

Adam Cumiskey  
David Hand  
Justin Wagner  
Marcelo Mazzotti Paes de Almeida  
Steven Zilberberg  
Tim Honeywell

Github: [github.com/srz2/CSC470](https://github.com/srz2/CSC470)  
Team: iOS  
MyM - Map your Moments

## Stage II: Requirements Modeling and Analysis

### Legal Issues:

#### **Security**

The application will provide different account modes for regular users and band or celebrity accounts (subscription users). Users will not be able to access data that others have posted without friending (other user accounts) or following (subscription accounts) that account first. Accounts will be password protected and use proven encryption techniques (we will not be reinventing the wheel). User data will be protected against malicious software and will not be at risk to be either compromised or deleted. The application will be protected against outside threats towards the server (mainly bad/invalid user input) in order to continue running and provide persistent service to users.

The application will be constructed under the premise of a “dumb” client. Heavy processing will be handled on the server side, allowing for validation of data in the server environment (gaining access to the server is much more difficult for hackers to accomplish). Data integration will be handled exclusively on server-side. The server will be fitted with strong access controls and session management to reduce stealing of permissions and session hijacking, respectively. The User Interface will be designed in order to reduce the amount of invalid data that could be passed to the server, although all data will be validated server-side as well. Most security protocols and testing procedures implemented will be taken directly from OWASP:

([https://www.owasp.org/index.php/OWASP\\_Mobile\\_Security\\_Project](https://www.owasp.org/index.php/OWASP_Mobile_Security_Project)).

#### **Backup and recovery**

All files uploaded to the application will be hosted and backed up on AWS S3 storage. The system will have at least two copies of all user content in order to prevent loss of data from an outside attack. Files will be recoverable and users will continue to have access to content once the server has recovered. Users will be able to quickly and easily terminate and erase their account, along with all attached content. All data will be deleted from storage, meaning accounts will not be recoverable whatsoever once they are terminated.

### **Legal issues**

The application will abide by several U.S. standards in regards to privacy of information. The Personal Data Privacy and Security Act of 2009 states that all private and confidential user information stored by a site should not be accessible to other users. Personal data and content on the application will be protected from unwarranted access. The application will abide by the Children's Online Privacy Protection Act of 1998 as well, which protects children under the age of 13 from releasing personal information that could be harmful to themselves or family members and friends.

The application will provide ways to block and filter illicit content, although a user's main line of defense will be to unfriend or unfollow users who upload content they do not agree with. Location tracking is an integral part of the application and users will be prompted to agree to tracking when opening the application for the first time. The application will provide a setting to adjust location tracking settings at a later time, despite the fact that disabling tracking removes most of the application's functionality.

The application will provide both a short form and long form privacy disclosure in order to inform users of the application's terms of service policy.

## Possible Applications of the System:

The primary intent of MyM is to serve as a location-based blog. Users will be given the opportunity to upload pictures, video, sound clips, and text posts, attaching them to specific geographic locations. This content is then displayed in the form of pins on a map.

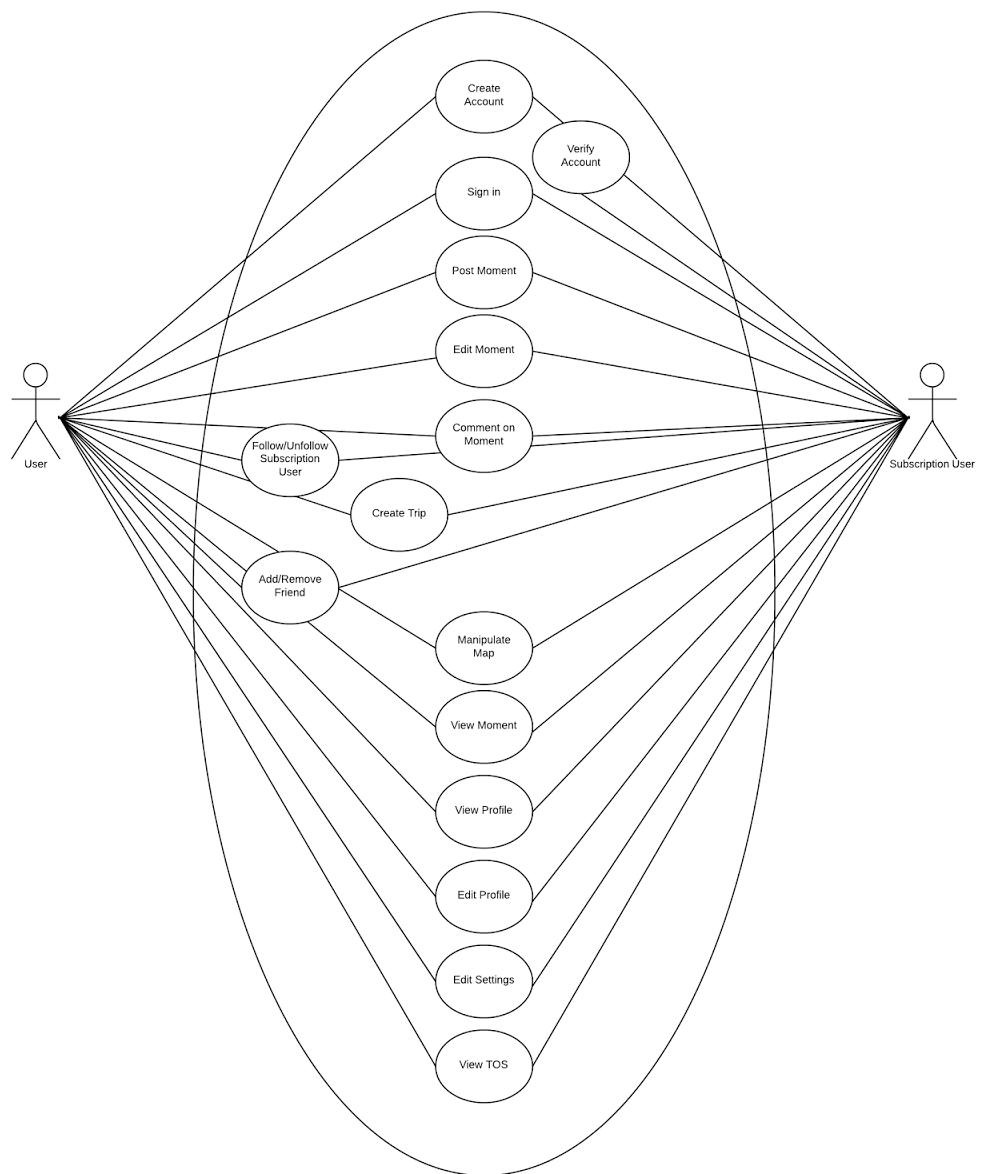
MyM lends itself to use as a digital travelogue. Because of its design as a location-based blogging platform, every piece of content a user captures on a trip can be used to show the progression of the trip through time. MyM specifically supports this idea by allowing users to organize different “moments” into collections called “trips,” which visually show the progress of a user by connecting moments on the map.

MyM would be an ideal tool for real-estate agents. The agent could photograph the houses currently on the market and write text posts containing their details. Text and photos for a particular house would naturally be grouped into a collection over its location on the map. Users could then follow the real-estate agent to have access to a dynamic catalog of houses.

Similarly, travel agents would be able to use MyM to advertise and promote travel destinations, adding photographs and videos to show off the locations and text to describe the vacations in more detail.

Bands and celebrities would benefit greatly from the “trips” functionality. While on tour, all the photos and reflections that are often posted to a traditional blog could be posted to a “trip” specific to the tour. Users following the band or celebrity would be able to track their progress and “hype up” upcoming events.

## Use Case Diagram for System:



## Use Case Descriptions:

Two actors:

- user
- subscription user (celebrity, band, speaker, travel agent, real-estate agent, etc)
  - subscription user is a verified account
  - subscription users are a subset of users

### Post Moment - Existing Photo

Primary actor: User/Subscription User

Goal in context: To post an existing photo as a moment in the application.

Preconditions: System is fully configured and app is installed on iPhone; user has logged in and photo to upload has already been taken.

Trigger: The user wants to upload a photo.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user will be navigated to the default application screen (the map).
3. The user will select the '+' icon at the bottom of the screen.
4. The icon will expand to display several additional icons: 'Camera,' 'Note,' and 'Sound.'
5. The user will select the 'Camera' icon.
6. The application will navigate to a camera interface, where the user can select to choose from existing photos/videos by selecting the bottom right button.
7. The user will select the existing photo to upload from his library.
8. The application will navigate back to a post screen, where the user can select to add a caption, tag people, and add the photo to an existing trip or create a new trip.
9. The application will then navigate to the map screen, where the new moment will be displayed in its location through use of the photo's metadata.

Exceptions:

1. No existing photos.
2. User decides not to post photo at any point during process.

Priority: Very high

When available: First iteration

Frequency of use: High

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. How many photos can a user choose to upload at once?
2. Can a photo be added to an existing trip and a new trip?
3. Who can a user tag?

## Post Moment - New Photo

Primary actor: User/Subscription User

Goal in context: To take a photo and post it as a moment in the application.

Preconditions: System is fully configured and app is installed on iPhone; user has logged in and photo to upload has not already been taken.

Trigger: The user wants to take and upload a photo.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user will be navigated to the default application screen (the map).
3. The user will select the '+' icon at the bottom of the screen.
4. The icon will expand to display several additional icons: 'Camera,' 'Note,' and 'Sound.'
5. The user will select the 'Camera' icon.
6. The application will navigate to a camera interface, where the user can take a photo using the capture button at the bottom of the screen.
7. The application will prompt the user to ask if he/she would like to upload the photo just taken.
8. The user will select 'Yes' to continue, and 'No' in order to return to the camera interface and try again.
9. The application will navigate back to a post screen, where the user can select to add a caption, tag people, and add the photo to an existing trip or create a new trip.
10. The application will then navigate to the map screen, where the new moment will be displayed in its location through use of the photo's metadata.

Exceptions:

1. User decides not to post photo at any point during process.

Priority: Very high

When available: First iteration

Frequency of use: High

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. How many photos can a user take before uploading?
2. Can a photo be added to an existing trip and a new trip?
3. Who can a user tag?

### Post Moment - Existing Video

Primary actor: User/Subscription User

Goal in context: To post an existing video as a moment in the application.

Preconditions: System is fully configured and app is installed on iPhone; user has logged in and video to upload has already been taken.

Trigger: The user wants to upload a video.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user will be navigated to the default application screen (the map).
3. The user will select the '+' icon at the bottom of the screen.
4. The icon will expand to display several additional icons: 'Camera,' 'Note,' and 'Sound.'
5. The user will select the 'Camera' icon.
6. The application will navigate to a camera interface, where the user can select to choose from existing photos/videos by selecting the bottom right button.
7. The user will select the existing video to upload from his library.
8. The application will navigate back to a post screen, where the user can select to add a caption, tag people, and add the video to an existing trip or create a new trip.
9. The application will then navigate to the map screen, where the new moment will be displayed in its location through use of the video's metadata.

Exceptions:

1. No existing videos.
2. User decides not to post video at any point during process.

Priority: Very high

When available: First iteration

Frequency of use: High

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. How many videos can a user choose to upload at once?
2. Can a videos be added to an existing trip and a new trip?
3. Who can a user tag?

## Post Moment - New Video

Primary actor: User/Subscription User

Goal in context: To record a video and post it as a moment in the application.

Preconditions: System is fully configured and app is installed on iPhone; user has logged in and video to upload has not already been taken.

Trigger: The user wants to take and upload a video.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user will be navigated to the default application screen (the map).
3. The user will select the '+' icon at the bottom of the screen.
4. The icon will expand to display several additional icons: 'Camera,' 'Note,' and 'Sound.'
5. The user will select the 'Camera' icon.
6. The application will navigate to a camera interface, where the user can take a video using the record button at the bottom of the screen.
7. The application will prompt the user to ask if he/she would like to upload the video just taken.
8. The user will select 'Yes' to continue, and 'No' in order to return to the camera interface and try again.
9. The application will navigate back to a post screen, where the user can select to add a caption, tag people, and add the video to an existing trip or create a new trip.
10. The application will then navigate to the map screen, where the new moment will be displayed in its location through use of the video's metadata.

Exceptions:

1. User decides not to post video at any point during process.

Priority: Very high

When available: First iteration

Frequency of use: High

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. How many videos can a user take before uploading?
2. Can a video be added to an existing trip and a new trip?
3. Who can a user tag?

### Post Moment - Soundbite

Primary actor: User/Subscription User

Goal in context: To record a soundbite and post it as a moment in the application.

Preconditions: System is fully configured and app is installed on iPhone; user has logged in to the application.

Trigger: The user wants to record and upload a soundbite.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user will be navigated to the default application screen (the map).
3. The user will select the '+' icon at the bottom of the screen.
4. The icon will expand to display several additional icons: 'Camera,' 'Note,' and 'Sound.'
5. The user will select the 'Sound' icon.
6. The application will navigate to a post screen, where the user will record his soundbite.  
The user can select to add a caption, tag people, and add the soundbite to an existing trip or create a new trip.
7. The application will then navigate to the map screen, where the new moment will be displayed in its location through use of the location tracking.

Exceptions:

1. User decides not to post soundbite at any point during process.

Priority: Very high

When available: First iteration

Frequency of use: High

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. How many soundbites can a user take before uploading?
2. Can a soundbite be added to an existing trip and a new trip?
3. Who can a user tag?

## Post Moment - Note

Primary actor: User/Subscription User

Goal in context: To create a note and post it as a moment in the application.

Preconditions: System is fully configured and app is installed on iPhone; user has logged in to the application.

Trigger: The user wants to create and upload a note.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user will be navigated to the default application screen (the map).
3. The user will select the '+' icon at the bottom of the screen.
4. The icon will expand to display several additional icons: 'Camera,' 'Note,' and 'Sound.'
5. The user will select the 'Note' icon.
6. The application will navigate to a post screen, where the user can create his note. The user can select to add a caption, tag people, and add the note to an existing trip or create a new trip.
7. The application will then navigate to the map screen, where the new moment will be displayed in its location through use of location tracking.

Exceptions:

1. User decides not to post note at any point during process.

Priority: Very high

When available: First iteration

Frequency of use: High

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. How many notes can a user upload at a time?
2. Can a note be added to an existing trip and a new trip?
3. Who can a user tag?

### Alter Moment - Comment on Moment

Primary Actor: User/Subscription User

Goal in context: To comment on a friends post

Preconditions: System is fully configured and app is installed on iPhone; user has logged in to the application. User has friends who have posted content.

Trigger: The user wants to comment on a friends post

Scenario:

1. The user will click on a friend's post on the map.
2. The post will expand to show a detail view of the post.
3. There will be a text field under the content of a note where text can be entered.
4. After the user enters text, they can press the 'post' button.
5. Their comments will appear along with their name and the date as static text under the content.

Exceptions:

1. The user dismisses the detail view of the post before pressing the 'post' button.

Priority: Medium

When available: Second Iteration

Frequency of use: High

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open Issues:

1. Character limit for the message
2. Space issue within the post's detail view

### Alter Moment - Delete Comment

Primary Actor: User/Subscription User

Goal in context: To delete a comment on a post

Preconditions: System is fully configured and app is installed on iPhone; user has logged in to the application. User has commented on a post, or has comments on one of their posts.

Trigger: The user wants to delete a comment

Scenario:

1. The user will navigate to the post's detail view where the comment was made
2. If the user owns the post, every post will have a little x in the top right corner
3. If the user wants to delete a comment on a friend's post, only an x will appear in the top right corner of their post
4. After clicking the x, an alert will appear to confirm the deletion
5. If the user presses 'Ok' the comment will be deleted

Exceptions:

1. The user presses cancel when asked to confirm

Priority: Medium

When available: Second Iteration

Frequency of use: High

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open Issues:

1. Need to ensure the list of comments reorders itself after deletion

## Alter Moment - Edit Moment

Primary Actor: User/Subscription User

Goal in context: To edit a post made by the user

Preconditions: System is fully configured and app is installed on iPhone; user has logged in to the application. User has a post on the map.

Trigger: The user wants to edit their post

Scenario:

1. The user clicks on a post that they have made, bringing up the detail view for that post.
2. User presses the 'Edit Post' button
3. The title field and description become editable text fields which the user can change. An 'Add to trip' button will also appear that will allow a user to group the moment with an existing trip. In addition, a delete button will appear on the screen.
4. If the user makes changes or deletes, an alert appears asking to confirm when finished.
5. The user presses the 'Ok' button to make changes
6. The user is returned to the map screen with the changes made to the post.

Exceptions:

1. The user presses the cancel button when prompted to confirm changes
2. No trips exist

Priority: Very High

When available: First Iteration

Frequency of use: High

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open Issues:

1. For deletion, make sure that the post is removed from the map properly when returning from the deletion
2. Make sure that the post is changes on both the map and the server.

## View Moment

Primary Actor: User/Subscription User

Goal in context: To bring up the detail view for a post

Preconditions: There is a post on a user's map

Trigger: User wants to see more information about a moment on the map

Scenario:

1. The user is logged in and is currently on the map screen
2. The user clicks on a moment that is on the map
3. The application navigates to the detail view for that moment

Exceptions:

1. Error retrieving information from server

Priority: Very high

When available: First iteration

Frequency of use: Very high

Channel to actor: iPhone

Secondary actors: none

Channel to secondary actors: N/A

Open issues:

1. Will the detail view just expand as a big annotation to the map, or will it transfer the user to a new screen?
2. What animation will occur when transitioning from the map to the detail view?

## Follow/Friend another User

Primary actor: User

Goal in context: To friend another user or to follow a subscription user

Preconditions: The user is logged in with a valid account

Trigger: The user wants to add a friend/start following someone

Scenario:

1. The user has navigated to another user's profile page
2. The user clicks a '+' icon on the user's profile
3. If the user is not a subscription user, an alert will appear asking the user to confirm adding the friend. Subscription users are added instantly
4. If the user presses 'Ok' to the alert, a request will be sent to the other user

Exceptions:

1. The other user denies the request.

Priority: High

When available: Second Iteration

Frequency of Use: Medium

Channel to actor: iPhone

Secondary actors: Other user

Channel to secondary actors: iPhone

Open Issues:

1. Implement request blocking
2. Prevent spam requests

### Stop following/Unfriend another user

Primary actor: User

Goal in context: To stop interactions with another user

Preconditions: User has a friend or is following someone

Trigger: User wants to remove a friend/stop following someone

Scenario:

1. User navigates to another user's profile page
2. The user clicks a '-' button that replaced the add button previously
3. If the user is not a subscription user, an alert will appear asking the user to confirm deleting the friend. Subscription users are unfollowed instantly
4. If the user clicks 'Ok' the other user will be removed from friends

Exceptions:

1. The user cancels when asked for confirmation

Priority: High

When available: Second Iteration

Channel to actor: iPhone

Secondary actors: other user

Channel to secondary actors: iPhone

Open Issues:

1. Add a blocking feature

## Search

Primary actor: User/Subscription User

Goal in context: To find people, trips, or posts within the app

Preconditions: There are things within the app that are searchable

Trigger: User wants to search for something within the map

Scenario:

1. From the main screen, the user swipes right to left to move over the search menu
2. The user will enter text into the search box that will be searched
3. A segmented controller with three buttons will determine whether the search will look for people trips, posts, or a combination of several.
4. After the user finished entering text, a list of results will appear under the search box
5. The user can select an item to be brought to a detail view for that item

Exceptions:

1. The item being searched does not exist or cannot be found
2. The user mistypes the search

Priority: Medium

When available: Second Iteration

Frequency of use: Medium

Channel to actor: iPhone

Secondary actors: None

Channel to secondary actors: N/A

Open Issues:

1. Items within the app need to be searchable

## Filter Map

Primary actor: User/Subscription User

Goal in context: To filter posts on a map to make it easier to see certain details

Preconditions: There are posts on the map which can be filtered

Trigger: The user wants to narrow the amount of content they see on the map

Scenario:

1. From the map screen, the user swipes from left to right to show the menu
2. The user then can select the 'Filter map' item from the menu list
3. Selecting this will bring the user to the Filter Map View
4. From here the user can choose to hide or show posts based on date, user, recent activity, and location.
5. After selecting the appropriate options, the user can press filter at the bottom of the screen
6. This will return the user to the map screen with only the selected posts displayed

Exceptions:

1. The user filters out all posts made, leaving them with an empty map
2. The user presses 'Back' before choosing any filters

Priority: Medium

When available: Second iteration

Frequency of use: Medium

Channel to actor: iPhone

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. Need to implement a way to organize different posts by their meta-data

## Expand Collection

Primary actor: User/Subscription User

Goal in context: To expand a collection of grouped moments on the map

Preconditions: There is a collection of posts that can be expanded

Trigger: The user wants to see the individual posts that make up a collection

Scenario:

1. The user is on the map screen with a collection icon displayed on the map
2. The user clicks the collection icon
3. The collection expands into individual posts in a circle around where the original icon was

Exceptions:

1. Posts may start to overlap if there is too much clutter in one area

Priority: Medium

When available: Second iteration

Frequency of use: Medium

Channel to actor: iPhone

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. How do we deal with collections that are close to each other?
2. Should collections auto expand/contract as the user zooms in/out

## Start/Stop Trip

Primary actor: User/Subscription User

Goal in context: To create a trip within the app

Preconditions: User is logged in with a valid account

Trigger: User wants to start a trip

Scenario:

1. User presses a button named 'Start Trip' on the map screen
2. A prompt appears to ask for a name and description of the trip
3. The 'Start Trip' button becomes a 'Stop Trip' button and now all posts made by the user will be grouped by this trip.
4. Pressing the 'Stop Trip' button will ask to confirm if the trip is over
5. If the user presses 'Ok' the app goes back into the default mode

Exceptions:

1. The user cancels starting or stopping a trip

Priority: High

When available: First Iteration

Frequency of use: High

Channel to actor: iPhone

Secondary actors: none

Channel to secondary actors: N/A

Open Issues:

1. Making sure that users can stop a trip and resume it at a later time
2. Making sure that the user can tell whether they are in a trip in an easily recognizable way

## Edit Trip

Primary actor: User/Subscription User

Goal in context: To edit a trip within the app

Preconditions: User has an existing trip

Trigger: User wants to edit a trip

Scenario:

1. User clicks on a trip's detail view
2. User clicks the 'Edit Trip' button
3. The name and description become editable text fields which the user can change. A delete button also appears.
4. After finishing changes, an alert will appear asking if the user wants to confirm the change
5. If the user presses 'Ok', save the changes and return to the map screen

Exceptions:

1. The user cancels the confirmation

Priority: Medium

When available: First Iteration

Frequency of use: Low

Channel to actor: iPhone

Secondary actors: none

Channel to secondary actors: N/A

Open Issues:

1. Need to decide how the user will go about accessing the detail view for a trip and what other information it will display

## Create Account

Primary actor: Prospective User

Goal in context: To create a MyM account

Preconditions: System is fully configured and app is installed on iPhone.

Trigger: The prospective user wants to create an account.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user will be presented with a splash screen.
3. The user enters a new username, email address and password in the appropriate fields, checks a box indicating agreement to the TOS, and selects a button labeled "Create Account."
4. The user will be presented with a screen notifying them that their account is pending email verification.
5. Upon returning to the app with the verification code sent via email, the user enters the code into the appropriate field and selects a button labeled "Verify Account."
6. The user will be navigated to the first use sequence.

Exceptions:

1. The user enters an existing username.
2. The user enters an existing email address.
3. The user enters a non-existent email address.
4. The user enters an incorrect verification code.
5. The user does not agree to the TOS.

Priority: Very high

When available: First iteration

Frequency of use: Medium

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. Present option to log in?
2. Present retry screens as necessary?

## Sign In

Primary actor: User/Subscription User

Goal in context: To sign in to a user's account.

Preconditions: System is fully configured and app is installed on iPhone.

Trigger: The user wants to use the application.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user will be presented with a splash screen.
3. The user enters their username and password in appropriate fields and selects a button labeled "Log In."
4. The user will be navigated to the default application screen (the map).

Exceptions:

1. The user does not have an account.
2. The user does not enter their account credentials appropriately.

Priority: Very high

When available: First iteration

Frequency of use: Low

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. Present option to create account?
2. Present retry login screen?

## First Use

Primary actor: User

Goal in context: To learn how to use MyM.

Preconditions: System is fully configured and app is installed on iPhone; user has logged in to MyM.

Trigger: The user has just logged in to MyM for the first time.

Scenario:

1. The user launches the application from the iPhone menu for the first time.
2. The user creates an account.
3. The user will be presented with a screen asking them whether they will permit their location to be tracked.
4. The user selects either 'Yes' or 'No' to set the setting.
5. A short sequential overlay tutorial sequence will appear.
6. The user will be navigated to the default application screen (the map).

Exceptions:

1. The user does not allow their location to be tracked.

Priority: High

When available: Second iteration

Frequency of use: Low

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. None.

## View Profile

Primary actor: User/Subscription User

Goal in context: To view the profile for a user.

Preconditions: System is fully configured and app is installed on iPhone, user is logged into the application.

Trigger: The user wants to view his/her profile.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user swipes left to right on the application's main screen, bringing up a sidebar display.
3. The user selects 'Profile' on the sidebar.
4. The system displays the user's profile for viewing, with option to 'Edit Profile.'

Exceptions:

1. The user decides not to view his/her profile.

Priority: High

When available: First iteration

Frequency of use: Medium

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

None

## Edit Profile

Primary actor: User/Subscription User

Goal in context: To edit the profile for a user.

Preconditions: System is fully configured and app is installed on iPhone, user is logged into the application.

Trigger: The user wants to edit his/her profile.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user swipes left to right on the application's main screen, bringing up a sidebar display.
3. The user selects 'Profile' on the sidebar.
4. The system displays the user's profile for viewing, with option to 'Edit Profile.'
5. User selects option to 'Edit Profile.'
6. User edits desired portion of profile and selects 'Save Changes' to make changes or the 'X' icon to revert to 'Profile' screen.
7. If user selects 'Save Changes,' changes are made and system reverts to 'Profile' screen.

Exceptions:

1. The user decides not to edit his/her profile.

Priority: High

When available: First iteration

Frequency of use: Medium

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. Should we require user to enter credentials again before editing?

## View Settings

Primary actor: User/Subscription User

Goal in context: To view the settings for a particular user.

Preconditions: System is fully configured and app is installed on iPhone, user is logged into the application.

Trigger: The user wants to view his current settings for the application.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user swipes left to right on the application's main screen, bringing up a sidebar display.
3. The user selects 'Settings' on the sidebar.
4. The user's settings are displayed on the screen, along with an 'Edit Settings' choice.

Exceptions:

1. The user decides not to view his settings.

Priority: Very high

When available: First iteration

Frequency of use: Medium

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. Should all settings be viewable on the screen?

## View Settings - View TOS

Primary actor: User/Subscription User

Goal in context: To view the MyM Terms of Service.

Preconditions: System is fully configured and app is installed on iPhone, user is logged into the application.

Trigger: The user wants to view the terms of service they have agreed to.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user swipes left to right on the application's main screen, bringing up a sidebar display.
3. The user selects 'Settings' on the sidebar.
4. The user's settings are displayed on the screen, along with 'Edit Settings' and 'Terms of Service' choices.
5. The user selects 'Terms of Service.'
6. The terms of service are displayed on the screen.

Exceptions:

1. The user decides not to view the TOS.

Priority: High

When available: First iteration

Frequency of use: Low

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. Should all settings be viewable on the screen?

## Edit Settings - Sign Out

Primary actor: User/Subscription User

Goal in context: To sign out of a user's account.

Preconditions: System is fully configured and app is installed on iPhone, user is logged into the application.

Trigger: The user wants to sign out of the application.

Scenario:

5. The user launches the application from the iPhone menu.
6. The user swipes left to right on the application's main screen, bringing up a sidebar display.
7. The user selects 'Settings' on the sidebar.
8. The user selects 'Sign Out' on the screen.
9. The system prompts the user to see if they are sure they would like to sign out.
10. The user selects 'Logout' to sign out of the application or 'No' in order to return to the 'Settings' sidebar.
11. If the user signs out, the application returns to the login screen.

Exceptions:

2. The user decides not to logout of the system.

Priority: Very high

When available: First iteration

Frequency of use: Medium

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. Should system prompt users to ask if sure?
2. What should be saved on logout?
3. Should editing the settings require the user to login again?

## Edit Settings - Right/Left Hand

Primary actor: User/Subscription User

Goal in context: To change the application layout so that buttons are more easily accessible based on the user's right/left handedness.

Preconditions: System is fully configured and app is installed on iPhone, user is logged into the application.

Trigger: The user wants to change the application orientation.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user swipes left to right on the application's main screen, bringing up a sidebar display.
3. The user selects 'Settings' on the sidebar.
4. The user selects 'Edit Settings' on the menu.
5. The user selects 'Right/Left' on the screen.
6. The user selects their desired orientation for the application.
7. The system prompts the user to see if they are sure they would like to adjust the orientation.
8. The user selects 'Yes' to change orientation or 'No' to return to the 'Settings' sidebar.
9. If the user changes orientation, the application adjusts and returns to the map screen.

Exceptions:

1. The user decides not to change the orientation of the system.

Priority: Medium

When available: First iteration

Frequency of use: Low

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. Should system prompt users to ask if sure?
2. Should the system offer additional or custom orientations?
3. Should editing the settings require the user to login again?

## Edit Settings - Location Tracking

Primary actor: User/Subscription User

Goal in context: To change the application's ability to track location.

Preconditions: System is fully configured and app is installed on iPhone, user is logged into the application.

Trigger: The user wants to change the application's ability to track their location.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user swipes left to right on the application's main screen, bringing up a sidebar display.
3. The user selects 'Settings' on the sidebar.
4. The user selects 'Edit Settings' on the menu.
5. The user selects 'Location Tracking' on the screen.
6. The user selects to turn location tracking on or off for the application.
7. The system prompts the user to see if they are sure they would like to adjust the current setting.
8. The user selects 'Yes' to change the current setting or 'No' to return to the 'Settings' sidebar.
9. If the user changes the current setting, the application adjusts and returns to the map screen.

Exceptions:

1. The user decides not to change location tracking for the system.

Priority: High

When available: First iteration

Frequency of use: Low

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. Should system prompt users to ask if sure?
2. Should editing the settings require the user to login again?

## Edit Settings - Upgrade Account

Primary actor: User

Goal in context: To upgrade the user account to a subscription user account.

Preconditions: System is fully configured and app is installed on iPhone, user is logged into the application.

Trigger: The user wants to upgrade his account.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user swipes left to right on the application's main screen, bringing up a sidebar display.
3. The user selects 'Settings' on the sidebar.
4. The user selects 'Edit Settings' on the menu.
5. The user selects 'Upgrade Account' on the screen.
6. The user selects to upgrade to a subscription user account.
7. The system prompts the user to see if they are sure they would like to upgrade their account.
8. The user selects 'Yes' to upgrade or 'No' to return to the 'Settings' sidebar.
9. If the user selects 'Yes,' the user's account is upgraded and the user is then returned to the main screen. This may take some time to complete.

Exceptions:

1. The user decides not to change the account orientation.

Priority: Medium

When available: First iteration

Frequency of use: Low

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. Should all users be able to upgrade their accounts?
2. What exactly needs to be done on the back-end to accomplish this?
3. Should editing the settings require the user to login again?

## Edit Settings - Downgrade Account

Primary actor: Subscription User

Goal in context: To downgrade the subscription user account to a normal user account.

Preconditions: System is fully configured and app is installed on iPhone, user is logged into the application.

Trigger: The subscription user wants to downgrade his account.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user swipes left to right on the application's main screen, bringing up a sidebar display.
3. The user selects 'Settings' on the sidebar.
4. The user selects 'Edit Settings' on the menu.
5. The user selects 'Downgrade Account' on the screen.
6. The user selects to downgrade to a normal user account.
7. The system prompts the user to see if they are sure they would like to downgrade their account.
8. The user selects 'Yes' to downgrade or 'No' to return to the 'Settings' sidebar.
9. If the user selects 'Yes,' the user's account is downgraded and the user is then returned to the main screen. This may take some time to complete.

Exceptions:

1. The user decides not to change the account orientation.

Priority: Medium

When available: First iteration

Frequency of use: Low

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. What exactly needs to be done on the back-end to accomplish this?
2. Should editing the settings require the user to login again?

## Edit Settings - Delete Account

Primary actor: User/Subscription User

Goal in context: To delete the user account.

Preconditions: System is fully configured and app is installed on iPhone, user is logged into the application.

Trigger: The subscription user wants to delete his account.

Scenario:

1. The user launches the application from the iPhone menu.
2. The user swipes left to right on the application's main screen, bringing up a sidebar display.
3. The user selects 'Settings' on the sidebar.
4. The user selects 'Edit Settings' on the menu.
5. The user selects 'Delete Account' on the screen.
6. The user is presented with an authentication screen, requiring them to input their username and password, and then select the 'Delete Account' button.
7. The system prompts the user to confirm they would like to delete their account.
8. The user selects 'Delete Account' to delete their account or 'Cancel' to return to the 'Settings' sidebar.
9. If the user selects 'Delete Account,' the user's account is deleted, all connections to the user's account are deleted from the MyM system, and the user is then returned to the splash screen. This may take some time to complete; an animated sequence will be shown as it takes place.

Exceptions:

1. The user decides not to delete their account.
2. The user does not authenticate their account.

Priority: High

When available: Second iteration

Frequency of use: Very Low

Channel to actor: Via iPhone display

Secondary actors: None

Channel to secondary actors: N/A

Open issues:

1. What exactly needs to be done on the back-end to accomplish this?
2. Is any additional verification needed for account deletion?

## Mock Up Images:



### LEFT.

In the login screen, the user will be able to login to an existing account, or register for a new account on the top left. At the bottom left, the user can access the About information.

### RIGHT.

This is the settings screen. Users can access and change information about their account such as their password in the Account tab. You can delete your account inside the General tab. The Sharing tab allows the user to set their options for sharing their maps and their moments. The Privacy tab allows a user to set who can view and follow their Maps and information about their account. The Notifications tab allows a user to set what notifications will be shown to them in their recent activity. Notifications could also be sent to a phone or an email. The Security tab allows a user to set up more authentication steps for logging into their account. The About tab displays information about the app, such as copyright information and the names of the app's creators.

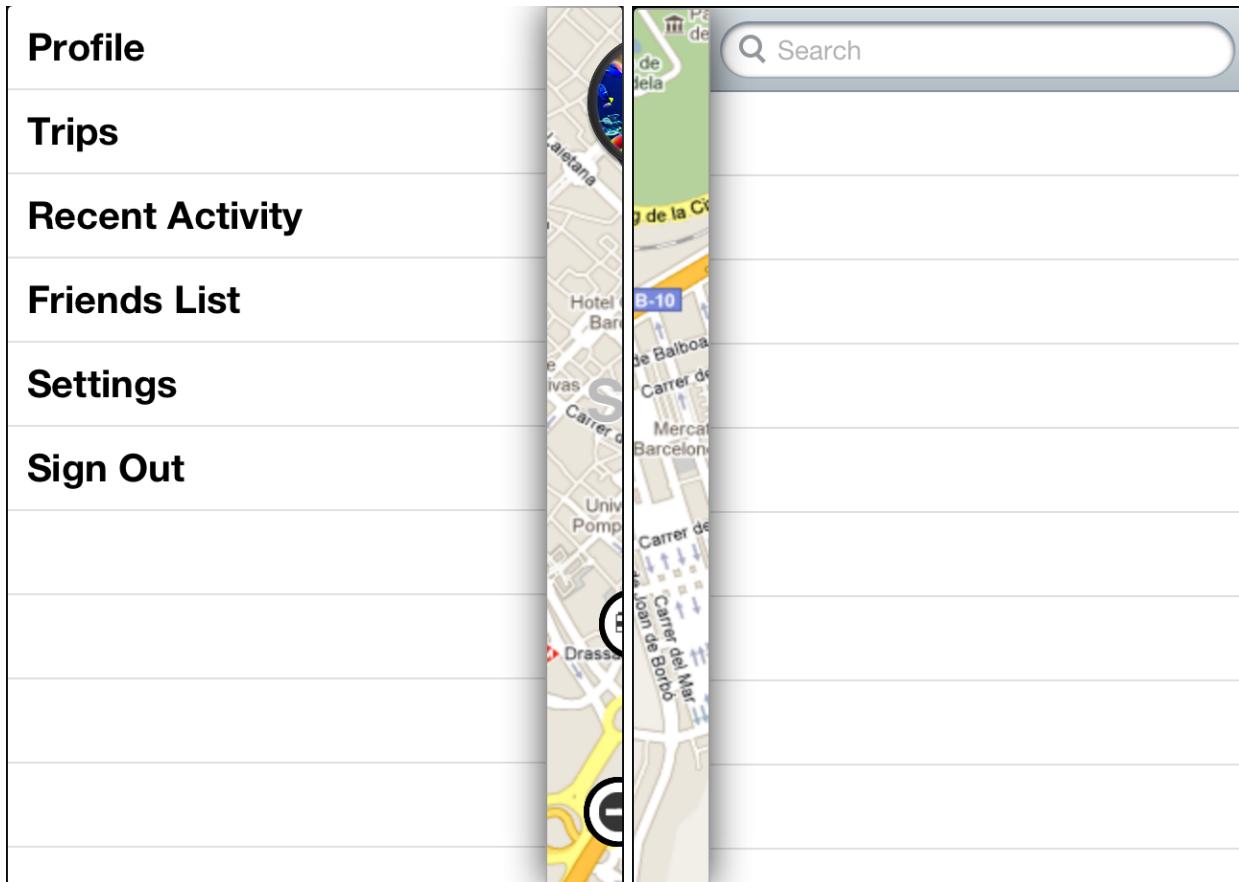


### LEFT.

This is the main Map page that a user will see upon successful login. The - symbol will be a + on start up, but can be expanded to display the various methods of adding moments to your Map. The Map shows all the moments on the current Trip that have been added since the Trip was created.

### RIGHT.

A user can click on a moment on their Map or their friends' Maps to display the moment as seen above. The user can click the small blue arrow to bring up more detailed information about the moment such as comments and tags.

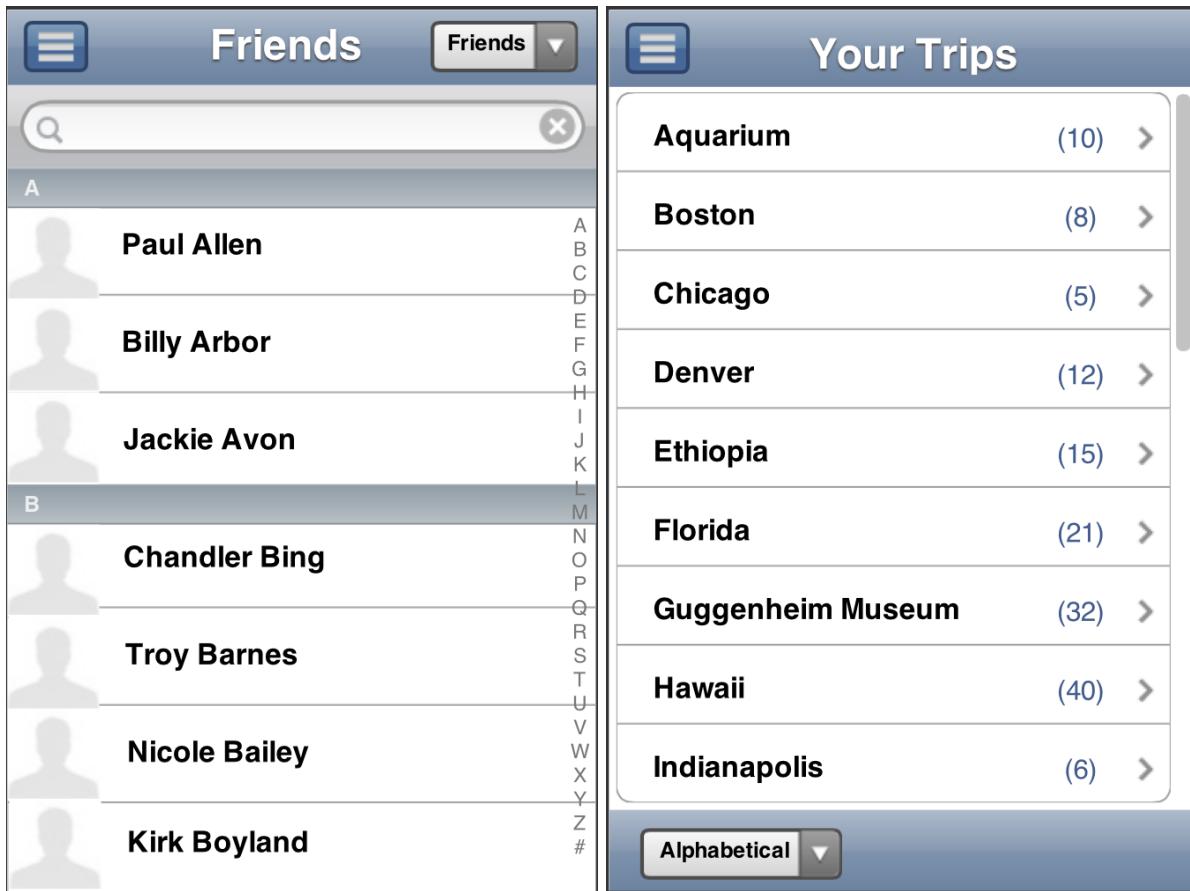


**LEFT.**

Swiping left to right brings up a menu of various options for the user to access. The user can view their profile information, view a listing of their trips, view their recent activities (notifications), view their friends list (includes who they are following), view their settings, and sign out from the session.

**RIGHT.**

Swiping right to left brings up a list of all of the user's friends and people they follow. Using the search bar they can search for a specific friend.

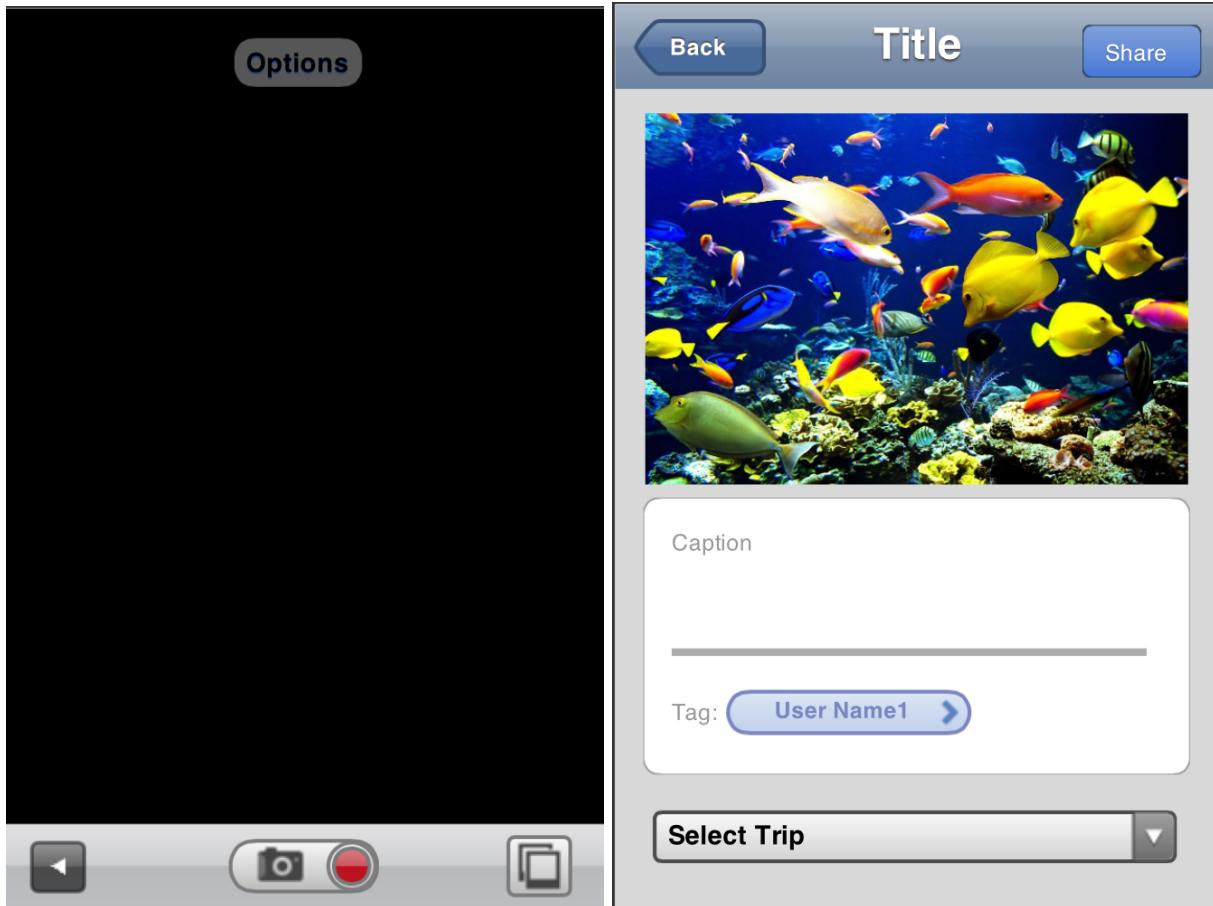


### LEFT.

This is the menu that appears when a user chooses Friends List from the menu shown by swiping right to left on the main Map screen. A user can click the top left icon to return to the map and can use the dropdown menu in the top right to filter between friends, followers, and people you are following. The search bar can be used to find a specific person.

### RIGHT.

This lists all of your trips. You can click on a trip to bring up a Map that shows all of the moments created during that trip. The number of moments is displayed right next to the name. The trips can be sorted through various methods (alphabetical, chronological, etc.). A user can swipe left to right on a trip to delete it. The top left icon will return the user to their current trip's Map.

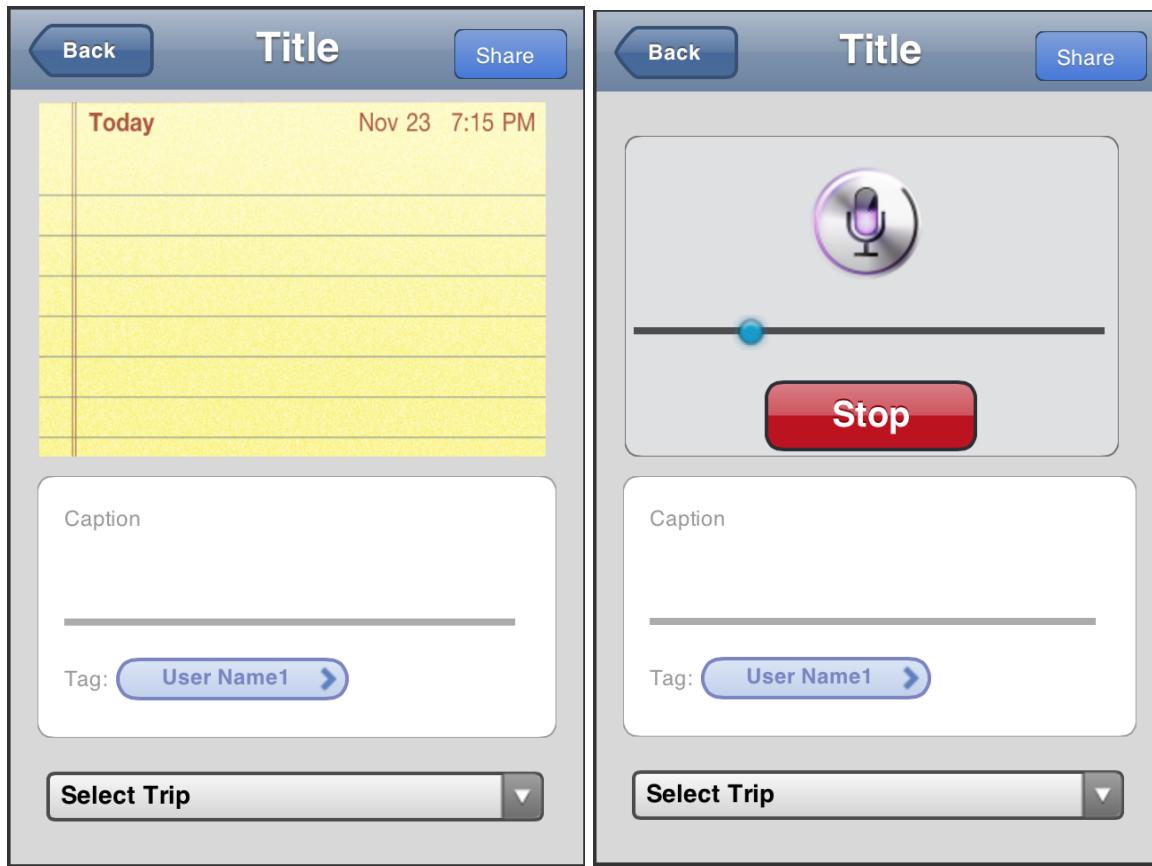


### LEFT.

This is the built in camera interface that launches when a user chooses to create a new picture/video moment from the expanded icon menu on the main Map. The options button allows a user to set flash to be on/off/auto, to switch between the front and back camera, and to turn gridding on/off. The bottom left arrow icon returns the user to the main Map. clicking the camera icon takes a picture whereas clicking the red dot starts a video recording. The picture icon in the bottom left allows a user to choose an image/video from their pre-existing camera roll.

### RIGHT.

After a photo has been taken for a created moment, the user chooses to add a caption, tag any friends, and selects a trip to add the moment to. Sharing the moment finishes its creation and pressing back exits the creation of the moment.

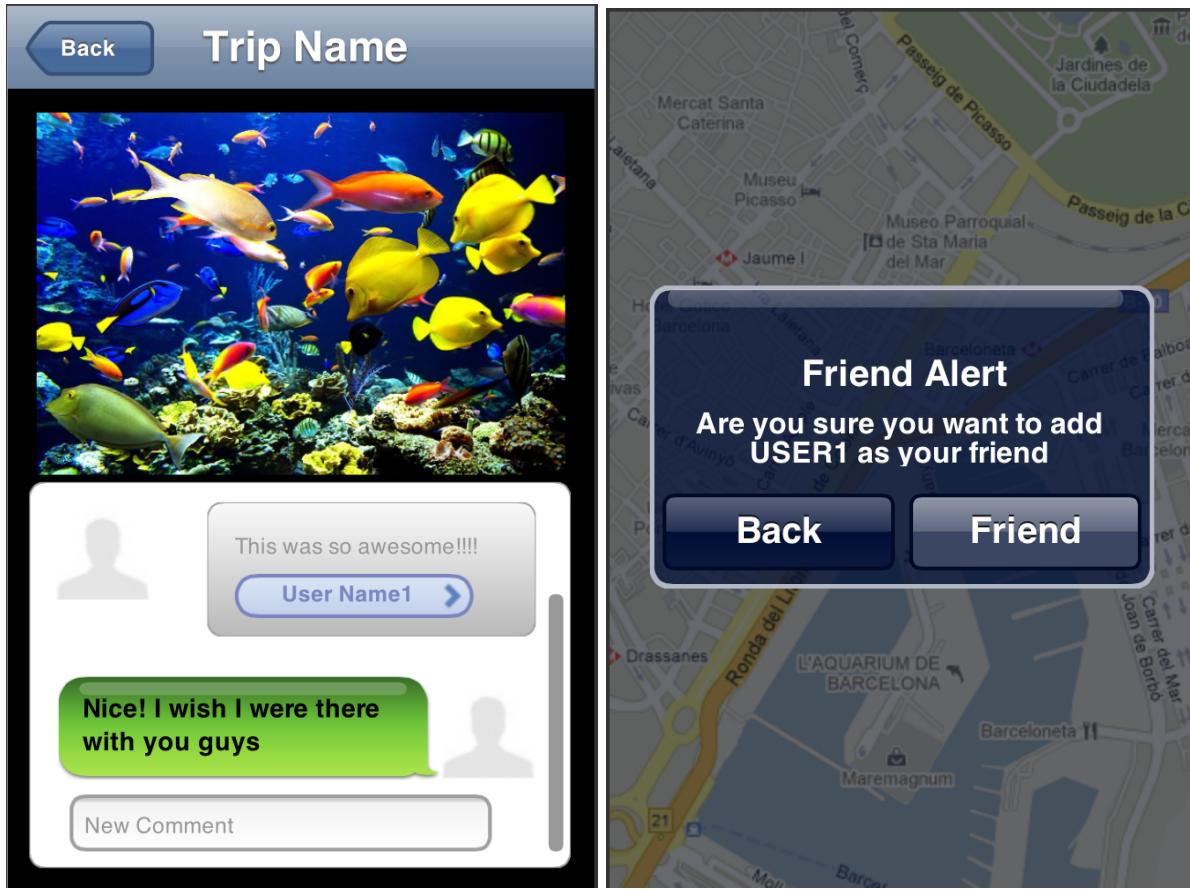


### LEFT.

This is the screen that a user is taken to when choosing to create a “note” moment. The user can type the note, add a caption, tag friends, and select a trip to add it to.

### RIGHT.

This is the screen that a user is taken to when choosing to create a “sound” moment. The user can record a sound clip, add a caption, tag friends, and select a trip to add it to.



### LEFT.

This is the screen that is shown when a user clicks the small blue arrow after clicking a moment on a map. The moment is displayed and the caption, friend tags, and comments are shown. A user or friend can create a new comment about the moment and the owner of the moment can edit its information.

### RIGHT.

This is the screen is when you select a user from the search list that is still not your friend. You won't be able to see the map unless the other person accepts your friend's frequent. If the user don't accept friend" request, only followers, you are going to be able to follow the user instead of friending him.