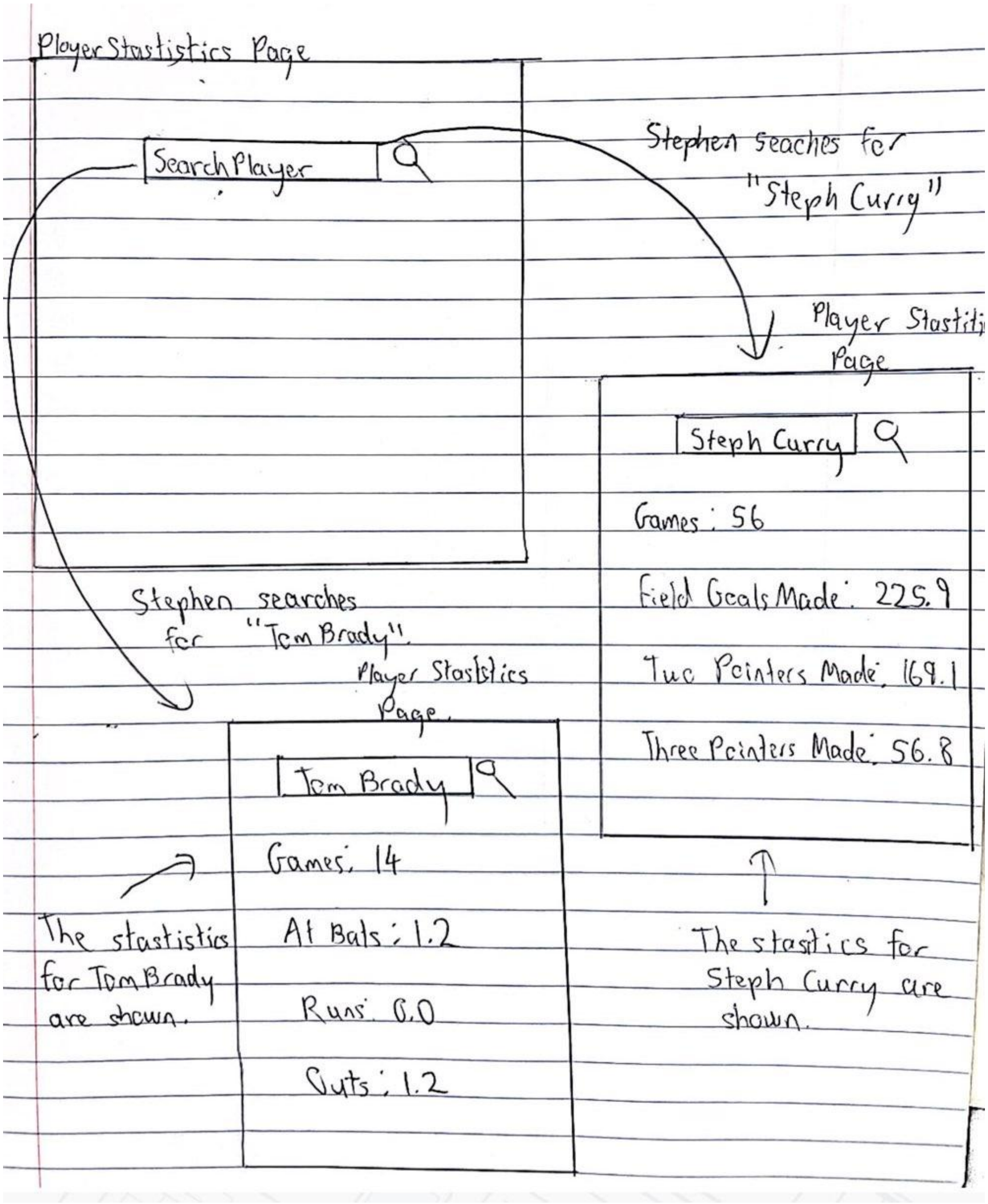
Following Sport Game

Wendy, a registered user, wants to be more involved with the Warrior vs Celtics game, so she comes to the YourSports site and opens the game's page. While she is on the site, she also went on to post her opinions on the game's forum where everyone else is also posting on.



Checking Player Statistics

John wants to check the updated statistics of Steph Curry and Tom Brady. He visits the player statistics page in the YourSports website, He first searches for Steph Curry, and sees the statistics like games Played, field goals made, two-pointers made, and three-pointers made. He then searches for Tom Brady, and views his statistics.


Player Stastistics Page
Search Player
Stephen seaches for "Steph Curry"
Player Stastiti Page
Steph Curry
Games : 56
Field Goals Made : 225.9
Two Pointers Made : 169.1
Three Pointers Made : 56.8
The stasitics for Steph Curry are shown.
Stephen searches for "Tom Brady".
Player Stastistics Page.
Tom Brady
Games: 14
At Bats : 1.2
Runs: 0.0
Outs : 1.2
The stastistics for Tom Brady are shown.

Database Architecture and Organization

Database Organization

Regular User

1. Regular users shall be able to view the live scores of a game.
2. Regular users shall be able to view the duration of a game.
3. Regular users shall be able to search games by the teams playing in the search bar.
4. Regular users shall be able to filter games by the sport type.
5. Regular users shall be able to filter games by their date.
6. Regular users shall be able to search for specific comments in the discussion forum.

Registered user

1. Registered users shall have the same privileges as regular users.
2. Registered users shall be able to post in the discussion forum of the game.
3. Registered users shall be able to delete their posts in the discussion forum.

Games

1. Games shall have an individual discussion forum.
2. Games shall be able to show live scores.
3. Games shall be able to show the date of the game.
4. Games shall be able to show the location of the game.

Discussion Forum

1. Discussion forum shall be able to be viewed by all users.
2. Discussion forum shall be able to show posts from users.

Admin Account

1. Admin shall be able to create discussion forums for upcoming games.

Business Rules

1. A regular user shall be or not be a registered user.
2. A registered user shall be able to use many devices to login the account.

3. A registered user shall be able to post many posts in game discussion forums.
4. A discussion forum shall contain many posts.
5. A post shall be able to be replied to many times.
6. An account shall be either a creator or an admin.
7. A game shall have at most one discussion forum under the games page.
8. An admin shall be able to manage many discussion forums.
9. An admin shall be able to manage many games.

Entities

Regular User (Strong)

Tracking_id (INT, PK, Not null, AI)

IP_Address (INT, Not null)

Account (weak)

Account_ID (INT, PK, Not null, AI)

Name (VARCHAR,Not null)

Username (VARCHAR,Unique,Not null)

Email (VARCHAR, Unique,Not null)

Password (VARCHAR, Not null)

Admin (weak)

Admin_ID (INT, PK, Not null, AI)

Account_id (INT, Not null, Unique, FK to Account(Account_ID))

Registered user (Weak)

Tracking_id (INT, Not null, FK to Regular User(Tracking_id))

Account_id (INT, Not null, Unique, FK to Account(Account_ID))

Game (Strong)

Game_ID (INT, PK, Not null, AI)

Game_Home_Team (VARCHAR, Not null)

Game_Away_Team (VARCHAR, Not null)

Game_Location (VARCHAR, Not null)

Home_Team_Score (INT, Not null)

Away_Team_Score (INT, Not null)

Duration (INT, Not null)

Discussion forum (Weak)

Forum_id (INT, PK, Unique, Not null, AI)

Game_id (INT, Unique, Not null, FK to Game(Game_ID))

Forum_Name (VARCHAR, Not null)

Posts (Weak)

Post_id (INT, PK, Unique, Not null, AI)

account_ID (INT, Not null, FK to Account(Account_ID))

Content (VARCHAR, Not null)

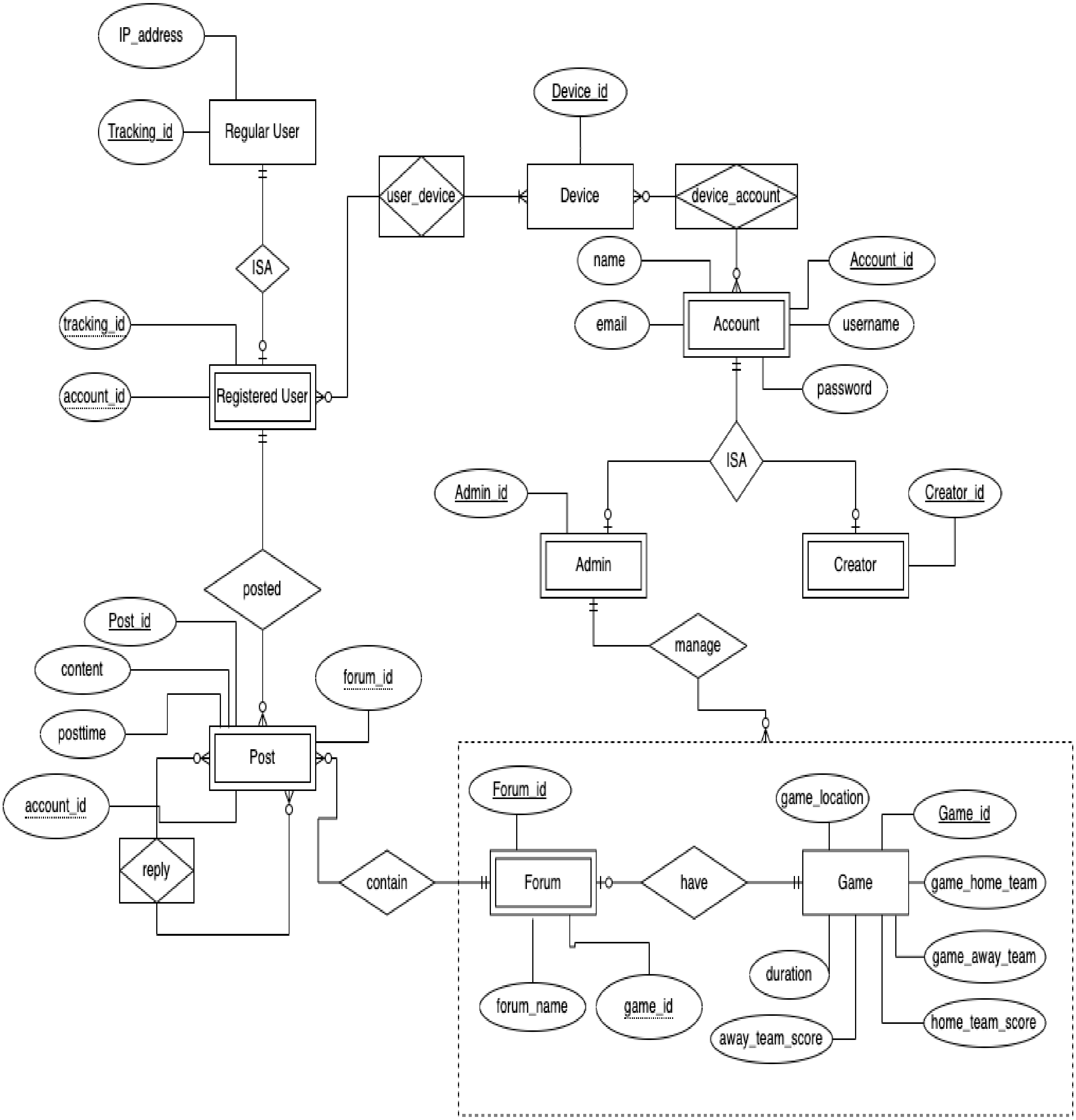
forum _id (INT, Not null, FK to Discussion Forum(Forum_id))

Post Time (DATE, Not null)

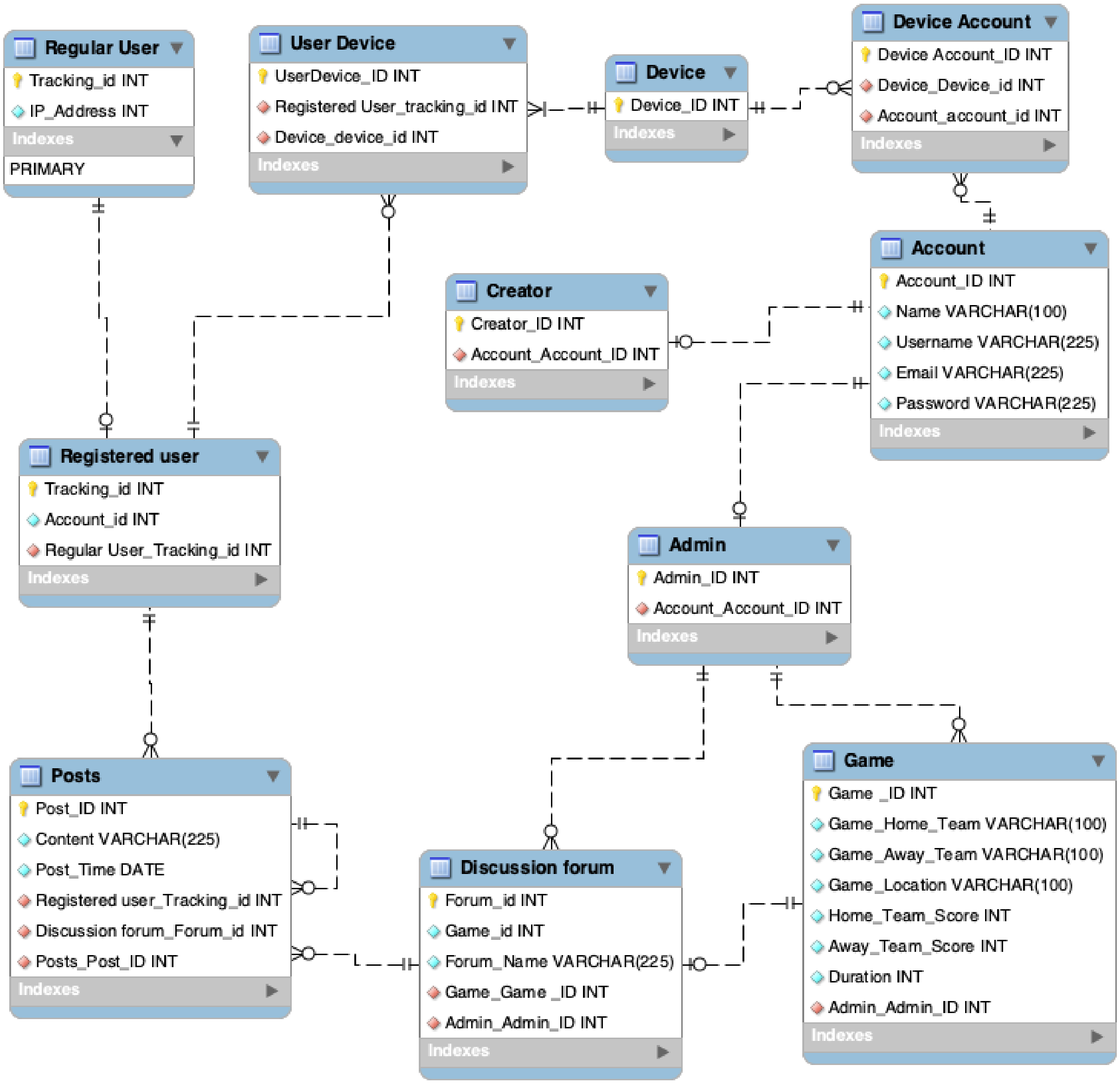
Device (Strong)

Device_ID (INT, PK, Not null)

Entity Relationship Diagram (ERD)



Database Model



Database that will be used

We will use MySQL database as it supports relational databases, and our database administrator is proficient with MySQL.

Media Storage

Our application will only deal with images, and text formats of data. The images will be stored in Google Cloud Storage. Each image will only be in a JPEG or PNG file format, with a maximum size of 5MB. The only entity that will require images are the Articles. Each record in the Article table will point to the URL of the image that is stored in the Google Cloud Storage bucket of our project.

Search architecture and implementation

Our search architecture will be implemented for searching articles.

To search our articles, we will use three constraints:

1. Search text specified in the search bar

If a search text is specified, the articles will be filtered according to the text provided in the search bar. To filter articles according to search text, we will use the like operator in our MySQL database to find articles that have a heading that have the same pattern specified as the text that is provided in the search bar.

```
SELECT image_URL,heading,subHeading,posttime,name as `Author`,sport FROM
(SELECT image_URL,heading,subHeading,fk_authorid,posttime,sport FROM Article
WHERE (heading LIKE CONCAT('%',inSearchText,'%') OR inSearchText is NULL) ) AS
Articles JOIN RegisteredUser ON user_id = Articles.fk_authorid ORDER BY
posttime DESC;
```

2. Date filter provided in the filter panel

If we only filter articles by date, we will return all the articles that have been posted after and including the date that has been specified in the filter.

```
SELECT image_URL,heading,subHeading,posttime,name as `Author`,sport FROM
(SELECT image_URL,heading,subHeading,fk_authorid,posttime,sport FROM Article
WHERE (posttime >= inPostdate OR inPostdate is NULL)) AS Articles JOIN
RegisteredUser ON user_id = Articles.fk_authorid ORDER BY posttime DESC;
```

3. Sport type filter provided in the filter panel

If we only filter articles via the type of sport, we will return all the articles that talk about the sport specified in the filter.

```
SELECT image_URL,heading,subHeading,posttime,name as `Author`,sport FROM
(SELECT image_URL,heading,subHeading,fk_authorid,posttime,sport FROM Article
WHERE (sport = inSport OR inSport is NULL )) AS Articles JOIN RegisteredUser ON
user_id = Articles.fk_authorid ORDER BY posttime DESC;
```

Combining the conditions for all the three filters will result in a search query like the one below.

```
SELECT image_URL,heading,subHeading,posttime,name as `Author`,sport FROM
(SELECT image_URL,heading,subHeading,fk_authorid,posttime,sport FROM Article
WHERE (posttime >= inPostdate OR inPostdate is NULL) AND (sport = inSport OR
inSport is NULL ) AND (heading LIKE CONCAT('%',inSearchText,'%') OR
inSearchText is NULL)) AS Articles JOIN RegisteredUser ON user_id =
Articles.fk_authorid ORDER BY posttime DESC;
```

If we don't specify any filters, then all articles in the database will be returned.

All articles returned from the backend service to the user will be organized according to the date they were posted in descending order. This basically means that users will be able to view the latest articles on the website.

The properties returned from the database for each article are the:

image_URL

> This is the thumbnail image of the article.

heading

> This is the title of the article.

subHeading

> This is the subHeading of the article.

posttime

> This is the date and time the article was posted.

sport

> This is the sport that the article is providing information for.

Author

> This is the name of the author who created the article.

High-Level API and Main Algorithm

YourSports API

Our Backend service will be hosted on this URL: http://34.136.124.189. The Backend service will provide access to the YourSports API. The YourSports API will have multiple endpoints with the following functions:

1. http://34.136.124.189:8080/api/account/signup:
    a. This function will accept the details of a user and sign them up if all details are valid.
    b. This is a POST request.
2. http://34.136.124.189:8080/api/account/login:
    a. This function will accept credentials from the user for authentication. If all the credentials are correct, the user will be logged in.
    b. This is a POST request.
3. http://34.136.124.189:8080/api/account/updateuserdetails:
    a. This function will update the details of the user like username, email, password, as well as their notification preferences.
    b. This is a POST request.
4. http://34.136.124.189:8080/api/account/sendemail:
    a. This function will send an email to the user-specified.
    b. This is a POST request.
5. http://34.136.124.189:8080/api/searchnews/search:
    a. This function will scan through the database to search for articles that cater to the user's search query. The results will be returned to the website so that the articles can be read by a user.
    b. This is a GET request.
6. http://34.136.124.189:8080/api/searchnews/markasread:
    a. This function will mark an article provided by the user as read.

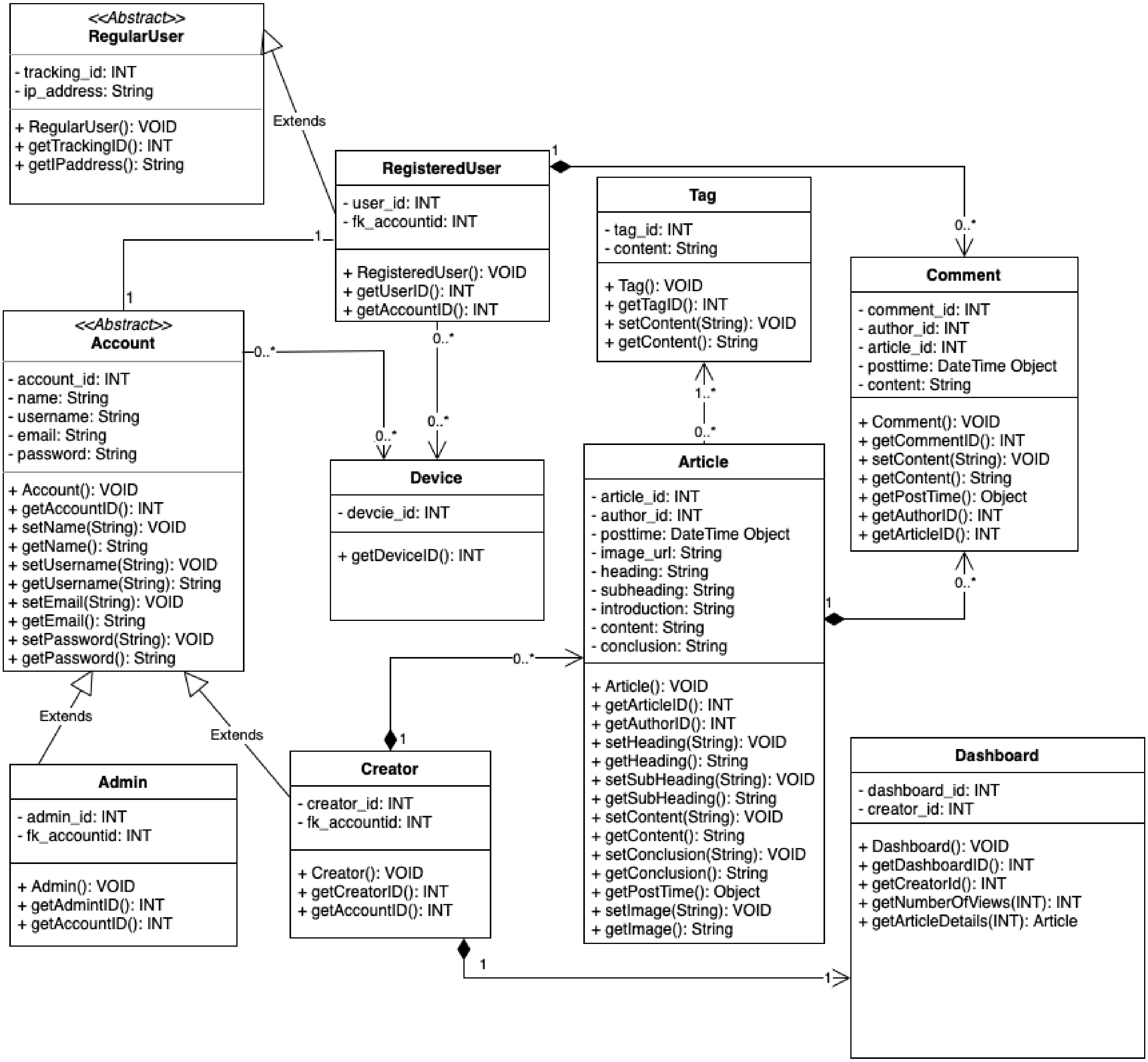
   b. This is a POST request.

7. http://34.136.124.189:8080/api/searchtweets/search:
   a. This function will use the Twitter API to scan through specific sports Twitter accounts, which will find tweets that are relevant to the user's search query, and return the results to the website being accessed by the user.
   b. This is a GET request.
8. http://34.136.124.189:8080/api/searchstatistics/search:
   a. This function will scan the database, to find the player statistics requested by the user. Once the record is found, it will return the results to the website to display to the user.
   b. This is a GET request.
9. http://34.136.124.189:8080/api/searchgames/search:
   a. This function will take a search query from a user through the website, scan through the database, of any past, future, and current sports games, and retrieve necessary details, like the teams playing and the venue of the game.
   b. This is a GET request.
10. http://34.136.124.189:8080/api/articles/publish:
    a. This function will accept a set of parameters that will form an article from the user through the website, and store the new article in the database.
    b. This is a POST request.
11. http://34.136.124.189:8080/api/articles/setaudience:
    a. This function will allow authors to select their target audience for respective articles.
    b. This is a POST request.
12. http://34.136.124.189:8080/api/comments/post:
    a. This function will accept content from a user through the website and create a new comment made by the user for the article, that the user wants to post a comment for.

    b. This is a POST request.

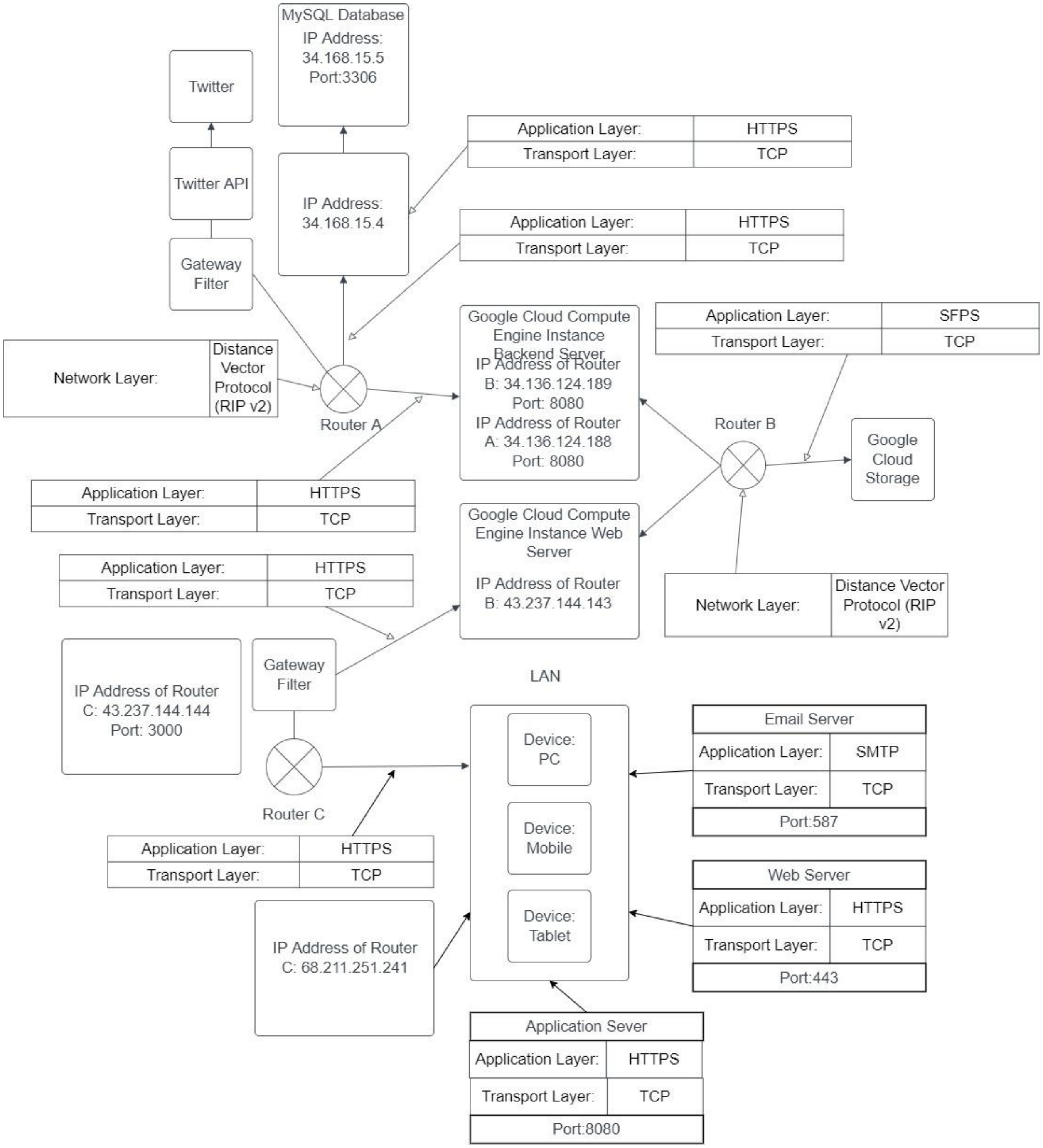
13. http://34.136.124.189:8080/api/games/getupdates:
    a. This function will run at repeated intervals to fetch live scores of a sports game so that users can have real-time access to the scores of a game.
    b. This is a GET request.
14. http://34.136.124.189:8080/api/games/getposts:
    a. This function will be run at repeated intervals, to fetch all the posts in the discussion forum of a game.
    b. This is a GET request.
15. http://34.136.124.189:8080/api/games/createpost:
    a. This function will accept the content from the user, and create a post in the database that will be reflected on the website being accessed by the user.
    b. This is a POST request.
16. http://34.136.124.189:8080/api/games/redflag:
    a. This function will red flag a post specified by the user.
    b. This is a POST request.
17. http://34.136.124.189:8080/api/games/markasread:
    a. This function will mark a post in the discussion forum as read.
    b. This is a POST request.

UML Diagram

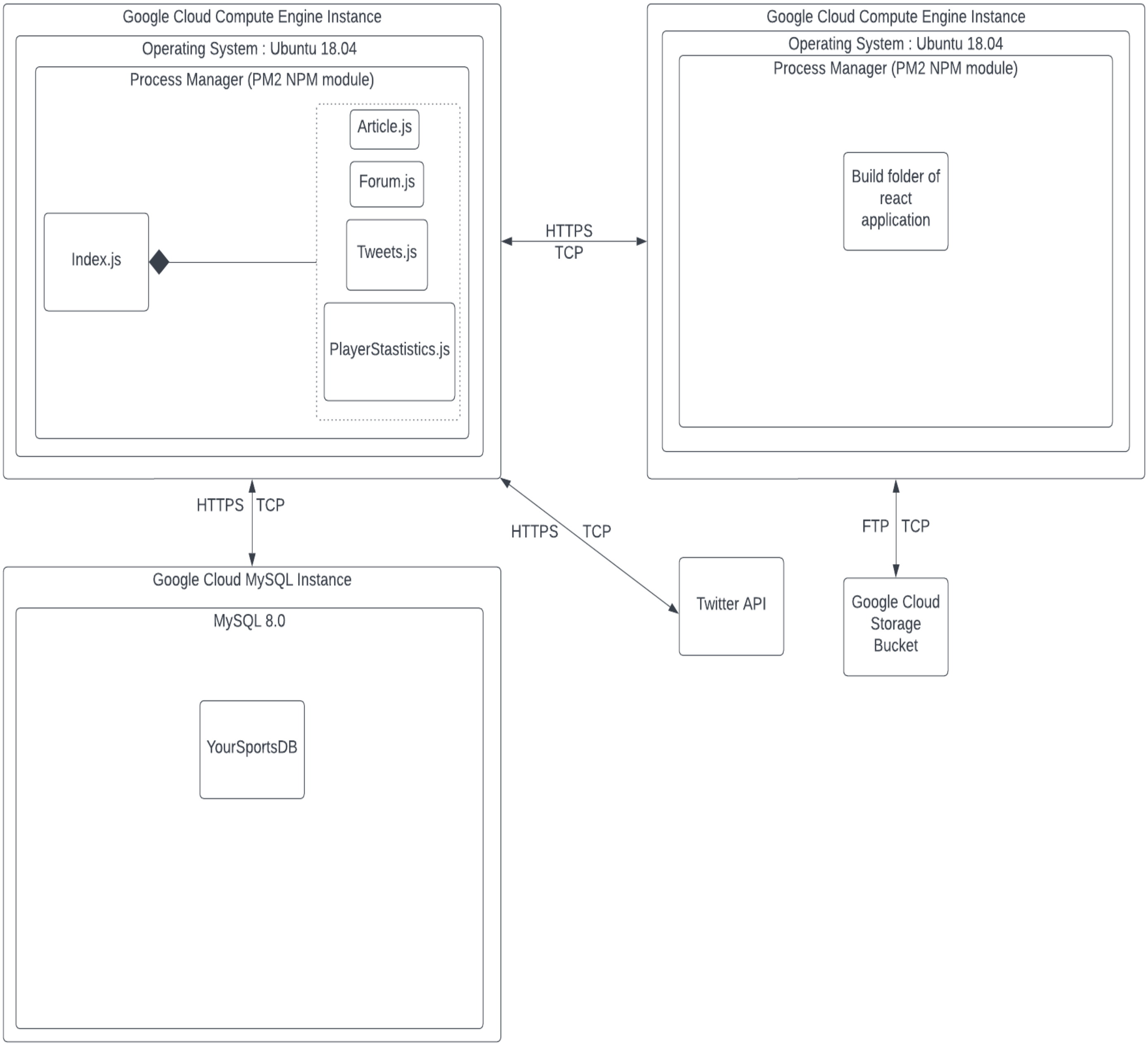

<<Abstract>>
RegularUser
- tracking_id: INT
- ip_address: String
+ RegularUser(): VOID
+ getTrackingID(): INT
+ getIPaddress(): String
Extends
RegisteredUser
- user_id: INT
- fk_accountid: INT
+ RegisteredUser(): VOID
+ getUserID(): INT
+ getAccountID(): INT
Tag
- tag_id: INT
- content: String
+ Tag(): VOID
+ getTagID(): INT
+ setContent(String): VOID
+ getContent(): String
Comment
- comment_id: INT
- author_id: INT
- article_id: INT
- posttime: DateTime Object
- content: String
+ Comment(): VOID
+ getCommentID(): INT
+ setContent(String): VOID
+ getContent(): String
+ getPostTime(): Object
+ getAuthorID(): INT
+ getArticleID(): INT
<<Abstract>>
Account
- account_id: INT
- name: String
- username: String
- email: String
- password: String
+ Account(): VOID
+ getAccountID(): INT
+ setName(String): VOID
+ getName(): String
+ setUsername(String): VOID
+ getUsername(String): String
+ setEmail(String): VOID
+ getEmail(): String
+ setPassword(String): VOID
+ getPassword(): String
Device
- devcie_id: INT
+ getDeviceID(): INT
Article
- article_id: INT
- author_id: INT
- posttime: DateTime Object
- image_url: String
- heading: String
- subheading: String
- introduction: String
- content: String
- conclusion: String
+ Article(): VOID
+ getArticleID(): INT
+ getAuthorID(): INT
+ setHeading(String): VOID
+ getHeading(): String
+ setSubHeading(String): VOID
+ getSubHeading(): String
+ setContent(String): VOID
+ getContent(): String
+ setConclusion(String): VOID
+ getConclusion(): String
+ getPostTime(): Object
+ setImage(String): VOID
+ getImage(): String
Admin
- admin_id: INT
- fk_accountid: INT
+ Admin(): VOID
+ getAdmintID(): INT
+ getAccountID(): INT
Creator
- creator_id: INT
- fk_accountid: INT
+ Creator(): VOID
+ getCreatorID(): INT
+ getAccountID(): INT
Dashboard
- dashboard_id: INT
- creator_id: INT
+ Dashboard(): VOID
+ getDashboardID(): INT
+ getCreatorId(): INT
+ getNumberOfViews(INT): INT
+ getArticleDetails(INT): Article
Extends
Extends
1
1
1
0..*
0..*
0..*
0..*
0..*
1..*
0..*
0..*
1
0..*
1
1
1

Application Network and Deployment Diagram

Application Network Diagram



Deployment Diagram


Google Cloud Compute Engine Instance
Operating System : Ubuntu 18.04
Process Manager (PM2 NPM module)
Article.js
Forum.js
Index.js
Tweets.js
PlayerStastistics.js
HTTPS
TCP
Google Cloud Compute Engine Instance
Operating System : Ubuntu 18.04
Process Manager (PM2 NPM module)
Build folder of react application
HTTPS
TCP
HTTPS
TCP
FTP
TCP
Google Cloud MySQL Instance
MySQL 8.0
YourSportsDB
Twitter API
Google Cloud Storage Bucket

Current Key Risks

Skills

Risks

- Most of the team does not have experience with cloud technologies
- Team members have varying skills, hence it is difficult to make decisions regarding technologies that fit everyone's requirements.

Proposed Solution

- Give team members enough time to learn the required technologies for our product.
- Make sure at least 3 members of the team can work with the cloud technologies that will be used in our project.

Schedule

Risks

- The priority one functional requirements may be difficult to complete by the end of the semester.

Proposed Solution

- Set internal deadlines to complete priority one requirements related to the features of our application.
- Delegate work according to the skillset of a team member.

Technical

Risks

- Many team members are not familiar with SQL which will be used to implement all of our search algorithms.

Proposed Solution

- Make sure every team member understands the functions implemented in the front-end and backend.
- Use the internet to search for optimal solutions that we will face when building our project

- Limit the number of different technologies used in our project.
- The team lead will make sure to review all the code from each branch before being pushed to the master branch.

Teamwork

Risks

- There is a possibility of conflict between team members who are working together on the same tasks.

Proposed Solution

- The team lead should delegate tasks to team members who have a similar set of skills, and make sure to resolve any misunderstandings between team members.

Legal/Content

Risks

- Users who are creators might upload inappropriate content in their articles.
- Users might use media/content that is not owned by them when they are trying to publish their articles.
- Setting up a terms and conditions agreement.

Proposed Solution

- Use an administrator account to monitor the content uploaded by creators.
- Create a legal document that states behavioral guidelines.
- Create a legal document that users who are creators agree to use only work owned by them when they are uploading articles.
- Use a ready made terms and conditions document that can be changed as per needed.

Project Management

Our project will be composed of three main components: front-end, back-end, and database system. Each component will be assigned to developers. The team lead will assign tasks to the people with the desired skill-set and set internal deadlines. The team lead will offer 24/7 support to make sure the tasks are completed by their given deadlines. The team lead will regularly pull from the Github repository, to make sure the developers are following proper coding standards, and making sure everything is going as per the requirements of the project. The front-end and back-end leads are responsible for setting their own deadlines, and plans of action to complete their given tasks. Trello will be used to track the tasks, the resources of the project will be accessed through Trello, and all the communication will be done through Discord. The team lead will set realistic deadlines, and consult with the front-end and back-end lead before assigning tasks and setting deadlines, to make sure realistic and achievable goals are set. The entire team will attend the scheduled meetings unless otherwise instructed, and update and ask for support from other team members to make sure tasks are completed by their deadline. The team lead will also assign tasks related to the documentation if they can be done individually, or complete the tasks together in a team meeting. Finally, the team lead will consult the rest of the team members before making any major decisions related to the project.

List of Team Contributions

Kshitiz Sareen (Team Lead, Back-end Lead)
Constructed the back-end API for searching news articles, and signing up. Deployed the front-end and backend applications on the server. Managed team meetings, and tasks progress of individual team members. Draws the “checking sport news” mockup storyboard.

Kevin Islas (Front-end Lead)
Contributed in designing the home page and the display for the articles. Implemented the search API front-end call function as well as the display of the data accordingly. Contributed in displaying information received from the sign up API and implemented the navbars. Draws the “posting news articles” mockup storyboard.

Shamar Ireland (Front-end Developer)
Contributed in designing the home and signup page as well as the formatting of the data displayed below the search bar. Contributed in implementing the signup API and prioritizing the use cases. Draws the “signing up” mockup storyboard.

Wenye Guo (Database Administrator)
Contributed in drawing the high level UML class diagram, ERD, database model, and in database organization. Draws the “creating group chat” mockup storyboard.

Sabur Saigani (Functional Lead)
Contributed in drawing the networking diagram, high level UML class diagram, ERD, database model, and in database organization. Contributed in checking the documentation’s grammar. Draws the “twitter feeds” mockup storyboard.

Jonathan Ip (Functional Assistant)
Contributed in formatting and organizing the documentation’s layout as well as checking the documentation’s grammar. Draws the “following sport game” mockup storyboard.

Mathew O Abiola (Github Master)
Contributed in prioritizing the use cases. Draws the “fantasy betting” mockup storyboard.