

EEL4914C Computer Engineering
Senior Design Final Project Report

Mouthpiece
Social Media Application for Athletes

Ian Tyrell

Computer Engineering B.S. FSU

Group Lead & Front-end Engineer

Peyton Smith

Computer Engineering B.S. FSU

Back-end Engineer

Julien Denis

Computer Engineering B.S. FAMU

Back-end Engineer

Christan Robinson

Computer Engineering B.S. FSU

Front-end Engineer

Abstract

Mouthpiece is a social media app used to share posts, pictures, and videos for a community of those interested in sports. This app connects athletes, sports fans, coaches/recruiters, trainers, and tournament organizers throughout the world. Athletes can gain a fan base by posting pictures, videos, and highlight reels which give exposure to coaches/recruiters. As for the recruiters, they can find out players' capabilities and assess their statistics. Trainers can find and help develop players by professionally training them. Teams can use this app for player scouting, analyzing their strengths and weaknesses. The main purpose of the app is to highlight the skills shown by athletes, enabling the chance of scholarships or job opportunities.

As for the construction of the project, our group is separated into both front and back-end developers. Front-end developers are responsible for how the app looks to the end-user. They design the user experience such as plotting what each button click does throughout the app. Back-end developers manage how the app operates under-the-hood. Using Google's database, Firebase, they are responsible for storing data throughout the app such as user profile data, messages, posts, and tournament data. The app is for iOS devices starting out. Using an iOS exclusive IDE, Xcode, developers code the app using Apple's programming language, Swift.

Unlike other social media platforms such as Twitter, Instagram, and TikTok, this app is solely focused on sports-related content. This is made possible by an internal filter system in the app that monitors the type of content posted. Users are also able to sort the type of sports content they wish to see through embedded algorithms in the app. Overall, this app is a great choice for sports fans seeking convenient sports content without being overwhelmed by any other irrelevant content.

Disclaimer

The information presented in this document is for educational and informational purposes only and should not be construed as professional advice. The authors and contributors of this document make no representation or warranty, express or implied, as to the completeness, accuracy, reliability, suitability or availability of the information contained herein. Any reliance you place on such information is therefore strictly at your own risk. The authors and contributors of this document will not be liable for any loss or damage, including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising from loss of data or profits arising out of, or in connection with, the use of this document. Any opinions expressed in this document are those of the authors and contributors and do not necessarily reflect the views of any organizations or institutions they may be affiliated with. The mention of any product, service, or organization in this document does not imply endorsement by the authors and contributors or their affiliated institutions.

Acknowledgements

We would like to express our deepest appreciation to the FAMU-FSU College of Engineering, Dr. Noroozi, and Dr. Hooker for providing us with the opportunity to create the Mouthpiece application and for their guidance throughout the planning and development process. Their support and encouragement were invaluable to our team.

We would also like to thank the development team who worked tirelessly to create this innovative application. Their hard work and dedication have resulted in the product that we are proud to present.

Finally, we would like to extend our gratitude to all those who have viewed our demo and final presentations. Your support and feedback have been essential in helping us improve and refine the product.

Once again, we would like to thank everyone who has contributed to the development of the Mouthpiece application. Your support and dedication have been invaluable to our team.

List of Tables & Figures

- Figure 1: Functional Decomposition Chart
- Figure 2: Target Chart
- Figure 3: Concept Generation Chart
- Figure 4: Concept Selection Chart
- Figure 5: House of Selection Chart
- Figure 6: AHP
- Figure 7: ER Diagram
- Figure 8 : Updated ER
- Figure 9 : Figma
- Figure 10: XCODE
- Figure 11: ED 1
- Figure 12: ED 2
- Figure 13: ED 3
- Figure 14: ED 4

Chapter One

Fall Semester Planning

Project Scope

Description

Mouthpiece is a social media motivated application whose sole purpose is to seamlessly share posts, pictures, and videos in a community of athletes. This community will connect athletes, sport fans, teams, and team members. The services provided:

- **User Profiles**
 - Specialized profile editing.
 - Profile Pictures
 - Tiers
 - Stats (Unofficial and Official)
- **Social Sharing**
 - Post pictures, videos, strings
 - Commenting
 - Messaging
 - Sharing
 - Liking, reactions
 - Ratings
- **Geolocation**
 - **Court/Field Locator**
 - Show hot courts in the area.
 - Nearest Courts
 - **Tournaments/Games Locator**
 - Custom Map Markers
 - MP MAP
- **Competitive Factor**
 - Athlete tier

- S through D.
- Tier Panels
- **Rankings**
 - Winning/followers/Likes decides rankings.
 - Beating higher ranks boosts rank.
 - Tournament incentive
 - Incentivizes playing competitive.

- **Teams/Groups**
 - Team feature where users can create a team.
 - Can be a street team, rec team, content creator team, school team, or just a group.
 - Groups can have tiers and a page.
 - Tier Panel
 - Team can do the same social sharing.
 - Register for tournaments.

- **Goals**
 - **1- Year Goals**
 - Deploy a working prototype on the App store.
 - Share the application with highschool, recreational, college, and intramural teams in the area.
 - Use the data gathered from users to improve the app, fix bugs, and scale service performances.
 - **Fall Semester Goals**
 - Initiate Swift course and lesson plans for software engineers.
 - Survey athletes in the area in order to scale out matrices of importance for application features.
 - Create deadlines, micro deadlines, and immediate deadlines for paperwork, class forms, NDA agreements, and lesson plans.
 - Design a UX/UI wireframe that shall highlight the front end of the mobile application.

- Have a concrete plan for the functionality of the application, filling all holes.
- Start development with the code homed on a private git server.
- **Spring Semester Goals**
 - Create weekly, biweekly development deadlines that align with the progress of the application.
 - Stress test good, functionality test good, and minor bug reports.
 - Patreon and sponsor funding.
 - Working prototype whose features are working efficiently.
- **Primary and Secondary market**

The customers for mouthpiece include:

- 1. Athletes**
 - a. Under looked athletes
 - a. Athletes in underrepresented communities
 - a. Athletes who want to join teams in local areas.
- 2. Coaches**
 - a. Coaches who want to create/coach teams.
- 2. Recruiters**
 - a. Who wants to find talent across the world or via locally.
 - a. Wants to recruit talent from underrepresented communities.
 - a. May not be able to travel and see all players.
- 2. Casual Viewers**
 - a. People who enjoy watching sports but not necessarily play them.
 - a. Someone who used to play sports all their life and genuinely enjoys watching it.
 - a. Does not want to go on YouTube every time to look up talent/highlights.
 - a. Wants to keep up with local talent that's going to be big in the future.
- 2. Trainers**
 - a. Who wants to promote their teaching skills to athletes or teams.
 - a. Wants to connect with various athletes and teams.
- 2. Tournament Organizers**

- a. Tournament hosts that want to promote their tournament on the application to teams in the area or outside of the area.

2. Teams

- a. Promote their team.
- a. Be a part of the competitive factor that would improve their rankings.

- **Assumptions**

- Users have an iOS device (starting out)
- Users can operate a smart device to run the application on.
- Users can manage to download applications from the app store on said iOS device.
- Users have a form of identification (email/phone number with password) to sign into their respective accounts on the application.

- **Stakeholders**

- No current stakeholders, however, Patreon/stakeholder may be of conversation in the future.

Customer Needs & Requirements

Note: Since we don't have a sponsor for this project, we asked a potential consumer that is interested in the theory behind the app to answer a few questions about any features that we could implement to make the app the best it could be.

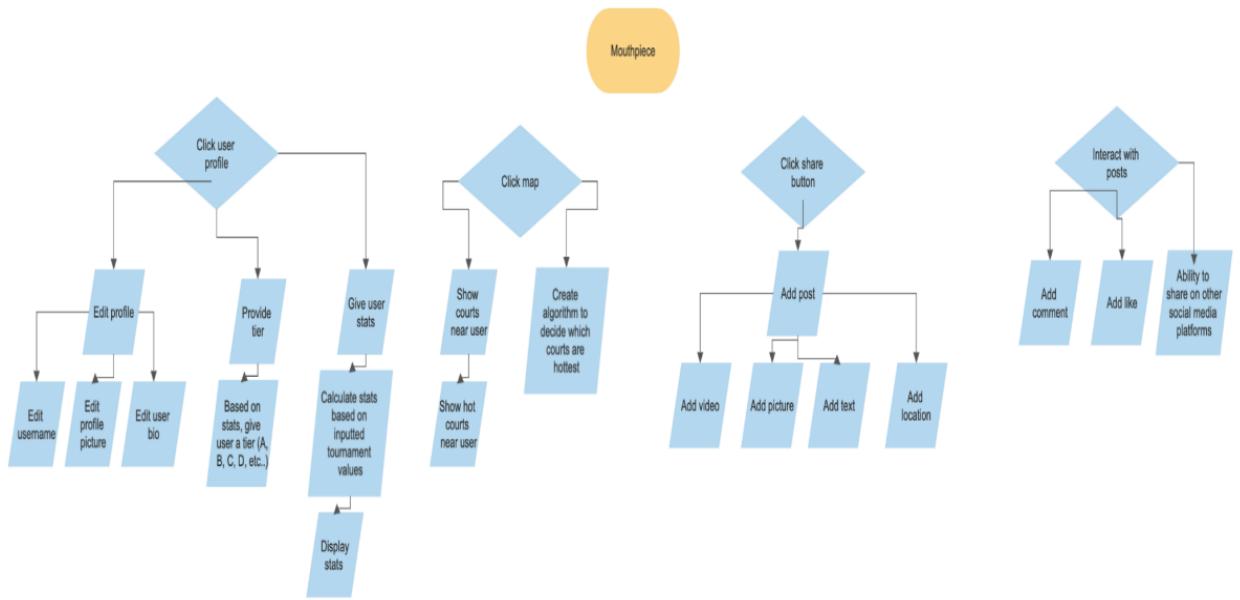
Questions given to the customers with the respective responses:

1. Question: What feature would you like to see in the app regarding picture/video sharing?
 - a. Response: I think it would be nice for other users on the app to interact with your shared pictures and videos just like how you can in any other social media app.
2. Question: How can we, as the developers of the app, bring the basketball community better together?
 - a. Response: Having some in-person events would be nice, maybe having a way for people of the app to play basketball together.
3. Question: What do you think is best for the recruitment-side of basketball from a recruiter's perspective?
 - a. Response: Have media be present and easily accessible for recruiters to see when they open the app.
4. Question: Any way for the app to integrate more of a competitive scene amongst players?
 - a. Response: Have a ranking system that ranks teams/individuals based on how well they perform in in-person events.
5. Question: Can you think of any other nice-to-have features for the app?
 - a. Response: Having statistics of a team or user would be nice, also helps the recruiters if they want more of an in-depth analysis of who they are looking for.

Interpreting customer needs:

1. Have a system where each post made by a user or team, whether that be a text post, video, or picture, has the ability to be explicitly “liked” or “loved”. Other users should also have the ability to comment or share the post with others.
2. Implement some geo-location functionality where users of the app can check nearby basketball courts, and closely monitor who and how many individuals are at said court.
3. Have a “discover” icon that shows videos and pictures of individuals that are growing in popularity on the app through their skill, giving recruiters the perfect leeway to recruit those that are seeking recruitment.
4. Have a tournament section of the app that allows teams, no matter the skill, to face other teams of similar-skill around the area to earn points towards a ranking system (S-D tiers), this incentivizes users to seek tournament play by implementing a more competitive aspect of the app.
5. When viewing the profile of a team page or an individual, users can view both unofficial and official statistics of said team or individual, such as recent games played alongside the scores, any tournaments that they’ve participated in and show how these statistics compare globally to other teams/individuals on the app.

Functional Decomposition



Explanation of Results:

We believe these are the fundamental components to creating a minimally viable product for Mouthpiece. These are the main components of a social media app, as well as components that will make our app unique.

The services provided:

User Profiles

- Specialized profile editing
- Profile Pictures
- Tiers
- Stats (Unofficial and Official)

Social Sharing

- Post pictures, videos, strings
- Commenting
- Messaging
- Sharing
- Liking, reactions

- Ratings

Geolocation

- Court/Field Locator
- Show hot courts in the area
- Nearest Courts

Connection to systems:

The columns provide an explanation of the steps our team needs to complete in order to produce a working function. As you go further down the graph, you will see that some of the functions build upon each other in order to provide more features. We start with the simplest implementation then proceed to complete harder tasks the further we get into the project.

Smart Integration:

The rows provide an explanation of all the functions our project will encompass. We believe these are the building blocks needed in order to create a minimal viable product. For example, every social media needs a way to edit your profile, post information, and comment and share. Once we complete all these functions, we can say we have a working product and can begin working on more advanced functions and features that our user may want.

Target Summary

Targets are formulated by features and requirements listed above in the functional decomposition chart.

Targets derived from requirements and features are given via the Pugh chart below:

ATTRIBUTES	PROFIT	TIME	EFFORT	TARGET
UI/UX	⊖	↓	↑	ALLOWS FOR EASY USE FOR CONSUMER AND EASIER DEVELOPMENT.
ACCESSABILITY	↑	↑	↓	HIGH ACCESSABILITY MIGHT BE HARDER TO IMPLEMENT WITHIN THE APP HOWEVER IT WILL SAVE TIME IN FIXING AND DEBUGGING.
INTERACTIVE	↑	↓	↑	INTERACTIVITY IS NEEDED FOR PROFIT AND WOULDN'T BE HARD TO IMPLEMENT, HOWEVER TIME MAY BE OF ISSUE.
GIOLOCATION	↑↑	↓	↓	HIGHLY PROFITABLE BUT REQUIRES BOTH EFFORT AND TIME, THIS MAY VARY AS THERE ARE SOURCE CODE FOR GOOGLE AND APPLE MAPS.
ALGORITHMIC FEATS	↑↑	↓↓	↓↓	HIGHLY PROFITABLE BUT REQUIRES MATHEMATICAL EFFORTS AND TIME. THIS WILL BE LUCRATIVE BUT DIFFICULT.
RANKING SYSTEMS	↑	↑↑	↑	WILL SAVE TIME FOR OTHER FEATURES WITHIN PLACEMENTS FOR USER PROFILES AND HELP ORGANIZATION.
BASIC SHARING/POST	⊖	⊖	⊖	BASIC FEATURES FOR APPLICATIONS ARE THE STANDARD.

Concept Generation

Shown below is a table of all the app components that we've gathered generations for ranging from the backend with which database to use to the type of feed content and of course the icons used throughout the app.

		Generations										
		App Component	1	2	3	4	5	6	7	8	9	10
Pegton		Database	A/S	SQLite	Realm	Apple's Core Data	Firebase					
Pegton		Feed Content	Videos only	Pictures only	Text posts only	Videos & pictures above	All of the above	Pictures & text posts				
Pegton		Geolocation Button										
Pegton		Profile Button										
Julien		Discover Button										
Julien		MyTeam Button										
Ian		User Title	@ handle + N	Username	Tier Panel			Player position	Team Affiliate	Verification st	Detailed profile	
Ian		Badge								Professional	Hammer Badge	
Chris		Share Button										
Chris		Comment Button										
Chris		Like Button										

Concept Selection

Pugh:

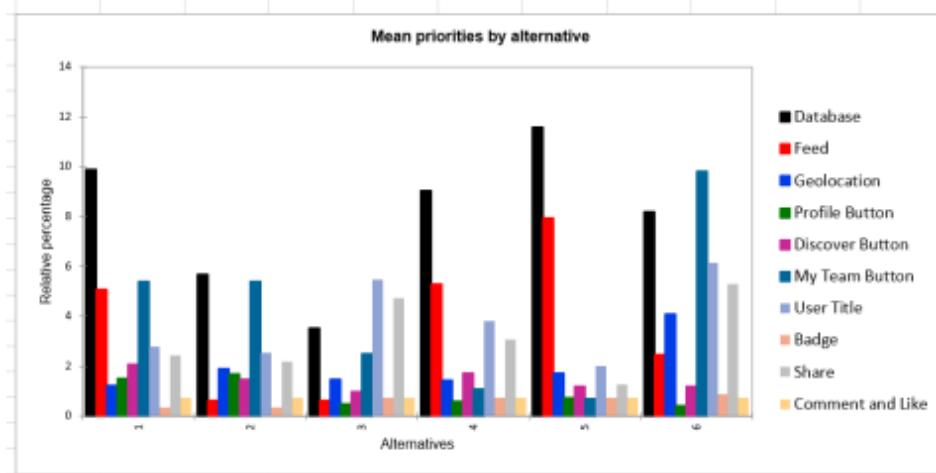
ATTRIBUTES	PROFIT	TIME	EFFORT
UI/UX	⊖	⬇️	⬆️
ACCESSABILITY	⬆️	⬆️	⬇️
INTERACTIVE	⬆️	⬇️	⬆️
GIOLOCATION	⬆️ ⬆️	⬇️	⬇️
ALGORITHMIC FEATS	⬆️ ⬆️	⬇️ ⬇️	⬇️ ⬇️
RANKING SYSTEMS	⬆️	⬆️ ⬆️	⬆️
BASIC SHARING/POST	⊖	⊖	⊖

House of Quality:

Row #	Max Relationship Value in Row	Relative Weight	Weight / Importance	Column #														
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Direction of Improvement: Minimize (▼), Maximize (▲), or Target (x)																		
1	9	18.5	1.0	UI/UX	▲	○	○	○	○	○	○	○	○	○	○	○	○	
2	9	9.3	0.5	Accessibility	▲	○	○	○	○	○	○	○	○	○	○	○	○	
3	9	14.8	0.8	Interactive	▲	○	○	○	○	○	○	○	○	○	○	○	○	
4	9	18.5	1.0	Geolocation	○	○	○	○	○	○	○	○	▲	▲	○	○		
5	9	16.7	0.9	Algorithmic Feats	○	○	○	○	○	○	○	○	○	○	○	○	○	
6	9	13.0	0.7	Ranking Systems	○	○	○	▲	○	○	○	○	○	○	○	○	○	
7	9	9.3	0.5	Basic Sharing/Posts	○	○	▲	○	▲	○	○	○	▲	○	○	○		
8																		
9																		
10																		
Target or Limit Value																		
Difficulty (0=Easy to Accomplish, 10=Extremely Difficult)																		
Max Relationship Value in Column																		
Weight / Importance																		
Relative Weight																		

AHP:

Crit./Alt.	UI/UX	Accessabi lity	GEOLOCA TION	Algo	Ranking	Basic Sharing post
Database	9.88	5.66	3.51	9.04	11.57	8.18
Feed	5.07	0.61	0.61	5.28	7.93	2.46
Geolocation	1.23	1.89	1.46	1.45	1.73	4.10
Profile But	1.52	1.69	0.47	0.58	0.73	0.43
Discover E	2.06	1.48	0.98	1.72	1.19	1.19
My Team	5.40	5.40	2.50	1.10	0.71	9.83
User Title	2.79	2.54	5.48	3.81	2.00	6.15
Badge	0.35	0.35	0.74	0.74	0.74	0.86
Share	2.44	2.19	4.74	3.06	1.25	5.29
Comment	0.74	0.74	0.74	0.74	0.74	0.74



Conclusive selections:

Developing the social media application “Mouthpiece”, we will use different features that would meet designer and customer needs. After analyzing data, feed content, discover section, and user profiles are the most important features to take into consideration. The other notable features that relate to those three features includes database, geolocation, and badges. Staying within course of the “top three” will also cover the other features.

Spring Project Plan

Completed Fall Semester Goals

- Initiate Swift course and lesson plans for software engineers.
- Survey athletes in the area in order to scale out matrices of importance for application features.
- Create deadlines, micro deadlines, and immediate deadlines for paperwork, class forms, NDA agreements, and lesson plans.
- Design a UX/UI wireframe that shall highlight the front end of the mobile application.
- Have a concrete plan for the functionality of the application, filling all holes.

Spring Semester Goals

- Continue development with the code homed on a private git server.
- Create weekly, biweekly development deadlines that align with the progress of the application.
- Finish prototype by April.
- Stress test good, functionality test good, and minor bug reports by April.
- Patreon and sponsor funding by March.
- Working prototype whose features are working efficiently by April.

Chapter Two

Spring Semester Development

Restated Project Definition and Scope

Description

Mouthpiece is a social media motivated application whose sole purpose is to seamlessly share posts, pictures, and videos in a community of athletes. This community will connect athletes, sport fans, teams, and team members. The services provided:

- **User Profiles**
 - Specialize profile editing.
 - Profile Pictures
 - Tiers
- **Social Sharing**
 - Post pictures, videos
 - Commenting
 - Messaging
 - Sharing
 - Liking
- **Geolocation**
 - **Court/Field Locator**
 - Show hot courts in the area.
 - Nearest Courts
 - **Tournaments/Games Locator**
 - Custom Map Markers
 - MP MAP
- **Competitive Factor**
 - Athlete tier
 - S through D.
 - Tier Panels

- **Rankings**

- Winning/followers/Likes decides rankings.
- Beating higher ranks boosts rank.
- Tournament incentive
- Incentivizes playing competitive.

- **Teams/Groups**

- Team feature where users can create a team.
 - Can be a street team, rec team, content creator team, school team, or just a group.
- Groups can have tiers and a page.
- Tier Panel
- Team can do the same social sharing.
- Register for tournaments.

- **Goals**

- **1- Year Goals**

- Deploy a working prototype on the App store.
 - Share the application with highschool, recreational, college, and intramural teams in the area.
 - Use the data gathered from users to improve the app, fix bugs, and scale service performances.

- **Fall Semester Goals**

- Initiate Swift course and lesson plans for software engineers.
 - Survey athletes in the area in order to scale out matrices of importance for application features.
 - Create deadlines, micro deadlines, and immediate deadlines for paperwork, class forms, NDA agreements, and lesson plans.
 - Design a UX/UI wireframe that shall highlight the front end of the mobile application.
 - Have a concrete plan for the functionality of the application, filling all holes.
 - Start development with the code homed on a private git server.

- **Spring Semester Goals**
 - Create weekly, biweekly development deadlines that align with the progress of the application.
 - Stress test good, functionality test good, and minor bug reports.
 - Patreon and sponsor funding.
 - Working prototype whose features are working efficiently.
- **Primary and Secondary market**

The customers for mouthpiece include:

- **Athletes**
 - Under looked athletes
 - Athletes in underrepresented communities
 - Athletes who want to join teams in local areas.
- **Coaches**
 - Coaches who want to create/coach teams.
- **Recruiters**
 - Who wants to find talent across the world or via locally.
 - Wants to recruit talent from underrepresented communities.
 - May not be able to travel and see all players.
- **Casual Viewers**
 - People who enjoy watching sports but not necessarily play them.
 - Someone who used to play sports all their life and genuinely enjoys watching it.
 - Does not want to go on YouTube every time to look up talent/highlights.
 - Wants to keep up with local talent that's going to be big in the future.
- **Trainers**
 - Who wants to promote their teaching skills to athletes or teams.
 - Wants to connect with various athletes and teams.
- **Tournament Organizers**
 - Tournament hosts that want to promote their tournament on the application to teams in the area or outside of the area.
- **Teams**
 - Promote their team.

- o Be a part of the competitive factor that would improve their rankings.

- **Assumptions**

- Users have an iOS device (starting out)
- Users can operate a smart device to run the application on.
- Users can manage to download applications from the app store on said iOS device.
- Users have a form of identification (email/phone number with password) to sign into their respective accounts on the application.

- **Stakeholders**

- No current stakeholders, however, Patreon/stakeholder may be of conversation in the future.

Detailed Design

Preliminary Detailed Design

I. Introduction

a. Problem Statement

- i. Develop an iOS mobile device-driven social media platform whose sole purpose is to seamlessly share posts, pictures, and videos for a community of athletes.

b. Motivation

- i. Having a centralized application aimed towards sports, more specifically basketball to help bring the community of athletes, sports fans, team members and recruiters together.

c. Requirements

- i. Have basic social-media application functionality such as sharing posts, pictures and videos as well as reacting to said content, following/friend system, messaging system
- ii. Have specializes user profiles with basic user profile data, tiers, and statistics about the user.
- iii. Geolocation functionality – Court/Field locator using Google Maps API, showing nearest/popular courts in the area.
- iv. Team/Group profiles – For various types of teams (street/recreational/content-creator/school/group of friends), teams can perform social sharing functionality like individuals.
- v. Competitive – In-game tournaments, tier panels/ranking system.

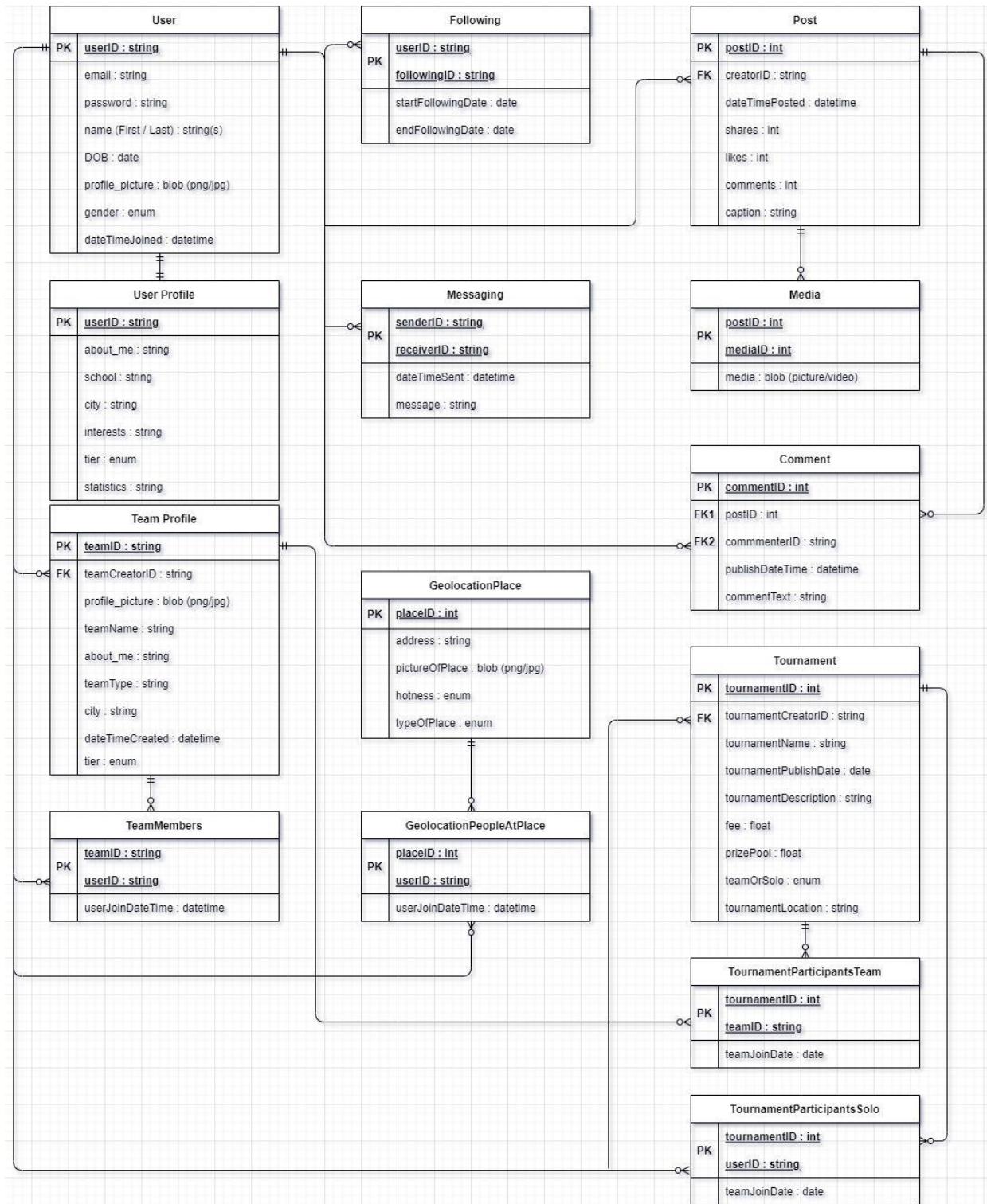
II. Selected Concept

- a. For our selected design concept, we have shown the full database schematic of the application. Going from basic user profile

data to tournament data, this design concept aims to show the full scope that this project hopes to implement.

III. Preliminary Design

a. Database ER (Entity-Relationship diagram)



- Using our choice of database, Amazon Web Services, we will create each of these entities as models represented as the core back-end of our project schema. With these models we will then store all the data/table entries using an AWS SQL (structured query language) server where we

can flexibly communicate with the database back-end whether that be updating ,deleting, or retrieving any data.

b. Component Requirements

i. For our core system component, we have our user entity, here we can store user data. Most importantly we can reference a user using a unique primary key, userID. This key can be used in all major aspects of the project database. Using the userID, we can gain access to user profile/badge and follower data. For our post block, we have a unique postID key that references a creatorID to the userID mentioned above. This postID key is used to retrieve any media associated with the post, whether that be a gif, video, or picture. In the comment block, it also uses the postID to fetch all comments associated to the post. Moving over to the message block, the userID is used for both the sender/receiverID properties. Switching to the team functionality, like a user profile, we also have a block dedicated to team profile data, represented by a unique identifier, teamID. From teamID, we can reference all the team members associated to the team, again using the userIDs. For our competitive requirement, we have both a tournament and game block. Both blocks have their own unique identifier, tournament/gameID with all the data associated with the two. In a tournament we have both team and solo play functionality, so we use the teamID/userID for tournament participants. Likewise with the game functionality, we use userID to reference all the game participants. With the game participants there is also the statistics associated with the game, per each participant. In the Statistics block, we use both the userID and gameID to reference all basketball statistics for a user in a particular game. Lastly with our geolocation requirement/functionality, we need to store court information, identified with a courtID, we can store all the data associated with a court. Moreover, with court information, we also store any people at the court, again using the userID input.

c. Function Implementation

i. As said in the introduction, in order to implement each and every one of these blocks/entities we first need to make model (code classes) with each of the properties listed in the diagram for each model as well as their data typing. With each of the model/classes implemented in our codebase, we can use AWS to reference these models to connect the two and store any instances of the models. With the database structured this way we can then easily communicate with the database whether that be with debugging or pulling manipulated data on the front-end.

d. Functional Decomposition Changes

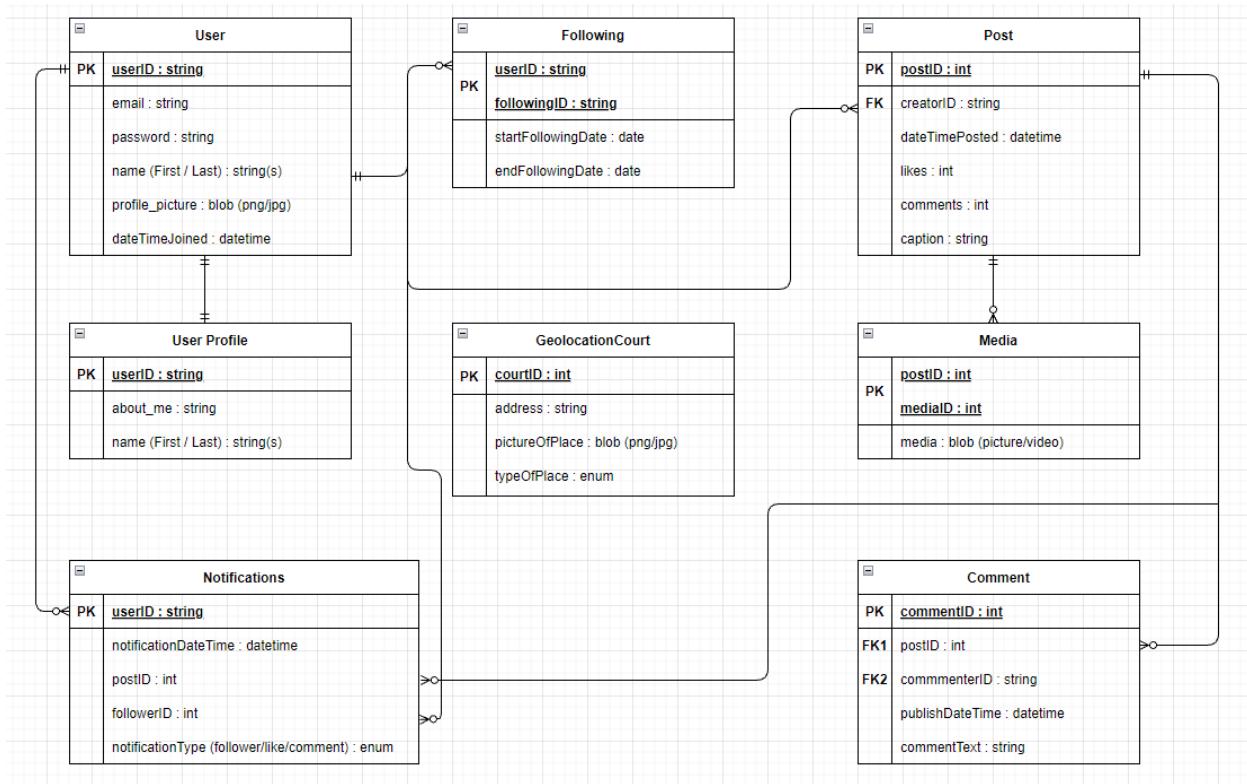
- i. No changes made between the functional decomposition and preliminary detailed design. All services/functionality are the same and implemented as such.

IV. Summary

- a. With the database fully plotted out, that gives the back-end developers a good understanding of the class/model implementation that we will use in our back-end code. This entity-relation diagram does just that. Each block in this ER diagram demonstrates functionality that is to be fully implemented in development. Starting from basic user/user-profile data we can gain access to other blocks like post/comment data, with their respective media embeds, messaging/following functionality to connect with other users of the app directly. Providing a more competitive nature of the app by implementing tournament/game functionality with the statistics of each player/team. Lastly by connecting users together in-person, having built-in geolocation functionality, we can store court data as well as all the people in said court. All in all, the preliminary design discussed will give us developers/engineers a better time on both the back/front-end when it comes to development of the application in order to implement project requirements.

Final Design

Updated ER Diagram



For the final design's back-end implementation, the database of choice is Google's Firebase. We've created each of these entities as models represented as the core back-end of our project schema. With these models we've stored all the data/table entries using an Firebase NoSQL (a non-relational structured query language) server where we can flexibly communicate with the database back-end whether that be updating , deleting, or retrieving any data. Component requirements and functional implementation applies the same as before in the preliminary detailed design.

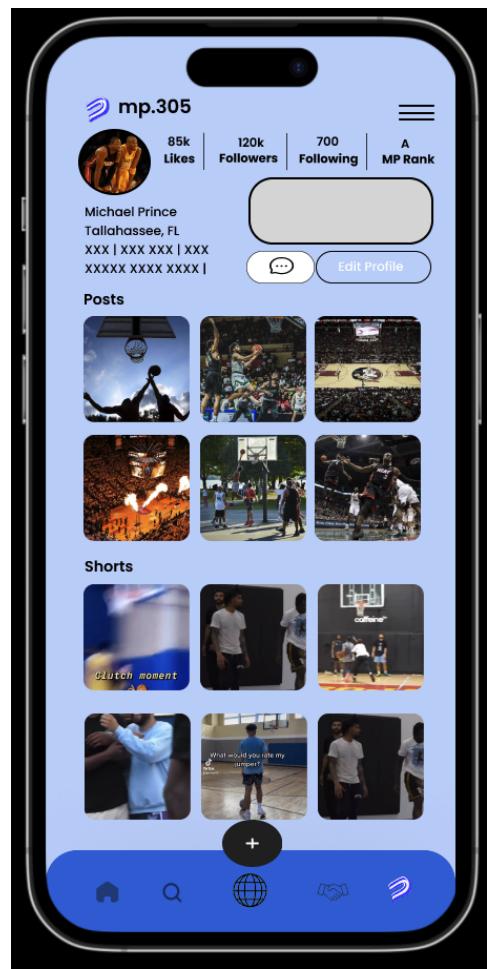
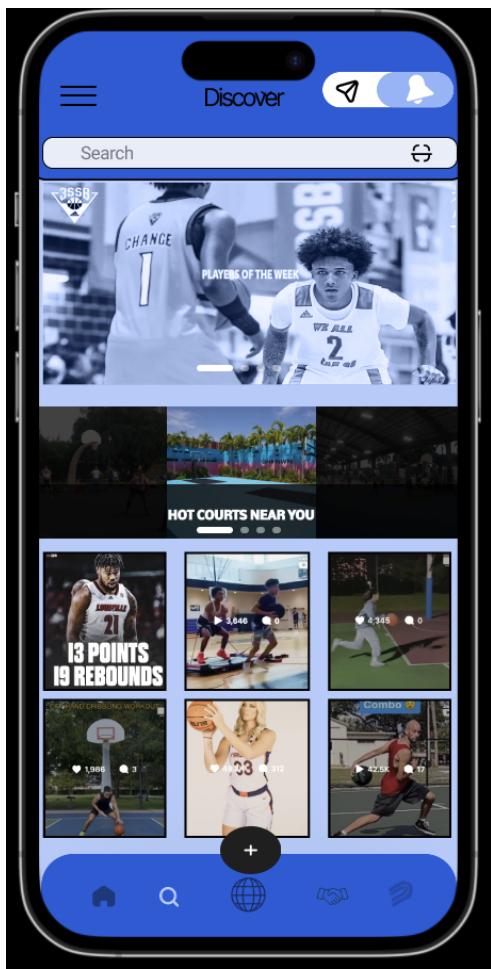
We've transitioned the preliminary to our final detailed design by fitting and reworking the project requirements within the allotted time frame. Doing so, we've managed to accomplish the underlying foundation that the social media app rests within the app itself as well as getting the core functionality of a social media app constructed (post sharing, notifications, profile editing). Our next biggest goal is getting one of our unique app features up and running which is our geolocation functionality. Using the iterative process of design while constructing the app, we've defined the problem at hand and what the user needs are, given our restraints. With this clear understanding in mind, the front-end and back-end engineers will do their parts in constructing the app's UI/UX in Figma to configuring the Firebase back-end, to finally connecting the two and integrating both within Xcode while coding our app. With a prototype done, we have

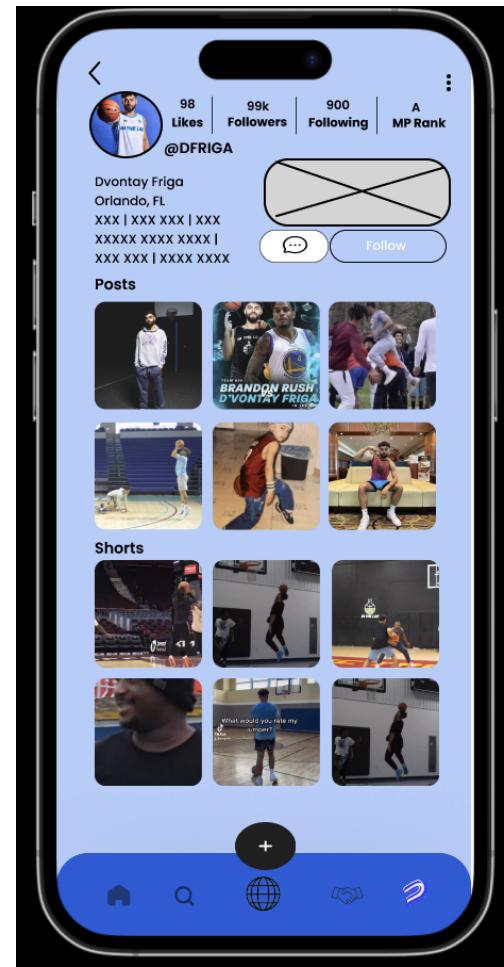
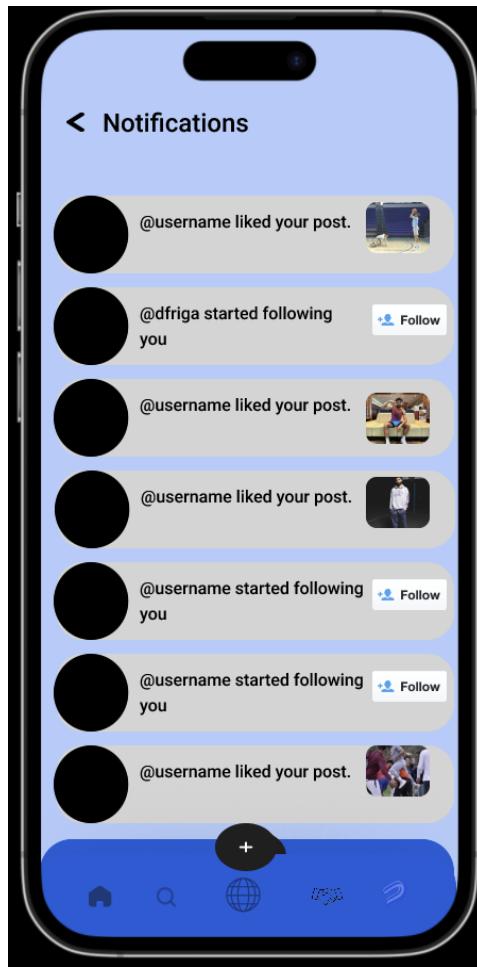
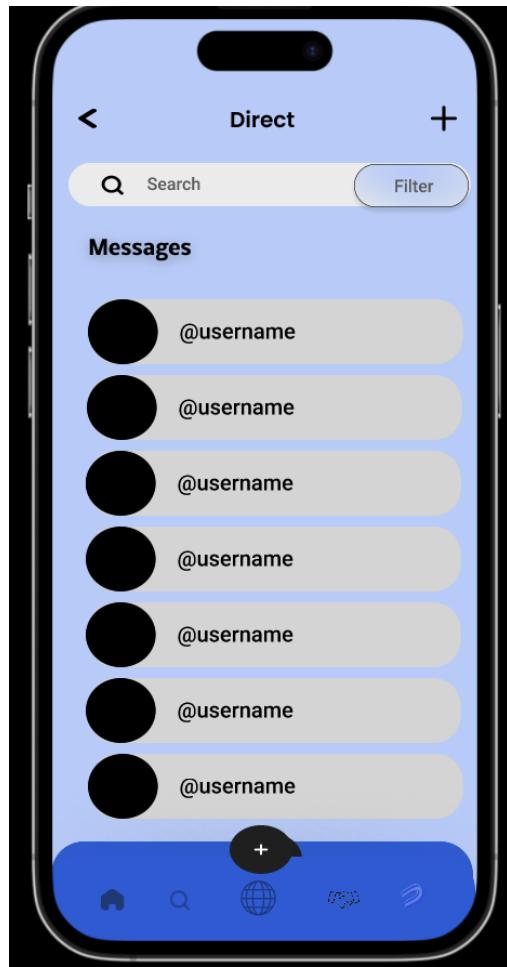
something to work with, going onto the next step in the process, user testing. Knowing that all of the project requirements are met, we can test each of these requirements, testing sheets are shown in the appendix of all the tests we've done. During this stage, we've tested all parts of the app's functionality and ironed out any bugs that have arose. Doing so, we can refine and iterate this until we have a functional working prototype within our project's limitations. At this point, we have applied the iterative process of design to construct a working prototype with the core foundation of a social media app complete and ready to use for pushing out other functionality we might want in the future.

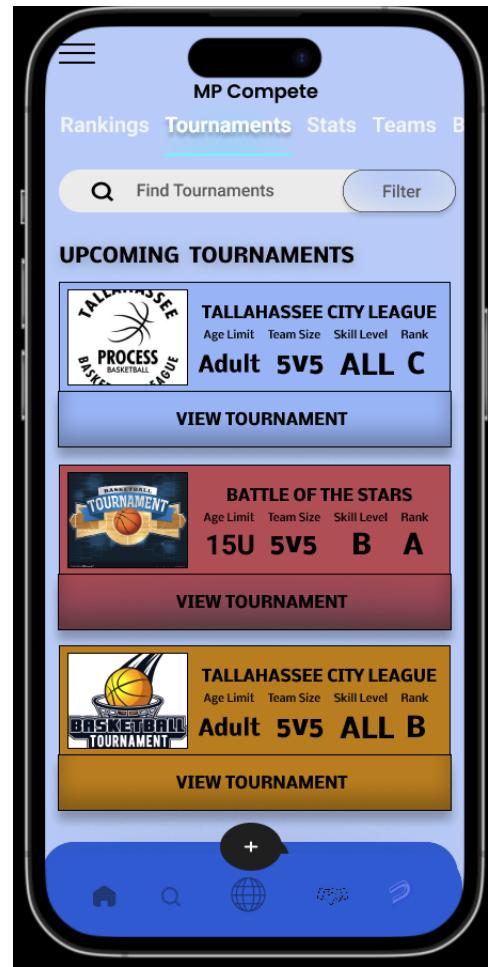
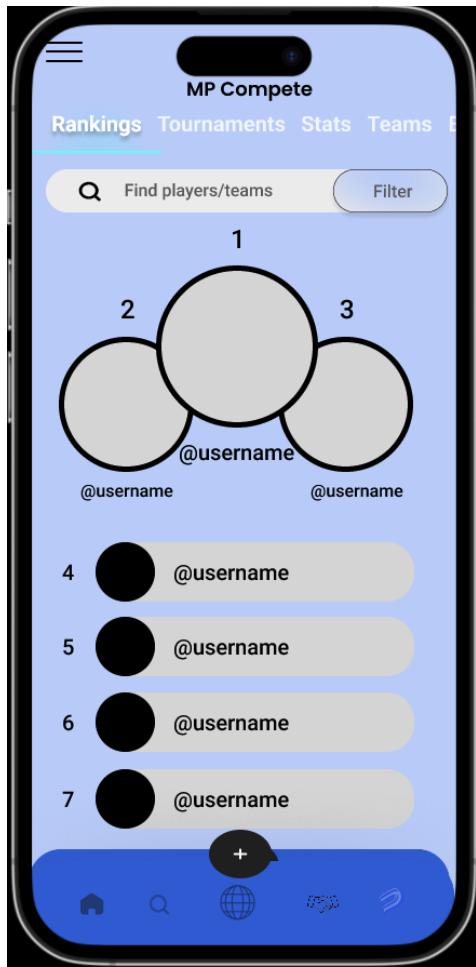
Results

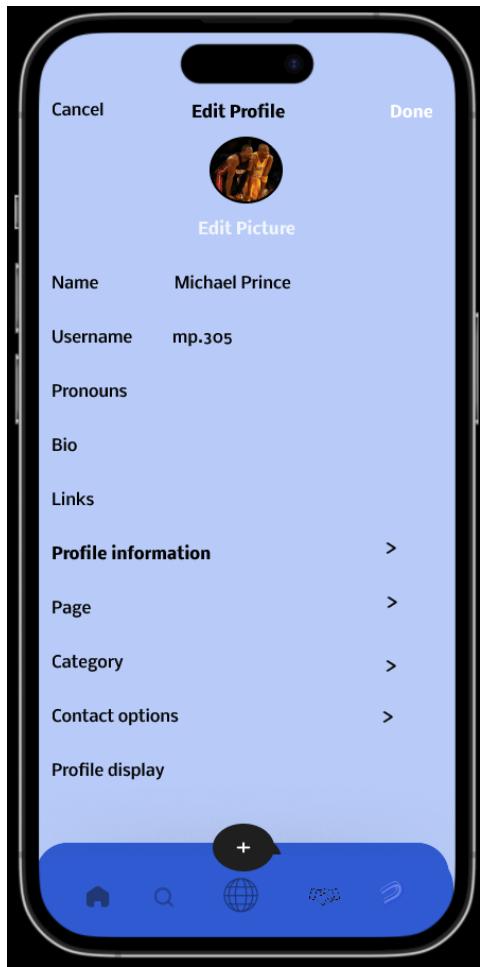
Figma Design





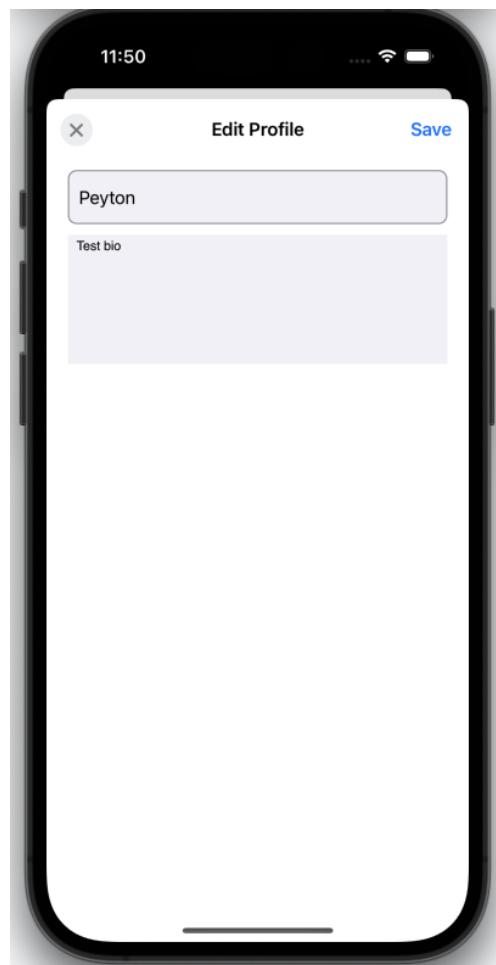
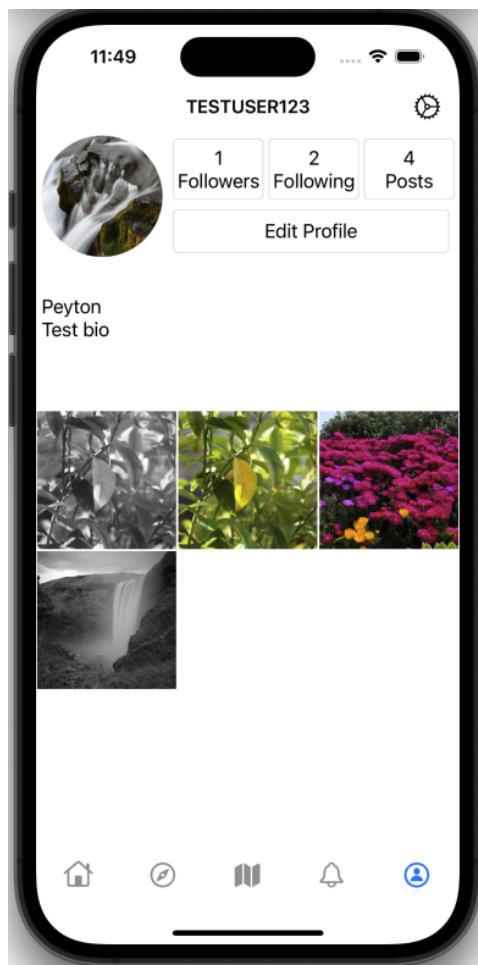
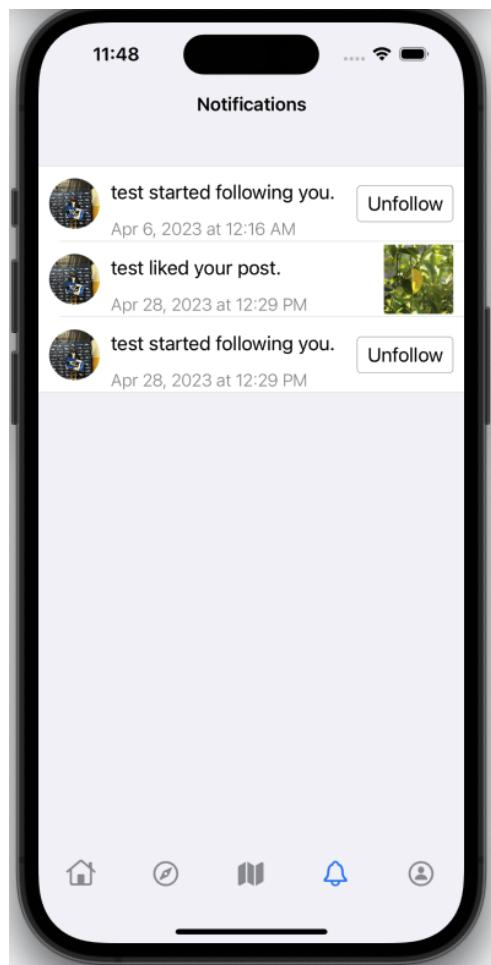


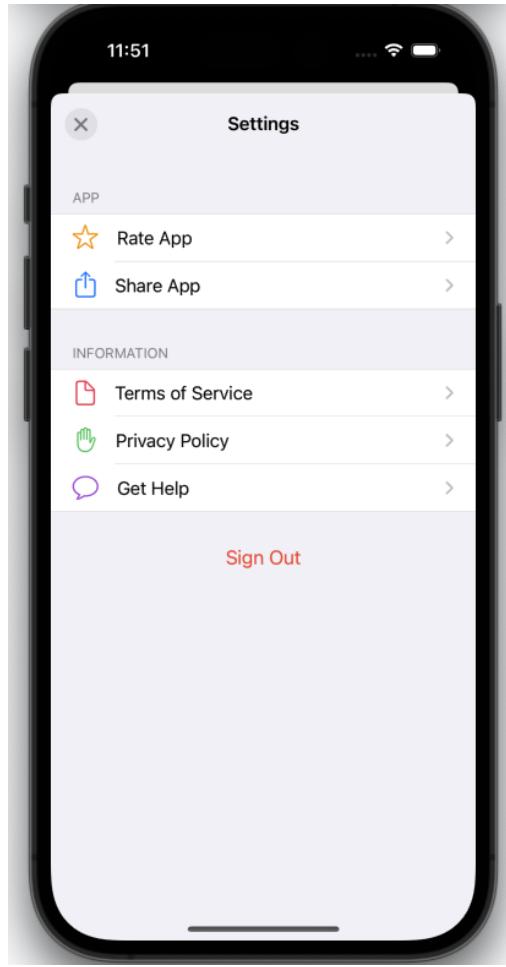
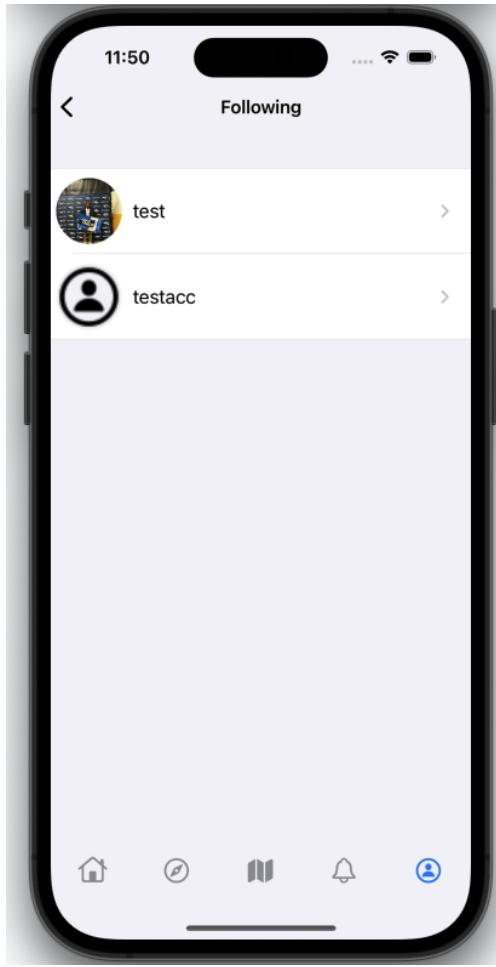


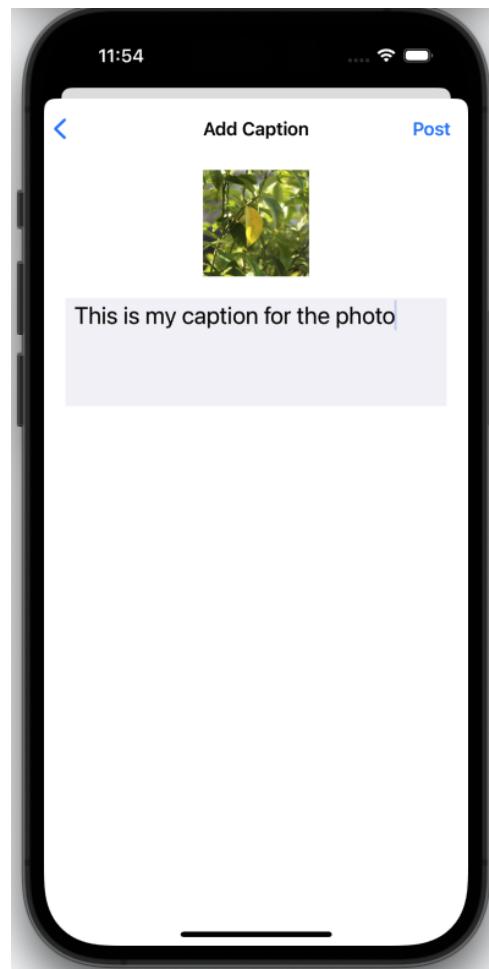
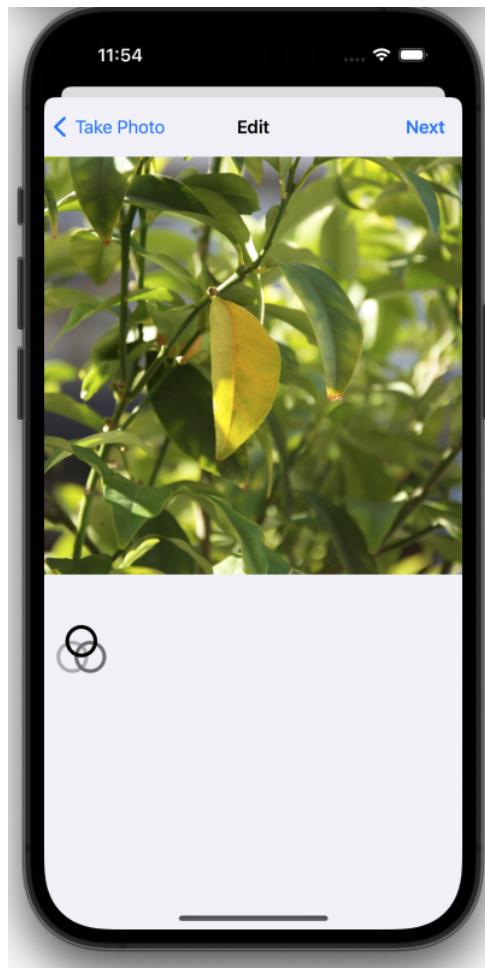


Xcode implementation









Discussion

To start, we can discuss some design decisions we've made throughout the entirety of the project, starting from basic planning to app development and polishing. Afterwards, we will discuss why we couldn't meet our initial project goals and some tips and critiques that would've helped us at the beginning.

Q: Why iOS only and not Android or both?

A: For the sake of the project we wanted to minimize the scope of where the app is presented to one platform. If we were to do this in Android, we would've had to refactor all the code to meet Android framework requirements and adjust any external packages that are only exclusive to iOS, which would've been cumbersome and outside of the scope of the project. Keeping Mouthpiece exclusively on one platform makes it easy for us developers to successfully meet project requirements without worrying about refactoring and repeating the pre-existing codebase to another platform, allowing for us to work on new features and bug testing. As for Android vs. iOS, we thought it was best to go with iOS platforms because many of the potential customers that were surveyed prior to the semester were predominately iPhone users. Because of this, the target audience is more geared to iOS users. As the project advances into a full fledged app available to the public, this gives it higher revenue potential with the bigger market. Another reason behind this decision is because of the IDE and Xcode environment as a whole is very simple to learn and easily accessible to debug and run code in testing.

Q: Why Firebase compared to the previous database of choice, AWS?

A: One factor that we've considered in this decision is the cost. Firebase gives a free plan that can be used to use most of the services provided on the platform. AWS, although it has a free tier, it requires more configuration and management to stay within these free limits. Also, many of the features provided with AWS are hidden behind a paywall. For the sake of the project, with it being small and not intensive on resource-usage, Firebase is perfect for us. Another factor is the integration with other Firebase services, such as user authentication, cloud storage, database querying, and most importantly the integration of our

Geolocation functionality using Google Maps API. Another reason is the overall ease of use. Firebase is known for its ease of use and simplicity. It has a user-friendly console and a well-documented API that makes it easy for us developers to test and debug the back-end. For AWS, it can be more complex and requires more technical expertise to set up and manage, making it cumbersome. Lastly, the Xcode environment as a whole works hand-in-hand with Firebase, making it easy to provide real-time updates from the client to the server. All in all, there were many factors that led us to this decision. There are pros of both sides, but for our purposes Firebase works exceptionally well and does the job.

Q: Why use Xcode's Coded UI vs. Storyboarding?

A: There are pros and cons of both sides. For us, we knew that we wanted to use Figma for the front-end design. Because of this, we decided to use Coded UI to easily get the foundation and layout of the app as a whole constructed, and worry about making it look visually pleasing later, integrating the Figma changes. Storyboarding, having a drag-and-drop style of making applications makes it easy to construct to a visually pleasing app using their interactive and easy-to-use GUI, but of course there are drawbacks. As we were learning Swift and Xcode in the Fall, we've learnt that debugging the layout when working with storyboard is very tedious. With all of this in mind, we decided to take the time to carefully construct the underlying layout and elements throughout the entire app programmatically so that in the end, we can simply integrate all of the UI/UX changes represented in Figma easily without worrying about any long-term difficulties with bugs that are prone to happening if we would've used Storyboarding.

Q: Why couldn't you meet the initial project requirements and how would you critique the execution of the project's development as a whole?

A: Having a well-crafted plan can be advantageous; nonetheless, excessive time allocation towards planning and learning may prove to be counterproductive. In our specific case, we dedicated an ample amount of time, the entire fall semester to be precise, towards acquiring proficiency in programming Swift through Udemy. This was a commendable effort aimed at ensuring that we possessed the requisite skills to develop the desired application without

encountering any impediments. Regrettably, despite our preparations, roadblocks did manifest themselves. One potential cause of suboptimality in our project was an unrealistic timeline for learning Swift programming and transitioning into app development. The decision to dedicate an entire semester to learning Swift on Udemy may have been well-intentioned, but it resulted in a significant delay in starting the app development process. This delay could have been exacerbated by a lack of prior experience in Swift programming and a failure to accurately estimate the amount of time needed to become proficient in the language. As a result, we may have set an unrealistic timeline for ourselves, leading to suboptimal outcomes in the project. For one of our project requirements, we wanted to add tournament functionality within the app to provide a competitive factor that can be tested amongst the users in the app. Being ambitious and naive of the task at hand, once developing the app we soon began to realize that this functionality would be very difficult to implement with the allotted time and knowledge that we had with Swift and other technologies needed. A tournament manager is a project by itself. We've learnt that not all project requirements will be met like we would've hoped but we've tried our best in building the other bits of functionality with the resources we had. All in all, poor planning with the project requirements to ensure that they are manageable and feasible to accomplish within the year is the main reason we couldn't meet initial project requirements. Not only this but with our limited knowledge of Swift programming language as a whole, as well as all the technologies we've included within the app made the learning curve that much harder, stacking all of these tasks of what we need to learn and apply in our app combined with our novice experience as a team in full-stack application development made it difficult to do everything we wanted to do. Nonetheless we got the underlying features completed and the basic social media functionality working. Going forward, we can add any additional features easily with the framework of the app constructed and the codebase organized to our liking.

Conclusion

In conclusion, Mouthpiece is a comprehensive sports platform that aims to revolutionize the way athletes connect and communicate with their teams and coaches. Through the use of advanced technologies such as Firebase, Google Maps API, and Xcode, we were able to create a user-friendly and efficient application that caters to the needs of athletes at all levels.

With a focus on team management, communication, and training, Mouthpiece has the potential to generate significant revenue through subscriptions, advertising, and e-commerce. The marketing strategy that we have outlined will help to increase the user base and brand awareness of Mouthpiece among the sports community.

While we have accomplished a great deal in the development and planning of Mouthpiece, there is still much work to be done. The future funding and development plans, as well as continued marketing efforts, will be critical in ensuring the long-term success of this platform.

We are grateful for the support and guidance of Florida State University, Dr. Noroozi, Dr. Hooker, and our dedicated development team in bringing Mouthpiece to life. We are confident that this platform will have a positive impact on the sports community and look forward to its continued growth and success.

Future Work

Looking ahead, there are several areas of future work that could help to ensure the long-term success of Mouthpiece. Firstly, additional funding will be necessary for continued development and improvement of the app. This could come from a variety of sources, including venture capital firms, angel investors, or even crowdfunding campaigns.

In terms of revenue potential, Mouthpiece has a number of promising avenues, including advertising revenue, e-commerce sales, and subscription fees. In order to fully realize these revenue streams, targeted marketing campaigns will be necessary. This could involve partnerships with local sports teams, collaborations with influencers in the sports community, and targeted online advertising campaigns.

Overall, the potential for Mouthpiece to become a leading platform for amateur athletes and sports enthusiasts is significant. With ongoing development, strategic marketing, and a focus on user experience, the app has the potential to become a go-to resource for athletes and sports enthusiasts across the country and beyond.

Appendix

Code and Standards

1. Respectful Communication: All communication within the team, with clients, and with users of the application must be respectful and professional. Any form of discrimination, harassment, or bullying will not be tolerated.
2. Quality Assurance: The development team is committed to delivering a high-quality product that meets the expectations of clients and users. All code must be thoroughly tested before release, and any bugs or issues must be promptly addressed.
3. Data Privacy and Security: The privacy and security of user data are of utmost importance. The development team will adhere to all relevant data protection laws and implement appropriate security measures to protect user data.
4. Transparency: The development team will maintain transparency throughout the development process, providing regular updates to clients and users on progress and any changes to the project scope.
5. Continuous Learning: The development team will stay up to date with the latest technologies and trends in app development, continually improving skills and knowledge to deliver the best possible product.

Code of Conduct

Mission Statement

Team 305 is committed to ensuring a positive work environment that supports professionalism, integrity, respect, and trust while also ensuring that weekly tasks are met accordingly in order to get a working prototype by the end of semester(s). Every member of the team will play a part in committing all effort possible to the creation of said prototype that best represents what we are all working hard to create.

Roles

Each team member is delegated a specific role based on their experience and skill sets and is responsible for all here-within:

Project Leader and Quality Assurance Representative –

Manages the team as a whole; develops a plan and timeline for the project, delegates tasks among group members according to their skill sets; finalizes all documents and provides input on other positions where needed. The team leader is responsible for promoting synergy and increased teamwork. If a problem arises, the team leader will act in the best interest of the project.

He keeps the communication flowing, both between team members and advisor. The team leader takes the lead in organizing, planning, and setting up meetings. In addition, he is responsible for keeping a record of all correspondence between the group and ‘minutes’ for the meetings. Finally he gives or facilitates presentations by individual team members and is responsible for overall project plans and progress

Team members:

Back-end

As part of the architecture of building applications, it is important to separate roles based on front and back-end. For members operating in the back-end role, these members will work behind the scenes in providing a framework and infrastructure that will layout the code so that it can be displayed and used by the front-end users to output the application for the end-user to see. Some duties include but aren’t limited to connecting to a database, managing the models of the code, and managing any business logic that the front-end users might need to use.

Front-end

For members operating in the front-end role, these members will work tangentially as a middle-ground with back-end users and the UI/UX designer to use the framework and infrastructure that they’ve built and create code that the end-users will see when loading

the application in regards to functionality. Some duties include but aren't limited to establishing front-end code that the UI/UX designer can output for the end-user to view.

UI/UX designer

This member will be working at the top of the stack, working with the front-end developers to use their functionality that they've used on the front-end to then output to the screen. Some duties include but aren't limited to managing how the app will look to the end-user through UI/UX elements (buttons/list/drop-down menus), app layout, etc.. While doing this, it is important for this designer to not only make a functional-looking app, but a good-looking modern-like application for the end-user to enjoy.

Writer

This member will be responsible for writing down meeting notes every week when the team meets to track progress over the year.

All Team Members:

- Work on certain tasks of the project
- Buys into the project goals and success
- Delivers on commitments
- Adopt team spirit
- Listen and contribute constructively (feedback)
- Be effective in trying to get message across
- Be open minded to others ideas
- Respect others roles and ideas

Communication

The main form of communication for the team will be over a Discord server. The benefits of this server will be to provide a group space for all the team members to speak to each other. It can also be used to announce meetings and/or any other important information that might arise, resources that can be used for the team members to get work done, a built-in video/calling chat system to communicate via voice/video and all-in-all provide a great, easy-to-use chatting system that works well with communicating to others. As a second form of communication we will be using SMS i.e. text messaging. This is another great way to easily get messages across if using the server doesn't work out.

Each group member must have Discord downloaded for the purposes of communication in general. It is important for all team members to check this server daily to check for important information and updates from the group.

If a meeting must be canceled, an announcement on the server must be sent to the group at least 24 hours in advance.

Any team member that cannot attend a meeting must give advance notice of 24 hours informing the group of his absence. Reason for absence will be appreciated but not required if personal. Repeated absences in violation with this agreement will not be tolerated.

Team Dynamics

The students will work as a team while allowing one another to feel free to make any suggestions or constructive criticisms without fear of being ridiculed and/or embarrassed. If any member on this team finds a task to be too difficult it is expected that the member should ask for help from the other teammates. If any member of the team feels they are not being respected or taken seriously, that member must bring it to the attention of the team in order for the issue to be resolved. We shall NOT let emotions dictate our actions. Everything done is for the benefit of the project and together everyone achieves more.

Ethics

Team members are required to be familiar with the NSPE Engineering Code of ethics as they are responsible for their obligations to the public, the client, the employer, and the profession. There will be stringent following of the NSPE Engineering Code of Ethics.

Dress Code

Team meetings will be held in casual attire, nothing specific. Advisor meetings and group presentations will also be casual attire with some-what of a professional/formal look, depending on the significance of the meeting/presentation as decided by the team per the event.

Attendance Policy

As discussed in the Communication section, it is important for all members of the team to meet during our weekly meetings, (On Fridays right after class lecture (~10:45am-11:45am)). Any team member that cannot attend a meeting must give advance notice of 24 hours informing the group of his absence. Reason for absence will be appreciated but not required if personal. Repeated absences in violation with this agreement will not be tolerated.

Weekly Tasks

Team members will participate in all meetings with the advisor and instructor. During said times ideas, project progress, budget, conflicts, timelines and due dates will be discussed. In addition, tasks will be delegated to team members during these meetings. Repeat absences will not be tolerated.

Engineering Drawings

Figure (ED.1) Entity-Relationship Diagram used for the database architecture:

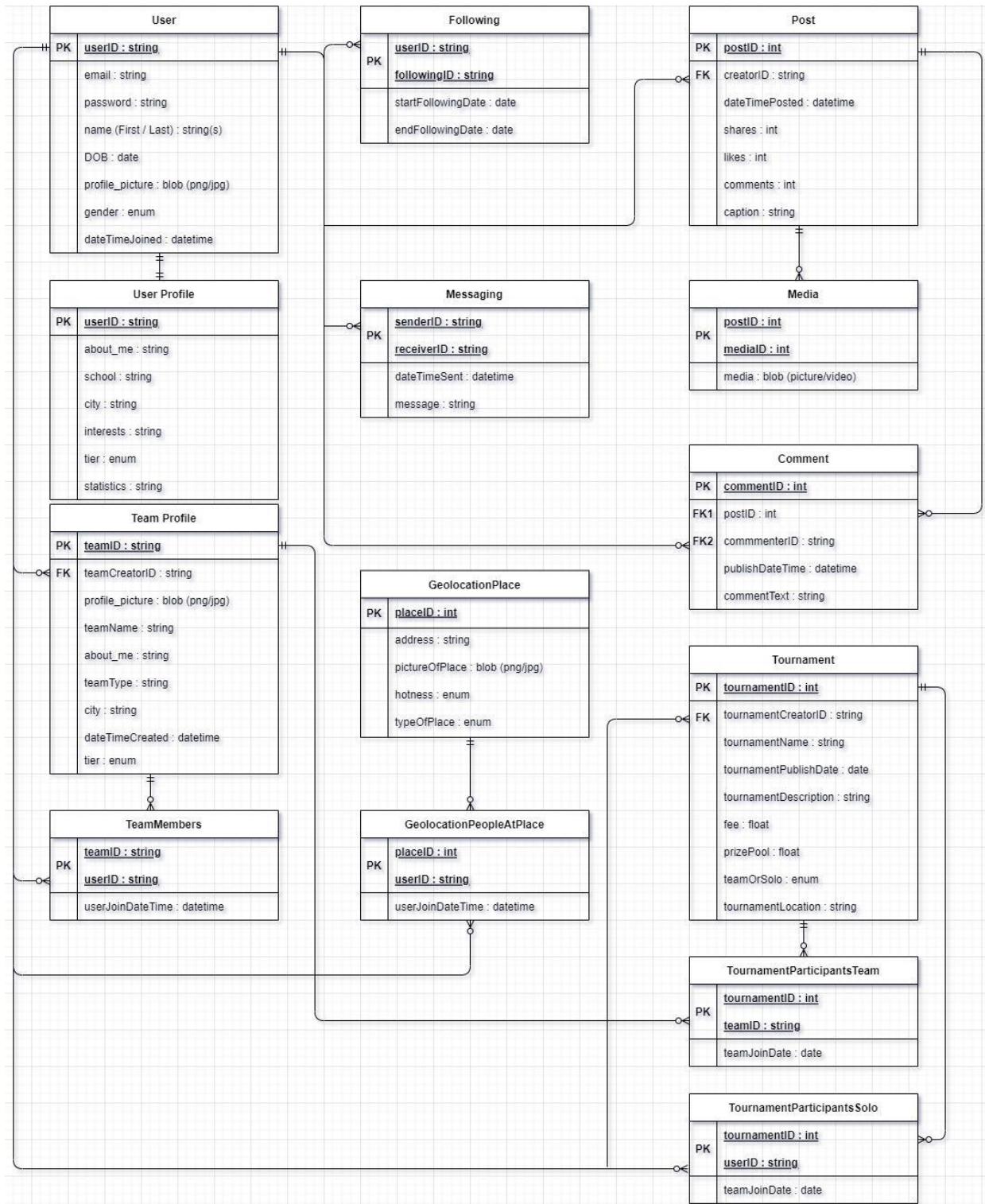


Figure (ED.2) Final Design's Entity-Relationship Diagram used for the database architecture:

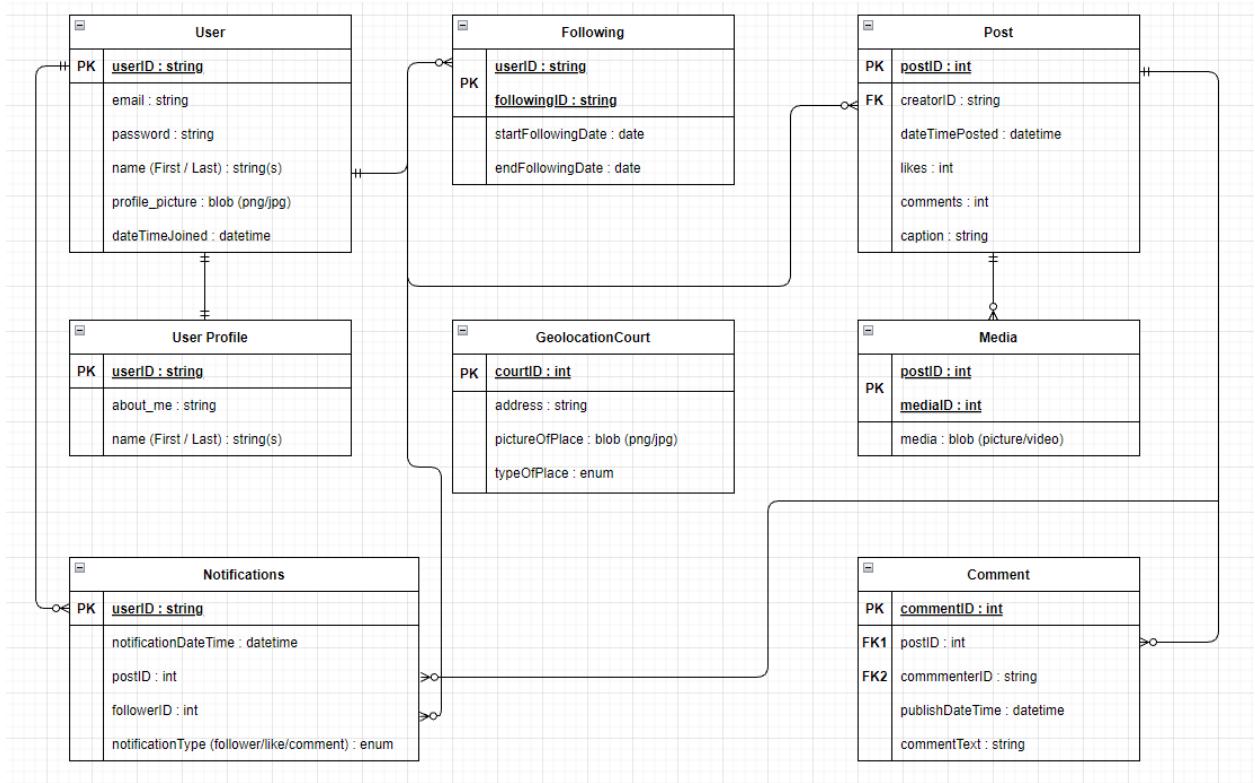


Figure (ED.3) Low-Fidelity Mock-Up

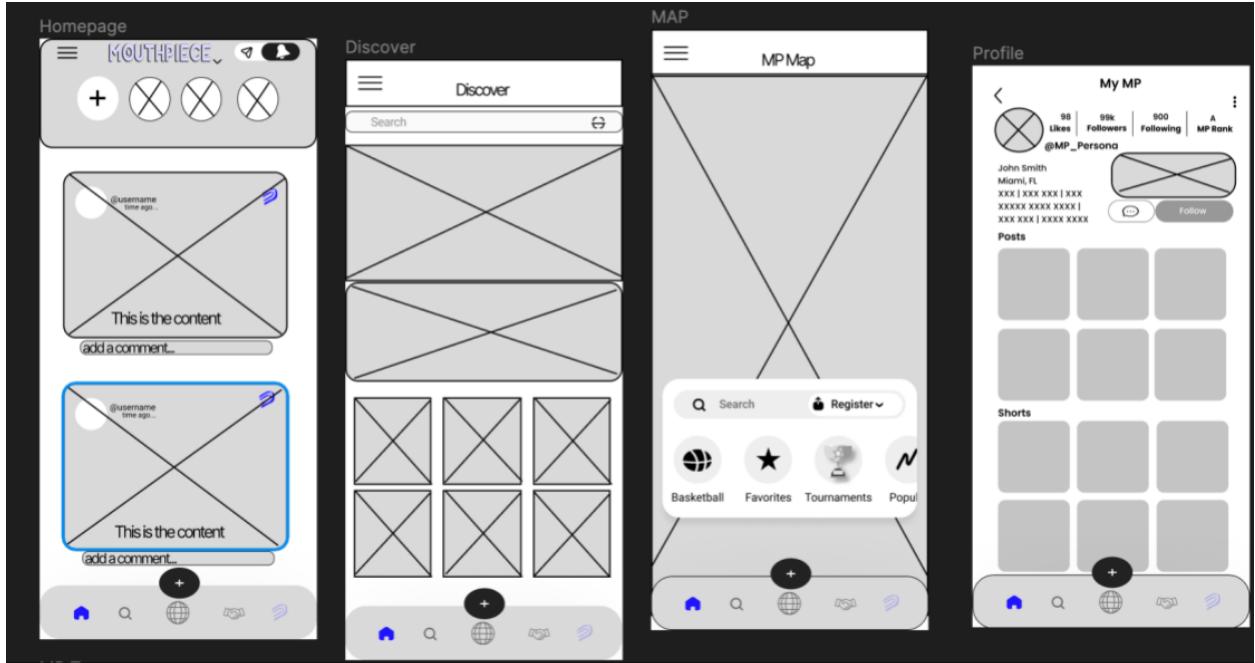
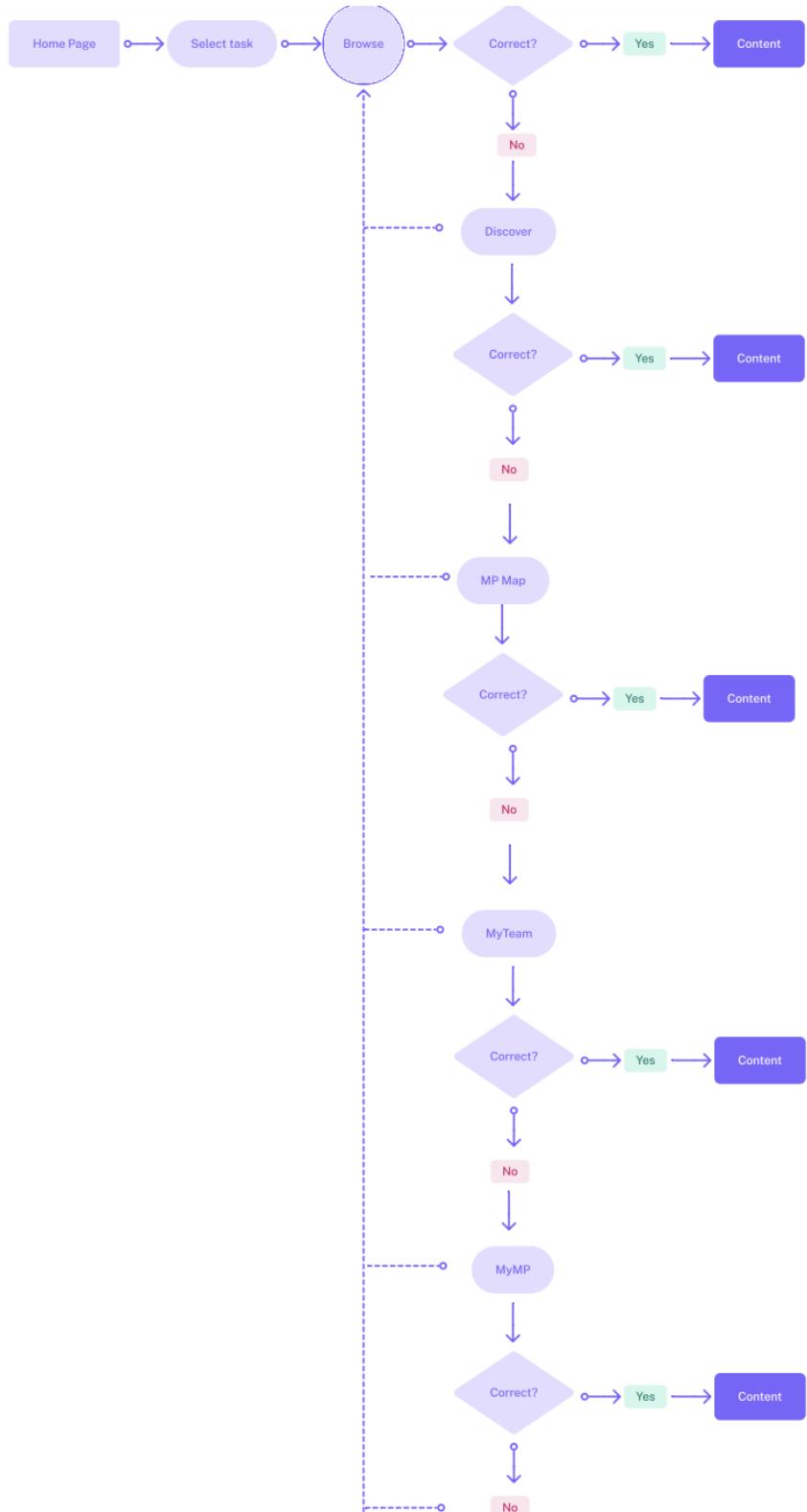


Figure (ED.4) User Task Flow Diagram



Risk Assessment

Project Hazard Assessment Worksheet								
PI/instructor: Babak Noroozi	Phone #: N/A	Dept.: Cmp. Eng	Start Date: August 2022	Revision number: N/A				
Project: Mouthpiece T305			Location(s): CoE					
Team member(s): Peyton Smith, Julien Denis, Christian Robinson, Ian Tyrell			Phone #: See Worksheet 2			Email: See Worksheet 2		
Experiment Steps	Location	Person assigned	Identify hazards or potential failure points	Control method	PPE	List proper method of hazardous waste disposal, if any.	Residual Risk	Specific rules based on the residual risk
Social Media – Psychological risk associated with use of the app.			Psychological hazard	Limit use / abstinence of use of application.			HAZARD: 1 CONSEQ: Minor Residual: Low	Safety controls are planned by both the worker and supervisor.
Geolocation – Privacy factor associated with use of location for geolocation functionality within the app			Psychological hazard – privacy risk	Disable location permission within the app or embedded in the smart device.			HAZARD: 1 CONSEQ: Minor Residual: Low	Safety controls are planned by both the worker and supervisor.
							HAZARD: CONSEQ: Residual:	
							HAZARD: CONSEQ: Residual:	
							HAZARD: CONSEQ: Residual:	

Project Hazard Assessment- Project Narrative

Name of Project: Mouthpiece T305	Date of submission: 04/20/2023	
Team member	Phone number	e-mail
Peyton Smith	8503700904	19peytonsmith@gmail.com
Ian Tyrell	9548048528	ianhtyrell@gmail.com
Julien Denis	8503450034	Ju.denis523@gmail.com
Christian Robinson	9546462390	Christanr00@gmail.com
Faculty mentor	Phone number	e-mail
Petru Andrei (Advisor)	N/A	pandrei@eng.famu.fsu.edu

Rewrite the project steps to include all safety measures taken for each step or combination of steps.
Be specific (don't just state "be careful").

Social Media – Psychological risk associated with use of the app. – Control Method: Limit use / abstinence of use of application.

Geolocation – Psychological/privacy risk factor associated with use of location for geolocation functionality within the app. – Control Method: Disable location permission within the app or embedded in the smart device.

Thinking about the accidents that have occurred or that you have identified as a risk, describe emergency response procedures to use.
--

If users experience psychological risks after use of the social media app, the most obvious response procedure would be to simply stop using the app. If it gets too serious, seek a therapist.

If users feel uncomfortable with the privacy factor associated with geolocation, an easy response would be to simply disable location so that the app is not tracking your where-abouts. This will ofcourse disable the geolocation functionality of the app.

List emergency response contact information:

- Call 911 for injuries, fires or other emergency situations
- Call your department representative to report a facility concern

Name	Phone number	Faculty or other COE emergency contact	Phone number
		Babak Noroozi (bnoroozi@eng.famu.fsu.edu)	N/A
		Petru Andrei (pandrei@eng.famu.fsu.edu)	N/A

Safety review signatures

Team member	Date	Faculty mentor	Date
Peyton Smith	04/20/2023	Petru Andrei	11/16/2022
Julien Denis	04/20/2023		
Christan Robinson	04/20/2023		

Revised 06-2019

Ian Tyrell	04/20/2023		

Report all accidents and near misses to the faculty mentor.

Testing Sheets

Test Writer Peyton Smith						
Test Case Name		Geolocation	Test ID #:	1		
Description		Tests the geolocation functionality of the app based off project requirements - Have a list of nearby courts with visible indicator of how many people are currently there.			Type	
Tester Information						
Name of Tester		Peyton Smith			Date	04/20/2023
Hardware/Software Ver.		Software			Time	4:00 PM
Setup		Start the application, log into the account, and load the geolocation tab, notice what happens. For testing purposes, simulate different locations and check.				
S t e p	Actions		Expected Results	P a s s	F a i l	N / A
1	Working map		When loaded it shows the google maps as a base and users can pan around the map to find different locations.	P		
2	Markers		With the map working, the markers are placed on the map to indicate nearby courts in the area.	P		
3	Crowd Meter		With each marker in place, a visible indicator for each marker is shown to indicate the activity at that location.	F		
Overall Test Result 2/3						

Test Writer Peyton Smith						
Test Case Name	Posting			Test ID #:	2	
Description	Core functionality for a social media app is being able to post content. Based off project requirements, users are to post pictures and videos to their account.			Type		
Tester Information						
Name of Tester	Peyton Smith			Date	04/20/2023	
Hardware/Software Ver.	Software			Time	4:20 PM	
Setup	<p>Start the application, log into the account, click the + button to add media to a post. Filter the media if needed, add your caption, and post the post.</p>					
S t e p	Actions	Expected Results	P a s s	F a i 1	N / A	Comments
1	Picture post from camera	Users can take a picture with their camera and post the post. The post will then save to the Firebase database and persist.	P			Works as intended
2	Picture post with preexisting picture	Users can choose a picture from your camera roll and post the post. The post will then save to the Firebase database and persist.	P			Works as intended, tested with different pictures of varying file sizes to test if the DB can handle it.
3	Video post	Users can either take a video or choose a pre-existing video from their camera roll to post.		F		Unable to take videos from the camera or choose a video from the camera roll. Internal error within Xcode tools.
Overall Test Result 2/3						

Test Writer Peyton Smith						
Test Case Name	Comments			Test ID #:	3	
Description	Core functionality for a social media app is being able to comment on posts. Based off project requirements, users are to be able to comment on a post that they've seen with no limitations.			Type		
Tester Information						
Name of Tester	Peyton Smith			Date	04/20/2023	
Hardware/Software Ver.	Software			Time	4:30 PM	
Setup	Start the application, log into the account, find a post and click the comment button and add your comment.					
S t e p	Actions	Expected Results	P a s s	F a i l	N / A	Comments
1	Add comment on any post	Users can add a comment to any post on the app. The comment will then save to the Firebase database and persist.	P			Works as intended, tested this with multiple posts, comments of various lengths and characters.
Overall Test Result 1/1						

Test Writer Peyton Smith						
Test Case Name	Liking			Test ID #:	4	
Description	Core functionality for a social media app is being able to like posts. Based off project requirements, users are to be able to like any post that they've seen with no limitations.			Type		
Tester Information						
Name of Tester	Peyton Smith			Date	04/20/2023	
Hardware/Software Ver.	Software			Time	4:40 PM	
Setup	Start the application, log into the account, find a post and click the like button.					
S t e p	Actions	Expected Results	P a s s	F a i l	N / A	
1	Like any post	Users can like any post on the app. The like will then save to the Firebase database and persist.	P			Works as intended, tested it with multiple posts and multiple times on a post, unliking will remove it from the database as expected.

Test Writer Peyton Smith			
Test Case Name	Liking	Test ID #:	4
Description	Core functionality for a social media app is being able to like posts. Based off project requirements, users are to be able to like any post that they've seen with no limitations.	Type	
Tester Information			

Name of Tester	Peyton Smith			Date	04/20/2023	
Hardware/Software Ver.	Software			Time	4:40 PM	
Setup	<p>Start the application, log into the account, find a post and click the like button.</p>					
S t e p	Actions	Expected Results	P a s s	F a i l	N / A	Comments
1	Like any post	Users can like any post on the app. The like will then save to the Firebase database and persist.	P			Works as intended, tested it with multiple posts and multiple times on a post, unliking will remove it from the database as expected.
Overall Test Result 1/1						

Test Writer Peyton Smith			
Test Case Name	Searching	Test ID #:	5
Description	Core functionality for a social media app is being able to search for users within the app. Based off project requirements, users are to be able to search for a specific user and it'll direct them to their respective page	Type	
Tester Information			
Name of Tester	Peyton Smith	Date	04/20/2023

Hardware/Software						
Ver.	Software	Time	4:40 PM			
Setup	Start the application, log into the account, go to the explore tab and click the search bar, type in a username, and go to their profile.					
S t e p	Actions	Expected Results	P a s s	F a i l	N / A	Comments
1	Search for a user	Users can type a username in the search bar and find a specific user on the app.	P			Works as intended, tested this with different types of users, the results list will dynamically update as users type in the name to fit potential options.
Overall Test Result 1/1						

Test Writer Peyton Smith			
Test Case Name	Notifications	Test ID #:	6
Description	Core functionality for a social media app is being able to notify users of alerts within the app. Based off project requirements, users are to be notified of users that follow them or when they like or comment on their post	Type	
Tester Information			
Name of Tester	Peyton Smith	Date	04/20/2023
Hardware/Software			
Ver.	Software	Time	5:00 PM

Setup		Start the application, log into the account, go to an account and follow them, like their post and comment to test all notification cases.				
S t e p	Actions	Expected Results	P a s s	F a i l	N / A	Comments
1	Following	Users are notified when a specific user follows them and redirects them to their account upon clicking.	P			Works as intended, prevented spam notifications when users follow/un-follow, tested with different accounts.
2	Like Post	Users are notified when a specific user likes one of their posts and redirects them to the respective post.	P			Works as intended, prevented spam notifications for when they like/un-like the post, tested with multiple accounts and various posts.
3	Comment Post	Users are notified when a specific user comments on one of their posts and redirects them to the respective post.	P			Works as intended, tested with multiple accounts and various posts.
Overall Test Result 3/3						

Test Writer Peyton Smith			
Test Case Name	Profile Editing	Test ID #:	7
Description	Core functionality for a social media app is having a unique profile, moreover being able to customize their profile. Based on project requirements, users can customize their info to be shown to other users on the app.	Type	
Tester Information			
Name of Tester	Peyton Smith	Date	04/20/2023
Hardware/Software Ver.	Software	Time	5:20 PM
Setup	Start the application, go to the profile section, and change your name/bio.		

S t e p	Actions	Expected Results	P a s s	F a i l	N / A	Comments
1	Change Name	Users can change the name displayed on their profile.	P			Works as intended, tested with names of varying lengths, persists on the Firebase DB as well.
2	Change Bio	Users can change the bio displayed on their profile.	P			Works as intended, tested with bios of varying lengths, persists on the Firebase DB as well.
Overall Test Result 2/2						

Test Writer Peyton Smith			
Test Case Name	Messaging	Test ID #:	8
Description	Core functionality for a social media app is being able to message other users on the app.	Type	
Tester Information			
Name of Tester	Peyton Smith	Date	04/20/2023
Hardware/Software Ver.	Software	Time	5:30 PM
Setup			

S t e p	Actions	Expected Results	P a s s	F a i l	N / A	Comments
1	Message User	Users can go to a profile and message them.			N/A	Time constraints prevented us from adding message functionality. Could've definitely been done with better management. Didn't necessarily fail, just incomplete.
Overall Test Result 0/1						

Test Writer Peyton Smith			
Test Case Name	Sign Up	Test ID #:	9
Description	Being able to sign up within a social media app is the first functionality that we needed to settle. Based on project requirements, users can enter their basic info and it'll curate a unique profile for them.	Type	
Tester Information			
Name of Tester	Peyton Smith	Date	04/20/2023
Hardware/Software Ver.	Software	Time	6:00 PM

Setup		Once the app loads, click the sign up button, enter the info and click create account.				
S t e p	Actions	Expected Results	P a s s	F a i l	N / A	Comments
1	Create Account	Users can add their info in the slots, and it'll successfully create the account for them.	P			Works as intended, users can put in their info, and it will persist on the Firebase DB and register the account. Fails for empty values for email/password as expected.
Overall Test Result 1/1						

Test Writer Peyton Smith			
Test Case Name	Explore Posts		
Description	One of our big goals for the app is to be able to discover new content within the app. Based on project requirements, users can find new posts within the app and even tailor their page with their own interests.	Type	
Tester Information			
Name of Tester	Peyton Smith	Date	04/20/2023
Hardware/Software Ver.	Software	Time	6:20 PM
Setup	Once the app loads, click on the explore tab and notice the posts.		

S t e p	Actions	Expected Results	P a s s	F a i l	N / A	Comments
1	See all posts within the app	Users can see a collection of all posts that are stored within the app.	P			Works as intended, tested with multiple accounts, added new posts to test the functionality.
2	See filtered posts based on personal interests	Users can see a tailor page of new posts based on their own interests.			N/A	Adding tags within the app is a goal that we've had and is necessary to complete this task, we can then use the tag system to display posts that use a specific tag. Although the test didn't necessarily fail, it isn't complete either.
Overall Test Result 1/2						

Test Writer Peyton Smith			
Test Case Name	Tournament	Test ID #:	9
Description	Our biggest goal for this app is to have built-in tournament functionality. Having the user see a list of nearby tournaments and being able to join them is based upon our initial project requirements.	Type	
Tester Information			
Name of Tester	Peyton Smith	Date	04/20/2023
Hardware/Software Ver.	Software	Time	6:30 PM
Setup			

S t e p	Actions	Expected Results	P a s s	F a i l	N / A	Comments
1	See Tournaments	Users can see a list of nearby tournaments in the area		F		Monumental task that we had no time in implementing. Huge overhead needed to even get the tournaments displayed much less functional.
2	Join Tournaments	Once the user finds a tournament, they can then join it.		F		
Overall Test Result 0/2						

Test Writer Peyton Smith			
Test Case Name	Geolocation	Test ID #:	1
Description	Tests the geolocation functionality of the app based off project requirements - Have a list of nearby courts with visible indicator of how many people are currently there.	Type	
Tester Information			
Name of Tester	Peyton Smith	Date	04/20/2023
Hardware/Software Ver.	Software	Time	4:00 PM

Setup		Start the application, log into the account, and load the geolocation tab, notice what happens. For testing purposes, simulate different locations and check.				
S t e p	Actions	Expected Results	P a s s	F a i l	N / A	Comments
1	Working map	When loaded it shows the google maps as a base and users can pan around the map to find different locations.	P			Works as intended
2	Markers	With the map working, the markers are placed on the map to indicate nearby courts in the area.	P			Works as intended, simulated with various locations, shows nearby basketball courts.
3	Crowd Meter	With each marker in place, a visible indicator for each marker is shown to indicate the activity at that location.		F		Time constraints made this unfeasible to setup, would combat overhead of joining/leaving a place.
Overall Test Result 2/3						

Code

Link to codebase:

https://drive.google.com/file/d/1p_-oBEHEQ8_v9FGqJjyx22s32Cu3RDBX/view?usp=share_link