Software Engineering

Team X – Project Report

‘FishON’ – Elaboration One Document

|  |  |
| --- | --- |
| **Team Members:** |  |
| **Semester:** |  |
| **Instructor:** | Brad Armitage |
| **Due Date:** |  |

**Table of Contents**

[Business Case: 3](#_Toc517183252)

[Risks: 4](#_Toc517183253)

[FURPS+: 5](#_Toc517183254)

[Diagrams: 7](#_Toc517183255)

[User Stories: 7](#_Toc517183256)

[Use Cases: 7](#_Toc517183257)

[Test Plans: 39](#_Toc517183258)

# Business Case:

There are several apps similar to our vision currently on the market.

**Fishbrain:**

Fishbrian is a social media fishing app on iOS and Android that connects anglers allowing them to post pictures with their fish, share fishing techniques, post spots where the caught fish, and shows lake info.

The app has premium and free accounts. Premium allows you to view lake contours and bait packages that work well with certain species of fish along with catch locations submitted by other users.

Fishbrain can link with Facebook to have friend lists synch up.

**Fishidy:**

Fishidy offers a contoured map that users can upload spot information along with pictures at the location of a fish caught.

Great for locations with high density of users but lacks a real public database beyond what users submit.

**Navionics:**

Navionics is GPS and lake contouring app that allows users to view and submit edits to lake contour.

There are no social media aspects, but it does offer the most accurate mapping info to users and is widely used in the fishing community.

All of these apps offer aspects of what we are trying to create but all fall short of executing our vision up to the standards we have set. None of these apps work off publicly available data, and often fall short in their execution of key features. For Fishbrain the map feature is poorly executed. Fishidy has social media aspects but they are limited and not as well implemented as Fishbrain. Navionics has no social media aspect however it does offer the best lake maps on the market, at a steep price to the user. To execute our vision we believe that developing our own app is the way to go rather than buying one of the apps currently available on the market.

# Risks:

The majority of our risks comes from working with external software, databases, and companies. To mitigate our risks, we will create more test cases in our test plans for these high risks items. We will also test the high risk cases more often than the lower risks cases. We're adding a "Report a Problem" feature so our users can report problems, especially with the high risks features. That way, we can troubleshoot for them and create test cases.

**Public data set**:

A big part of our app will be pulled from public data sets: lake maps, lake contours, lake temperature, and weather information. The risk is we don’t have control if their servers go down or if/when they make updates to their system that could affect how our app function. Which can affect our user experience and our competitive edge over other existing apps.

**Credit card payments platform:**

We are working with Dolphin Credit to handle credit cards payment transactions. The risk will be relying on Dolphin Credit to handle all payments and working with them on license/contracts.

**Working with Apple AppStore and Google Play Store:**

Since our application will work with iOS and Android, a part of our risk will be making sure we fulfill the terms and requirements set forth by Apple and Google Play Store, as well as maintain the fulfillment.

**Ads revenue from vendors:**

Some of our revenue will come from ads we charge companies to host on our sites. Working to get new companies, maintain current ones, and update the ads is a risk because we would have to work out contracts set forth by both parties.

**Integrating with email servers:**

Since users will be using their email address to establish an account with us, reset password, receive notifications and confirmations. The risk will be to find a relying on BlueWhale to be dependable and reliably send out emails we need them to.

**GPS system on users’ phone:**

We must work with the user’s phone GPS system to determine their location and populate their information if they so choose. Since it’s not through us directly, the reliability is unstable and poses a risk we must work with carefully. To mitigate this specific risks, when a user first logs in, we’ll ask them to enable our app’s access to their GPS. If we can’t locate their GPS location, we’ll ask them to check their settings to see if their phone’s GPS location is turned on. If all fails, we’ll use their input zip code to help us narrow down the location to their nearby area. These steps will help to increase the chance of a working GPS or narrow down the location for the ease of the user. The happier our users are, the more likely they’ll become a pro member and remain one.

# FURPS+:

The following section describes the FishON project Function, Usability, Reliability and Security related non-functional requirements. These help define the architectural requirements and key support features to efficiently run and maintain the application.

**Functional Requirements:**

1. Licensing
   1. The application will be deployed and maintained in both Apple App store and Google Play store.
   2. There will be a onetime fee to register with App store and Google Play store
   3. Upon registration, two administrators or staff members of the FishON IT team will maintain app store Unique ID and code deployment responsibility. (One primary and another as shadow backup)
   4. Credit cards and Payment transactions will be outsourced to ‘Dolphin Credit’ vendor
   5. The licensing and contracts with ‘Dolphin Credit’ will be reviewed and renewed on an annual basis
   6. All email transactions will be outsourced to ‘BlueWhale’ email service provider
2. Localization
   1. The application will be maintained and available to users in ‘English’ language only
3. Email Servies
   1. The FishON application will interact with ‘BlueWhale’ email service provider. The email service will handle all the email transactions to
4. Security
   1. The app will support ‘https’ protocol
   2. All user interactions with system will be on ‘https’ protocol on 4G LTE or WiFi networks
   3. The database will be encrypted and protected by passwords
   4. The passwords will be regularly changed every 90 days by the System Administrators
   5. Minimum two system administrators will be authorized to access the protected infrastructures like Database, Licenses and app store unique IDs
   6. All software licenses will be maintained in electronic vault for auditing and maintenance purposes
5. Replication
   1. FishON IT team will maintain 3 backups of the system information including databases

**Usability Requirements:**

1. Accessibility
   1. The system will be designed and developed to follow the app store & Google Play store guidelines
2. Aesthetics
   1. The development team will consider using the best practices used in User interface designs
   2. The look and feel of the application will be native to iOS or Android devices in which it is running
   3. The user interface will have consistent behavior in both iOS and Android devices. (Example: The home screen, menu options)

**Reliability Requirements:**

1. Availability
   1. The app will be deployed in App store and Google Play store
   2. The availability will depend on the app store uptime and availability.
   3. The database and related information will be available up to 99.95% of time
2. Recoverability
   1. When system detects a failure, the Recovery Point Objective (RPO) will be set to 1 hr from the time of failure
   2. When system detects a failure , the Recovery Time Objective (RTO) will be set to 15 mins from the time of failure

**Performance Requirements:**

1. Response Time
   1. When system interacts with other external systems such as payment processing system and database, the transaction will be complete in maximum of 10ms
   2. An appropriate error message will be displayed to user in case of timeout delays
2. Throughput
   1. The system will have a load balancing mechanism
   2. System will support maximum of 100 ‘Concurrent user’ transactions at a time

**Supportability Requirements:**

1. Auditability
   1. System will maintain the audit trail of
      1. User payment transactions
      2. User account deletion
2. Compatibility
   1. The application will run on iOS 9+ versions
   2. The application will run on 5.0 + (Lollipop) version onwards
3. Testability
   1. The app will be testable on local developer machine with iOS and Android development environment
   2. The app will be testable on iOS and Google Play Store sandboxes before being deployed to production

**“+” Design Constraints:**

* + 1. Database

1. The transaction data will be maintained in ‘Oracle 11G Release 2’ version
2. The social interaction data such as friends and contacts will be maintained in graph database management system Neo4

**“+” Implementation Constraints:**

1. External System – BlueWhale email service provider
   1. System will interface with email server to process and the email transactions
2. External System – ‘Dolphin Credit’ payment processing system
   1. System will interface with payment processing system to handle all the credit card payments
3. Application platforms
   1. The application will be deployed and run in iOS and Android operating system compatible devices

# Diagrams:

For our diagrams view the accompanying artifact (FishON Diagrams.pdf)

# User Stories:

For a list of user stories view the accompanying artifact (FishON User Stories.xlsx)

# Use Cases:

**List of use cases**

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID | Use Case Name | Description | Page |
| UC\_SI\_01 | Send Friend Request | This use case describes to send a friend request on the app |  |
| UC\_SI\_02 | Accept friend request | This use case describes accepting a friend request on the app |  |
| UC\_SI\_03 | Delete Friend | This use case describes how to delete a friend |  |
| UC\_SI\_05 | Comment on friend’s photo | This use case describes how to comment on another friend’s photo |  |
| UC\_SI\_10 | Post Photo to profile repository | This use case describes posting a photo to the user’s profile |  |
| UC\_SI\_15 | Post new location information | This use case describes how to upload information on a location |  |
| UC\_SI\_16 | Post new location information with access | This use case describes how to upload information on a location |  |
| UC\_SI\_17 | Add fish to location | This use case describes how to upload fish information on a location |  |
| UC\_PM\_01 | Creating a profile | This use case describes how a user can create a profile and potentially create a membership subscription |  |
| UC\_PM\_02 | Login | This use case describes how user logs in to the app using an email address and password |  |
| UC\_PM\_03 | Retrieves forgotten email address | This use case describes how user can get email address through "Forgot Email Address" |  |
| UC\_PM\_05 | Reset forgotten password | User can reset password through pressing Forgot Password on sign in page |  |
| UC\_PM\_07 | Update profile information | This use case describes how user can update basic profile information: First name, Last name, Phone number, Address, City, State, Zip code. |  |
| UC\_PM\_08 | Change password | This use case describes how user can change password from profile Account Settings page |  |
| UC\_PM\_11 | Ending Pro membership | This use case describes how user can change account from Pro to Enthusiast by ending Pro membership |  |
| UC\_PM\_12 | Upgrade to Pro | This use case describes how user can upgrade to a Pro membership by paying with credit card |  |
| UC\_PM\_16 | First time user | This use case describes how user can open app for the first time and enable GPS |  |
| UC\_SA\_01 | Delete User Account | This use case describes how a system administrator deletes an account from the system |  |
| UC\_SA\_02 | Review Event Log | This use case describes how the Systems Administrator review event log |  |
| UC\_SA\_03 | Monitor System | This use case describes how the Systems Administrator monitors system |  |
| UC\_SA\_04 | Add, Update, Delete Ads | This use case describes the Systems Administrator adding, deleting or updating an ad |  |
| UC\_SA\_05 | Generate Revenue Reports | This use describes how the System Administrator generates revenue reports |  |
| UC\_BI\_01 | View Area Map | This use case describes how a user will be able to view Area Map of current geo location |  |
| UC\_BI\_02 | View Lake Information | This use case describes how a user will be able to view lake information |  |
| UC\_BI\_03 | View Pro user feature | This use case describes system actions to determining the user type and allow user to view the relevant content allowed only to pro users |  |
| UC\_BI\_04 | View Activity Feed/Activity Dashboard | This use case describes how a user will be able to view Activity Feed/Activity Dashboard |  |
| UC\_BI\_05 | View weather information | This use case describes how a user will be able to view weather information for a specific address or zip code |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SI\_01 | | | | |
| Use Case Name | | Send Friend Request | | | | |
| Description | | This use case describes to send a friend request on the app | | | | |
| Actors | | User A (Tier 2 or 3), User B (Tier 2 or 3) | | | | |
| Trigger | | User A selects “add friend” button on User B’s profile | | | | |
| Pre-conditions | | User A and User B are not currently friends, User A is on User B’s profile | | | | |
| Post conditions | | User B receives friend request | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User A selects “add friend” button | | |  | |
| 2 | | System notifies User B that User A has added them as a friend | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
| UC\_SI\_02 | UC\_PM\_01 | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SI\_02 | | | | |
| Use Case Name | | Accept friend request | | | | |
| Description | | This use case describes accepting a friend request on the app | | | | |
| Actors | | User A (Tier 2 or 3), User B (Tier 2 or 3) | | | | |
| Trigger | | User A selects “add friend” button on User B’s profile | | | | |
| Pre-conditions | | User A and User B are not currently friends, User A has sent a friend request. Refer to UC\_SI\_01 | | | | |
| Post conditions | | User B accepts or denies the User A friend request | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | System sends User B a notification User A has added them as a friend | | |  | |
| 2 | | User B selects “accept” button | | | A1. User B selects “deny button” | |
| 3 | | System correlates User A and User B as friends in their contact list | | |  | |
| 4 | | System removes friend request | | |  | |
| Alternate Flows | | | | | | |
| A1 | | **User B denies the friend request** | | | | |
| Step | | **Description** | | | | |
| 1 | | User B selects “deny button” | | | | |
| 2 | | System removes the notification from User B | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
| UC\_SI\_01 | UC\_PM\_01 | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SI\_03 | | | | |
| Use Case Name | | Delete Friend | | | | |
| Description | | This use case describes how to delete a friend | | | | |
| Actors | | User A (Tier 2 or 3), User B (Tier 2 or 3) | | | | |
| Trigger | | User A selects “Delete Friend” button on User B’s profile | | | | |
| Pre-conditions | | User A and User B are currently friends, User A is on User B’s profile, User A is logged in | | | | |
| Post conditions | | User A and User B are no longer connected as friends | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User A selects “Delete Friend” button | | |  | |
| 2 | | System no longer correlates User A and User B as friends | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
| UC\_PM\_01 | UC\_SI\_01 | | UC\_SI\_02 |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SI\_05 | | | | |
| Use Case Name | | Comment on friend’s photo | | | | |
| Description | | This use case describes how to comment on another friend’s photo | | | | |
| Actors | | User A (Tier 2 or 3), User B (Tier 2 or 3) | | | | |
| Trigger | | User A selects “Comment” button on User B’s post | | | | |
| Pre-conditions | | User A and User B are friends, User A is viewing User B’s photo | | | | |
| Post conditions | | User B is notified of User A’s comment | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User A selects “Comment” button while viewing User B’s photo | | |  | |
| 2 | | System prompts User A with a text box to fill in | | |  | |
| 3 | | User A enters text into prompt | | |  | |
| 4 | | User A selects “Send” button | | | A1. User A selects “Cancel” button | |
| 5 | | System correlates comment with photo | | |  | |
| 6 | | System sends notification to User B that User A has posted a comment on their photo | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | **User A cancels comment** | | | | |
| 1 | | User A selects “Cancel” button | | | | |
| 2 | | System returns User A to photo viewing | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
| UC\_SI\_01 | UC\_SI\_02 | | UC\_PM\_01 | UC\_SI\_10 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SI\_10 | | | | |
| Use Case Name | | Post Photo to profile repository | | | | |
| Description | | This use case describes posting a photo to the user’s profile | | | | |
| Actors | | User (Tier 2 or 3) | | | | |
| Trigger | | User selects “add photo” button | | | | |
| Pre-conditions | | User is logged in, User is currently on their profile | | | | |
| Post conditions | | User successfully uploads photos to user profile repository | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects “add photo” button | | |  | |
| 2 | | System accesses the phone’s gallery | | | E1. There is no gallery | |
| 3 | | System prompts user to select a photo | | |  | |
| 4 | | User selects a photo from their gallery | | | E2. User exits gallery | |
| 5 | | System uploads photo to user Profile photo repository. | | |  | |
| 6 | | System prompts the user to make the photo ‘Private’ or ‘Public’ | | | A1. The user selects to make the photo remain as private in their collection | |
| 7 | | System makes the recently uploaded photo visible to all friends | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | The user selects to make the photo remain as private in their collection | | | | |
| 1 | | The uploaded photo remains as private. | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1. There is no pictures in the phone gallery | | | | | | |
| 1 | | System alerts user that gallery is missing | | | | |
| 2 | | System displays a message to close the app and restart | | | | |
| E2. User Exits Gallery | | | | | | |
| 1 | | System aborts process | | | | |
| 2 | | User is returned to their profile | | | | |
| Related Use Cases | | | | | | |
| UC\_PM\_01 |  | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SI\_15 | | | | |
| Use Case Name | | Post new location information | | | | |
| Description | | This use case describes how to upload information on a location | | | | |
| Actors | | User (Tier 2) | | | | |
| Trigger | | User selects “Add new location” button | | | | |
| Pre-conditions | | User is signed in | | | | |
| Post conditions | | User successfully uploads information to user generated database | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects “Add new location” button | | |  | |
| 2 | | System prompts user for GPS coordinates | | |  | |
| 3 | | User selects “Current Location” button | | | A1. User selects “Manually Enter” button  A2. User selects “Cancel” button | |
| 4 | | System accesses the phone’s GPS system | | |  | |
| 5 | | System fills GPS coordinates into location field | | |  | |
| 6 | | System prompts user for name | | |  | |
| 7 | | User enters name into prompt | | |  | |
| 8 | | System makes location available for all to see | | |  | |
| 9 | | System takes user to new location page | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | User manually enters GPS coordinates | | | | |
| 1 | | User selects “Manually Enter” button | | | | |
| 2 | | System prompts user for GPS coordinates | | | | |
| 3 | | User fills in fields for GPS coordinates | | | | |
| 4 | | Return to basic flow 6 | | | | |
| A2 | | User cancel’s process | | | | |
| 1 | | User selects “cancel” button | | | | |
| 2 | | System returns user to home page | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
| UC\_SI\_16 | UC\_SI\_17 | | UC\_PM\_02 |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SI\_16 | | | | |
| Use Case Name | | Post new location information with access | | | | |
| Description | | This use case describes how to upload information on a location | | | | |
| Actors | | User (Tier 3) | | | | |
| Trigger | | User selects “Add new location” button | | | | |
| Pre-conditions | | User is signed in | | | | |
| Post conditions | | User successfully uploads information to user generated database | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects “Add new location” button | | |  | |
| 2 | | System prompts user for GPS coordinates | | |  | |
| 3 | | User selects “Current Location” button | | | A1. User selects “Manually Enter” button  A2. User selects “Cancel” button | |
| 4 | | System accesses the phone’s GPS system | | |  | |
| 5 | | System fills GPS coordinates into location field | | |  | |
| 6 | | System prompts user for name | | |  | |
| 7 | | User enters name into prompt | | |  | |
| 8 | | System prompts user to select “public”, “private”, or “group” access | | | A3. User selects “private”  A4. User selects “group” | |
| 9 | | User selects “public” | | |  | |
| 10 | | System makes location available for all to see | | |  | |
| 11 | | System takes user to new location page | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | User manually enters GPS coordinates | | | | |
| 1 | | User selects “Manually Enter” button | | | | |
| 2 | | System prompts user for GPS coordinates | | | | |
| 3 | | User fills in fields for GPS coordinates | | | | |
| 4 | | Return to basic flow 6 | | | | |
| A2 | | User cancel’s process | | | | |
| 1 | | User selects “cancel” button | | | | |
| 2 | | System returns user to home page | | | | |
| A3 | | Making location private | | | | |
| 1 | | User selects “private” button | | | | |
| 2 | | System makes location only available to user | | | | |
| 3 | | System takes user to new private page | | | | |
| A4 | | Making location available to a group | | | | |
| 1 | | User selects “group” button | | | | |
| 2 | | System prompts user with available groups they are a part of | | | | |
| 3 | | User selects a group | | | | |
| 4 | | System makes location available to all users in the group | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
| UC\_SI\_15 | UC\_SI\_17 | | UC\_PM\_02 |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SI\_17 | | | | |
| Use Case Name | | Add fish to location | | | | |
| Description | | This use case describes how to upload fish information on a location | | | | |
| Actors | | User (Tier 2 and Tier 3) | | | | |
| Trigger | | User selects “Add fish” button | | | | |
| Pre-conditions | | User is signed in, location is created, user is on location page | | | | |
| Post conditions | | User successfully uploads fish information to location | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects “Add fish” button | | |  | |
| 2 | | System prompts user for time and | | |  | |
| 3 | | User selects “Current Time” button | | | A1. User selects “Skip”  A8. User selects “Manually Enter” Button | |
| 4 | | System accesses the phone’s internal clock | | |  | |
| 5 | | System fills in time and date field with information from the phone | | |  | |
| 6 | | System prompts the user for type of bait used | | | A2. User Selects “skip” | |
| 7 | | User types in type of bait | | |  | |
| 8 | | System prompts user for length of fish | | | A3. User Selects “skip” | |
| 9 | | User types in length of fish | | |  | |
| 10 | | System prompts user for weight of fish | | |  | |
| 11 | | User types in weight of fish | | |  | |
| 12 | | System prompts user to add photo | | | A4. User Selects “skip” | |
| 13 | | User selects “add photo” | | |  | |
| 14 | | System accesses the phone’s gallery | | | E1. There is no gallery | |
| 15 | | System prompts user to select a photo | | |  | |
| 16 | | User selects a photo from their gallery | | | E2. User exits gallery | |
| 17 | | System stores fish information in location | | |  | |
|  | |  | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | Skip entering information | | | | |
| 1 | | User selects “skip” button | | | | |
| 2 | | System leaves Time field empty | | | | |
| 3 | | System moves to basic flow 6 | | | | |
| A2 | | Skip entering information | | | | |
| 1 | | User selects “skip” button | | | | |
| 2 | | System leaves Bait field empty | | | | |
| 3 | | System moves to basic flow 8 | | | | |
| A3 | | Skip entering information | | | | |
| 1 | | User selects “skip” button | | | | |
| 2 | | System leaves length field blank | | | | |
| 3 | | System moves to basic flow 12 | | | | |
| A4 | | Skip posting photo | | | | |
| 1 | | User selects “skip” button | | | | |
| 2 | | System moves to basic flow 17 | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | There is no pictures in the phone gallery | | | | |
| 1 | | System alerts user that gallery is missing | | | | |
| 2 | | System displays a message to close the app and restart | | | | |
| E2 | | User Exits Gallery | | | | |
| 1 | | System aborts process | | | | |
| 2 | | User is returned to their profile | | | | |
| Related Use Cases | | | | | | |
| UC\_SI\_15 | UC\_SI\_16 | | UC\_PM\_02 |  |  |  |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_PM\_01 | |
| Use Case Name | User creating a profile | |
| Description | This use case describes how a user can create a profile and potentially create a membership subscription | |
| Actors | User (Tier 1) | |
| Trigger | User selects "Create a profile" | |
| Pre-conditions | 1. User downloaded 2. User opened the app | |
| Post conditions | Tier 1 User becomes Tier 2 | |
| Process Flow / Data Flow | | |
| <Place for flow diagrams> | | |
| Basic Flow | | |
| Step | **Description** | **Alternate Flow/ Error Flow #** |
| 1 | System prompts user to “Sign in”, "Create a profile", or "Skip for now" | A1 User selects “Skip for now”  A2 User selects “Sign in” |
| 2 | User selects "Create a profile" |  |
| 3 | System prompts user with terms and agreements with “I accept” or “I decline” options |  |
| 4 | User selects “I accept” | A3 User selects “I decline” |
| 5 | System takes user to Create a profile page. System prompts user for the following required fields   * First Name * Last Name * Phone Number * Address * City * State * Zip Code   With "Confirm" and "Cancel" buttons |  |
| 6 | User provides the required fields |  |
| 7 | User selects "Confirm" | E1 Missing required field(s)  A4 User selects "Cancel" |
| 8 | System takes user to “Do you want to take the next step to become a pro member to enjoy exclusive features such as quality lake contour and more?”  With “Yes, take me to the next level” and “No, take me back” buttons |  |
| 9 | User selects “No, take me back” | A5 User selects “Yes, take me to the next level” |
| 10 | System changes user’s account to Tier 2 |  |
| Alternate Flows | | |
| Step | **Description** | |
| A1 | **User selects “Skip for now”** | |
| 1 | System takes user to Tier 1 main page | |
| A2 | **User selects “Sign in”** | |
| 1 | Go to use case UC\_PM\_02 | |
| A3 | **User selects “I decline”** | |
| 1 | System takes user to Tier 1 main page | |
| A4 | **User selects "Cancel"** | |
| 1 | System takes user back to main homepage with three selection "Sign in", "Create a profile" or "Skip for now" | |
| A5 | **User selects “Yes, take me to the next level”** | |
| 1 | Go to use case UC\_PM\_12 Basic Flow 4 | |
| Exception / Error Flows | | |
| Step | **Description** | |
| E1 | **Missing required field(s)** | |
| 1 | System displays error message with missing information highlighted | |
| 2 | Go to Basic Flow 3 | |
| Extension Points | | |
|  | | |

|  |  |  |
| --- | --- | --- |
| Use Case ID | UC\_PM\_02 | |
| Use Case Name | User Login | |
| Description | This use case describes how user logs in to the app using an email address and password | |
| Actors | User (Tier 2 or 3) | |
| Trigger | User selects “Sign in” to access account | |
| Pre-conditions | User already created a profile | |
| Post conditions | User is logged into system and can view homepage. | |
| Process Flow / Data Flow | | |
| <Place for flow diagrams> | | |
| Basic Flow | | |
| Step | **Description** | **Alternate Flow/ Error Flow #** |
| 1 | System prompts user to “Sign in”, "Create a profile", or "Skip for now" |  |
| 1 | User selects “Sign in” | A1 User selects “Create a profile”  A2 User selects “Skip for now” |
| 2 | System prompts user for email address and password with “Sign In” or “Cancel” buttons |  |
| 3 | User enters email address and password |  |
| 4 | User selects “Sign In” | E1 Missing required fields  E2 User enters incorrect email address and or password  A3 User selects “Cancel” |
| 5 | System validates the entered email address and password |  |
| 6 | System logs user in |  |
| Alternate Flows | | |
| Step | **Description** | |
| A1 | **User selects “Create a profile”** | |
| 1 | Go to use case UC\_PM\_02 Basic Flow 3 | |
| A2 | **User selects “Skip for now”** | |
| 1 | System takes user to Tier 1 main page | |
| A3 | **User selects "Cancel"** | |
| 1 | System takes user back to main homepage with three selection "Sign in", "Create a profile" or "Skip for now" | |
|  |  | |
| Exception / Error Flows | | |
| Step | **Description** | |
| E1 | **Missing required fields** | |
| 1 | System prompts for missing fields | |
| 2 | Go to Basic Flow 3 | |
| E2 | **User enters incorrect email address and or password** | |
| 1 | System displays error message | |
| 2 | System prompts user to reenter email address and or password | |
| 3 | Go to Basic Flow 3 | |
| Extension Points | | |
|  | | |
|  | | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_PM\_03 | | | | |
| Use Case Name | | Retrieves forgotten email address | | | | |
| Description | | This use case describes how user can get email address through "Forgot Email Address" | | | | |
| Actors | | User (Tier 2 & 3) | | | | |
| Trigger | | User selects “Forgot Email” | | | | |
| Pre-conditions | | 1. User has previously created a profile, UC\_PM\_01 2. User selected “Sign in” on main prompt page | | | | |
| Post conditions | | User retrieved email address associated with account and is given option to go back to sign in | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects “Forgot Email” from the sign in page. | | |  | |
| 2 | | System prompts user for required account information to connect with existing email address   * First Name * Last Name * Phone Number * Zip Code   With “Confirm” or “Cancel” options | | |  | |
| 3 | | User provides required fields | | |  | |
| 4 | | User selects “Confirm” button | | | E1 Missing required field(s)  A1 User selects “Cancel” button | |
| 5 | | System connects provided information to existing account and retrieves email address | | | E2 System cannot connect provided information to existing account | |
| 6 | | System displays email address for user with button to “Go to sign in page” | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | **A1 User selects “Cancel” button** | | | | |
| 1 | | System takes user to sign in page | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | **Missing required field(s)** | | | | |
| 1 | | System displays error message with missing information highlighted | | | | |
| 2 | | Go to Basic Flow 3 | | | | |
| E2 | | **System cannot connect provided information to existing account** | | | | |
| 1 | | System displays message “The provided information does not match existing account. Please check the information and re-enter” | | | | |
| 2 | | Go to Basic Flow 3 | | | | |
| Related Use Case(s) | | | | | | |
| UC\_PM\_01 |  | |  |  | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_PM\_05 | | | | |
| Use Case Name | | Reset forgotten password | | | | |
| Description | | User can reset password through selecting Forgot Password on sign in page | | | | |
| Actors | | User (Tier 2 or 3) | | | | |
| Trigger | | User selects the ‘Forgot Password’ option on Sign in page. | | | | |
| Pre-conditions | | User is on the sign-in page | | | | |
| Post conditions | | User has a new password and is able to log in | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects “Forgot Password” from the sign in page. | | |  | |
| 2 | | System displays message verifying that user wants an email to be sent to their email address with options to reset password, “Continue” or “Cancel” | | |  | |
| 3 | | User selects “Continue” | | | A1 User selects “Cancel” | |
| 4 | | System prompts user to enter the user’s email address | | |  | |
| 5 | | User enters email address and selects “Enter” | | |  | |
| 6 | | System verifies the email address is in database | | | E1 Missing or wrong email address field | |
| 7 | | System sends the email to the specified email address with a link to change password | | |  | |
| 8 | | User click on link from email (External event outside of the app) | | |  | |
| 9 | | System takes user to Change Password page | | |  | |
| 10 | | System prompts the user to enter new password and reenter new password | | |  | |
| 11 | | User provides required fields and selects “Enter” | | | E2 Missing fields or non-matching fields | |
| 12 | | System validates that the two password fields are the same | | |  | |
| 13 | | System updates user password | | |  | |
| 14 | | System display message to user that their password is updated | | |  | |
| 15 | | System takes user to sign in page | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | **User cancels Forgot Password flow** | | | | |
| 1 | | User is taken back to the sign in page | | | | |
| 2 | | User can then enter correct email address and password to go to Homepage or go back to Basic Flow 1 | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | **User provides the wrong email address or doesn’t provide one** | | | | |
| 1 | | System displays error message | | | | |
| 2 | | Go to Basic Flow 4 | | | | |
| E2 | | **User provides non-matching fields or doesn’t provide both fields** | | | | |
| 1 | | System displays error message | | | | |
| 2 | | Go to Basic Flow 4 | | | | |
| Related Use Cases | | | | | | |
|  |  | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_PM\_07 | | | | |
| Use Case Name | | Update profile information | | | | |
| Description | | This use case describes how user can update basic profile information: First name, Last name, Phone number, Address, City, State, Zip code. | | | | |
| Actors | | User (Tier 2 & 3) | | | | |
| Trigger | | User selects “Update Profile Information” | | | | |
| Pre-conditions | | 1. User already created a profile 2. User is on “Manage Account” page | | | | |
| Post conditions | | User’s basic profile information is updated | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects “Update Profile Information while on “Manage Account” page | | |  | |
| 2 | | System takes user to Update Profile Information page | | |  | |
| 3 | | System auto-populates existing information with editable required fields   * First name * Last name * Phone number * Address * City * State * Zip code   With “Save” and “Cancel” options | | |  | |
| 4 | | User provides the new information into the required fields | | |  | |
| 5 | | User selects “Save” | | | E1 Missing required field(s) or invalid inputs for phone number, state, or zip code  A1 User selects “Cancel” | |
| 6 | | System updates user account information | | |  | |
| 7 | | System takes user back to Manage Account page | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | **User selects “Cancel”** | | | | |
| 1 | | System takes user back to Manage Account page | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | **Missing required field(s)** | | | | |
| 1 | | System displays error message with missing/invalid information highlighted | | | | |
| 2 | | Go to Basic Flow 8 | | | | |
| Related Use Case(s) | | | | | | |
|  |  | |  |  | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_PM\_08 | | | | |
| Use Case Name | | Change password | | | | |
| Description | | This use case describes how user can change password from profile Account Settings page | | | | |
| Actors | | User (Tier 2 & 3) | | | | |
| Trigger | | User selects “Change Password” | | | | |
| Pre-conditions | | 1. User has created a profile 2. User is on Manage Account Page | | | | |
| Post conditions | | 1. User’s password is updated in the system 2. System takes user back to Manage Account page | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects “Change Password” from Manage Account page | | |  | |
| 2 | | System takes user to Password Change page with required prompts   * Current password * New password * Re-enter new password   With “Confirm change” and “Cancel” options | | |  | |
| 3 | | User provides required fields | | | E1 Re-enter new password does not match new password | |
| 4 | | User selects “Confirm change” | | | A1 User selects “Cancel” | |
| 5 | | System updates user’s password | | |  | |
| 6 | | System takes user back to Manage Account page | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | **User selects “Cancel”** | | | | |
| 1 | | System takes user to Manage Account page | | | | |
| 2 | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | **Re-enter new password does not match new password** | | | | |
| 1 | | System displays new passwords don’t match on current Password Change page | | | | |
| 2 | | Go to Basic Flow 3 | | | | |
| Related Use Case(s) | | | | | | |
| UC\_PM\_02 | UC\_PM\_03 | |  |  | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_PM\_11 | | | | |
| Use Case Name | | Ending Pro membership | | | | |
| Description | | This use case describes how user can change account from Pro to Enthusiast by ending Pro membership | | | | |
| Actors | | User (Tier 3) | | | | |
| Trigger | | User selects “End Pro membership” | | | | |
| Pre-conditions | | 1. User is a Pro member, UC\_PM\_12 2. User is on Membership page | | | | |
| Post conditions | | User account’s level is now Enthusiast (Tier 2) | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects “End Pro Membership” from Membership page | | |  | |
| 2 | | System takes user to End Membership page 1 with message “Are you sure you want to end your pro membership with access to exclusive features?” with “Yes” and “No” options | | |  | |
| 3 | | User selects “Yes” | | | A1 User selects “No” | |
| 4 | | System takes user to End Membership page 2 with message “You will have Pro membership access until \_date\_\* and your credit card will not be charged for next month.”(\*when membership was purchased) with “Confirm” or “Cancel” options | | |  | |
| 5 | | User selects “Confirm” | | | A2 User selects “Cancel” | |
| 6 | | System updates user account’s level from Pro membership (Tier 3) to Enthusiast (Tier 2) | | |  | |
| 7 | | System stores user payment information for potential future upgrade | | |  | |
| 8 | | System takes user back to Membership page | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | **User selects “No”** | | | | |
| 1 | | System takes user back to Membership page | | | | |
| A2 | | **User selects “Cancel”** | | | | |
| 1 | | System takes user back to Membership page | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
|  | |  | | | | |
| Related Use Case(s) | | | | | | |
|  |  | |  |  | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_PM\_12 | | | | |
| Use Case Name | | Upgrade to Pro | | | | |
| Description | | This use case describes how user can upgrade to a Pro membership by paying with credit card | | | | |
| Actors | | User (Tier 2) | | | | |
| Trigger | | User wants to upgrade to Pro membership | | | | |
| Pre-conditions | | 1. User already created a profile 2. User is on “Manage Account” page | | | | |
| Post conditions | | User becomes a Pro member with access to enhanced features | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects Membership button | | |  | |
| 2 | | System takes user to Membership page | | |  | |
| 3 | | User chooses “Upgrade to Pro membership” | | |  | |
| 4 | | System displays monthly subscription price, updated terms and agreements, and asks user to confirm in agreement to proceed | | |  | |
| 5 | | User selects “Confirm” | | | A1 User selects “Cancel” | |
| 6 | | System populates the following field for user   * First Name * Last Name * Phone Number * Address * City * State * Zip Code | | |  | |
| 7 | | System prompts the user to enter billing information with following fields   * Credit card number * Name on card * Card expiration date * Card security code * Billing Address Line 1 * Billing Address Line 2 * Billing City * Billing State * Billing Zip Code   With “Checkout” and “Cancel” buttons | | |  | |
| 8 | | User verifies auto-populated fields and provides required fields | | |  | |
| 9 | | User selects “Check out” | | | E1 Missing required field(s) or invalid inputs for phone number, state, or zip code  A2 User selects “Cancel” button | |
| 10 | | System authorizes credit card information with credit card system data | | | E2 User enters invalid credit card information | |
| 11 | | System informs user their card will be charged this day next month if membership is monthly unless membership is cancelled and prompts user to “Complete transaction” or “Cancel” | | |  | |
| 12 | | User selects “Complete transaction” | | | A3 User selects “Cancel” button | |
| 13 | | System updates user account to Tier 3 | | |  | |
| 14 | | System takes user to the thank you and congratulations page. | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | **User selects the “Cancel”** | | | | |
| 1 | | System returns the user to the membership page | | | | |
| A2 | | **User selects the “Cancel”** | | | | |
| 1 | | System returns the user to the membership page | | | | |
| A3 | | **User selects “Cancel” button** | | | | |
| 1 | | System returns the user to the membership page | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | Missing required field(s) or invalid inputs for phone number, state, or zip code | | | | |
| 1 | | System displays error message with missing/invalid information highlighted | | | | |
| 2 | | Go to Basic Flow 8 | | | | |
| E2 | | **User enter invalid credit card information** | | | | |
| 1 | | System displays error message | | | | |
| 2 | | System takes user back to Check Out screen as in Basic Flow 8, with missing required fields highlighted | | | | |
| 3 | | Go to Basic Flow 9 | | | | |
| Related Use Cases | | | | | | |
|  |  | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_PM\_16 | | | | |
| Use Case Name | | First time user | | | | |
| Description | | This use case describes how user can open app for the first time and enable GPS | | | | |
| Actors | | User (Tier 1) | | | | |
| Trigger | | User open the application | | | | |
| Pre-conditions | | User has downloaded the application | | | | |
| Post conditions | | User will be in the main prompt page with GPS access enabled for application | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User opens the application | | |  | |
| 2 | | System shows loading screen | | |  | |
| 3 | | System loads the application | | |  | |
| 4 | | System prompt user to allow application to access phone’s GPS system, with “Allow” or “Not now” options | | | A1 User selects “Not now” | |
| 5 | | User selects “Allow” | | |  | |
| 6 | | System connects to phone’s GPS to access location successfully | | | E1 System is unable to connect to phone’s GPS | |
| 7 | | System takes user to main prompt page | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | **User selects “Not now”** | | | | |
| 1 | | Go to Basic Flow 7 | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | **System is unable to connect to phone’s GPS** | | | | |
| 1 | | System display message to user to check if phone’s GPS in settings is enabled | | | | |
| 2 | | Go to Basic Flow 6 | | | | |
| Related Use Case(s) | | | | | | |
| UC\_PM\_01 | UC\_BI\_01 | | UC\_BI\_02 |  | |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SA\_01 | | | | |
| Use Case Name | | Delete User Account | | | | |
| Description | | This use case describes how a system administrator deletes an account from the system | | | | |
| Actors | | Systems Administrator | | | | |
| Trigger | | A staff member wants to delete an inactive user account | | | | |
| Pre-conditions | | * Systems Administrator has permission to access all user accounts * An inactive account exists | | | | |
| Post conditions | | * Inactive user membership as pro-user is terminated * User cannot access the account | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | Administrator logs in with systems administrator capabilities | | |  | |
| 2 | | Administrator selects the account to delete | | |  | |
| 3 | | System prompts the administrator to confirm delete | | | A1. Administrator selects “Cancel” option | |
| 4 | | System deletes the account | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | Administrator selects “Cancel” option | | | | |
| 1 | | System returns the user to user management page | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
|  |  | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SA\_02 | | | | |
| Use Case Name | | Review Event Log | | | | |
| Description | | This use case describes how the Systems Administrator review event log | | | | |
| Actors | | Systems Administrator | | | | |
| Trigger | | Request sent to Systems Administrator to look up event | | | | |
| Pre-conditions | | Systems Administrator logged in with administrator capabilities | | | | |
| Post conditions | | Administrator is able conduct a review of events | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | Systems Administrator gets notified of an event | | |  | |
| 2 | | Administrator signs in with systems administrator capabilities | | |  | |
| 3 | | Administrator pulls events log | | |  | |
| 4 | | Administrator is able to review the specific event requested | | |  | |
| 5 | | Administrator provides feedback response | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
|  |  | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SA\_03 | | | | |
| Use Case Name | | Monitor System | | | | |
| Description | | This use case describes how the Systems Administrator monitors the system | | | | |
| Actors | | Systems Administrator | | | | |
| Trigger | | Monthly systems check requirement | | | | |
| Pre-conditions | | Systems Administrator logged in with administrator capabilities | | | | |
| Post conditions | | Administrator is able conduct an inspection of system | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | Systems Administrator receives prompt to checks the systems performance | | |  | |
| 2 | | Administrator signs in with systems administrator capabilities | | |  | |
| 3 | | Administrator runs a systems updates check | | |  | |
| 4 | | Administrator updates systems | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
|  |  | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SA\_04 | | | | |
| Use Case Name | | Add, Update, Delete Ads | | | | |
| Description | | This use case describes the Systems Administrator’s capabilities to add, update, or delete an ad | | | | |
| Actors | | Systems Administrator | | | | |
| Trigger | | Request to add update or delete an ad received | | | | |
| Pre-conditions | | Systems Administrator logged in with administrator capabilities | | | | |
| Post conditions | | Administrator is able to add, delete or update an ad | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | Systems Administrator gets notified of an event to add, delete, or update an ad | | |  | |
| 2 | | Administrator signs in with systems administrator capabilities | | |  | |
| 3 | | Administrator is able to view the specific event requested | | |  | |
| 4 | | Administrator adds, updates, or deletes ad | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
|  |  | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_SA\_05 | | | | |
| Use Case Name | | Generate Revenue Reports | | | | |
| Description | | This use case describes how the Systems Administrator generates revenue reports | | | | |
| Actors | | Systems Administrator | | | | |
| Trigger | | Request by management | | | | |
| Pre-conditions | | Systems Administrator logged in with administrator capabilities | | | | |
| Post conditions | | Administrator generates the revenue report | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | Systems Administrator receives request to generate a revenue report | | |  | |
| 2 | | Administrator signs in with systems administrator capabilities | | |  | |
| 3 | | Administrator runs finance report module | | |  | |
| 4 | | System generates the requested report | | |  | |
| 5 | | Administrator reviews results of system generated report | | |  | |
| 6 | | Administrator provides feedback response to upper management | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
|  |  | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_BI\_01 | | | | |
| Use Case Name | | View Area Map | | | | |
| Description | | This use case describes how a user will be able to view Area Map of current geo location | | | | |
| Actors | | User (Tier 1, 2 & 3) | | | | |
| Trigger | | User selects view map option in application | | | | |
| Pre-conditions | | 1. User has enabled the GPS/ Location settings in the device UC\_PM\_16 | | | | |
| Post conditions | | User view the Area Map of current geo location | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects View Map option | | |  | |
| 2 | | System prompts the user with View Map Page | | |  | |
| 3 | | System determines location via GPS/Location settings from device | | | **E1. Unable to determine GPS/Location settings**  **E2 E1 has run, still unable to determine location** | |
| 4 | | User views the Area Map of the current location with a marker (‘You are here’) for the current Longitude and Latitude | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | **Unable to determine GPS/Location settings** | | | | |
| 1 | | System displays unable to determine GPS/Location message to user | | | | |
| 2 | | System displays message to user to make sure GPS is enabled on phone in settings | | | | |
| 3 | | Go to Basic Flow 3 | | | | |
| E2 | | **E1 has run, still unable to determine location** | | | | |
| 1 | | System displays unable to determine GPS/Location message to user | | | | |
| 2 | | System uses user address’ zip code and narrow map area to zip code | | | | |
| 3 | | User views map of zip code and search for desired area | | | | |
| Related Use Cases | | | | | | |
| UC\_BI\_02 | UC\_PM\_16 | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_BI\_02 | | | | |
| Use Case Name | | View Lake Information | | | | |
| Description | | This use case describes how a user will be able to view lake information | | | | |
| Actors | | User (Tier 2 & 3) | | | | |
| Trigger | | User interacts with Area Map | | | | |
| Pre-conditions | | 1. User has already created a profile 2. User (Tier 2 & 3) are logged in 3. User is a paid subscriber | | | | |
| Post conditions | | User views the Lake information | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User interreacts with Area Map to view Lake information | | | A1. Search for nearby lake information | |
| 2 | | User views the following basic lake information   1. Lake clarity 2. Lake contour 3. Lake Temperature 4. Lake foliage 5. Lake hot/cold 6. GPS co-ordinates information 7. Boat Launch/Docking info 8. Public Access | | |  | |
| 3 | | User views the following additional options over the lake area map   1. Type of Fish 2. State and Federal Regulations 3. Invasive Species 4. Weather information 5. News and Articles related to local lake(s) 6. Friends Activity Feed | | |  | |
| 4 | | User selects an option to view more information about lake | | |  | |
| 5 | | User views the content or information | | | **E1. If not logged in, Prompt to Login / Create Profile** | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
| A1 | | **Search nearby lake information** | | | | |
| 1 | | User interacts with map the find nearby lake information by scrolling the area map | | | | |
| 2 | | Go to Step 2 in Basic Flow | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | **Prompt to Login / Create Profile** | | | | |
| 1 | | System determines user is not logged in | | | | |
| 2 | | System prompts the user with login screen / Create Profile screen | | | | |
| Related Use Cases | | | | | | |
| UC\_BI\_01 | UC\_PM\_01 | | UC\_PM\_02 | UC\_BI\_05 |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_BI\_03 | | | | |
| Use Case Name | | View Pro user feature | | | | |
| Description | | This use case describes system actions to determining the user type and allow user to view the relevant content allowed only to pro users | | | | |
| Actors | | User (Tier 1, Tier 2 & Tier 3) | | | | |
| Trigger | | User selects a restricted option only allowed to pro user in the application | | | | |
| Pre-conditions | | 1. User has already created a profile 2. User is logged in 3. User is a paid subscriber | | | | |
| Post conditions | | User views the feature or information allowed to pro users | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects a Pro user feature from the application | | |  | |
| 2 | | System determines user is a paid subscriber | | | **E1. If not paid subscriber, then prompt user to Upgrade to pro** | |
| 3 | | User view the relevant content or information | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | **Prompt user to Upgrade to Pro** | | | | |
| 1 | | System identifies user is not a paid subscriber | | | | |
| 2 | | Go to Upgrade to Pro Use Case UC\_PM\_12 | | | | |
| Related Use Cases | | | | | | |
| UC\_PM\_12 | UC\_BI\_02 | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_BI\_04 | | | | |
| Use Case Name | | View Activity Feed/Activity Dashboard | | | | |
| Description | | This use case describes how a user will be able to view Activity Feed/Activity Dashboard | | | | |
| Actors | | User (Tier 3) | | | | |
| Trigger | | User selects an option allowed only to | | | | |
| Pre-conditions | | 1. User is logged in 2. User is a paid subscriber | | | | |
| Post conditions | | User views the Activity Feed/Activity Dashboard contents | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects Activity Feed/Account Dashboard option from the menu | | |  | |
| 2 | | Go to View Pro user feature Use case (UC\_BI\_03) | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Related Use Cases | | | | | | |
| UC\_BI\_03 |  | |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Use Case ID | | UC\_BI\_05 | | | | |
| Use Case Name | | View weather information | | | | |
| Description | | This use case describes how a user will be able to view weather information for a specific address or zip code | | | | |
| Actors | | User (Tier 1, 2 & 3) | | | | |
| Trigger | | User selects weather icon/option from menu to view weather information | | | | |
| Pre-conditions | | 1. User is viewing lake information 2. Tier 2 & Tier 3 user will be logged in 3. User selects ‘Weather’ option from the main menu | | | | |
| Post conditions | | User views the weather information for the specific address or zip code | | | | |
| Process Flow / Data Flow | | | | | | |
| <Place for flow diagrams> | | | | | | |
| Basic Flow | | | | | | |
| Step | | **Description** | | | **Alternate Flow/ Error Flow #** | |
| 1 | | User selects weather icon available on a lake view | | |  | |
| 2 | | System fetches weather information of current lake | | | E1. Unable to determine the current location information | |
| 3 | | System displays weather information in the overlay | | | E2. System unable to fetch the weather information | |
| 4 | | User views the following weather details on the overlay screen   * Current temperature * Day’s High and Low temperature * Days forecast (Sunny, Cloudy Rainy) * Precipitation % * Wind (MPH) * Pressure * 5 Day forecasts (Date – [day-date], Forecast, High | Low) | | |  | |
| Alternate Flows | | | | | | |
| Step | | **Description** | | | | |
|  | |  | | | | |
| Exception / Error Flows | | | | | | |
| Step | | **Description** | | | | |
| E1 | | **Unable to determine the current location information** | | | | |
| 1 | | System displays an error message indicating that there is no GPS info available. | | | | |
| 2 | | System returns the user to Lake view page | | | | |
| E2 | | System unable to fetch the weather information | | | | |
| 1 | | System displays an error message indicating that unable to fetch weather information. | | | | |
| 2 | | System returns the user to Lake view page | | | | |
| Related Use Cases | | | | | | |
|  |  | |  |  |  |  |

# Test Plans:

|  |
| --- |
| **FishON Test Plan 1:** Logging in  **Introduction**: This test plan will cover logging in from the application on iOS systems, making sure that all possibilities of a user signing in is considered.  **Related Use Case:** UC\_PM\_02  **Environment:** iOS  **In scope:** The scope will be all the scenarios of a user logging on after an account has been created.  **Test cases:**   1. Successful sign in - email address is already in system and password is correct 2. Unsuccessful sign in – email address is in system but password is not correct, user is taken back to sign in page 3. Unsuccessful sign in – email address is not in system or is incorrect, user is taken back to sign in page 4. Unsuccessful sign in – user cancels sign in process, user is taken back to sign in page   **Out of scope:**   * creating a profile * logging in from android systems * paying for membership * interacting with application   **People:**   * FishON testers with test user accounts   **Risks:**   * Unsuccessful test cases will result in reconfiguration of the application * Defects will delay other testing schedules/implementation   **Tools:**   * Test user accounts creation and set up * Tracking tools   **Timescales:**   * Initial tests – 2 days: one for writing out the test cases and one for executing them and noting down results * Each time an update is implemented, testers will run through the test cases again |
| **FishON Test Plan 2:** Adding fish to location  **Introduction**: This test plan will cover adding a fish to the location page on iOS systems.  **Related Use Case:** UC\_SI\_17  **Environment:** iOS  **In scope:** The scope will cover all scenarios of completing the “add fish” process  **Test cases:**   1. Skip – User skips various prompts 2. Cancel – User cancels the process and is taken back to the location page 3. Invalid input – User enters information that is outside the acceptable range 4. Successful post – User successfully enters info and generates a fish report on the location page.   **Out of scope:**   * Creating a location   **People:**   * FishON testers with test user accounts   **Risks:**   * Unsuccessful test cases will result in reconfiguration of the application * Defects will delay other testing schedules/implementation * If unsuccessful the feature may need to be removed   **Tools:**   * Tracking tools   **Timescales:**   * Initial tests – 2 days: one for writing out the test cases and one for executing them and noting down results * Each time an update is implemented, testers will run through the test cases again |