

Principles of Software Engineering Spring 2023

[FAU: CEN 4010]

Milestone 1 Project Proposal and High-level Description

Team 2

Group: Team 2

John(Team Lead)

Team members:

Eli Antoine | eantoine2021@fau.edu

Garret Bell | gbell2021@fau.edu

Jayde Shaw | jaydeshaw2021@fau.edu

John Kantlehner | jkantlehner2018@fau.edu

Manny David | edavid2021@fau.edu

Milestone 1

Date: 02/20/2023

Revision History

TABLE OF CONTENTS

Executive Summary.....	3
Competitive Analysis	4
Data Definition	5
Overview, Scenarios, and Use Cases	8
Initial List of High-Level Functional Specifications	21
List of Non-Functional Requirements	31
High-Level System Architecture	37
Team Roles	39
Checklist	40

1. Executive Summary

Introducing a new and revolutionary social platform called Hobbly. This social platform that we have designed serves as a way to connect and chat with hobbyists and enthusiasts alike! Hobbly is a unique and interactive platform that offers a personalized experience for users to explore, discuss, and even monetize their hobbies. Whether you're a seasoned veteran or a curious newcomer, Hobbly offers a space to connect with like-minded individuals in a community built around your interests.

Hobbly is divided into sections, each dedicated to a particular hobby, allowing users to easily navigate to topics that interest them. Not only does it provide a forum for discussion, but it also includes an e-commerce aspect, allowing users to buy and sell merchandise related to their hobbies. Users can even host their own stores on the platform, giving them a unique opportunity to monetize their passion. Hobbly features a sign-up and login system to ensure that users have a secure and personalized experience, with the ability to track their activity and engage with other users. The platform also includes features such as comments and likes, as well as a system for reviewing posts and measuring their helpfulness.

One of the key advantages of Hobbly is its focus on community-building. By connecting users with like-minded individuals, Hobbly offers a space to share knowledge, learn from others, and form meaningful relationships. Additionally, the e-commerce aspect of the platform provides a unique opportunity for users to turn their passion into a source of income. Hobbly is novel in its combination of social networking and e-commerce, offering a one-stop-shop for all things hobby-related. Furthermore, Hobbly's emphasis on community-building sets it apart from other social networking sites that may have a more general focus. It targets a broad market sector, appealing to anyone with a hobby or interest, from casual enthusiasts to dedicated professionals.

In summary, Hobbly is a social platform that caters to hobbyists and enthusiasts, providing a unique and interactive experience for discussion, commerce, and community-building. Whether you're looking to connect with others, buy and sell merchandise, or simply learn more about your hobbies, Hobbly has everything you need in one convenient location.

2.Competitive Analysis

Company Profile



Company Info

Name: Hobbly

Name: Facebook

Name: Twitter

Description

Hobbly is a social networking site that allows you to connect with people all over the world pertaining to the hobbies they like.

Facebook is a social networking site that makes it feasible for you to communicate and share with family and friends online.

Twitter is a social networking site for friends and family to communicate and stay connected through the exchange of quick, frequent messages.

Product & Services

Hobbly offers the user to:

- Forum for discussions.
- E-commerce: users can buy or sell products related to their hobbies.
- Users can have their own stores on Hobbly.
- A Sign-up and login system to so that users have a safe experience.

Facebook offers the user to:

- Facebook (including the Facebook mobile app and in-app browser)
- Meta View
- Messenger
- Instagram (including apps like Boomerang)
- Shops

Twitter offers the user to:

- Media Studio.
- TweetDeck.
- Twitter Analytics.
- Embedded Tweets
- Twitter blue

Strengths & Weaknesses

Strengths

- Allows users to easily access more information and communication about their hobby.
- Forum for discussion, e-commerce: allowing users to purchase or sell merchandise based on their hobby.
- "Family" structure, allows you to find the right family for you based on you hobby. Giving you a safe space to enjoy your hobby.

Weaknesses

- Still in production.

Strengths

- Home to profiles from people of all ages
- Businesses sell their merchandise through facebook ads and sponsored post.
- Sharing life, current thoughts and more through statuses, comments, and stories.

Weaknesses

- There are known to be some privacy concerns with Facebook.
- Unwanted Ads

Strengths

- Twitter's hashtags are the most impactful.
- Popular for news and marketing.
- Retweet feature to share your thoughts amongst the world.

Weaknesses

- Fake Accounts are easy to make
- The algorithm doesn't always show what you are interested in and not always user appropriate.

The analysis of competitors' web sites will focus on five main features (homepage, design, navigation, search, content) (ask a local, wish list, add item). The competitive analysis will utilize a numerical scale: (1=bad, 2=poor, 3=fair, 4=good, 5=outstanding) and consists of 3 web sites chosen.

Features	HOBBLY	FACEBOOK	TWITTER
Homepage	5	4	4
Design	5	4	4
Navigation	4	5	4
Search	5	3	5
Content	5	2	5

Hobbly

The homepage for Hobbly is very attention grabbing, you will be able to understand the site upon arrival. There will be tips on how to understand the site should there be any difficulties. The site navigation is very efficient once you take the time to understand. The search is quick to help you get to the necessary information of your hobby of choice. The content is focused on hobbies and allows the users to have a community within their hobbies.

Facebook

The homepage for Facebook is straightforward, there is not really any eye catching design besides being able to use dark mode/ light mode. The site navigation is good when showing you who you may know, and things you may like. The search is easy to use. The content is focused on connecting with friends and family, sponsoring ads, facebook market, etc. The only downfall to this is many features you won't be granted access to until you have a certain amount of friends.

Twitter

The homepage for Twitter is simple, you cannot browse the site before making an account. The site navigation is consistent. The search is quick and gives you exactly what you are looking for when needed. The content is focused on people's everyday thoughts or daily news. Hashtags are useful when searching for a certain topic

Planned Advantages for

Hobbly

While there are many different social networking apps that can be chosen. Hobbyly has multiple features that take connecting and communicating with people all over the world to the next level. Hobbyly will allow the user to be safe and secure when entering sensitive information. All costs will be at a minimum when starting your own store for your favorite hobby, ensuring you make a good profit from the start. Hobbyly is open to everyone as we give you a variety of different hobbies so that what you want is what you get when you launch it. For users that are looking for a “safe space” to protect the love of your hobbies, Hobbyly is the perfect site for you.

This section serves as the “dictionary” of your document. It defines main terms, data structures and “items” or “entities” *at high or logical (not implementation) level* (e.g. name, meaning, usage, and NOT how the data is stored in memory) so it is easier to refer to them in the document. Focus on key terms (main data elements, actors, types of users etc.) specific for your application and not on general well know terms. These terms and their names must be used consistently from then on in all documents, user interface, in naming software components and database elements etc. In later milestones, you will add more implementation details for each item. You will later expand this section with more details.

3. Data Definitions

1. **Visitor (Actor / Entity)**
 - Platform user that is not logged in or has not created an account and has limited privileges.
 - Cannot view/post public content.
2. **User (Actor / Entity)**
 - Platform user with created account and credentials that is able to post, comment, like, shop, view public content, and more.
3. **Admin (Actor / Entity)**
 - Platform administrator with increased privileges that is able to moderate content, manage users, and view analytics
4. **Profile (Data Structure / Entity)**
 - Data structure of a user’s information that displays information about the user.
5. **Posts (Data Structure / Entity)**
 - Content posted by users that can be categorized using hobby tags
 - May include images, videos, text
 - Contains elements like comments, likes, date/time, location
6. **Login credentials (Data Structure / Item)**
 - Set of information used to access a user account on the platform.

- Includes
 - UserID
 - password
 - optional security question / two-factor authentication
- 7. **UserID (Data Structure / Item)**
 - username stored in a set of login credentials
- 8. **Password (Data Structure / Item)**
 - password stored in a set of login credentials
- 9. **Security Question / Two-Factor Authentication (Data Structure / Item)**
 - authentication stored in a set of login credentials.
- 10. **Followers (Data Structure / Entity)**
 - set of users that are opted-in to a specific user or hobby. Can be used as data to calculate hobby popularity / trending measurement
- 11. **Notifications (Data Structure / Entity)**
 - set of messages or alerts that notify the user of certain events such as posts, comments, likes, purchases etc.
- 12. **Hobby (Data Structure / Entity)**
 - An activity or interest that users can follow on the platform.
 - Includes hobby info, posts, forums, shop
- 13. **Tags (Data Structure / Item)**
 - Labels that can be added to categorize posts and products
 - Makes content easier to find on the platform
- 14. **Comments(Data Structure / Entity)**
 - Feedback posted on content by a user.
 - Allows for engagement between users.
 - Users can be tagged directly using @
- 15. **Likes (Data Structure / Item)**
 - A way to show appreciation or interest in a post.
 - Can be used to track popularity of a post or user on the platform.
- 16. **Report (Data Structure / Item)**
 - System for users to report inappropriate content on the platform.
 - Can include information about the report such as the reported content, violation reason, description, post date.
- 17. **Location (Data Structure / Item)**
 - represents a geographical location that can be linked to a post, profile, spot, etc.
- 18. **Spots (Data Structure / Item)**
 - Represents specific locations that are relevant to certain hobbies
 - Feedback can be provided in the form of likes and comments.
 - This feedback can be used to calculate a spot's popularity / rating
- 19. **Date/Time (Data Structure / Item)**
 - Data representing the time of a post, comment, etc.
 - Can be used to sort content.
- 20. **Data and analytics (Data Structure / Item)**
 - Data tracked and analyzed for:
 - Posts, Profiles/Users, Hobbies, Spots, Shops
 - Allows for sorting by popularity, time, etc.

- 21. **Trending (Data Structure / Item)**
 - Measurement used to track popularity of a user, post, hobby, spot, etc.
 - 22. **Forums (Data Structure / Entity)**
 - Message Board used for discussing specific hobby topics
 - 23. **Shop (Data Structure / Entity)**
 - Contains products pertaining to a certain hobby
 - 24. **Product (Data Structure / Entity)**
 - Item sold on the shop platform
 - 25. **Category (Data Structure / Item)**
 - Tag/Label used to categorize products
 - 26. **Cart (Data Structure / Entity)**
 - Data Structure to keep track of desired items and their quantities
 - 27. **Transaction (Data Structure / Entity)**
 - Data of the payment process including the receipt, payment method, and approval.
 - 28. **Shipping (Data Structure / Entity)**
- Shipping Information for a specific transaction**

4. Overview, scenarios, and use cases

Project Overview

Our project aims to create a user-friendly website that serves as a one-stop destination for hobby enthusiasts to connect, share knowledge, and discover new hobbies. The website will provide a platform for users to join hobby forums, search for specific hobby-related information, and connect with like-minded individuals. The primary objective of the website is to offer a personalized and engaging user experience that encourages users to share their passion for hobbies and explore new interests.

The website will have a simple and intuitive design that makes it easy for users to navigate and find what they need. The home page will feature a search bar that allows users to search for hobby-related information or specific forums. Additionally, users can create their profiles and connect with others who share similar interests. The website will also have a section where users can browse and discover new hobbies. All these aspects will be made interactive to enhance user experience through the use of responsive design techniques. This includes the implementation of visual effects, animations, and chatbots to assist users and answer any of their questions.

The website's content will be user-generated, and users will be able to share their knowledge and expertise with others. The platform will allow users to create and moderate their own hobby forums, where they can discuss various topics, ask questions, and share their experiences. The forums will be categorized by interest, making it easy for users to find topics that interest them. Also, to really dive into the user-generated aspect of this website, we encourage users to utilize our rating system which allows users to rate or review forums and user-generated content to help others find the best quality and most helpful information. The website will also have a robust search feature that allows users to filter their search results by various criteria, such as location, activity level, and popularity. Users will also be able to save their searches and receive notifications when new posts are added to a particular forum.

Our website is designed for hobby enthusiasts of all skill levels who are looking for a personalized and engaging platform to connect, share knowledge, and discover new interests. Users can easily create their profiles and start browsing through various forums, where they can interact with like-minded individuals and share their experiences. For example, a beginner-level user interested in photography can simply type "photography" in the search bar on the home page and find forums dedicated to photography basics, camera equipment, and even specific genres of photography such as wildlife or portrait photography. They can join these forums, ask questions, and receive guidance from more experienced users. On the other hand, a professional-level user looking to monetize their passion for art can create their own store on our platform and start selling their artwork to a wider audience. They can also participate in forums dedicated to art marketing, networking, and other business-related topics.

This range in potential users is what makes Hobbly such a community-building platform since inexperienced users on a specific hobby are able to gain knowledge and

experience from experienced users and thus expanding the community of that hobby. This in return also can create new bonds and friendships which furthers emulates Hoobly's initiative to make this platform a place where people feel safe and can be able to create a community full of hobbists and enthusiasts alike.

Overall, our project aims to provide a fun, engaging, and interactive platform for hobby enthusiasts to connect, share, and discover new interests. Our goal is to create a community-driven website that caters to users of all skill levels and interests.

Use Cases and Scenarios

Use Case- User Registration and Login

Description: This use case describes how the user can register and login to the website in order to access additional features and personalize their experience.

Scenario: The user arrives at the website and wants to create an account to save their favorite hobby forums and participate in discussions.

Actors:

1. User
2. System

Preconditions:

1. User has an active internet connection
2. System is available User is on the website's homepage

Primary Flow of Events:

1. User clicks on the "Register" or "Login" button on the homepage.
2. User is directed to the registration/login page.
3. User fills out the registration form, including their name, email address, and password.
4. System verifies the user's email address and creates a new account.
5. User logs in with their email address and password.
6. System verifies the user's account information and grants access to additional features.
7. User is redirected to their personalized homepage.

Alternate Flows

A1 User Enters Incorrect Login Credentials

1. If in step 5, the user enters an incorrect email address or password.
2. System displays a message informing the user that their login credentials were incorrect.
3. User can retry entering the correct login credentials or reset their password.

Postconditions:

1. The user has successfully registered and logged in to the website.
2. The user has access to additional features, such as saving favorite hobby forums and participating in discussions.
3. The user's personalized homepage is displayed with relevant information based on their interests.

Use Case - User Profile Creation

Description: This use case outlines the process for a user to create a profile on the website, including providing personal information and interests.

Actors:

1. User
2. System

Preconditions:

1. User has an active internet connection
2. System is available
3. User is on the website's homepage

Primary Flow of Events:

1. User navigates to the website and clicks on the "Sign Up" button.
2. User enters their name, email address, and creates a password
3. User fills out their profile information, including their location, hobbies, and interests
4. User clicks on the "Create Profile" button
5. System creates the user's profile and redirects them to their dashboard

Alternate Flows:

A1. User Enters Invalid Information

1. In step 4, if there are errors in the information provided by the user, the system displays an error message and prompts the user to correct the errors.
2. The system may specify which fields contain invalid information and what corrections are necessary.

3. The flow continues from step 2.

A2. User Already Has an Account

1. In step 5, if the user's information matches an existing profile, the system displays an error message indicating that the user already has an account.
2. The flow terminates.

Postconditions:

1. User has a new profile on the website with their personal information stored in the database.
2. User can log in to the website using their credentials.

Use Case - Search

Description: Use case describes the process of how the User will utilize the search feature of the system in order to search various hobby forums and e-commerce shops within various hobby forums

Scenario: The user comes to the website and wants to use the search capabilities to explore their hobbies and interests.

User arrives at the Hobbyly Home Page and uses the search bar field to enter their search criteria, such as the name of a specific hobby or related merchandise. The system displays the search results and the user chooses among the selections, which could include discussion threads, e-commerce listings, and user-generated content related to the search query.

Actors:

1. User
2. System

Preconditions:

1. User has an active internet connection
2. System is available
3. User is on the website's homepage

Primary Flow of Events:

1. User arrives at the Home Page of the website.
2. User navigates to the search bar field to enter search criteria for the hobby forums.
3. User enters keywords related to the hobby forum they are interested in and clicks on the search button.

4. The system displays the relevant search results based on the user's entered keywords.
5. User selects one of the hobby forums among the search results.
6. The system displays the selected hobby forum.
7. User can browse and participate in the hobby forum as desired.
8. Terminate Use Case: Search.

Alternate Flows:

A1. User Enters Prohibited Characters/Format into Search Bar

1. If in step 2, user enters prohibited characters or a prohibited format in the search bar. (User enters all numbers or punctuation)
2. Web site notifies user that the characters they used are prohibited
3. Return to step 1

A2. User Enters Unknown Location/Activity

1. If in step 2, user enters a location/activity that doesn't exist in the database.
2. System displays a message informing the user that the search results could not be found.
3. System suggests alternative search options or prompts the user to refine their search criteria.
4. User modifies their search criteria and returns to step 2.

A3. Search Results Return No Matches

1. If in step 3, the system returns no search results that match the user's criteria.
2. System displays a message informing the user that no search results could be found.
3. System suggests alternative search options or prompts the user to refine their search criteria.
4. User modifies their search criteria and returns to step 2.

A4. User Experiences Technical Difficulties

1. If at any point during the search process the user experiences technical difficulties.
2. System displays a message informing the user that technical difficulties have been encountered.
3. System suggests possible solutions or prompts the user to contact customer support for assistance.
4. User follows the instructions provided and either resolves the issue or terminates the use case.

Postconditions:

1. The system displays search results that are relevant and accurate to the user's search query.
2. The user is able to easily navigate and filter through the search results.
3. The user is able to access and view the selected discussion threads, e-commerce listings, or user-generated content related to their search query.
4. The user can save their search results and receive notifications when new posts related to their search query are added.
5. The user can rate or review the search results to help others find the best quality and most helpful information.

Use Case - Forum Creation

Description: This use case outlines the process for a user to create a new hobby forum on the website, including providing a forum name, description, and setting up forum rules.

Scenario:

A possible user may be an avid collector of vintage comic books or manga collector, and they want to create a forum where they can connect with other collectors, share their collection, and discuss various topics related to vintage comic books. They decide to use the hobby website to create his forum. They navigate to the hobby website and log in to their account. They then click on the "Create Forum" button on the website's home page to begin the process of creating a forum.

Actors:

1. User
2. System

Preconditions:

1. User has an active internet connection
2. System is available
3. User is logged in to their account

Primary Flow of Events:

1. User navigates to the "Create Forum" section of the website User enters the forum name and description
2. User sets up forum rules, including moderation policies and content guidelines

3. User clicks on the "Create Forum" button System creates the new forum and adds it to the list of available forums
4. User is redirected to the forum page where they can start adding content and inviting other users to join the forum

Alternate Flows:

A1. Forum Name Already Exists

1. In step 2, if the forum name entered by the user already exists, the system displays an error message and prompts the user to choose a different name.
2. The flow continues from step 2.

Postconditions:

1. User has successfully created a new hobby forum on the website with their personal information and rules stored in the database.
2. The new forum is added to the list of available forums. Other users can search for and join the new forum.

Use Case: Forum Post Creation

Description: This use case describes the process of how the User will create a new forum post in a hobby forum.

Scenario: User wants to create a post on a forum to share their knowledge and expertise regarding a hobby or they may want to get some feedback and help regarding something related to a hobby.

Actors:

1. User
2. System

Preconditions:

1. User has an active internet connection
2. System is available
3. User is logged in to the website

Primary Flow of Events

1. User navigates to the hobby forum they want to create a post in.
2. User clicks on the "Create Post" button.
3. User enters a title for the post.
4. User enters the post content in the text editor.
5. User can add attachments such as images or videos.

6. User can preview the post before submitting it.
7. User submits the post.
8. The system displays a confirmation message and the newly created post is visible in the forum.

Alternate Flows:**A1. User tries to submit the post without a title or content**

The system displays an error message and prompts the user to enter the required information.

A2. User encounters technical difficulties during the post creation process

The system displays a message informing the user that technical difficulties have been encountered and suggests possible solutions or prompts the user to contact customer support for assistance.

Postconditions:

1. A new forum post has been created.
2. The post can be viewed by other users in the hobby forum.
3. The user who created the post can edit or delete the post.

Use Case: Event Creation

Description: This use case describes the process of how the user will create a new event related to a hobby.

Actors:

1. User
2. System

Preconditions:

1. User has an active internet connection
2. System is available
3. User is logged in to the website

Scenario: Use wants to create a public or private event for the hobby forum in order to bring awareness to their hobby and inform people about their hobby. These events can also be a way of of them showcasing their hobby.

Primary Flow of Events:

1. User navigates to the "Events" section of the website.

2. User clicks on the "Create Event" button. User enters the event title, date, time, location, and description.
3. User can add an image or a video to the event page.
4. User can preview the event before submitting it.
5. User submits the event. The system displays a confirmation message and the newly created event is visible in the events section.

Alternate Flows:**A1. User tries to submit the event without providing the required information**

1. The system displays an error message and prompts the user to enter the required information.
2. User encounters technical difficulties during the event creation process.
3. The system displays a message informing the user that technical difficulties have been encountered and suggests possible solutions or prompts the user to contact customer support for assistance.

Postconditions:

1. A new event related to the hobby has been created.
2. The event can be viewed by other users in the events section of the website.
3. The user who created the event can edit or delete the event.

Use Case: Adding a Product to Cart

Description: This use case describes the process of how a user can add a product to their cart to make a purchase on the e-commerce section of the website.

Scenario: The user wants to buy a product from one of the e-commerce stores listed on the website.

Actors:

1. User
2. System

Preconditions:

1. User has an active internet connection
2. System is available
3. User has navigated to the product page on the e-commerce store

Primary Flow of Events:

1. User navigates to the product page of the e-commerce store.
2. User clicks the "Add to Cart" button.
3. The system adds the selected product to the user's cart and updates the cart icon to reflect the added item.
4. User can continue shopping or proceed to the checkout page to complete their purchase.

Alternate Flows:**User Attempts to Add Out-of-Stock Product to Cart**

1. If in step 2, the user attempts to add an out-of-stock product to their cart.
2. The system displays a message informing the user that the selected product is currently out of stock.
3. User can select another product or choose to wait for the product to become available again. User Removes Item from Cart If in step 4, the user decides to remove an item from their cart. The system removes the selected item from the user's cart and updates the cart icon to reflect the removed item.
4. User can continue shopping or choose to checkout with the remaining items in their cart.

Postconditions:

1. The system updates the user's cart to reflect the added item.
2. The user can easily view their current cart status and make changes as desired.
3. The user can proceed to the checkout page to complete their purchase.

Use Case: Following Other Users

Description: This use case describes the process of how a user can follow other users on the website, allowing them to see updates and notifications from those users.

Scenario: User wants to add another user's account to their following list in order to have their content on their feed and keep up with things that they are doing.

Actors:

1. User
2. System

Preconditions:

1. User has an active account on the website.
2. User is logged into their account on the website.
3. The user whose profile is being followed has an active account on the website.

Primary Flow of Events:

1. User logs into their account on the website.
2. User navigates to the profile page of the user they wish to follow.
3. User clicks the "Follow" button on the user's profile page.
4. The system adds the user to the list of users being followed by the user.
5. The system sends a notification to the user when the user they are following creates a new post or updates their profile.
6. The user can view updates from the users they follow in their news feed.
7. User can unfollow users at any time by clicking the "Unfollow" button on the user's profile page.

Alternate Flows:**A1. User attempts to follow a user they are already following**

The system does not add the user to the list of users being followed since they are already following the user.

A2. User attempts to follow a user who has blocked them.

The system does not add the user to the list of users being followed and displays a message to the user informing them that they cannot follow the user who has blocked them.

Postconditions:

The user is following the selected user's profile and receiving updates from them in their news feed.

Use Case: Flagging Inappropriate Content

Description: This use case describes the process of how a user can flag inappropriate content on the website, alerting the system moderators to review and potentially remove the content.

Scenario: User views a post or comment on the website that contains inappropriate content.

Actors:

1. User
2. System
3. Moderators

Preconditions:

1. User is logged into their account on the website.
2. Inappropriate content is present on the website.

Primary Flow of Events:

1. User views a post or comment on the website that contains inappropriate content.
2. User clicks the "Flag" button located next to the post or comment.
3. The system sends an alert to the website moderators, notifying them of the flagged content.
4. The moderators review the flagged content and take appropriate action, which may include removing the content or taking disciplinary action against the user who created the content.
5. The user receives a notification informing them of the action taken by the moderators.

Alternate Flows:**A1. The user clicks on "No" in step 3**

1. The system returns to the previous page.
2. The user can choose to flag the content again or continue browsing the website.

A2. The user is not able to locate the "flag" button in step 2.

1. The user clicks on "Report" or "Contact Us" button instead.
2. The system directs the user to a form where they can report the inappropriate content.
3. The user fills out the form and submits it.
4. The system sends a notification to the moderators.
5. The moderators review the reported content and take appropriate action.
6. The system notifies the user of the outcome.

Post-Conditions:

1. The inappropriate content has been flagged and brought to the attention of the moderators.
2. The moderators review the content and take appropriate action.
3. The user receives a notification of the outcome.

4. The system logs the flagging action for future reference.

Alternate Post-Conditions:

1. The moderators determine that the content is not inappropriate and take no action.
2. The moderators determine that the content is inappropriate and take action, such as deleting the content or suspending the user who posted it.
3. The user who posted the inappropriate content receives a warning or suspension.
4. The user who flagged the content receives a notification of the outcome.

5. High-level system requirements

Non-Member Expectation

1. Creating Account

○ 1.1 Overview

The system allows the user to create an account by storing UserID, Password, Date of Birth, First Name, Last name, Location, and phone number. The system checks for UserID duplicates and won't allow the User to create an account if the desired ID chosen by the User already exists in the System's Database. The system will also prevent the user from creating an account if the chosen password doesn't meet the minimum character requirements or if it doesn't match the re-enter password field. System shall prevent the creation of the user's account if the following fields aren't filled. Fields that have to be filled are First Name, Last Name, Location, UserID, Password, Re-enter Password, Phone number, and Date of Birth.

○ 1.2 Stimulus/Response Sequence - Create

1. User enters a UserID
2. System checks if UserID is available
3. User enters a Password
4. User re-enters Password for confirmation
5. System validates the Password to make sure it fits the password requirements
6. The user enters their First and Last Name
7. The user enters their Location.
8. The user enters their date of birth.
9. The user will hit 'create account' and the system will store all recorded information.
10. The system will confirm that the account was created to the user through an alert.
11. System will redirect the user back to the home page after the account has been created.

○ 1.3 Function requirement label

1. REQ 1 Creating Account

2. Browse the Hobbies

○ 2.1 Overview

The user will be able to browse by selecting “hobbies” in the menu bar. This will show the user the top 10 trending hobbies along with a link to the hobbies indexed alphabetically below. The system can let the user filter out the list that has been provided by using predetermined tags or a search bar. The user will not be able to add their hobbies since they are non-members.

○ 2.2 Stimulus/Response Sequence - Browse

1. User clicks “hobbies” in the menu bar.
2. System shall show the top 5 trending hobbies along with the link to all the hobbies indexed alphabetically below.
3. If the user clicks the link, it will take him to a page that shows all the hobbies organized alphabetically and their respective links.
4. The user can always navigate back to the home page via the navbar on top

○ 2.4 Stimulus/Response Sequence - Filter/Search

1. The user will be able to enter a hobby into the search bar.
2. System will give the user a list of hobbies with the most similar one on the top.
3. The user can always navigate back to the home page via the navbar on the top.

○ 2.5 Function requirement label

1. REQ 2.2 Browse
2. REQ 2.3 Filter/Search

3. View the Specific Hobby

○ 3.1 Overview

The User has the option to click on a specific hobby to then have an option to choose whether they wish to look at the forums, posts, and merchandise. Since the user isn't a member yet, they won't be able to edit or add anything.

○ 3.2 Stimulus/Response Sequence - View

1. The user will be able to click on a hobby they're interested in.
2. System will provide the user options and let them choose if they want to view the forums, posts or merchandise.
3. System shall have a button to allow the user to return to the index page.

○ 3.3 Stimulus/Response Sequence - Forums

1. The user will be able to click the forum of their choice and the page will take them to it.
2. Since the page is view only since the user is a non-member, they will not be able to add or make edits on anything.

○ 3.4 Stimulus/Response Sequence - Posts

1. The user will be able to view various posts related to the hobby as well as their respective comments.
2. Since the page is "view only" because the user is a non-member, they will not be able to create new posts. They will however be able to comment on other posts. Their comments will have a randomly generated name attached to it for the public to see

○ 3.5 Stimulus/Response Sequence - Merchandise

1. The user will be able to view all the merchandise associated with the hobby
2. If the user wishes to purchase an item, they can add it to cart and check out.

- **3.6 Stimulus/Response Sequence - Spots**

1. The user will click the “Spots” option when they click on a hobby
2. The user can then browser the different spots and see their ratings
3. The user will be able to rate different spots by clicking one out of the 5 stars under each spot if they’ve been there.
4. They will be prevented from creating a new rating but they will be able to edit an existing rating.

- **3.7 Function requirement label**

1. REQ 3.2 View
2. REQ 3.3 Forums
3. REQ 3.4 Posts
4. REQ 3.5 Merchandise

4. About Us

- **4.1 Overview**

The system would provide information to the guest user about all the different things the website has to offer. The system would also provide the user with links that will direct them to the top 10 hobbies or the hobby index.

- **4.2 Stimulus/Responsive Sequence - About**

1. The user clicks “About Us” in the navigation bar.
2. The user is redirected to the “About Us” page.
3. The user sees links with hobbies at the top of the page.
4. The user clicks a hobby and gets redirected to that hobby page.
5. The user can go back to the home page by clicking “Home” in the navigation bar.

- **4.3 Function requirement label**

1. REQ 4.2 About

Members Expectations

Pre: The members will be able to utilize the above features in addition to:

5. Edit Profile

○ 5.1 Overview

The user will be able to edit aspects of their profile like their UserID, profile picture, and a brief summary about themselves. The System will store the updated UserID, profile picture, and summary when the user clicks the save button. The user has type the information in a valid format in order for the system to store the information.

○ 5.2 Stimulus/Responsive Sequence - Edit

1. The user will go to “my account” from the nav bar and a drop down menu will be activated
2. The user will click on edit profile
3. The user will enter new information that they wish to be updated in their respective fields.
4. User shall click save
5. System will store their new ID, Profile Picture, and/or summary
6. System will refresh to user profile with updated information as a confirmation
7. System will automatically redirect the user back to their account page.

○ 5.3 Function requirement label

- i. REQ 5.2 Edit

6. Comment/Likes

○ 6.1 Overview

The user will be able to comment (within 255 characters) under any post page. The user will also be able to like any post or comment by double clicking it or clicking the like button.

- **6.2 Stimulus/Responsive Sequence - Comment**

1. The user will click the text box if they wish to make a comment.
2. The user will be able to write whatever comment they want within 250 characters.
3. In the event that the comment exceeds the character limit, an error message will appear and they will be prevented from typing anything else.
4. When they are ready to post their comment, they will be able to hit “post comment” and the comment will live.

- **6.3 Stimulus/Responsive Sequence - Edit Comment**

1. If the user wishes to edit a comment, they can right click it and click edit.
2. After the edit, they will click post and the edit will be posted.

- **6.4 Stimulus/Responsive Sequence - Like**

1. If the user double clicks a post or comment, a heart animation will show to indicate that they have liked the entity.
2. If the user clicks the like button, the animation will also play to indicate that they have liked the entity.

- **6.5 Stimulus/Responsive Sequence - Delete**

1. The user will right click the comment and click delete.
2. They will be prompted to make sure they want to delete.
3. If after clicking “delete” the second time, the comment will be deleted.

- **6.6 Function requirement label**

1. REQ 6.2 Comment
2. REQ 6.3 Edit Comment
3. REQ 6.4 Like
4. REQ 6.5 Delete

7. Posts

- **7.1 Overview**

The user will be able to create posts and create captions for them. In the event that revisions need to be made, they will be able to edit the posts. If need be, they will also be able to delete them.

○ **7.2 Stimulus/Responsive Sequence - Create**

1. The user will click the “Create” icon in the navbar.
2. The user will then be prompted on whether they wish to create a new post, a new forum, or a spot.
3. The user will select post and either be able to take a picture/video or select up to 5 pictures and videos in total to upload.
4. The user will then be prompted to select which hobby the post is about so that the post will be more easily accessible to the other users.
5. The user will then be prompted to create a caption for their post.
6. There will be an option to also use the user’s location to add a new to “Spot” associated with the hobby.
7. The user will see a demo of the post before they confirm it.
8. The user will then click “post” and will be automatically redirected to a page with the actual post.

○ **7.3 Stimulus/Responsive Sequence - Edit**

1. If the user wishes to edit, they may click the menu dropdown button on the post (...) and hit edit.
2. The user can then edit the comments, location, or tag associated with the post. The user will not be able to edit the associated media after posting.
3. The user will see a demo of the changes while being prompted if they want to make the edits final.
4. The user will then click “post” and the page will automatically refresh and the edits will be made.

○ **7.4 Stimulus/Responsive Sequence - Delete**

1. The user will click the menu dropdown on the post (...) and click delete.
2. They will be prompted to make sure they want to delete.
3. If after clicking “delete” the second time, the post will be deleted.

- **7.5 Function requirement label**

1. REQ 7.2 Create
2. REQ 7.3 Edit
3. REQ 7.4 Delete

8. Forums

- **8.1 Overview**

The user will be able to create forums for users to comment on. When someone responds to a comment, the comment they responded to will show attached to their comment above it.

- **8.2 Stimulus/Responsive Sequence - Create**

1. The user will click the “Create” icon in the navbar.
2. The user will then be prompted on whether they wish to create a new post, a new forum, or a new spot.
3. The user will select forum and be prompted to give it a Title and a Subject (what the forum is regarding).
4. The user will then be prompted to select which hobby the forum is about so that the forum will be more easily accessible to the other users.
5. The user will see a demo of the forum before they confirm it.
6. The user will then click “post” and will be automatically redirected to a page with the actual post.

- **8.3 Stimulus/Responsive Sequence - Reply**

1. The user will click “respond” at the bottom of the thread.
2. The user will then be able to provide a response within 500 characters.
3. If the response is longer than 500 characters then the user will be alerted and prevented from typing anything else until some characters are deleted.
4. The user will click post when they’re finished.
5. If the user wishes to edit a comment, they can click the (...) to the left of the comment and click edit.
6. After the edit, they will click post and the edit will be posted.
7. If the user wishes to delete a comment, they can click the (...) to the left of the comment and click delete.

8. Once they click delete, they will be prompted to ensure their decision.
9. Once they click delete for the second time, the comment will be deleted.

- **8.4 Stimulus/Responsive Sequence - Delete**

1. The user will click the menu dropdown at the top of the forum and click delete.
2. They will be prompted to make sure they want to delete.
3. If after clicking “delete” the second time, the entire forum thread will be deleted.

- **8.4 Function requirement label**

1. REQ 8.2 Create
2. REQ 8.3 Reply
3. REQ 8.4 Delete

9. Adding Spots

- **9.1 Overview**

User will be able to add “Spots” or places where other users may enjoy partaking in a hobby.

- **9.2 Stimulus/Responsive Sequence - Create**

1. The user will click the “Create” icon in the navbar.
2. The user will then be prompted on whether they wish to create a new post, a new forum, or a new spot.
3. The user will select “Spots” and will be prompted to enter a location.
4. Once the location has been selected, the user will be prompted to give a description of the spot and the activities and hobbies you can do there.
5. The user will then be prompted to select tags associated with the Spot to make it easier for the users to access.
6. The user will click create and they will be shown a demo of their Spot being added to their page
7. Once the user clicks create for the second time, the page will refresh and they will be taken to their Spots’ page.

- **9.3 Function requirement label**

1. REQ 9.2 Create

10. Follow Users and Hobbies

- **10.1 Overview**

User will be allowed to contact the developers via email. user shall type in their name, email, and their query. System shall store these fields and submit it once the user click the send button. System shall prevent the User from contacting developers directly. User have to submit a ticket and the ticket shall be redirected to the correct personnel.

- **10.2 Stimulus/Responsive Sequence - Search Users/Hobbies**

1. The user will click the search bar on the top.
2. They will type either a UserID or a hobby and the results will be organized based on relevance.
3. The user can then select the user or hobby and it will take them to the respective page.

- **10.2 Stimulus/Responsive Sequence - Follow**

1. The user will click the “follow” button at the desired user/hobby page.
2. The user will now be prompted to confirm
3. If the user confirms then they will now be following the new page.
4. If the user does not confirm then do not follow.

- **10.4 Function requirement label**

1. REQ 10.2 Search
2. REQ 10.3 Follow

6. Non-functional Requirements

Performance:

1. Responsiveness: The system will be responsive, being able to be seen on various monitor sizes. The system also acknowledges and designates acceptance of user input.
2. Test Requirements: The system will be tested and updated regularly. Regular Quality assurance through the site daily.
3. Reliability: The mean time between failures is that it must have 1 hour or less of downtime in a total of 2 months. This will show how reliable the system is without failing.
4. Minimum Bug Counts: Testing the system regular will ensure bugs will not repeat themselves.
 - No more than 5 bugs in the system during integration and testing.
 - No more than 3 bugs can remain in the system after delivery.
5. Execution Speed: Execution speed of the initial home page on a high-speed internet connection should load within 200-300 milliseconds, depending on the current cycle time.
6. Learnability: The system will be straightforward and have tips on how to navigate throughout the site.
7. Accessibility: The system should be accessible to users of all levels providing features to maximize accessibility.
8. Response time: The website should respond to user requests within 3 seconds on average to ensure a smooth user experience.
9. Throughput: The system should be capable of handling a minimum of 1000 concurrent users at any given time.
10. Scalability: The system should be scalable enough to handle a 50% increase in traffic without any significant impact on performance.
11. Availability: The website should have an uptime of 99.9% in any given month to ensure users can access the site whenever they need to.
12. Resource utilization: The website should be optimized to use system resources efficiently to minimize the risk of server overload and ensure a consistent level of performance for all users.

Ease of use:

1. Intuitive Navigation: The website should have an easy-to-use and intuitive navigation system that allows users to find what they are looking for quickly and easily.
2. Clear and Concise Language: The language used on the website should be clear, concise, and easy to understand, regardless of the user's skill level or background.
3. Consistent Layout: The website should have a consistent layout and design throughout, making it easier for users to navigate and find information.
4. Customizable Settings: The website should allow users to customize their settings and preferences, such as font size, color scheme, and layout, to make the website more user-friendly for them.
5. User Feedback: The website should provide users with a way to provide feedback on their experience, such as through a feedback form or survey, and use this feedback to improve the ease of use of the website.

Interoperability:

1. Compatibility with different browsers: The system should be able to operate smoothly and consistently across different browsers, including popular options such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari.
2. Integration with other systems: The system should be designed to integrate with other systems, such as external databases or APIs, to ensure seamless data exchange and smooth communication between different platforms.
3. Interoperability with different devices: The system should be able to function and provide the same user experience on different devices, including desktops, laptops, tablets, and smartphones.
4. Compliance with industry standards: The system should comply with established industry standards and protocols, such as HTTP, HTTPS, and REST, to facilitate data exchange and ensure interoperability with other systems.
5. Support for different file formats: The system should be able to read and interpret different file formats, such as images, videos, and documents, to ensure that users can upload and share various types of content seamlessly.

Security:

1. Authentication: The system should have strong authentication mechanisms to ensure the identity of users. This can be measured by requiring a minimum password length, enforcing password complexity rules, and using multi-factor authentication.
2. Encryption: The system should use encryption to protect sensitive data such as user credentials and personal information. This can be measured by using industry-standard encryption algorithms and protocols.
3. Access control: The system should have granular access control mechanisms to ensure that users can only access data and resources that they are authorized to use. This can be measured by implementing role-based access control (RBAC) and tracking access attempts.
4. Auditing and logging: The system should generate audit logs for all user activity and system events, allowing administrators to investigate security incidents and maintain compliance. This can be measured by implementing a logging framework that captures relevant events and storing logs for a defined retention period.
5. Vulnerability management: The system should be regularly scanned for vulnerabilities and updated with security patches as needed. This can be measured by conducting regular vulnerability assessments and ensuring that all patches and updates are applied in a timely manner.

Maintainability:

1. Code Maintainability: The code of the website should be easy to maintain and modify, with a maximum cyclomatic complexity of 10 for any individual function. This ensures that the code remains easy to understand and debug in the future.
2. Documentation: All code and system documentation should be up-to-date and easily accessible, with a documentation coverage of at least 80%. This will ensure that future maintenance and development efforts can proceed without delays caused by insufficient documentation.
3. Code Reusability: The code should be written in a modular and reusable manner, with at least 50% of the code being reusable across different parts of the website. This will ensure that new features and functionality can be added to the website without the need to rewrite large portions of the codebase.
4. Testability: The code should be designed with testability in mind, with a test coverage of at least 80%. This will ensure that future maintenance and development efforts can proceed with confidence, knowing that changes to the code are not introducing new bugs or issues.

5. Upgradability: The website should be designed with upgradability in mind, with the ability to easily update and replace components as needed. This includes a maximum dependency depth of 3, meaning that any given component or library should not have more than three layers of dependencies. This ensures that the website can be upgraded and maintained with minimal impact on other parts of the system.

Data Integrity:

1. Accuracy: The system must maintain a data accuracy rate of 99% or higher. This can be measured through regular data audits and accuracy checks.
2. Consistency: The system must maintain consistent data across all components and modules. Any discrepancies or inconsistencies must be detected and corrected promptly.
3. Completeness: The system must ensure that all required data is collected and stored in its entirety. Any missing data must be flagged and corrected promptly.
4. Validity: The system must ensure that all data entered into the system is valid and conforms to established data standards. Any invalid data must be flagged and corrected promptly.
5. Timeliness: The system must ensure that all data is entered and processed in a timely manner, according to established timelines and deadlines. The system should have the capability to track and report on any delays.

7. High-Level System Architecture

- Front End
 - React library used to make user interfaces
 - React router to manage routing
 - Axios package to make http requests and communicate with the back end
- Back End
 - Node.js is the language used for the back-end
 - Express.js is the framework for building the web application
 - handles requests from react in front end
 - Passport.js
 - used for authenticating requests and storing user info and hashed passwords on the MongoDB database
- Storage
 - MongoDB hosted on a cloud service to store data like posts, images, user information, etc.
- Network
 - Express.js framework
- Cloud
 - MongoDB Atlas
 - cloud database for storing platform data
 - Cloud service like Heroku or AWS to deploy the Node.js applications and store data.
- API
 - RESTful API using express and mongoose
- API Endpoints
 - User authentication
 - registering new users.
 - logging in a user
 - logging out a user
 - User profile
 - receiving and updating the profile information using requests

- Content management
 - requests for posting, viewing, updating, and deleting content such as posts or comments.
- Frameworks
 - React
 - Express.js
- Deployment
 - Potentially containerize the application using Docker / Kubernetes
- Supported Browsers
 - browsers that support HTML5, CSS3, and JavaScript
- Tools
 - VSCode
 - Github / Git for version control
 - Google Analytics or Mixpanel to track statistics on the platform

8. Team Roles

- Product Owner: Manny
- Team Lead: John
- Scrum master: Eli
- Full Stack Developer: Eli
- Front End Group
 - FE Lead: Jayde
 - FE Members: Jayde and Garrett
- Back End Group
 - BE Lead: John
 - BE Members: John, Manny
- Github master: Eli Antoine
- More masters to add as architecture is designed(SQL, etc)

9. Checklist

- a) Team decided on basic means of communications **DONE**
- b) Team found a time slot to meet outside of the class **DONE**
- c) Front and back end team leads chosen **DONE**
- d) Github master chosen **DONE**
- e) Team ready and able to use the chosen back and front-end frameworks
ON TRACK
- f) Skills of each team member defined and known to all **DONE**
- g) Team lead ensured that all team members read the final M1 and agree/understand it before submission **DONE**

10. Github Connected with M1 Assignment.

History Table

Revision	Objective #	Notes	Date
Revised executive summary	1	Eli	02/23/2023
Updated high-level system architecture	7	John	02/25/2023
Revised use cases	4	Eli	02/25/2023
Edited scenarios of use cases	4	Eli	02/25/2023
Added two new use cases: forum creation and forum post creation	4	Eli	02/25/2023
Updated section for initial list of high-level functional requirements	5	Manny	02/25/2023
Updated high-level system architecture	7	Manny	02/26/2023
Updated checklist section	9	Garrett	02/27/2023
Updated non-functional requirements section	6	Jayde	02/27/2023
Updated checklist section	9	Garrett	02/28/2023
Updated non-functional requirements section	6	Jayde	02/29/2023