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

Github: www.github.com/srz2/CSC470
Team: iOS
MyM - Map your Moments

# Stage III: Design

Shneiderman's Eight Golden Rules of Interface Design are a set of implied standards that should be taken into account with all design projects in order to strive to make the best quality project. These standards or relevant in our project because as projects like ours become more streamline such as in a mobile device, speed and quality of a solution becomes more and more important. Thinking about the eight rules helps us plan our project to deliver a high quality end result. The rules relate to; consistency, quick use through shortcuts, informative feedback, closure dialog, elegant error handling, reversal of actions, locus of control, reduced short term memory load. As each of these rules are explained further below, it becomes more apparent how not only necessary these qualities are in a project like ours but also how most of what they encompass in expected in a solutions like ours.

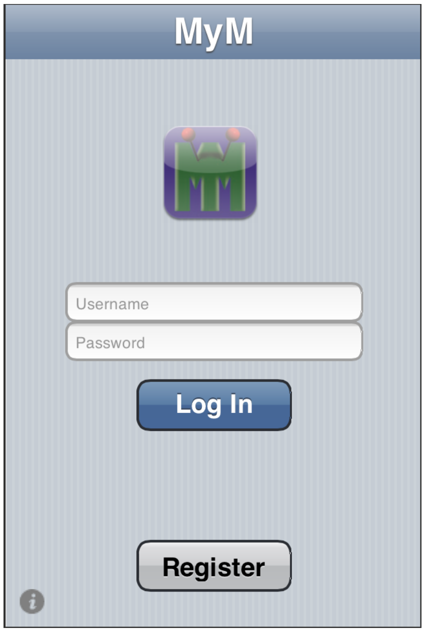
## Shneiderman's "Eight Golden Rules of Interface Design"

### 1 - Strive for consistency

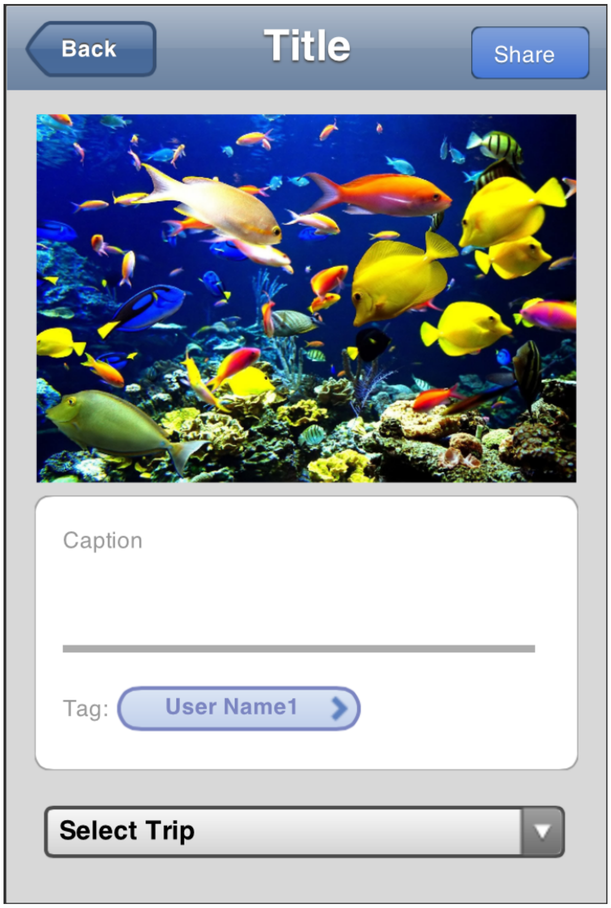
*Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.*

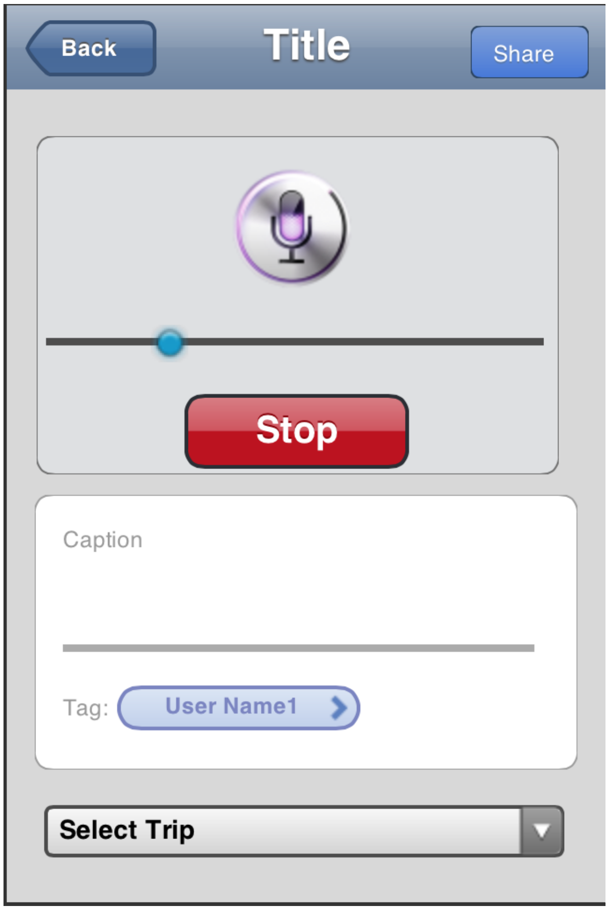
By keeping consistent with other applications, we ensure that the learning curve is going to be easier and ensure the user will enjoy using the application. As a result, the chances of the user not knowing what a specific button does is reduced. The idea is to build an application easy to learn and master.
By keeping the consistency within the application, we ensure the user is always getting the expect result out of an action.

For example, we have the Log In screen, that looks like any other Log In screen. This ensure that the user will be familiar with what it is presented to him, reducing the learning time.



An example of consistency within the MyM application is the screen to post a new Moment. The screens to post a new moment will look similar, regardless of the content the user posts. The only difference is the top half of the screen, which is going to depend on what kind of Moment the user wants to post. This is beneficial because the user will be familiar with the screen layout.



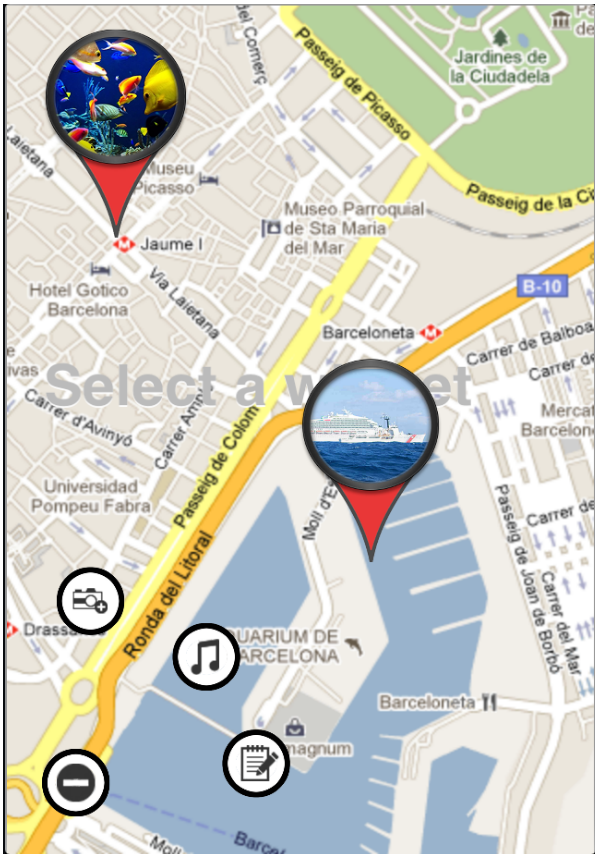


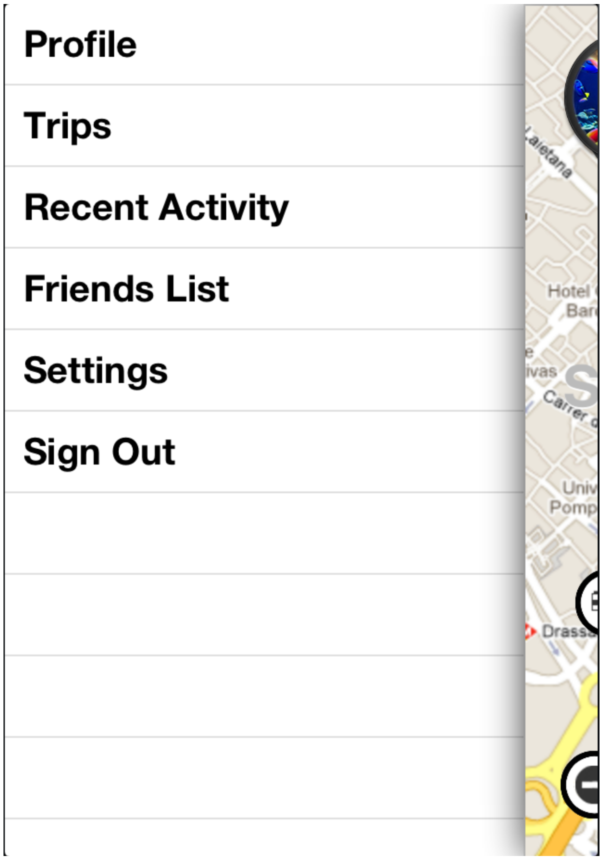
**2 - Enable frequent users to use shortcuts**

*As the frequency of use increases, so do the user’s desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations, function keys, hidden commands, and macro facilities are very helpful to an expert user.*

We want users to keep using our solution, which is why we are are offering every functionality two or three clicks away. We will also implement swipe actions, that are fast and intuitive to be used as shortcuts to common tasks.
The application will utilize a swiping menu on the left, that gives the user access to all main functionalities.
In addition, there will be a small collapsible menu at bottom that will provide fast access to the moment sharing screens.

As an example we have the little menu at the very bottom, that once it is clicked it will expand into three options: share picture, note or sound. This is really practical and save us a lot of space in the screen and let the user focus on what is important. On the right we have an example of the swiping menu that will provide fast access to main features.





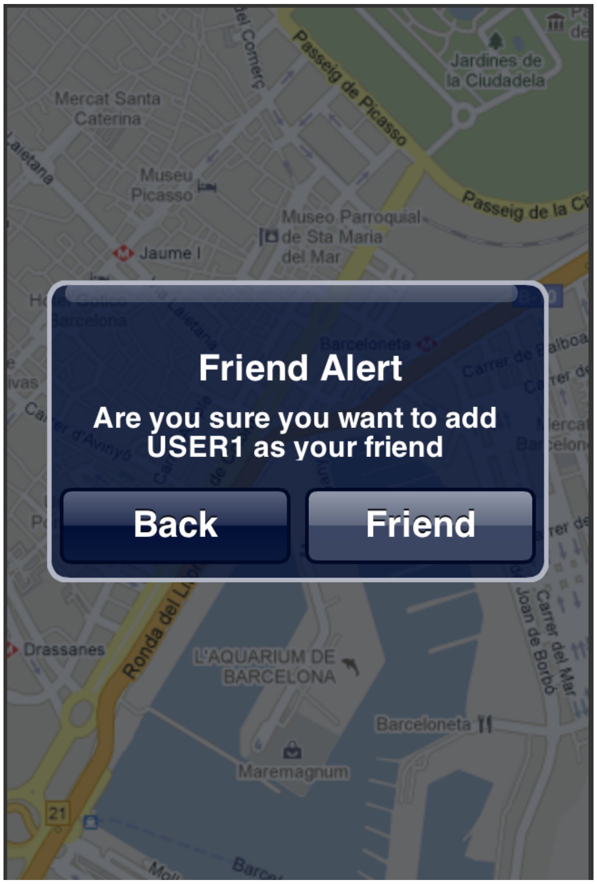
**3 - Offer informative feedback**

*For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.*

The application will be using a mix of colors, animations and pop-ups to give feedback to the user, leaving no ambiguity regarding the status of the action.

The user will always have full control of that actions. At many points we will offer the user the ability to cancel the current process.

As an example we have an Alert View that is going to pop-up when the user visits someones page that he is still not friends with. The Alert View allows him to add that user to his friend's list or simply go back.



Friend Alert

Are you sure you want to add USER1 as your friend

**4 - Design dialog to yield closure**

*Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.*

Our application has four major action sequences available to the user: sharing a moment, filtering their map, friending or following someone, and adding a moment to a trip. All action sequences end with the user viewing the main map screen in some state, whether with a new moment, a new trip, some other person's content, or with new filters applied.

Users will be directed through the action sequences if they so choose, but will know they have completed their chosen action when the map reappears with new content.

Post moment: (initialize post -> add content -> provide details -> post -> view on map)
Filter map: (swipe to filters -> select filters -> view map with applied filters)
Friend/follow person: (view person -> friend/follow person -> view person's map(if follow, friend pends confirmation))
Add moment to trip: (select moment -> select trip -> view trip on map with added moment)

***5 - Offer simple error handling***

*As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.*

There will be a large amount of redundancy built into the application to assure that serious errors are not incurred. Moments, for instance, will not be able to be shared with anything less than the required amount of content. The "Share" button will be greyed out until the appropriate content has been given to the post by the user.

Another potentialissue is network connectivity. Although not necessarily under the user's control, the app should not allow the user to enter or access content without a stable network connection. If it did, the user could encounter the frustrating case of a hanging or unresponsive app. If the user loses a network connection, the application will gracefully save its state, including any existing user input, and display a modal notifying the user of the dropped connection.

Network error
Disk space error
By design, posts cannot contain null content
By design, trips cannot contain null moments
Invalid credentials
Amazon-specific errors

**6 - Permit easy reversal of actions**

*This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.*

Users will always be able to undo any action. The application will allow the user to: Edit post, Edit trip, Edit account, Remove post, Remove trip and Remove account. This will let users fix/undo any undesired actions.

**7 - Support internal locus of control**

*Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.*

The default state of the application is the map view, with the last selected filters applied. The map view will not change unless the user takes an action: posting a moment, navigating the map, or altering the filters. If the user posts a moment, the map will zoom to the new moment. If the user navigates the map, it will respond appropriately. If the user alters the filters, the moments and trips displayed on the map will change as needed. The application will at no time bother the user to attend to notifications, bug them to post something, or otherwise get in the way of the user's experience. It will simply display appropriate options at each point in time, making them available yet unobtrusive.

**8 - Reduce short-term memory load**

*The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.*

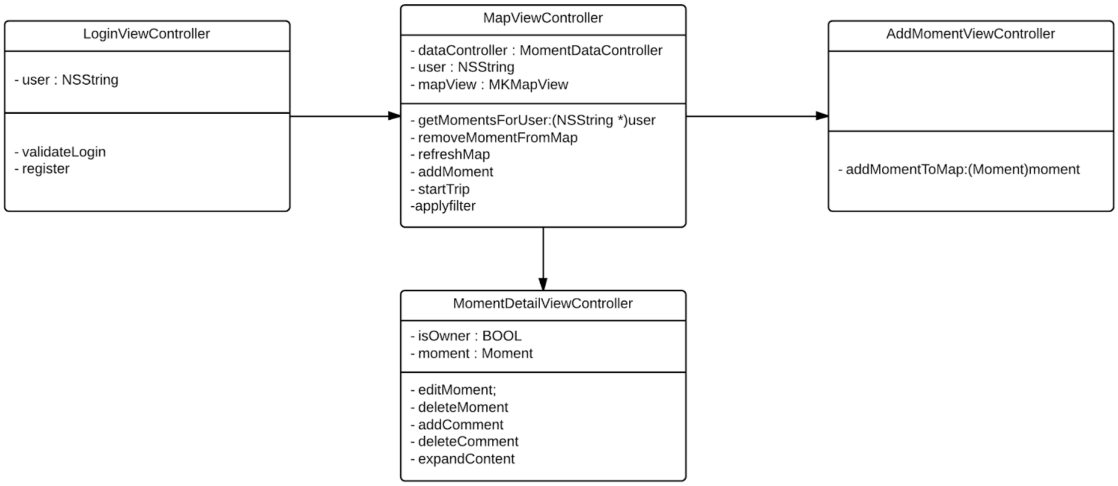
A user only ever has to worry about one action at a time, and each action is walked through in a logical order, so the user only has to input one thing at a time.

A user is always doing only one of the following:

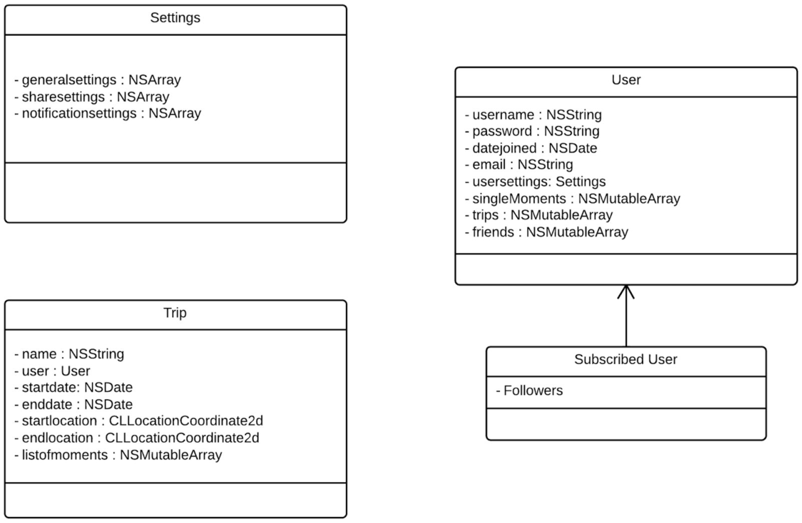
- Sharing a moment
- Viewing the map
- Applying a filter
- Editing the settings

# Class Diagrams

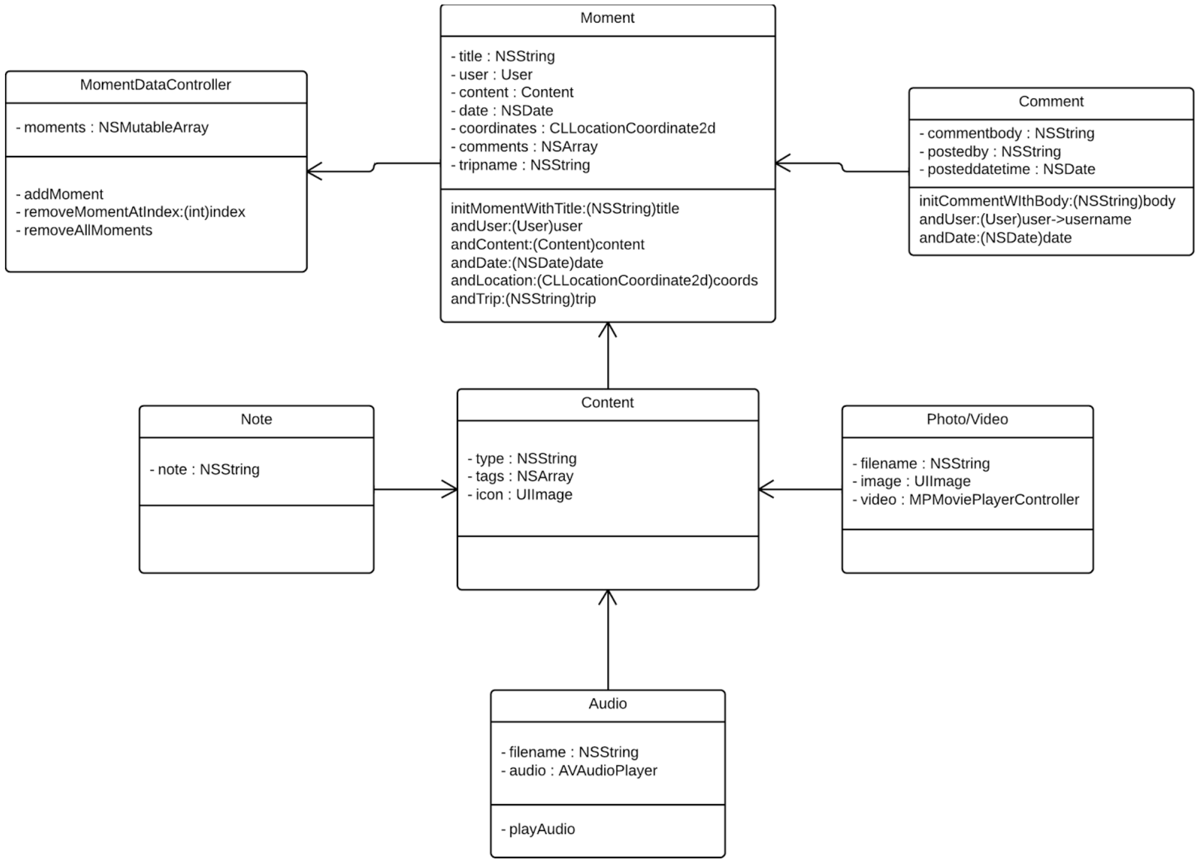
## View Controller Class Design Diagrams



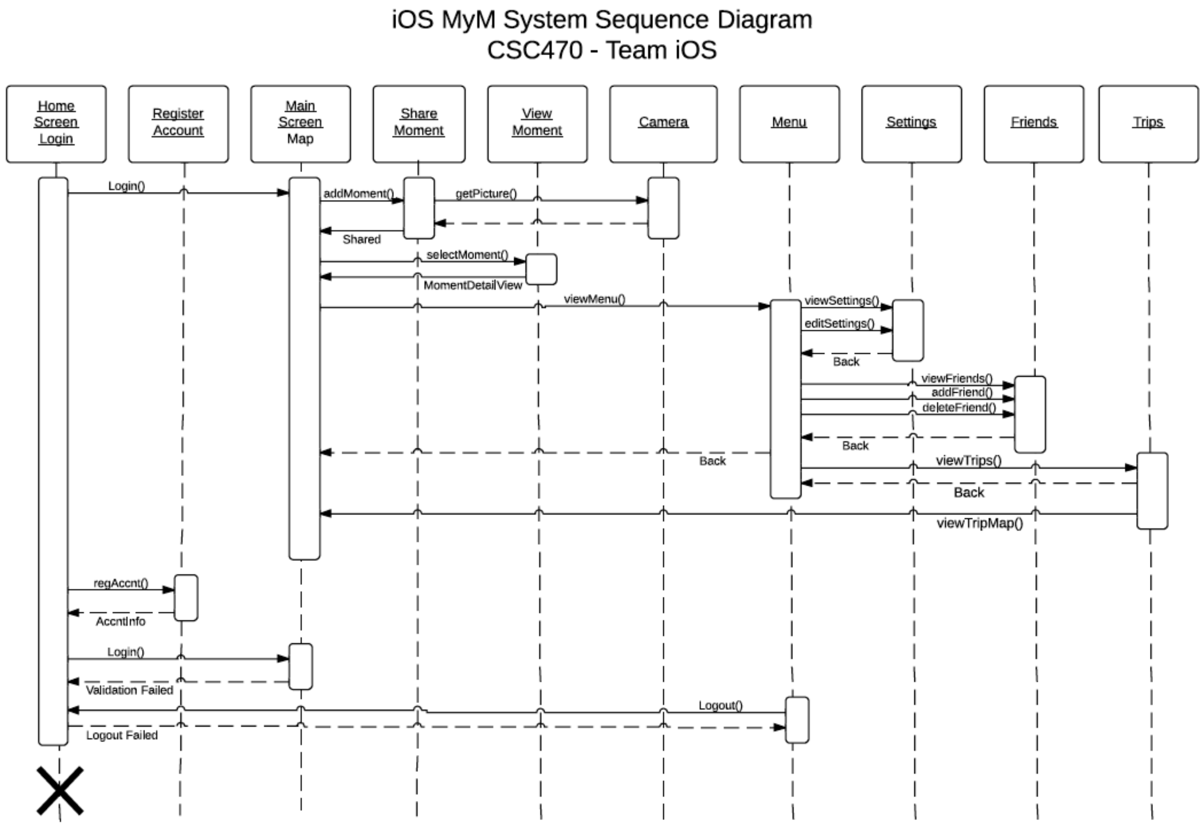
## Model Class Design Diagrams



Settings
- generalsettings : NSArray
- sharesettings : NSArray
- notificationsettings : NSArray
User
- username : NSString
- password : NSString
- datejoined : NSDate
- email : NSString
- usersettings: Settings
- singleMoments : NSMutableArray
- trips : NSMutableArray
- friends : NSMutableArray
Trip
- name : NSString
- user : User
- startdate: NSDate
- enddate : NSDate
- startlocation : CLLocationCoordinate2d
- endlocation : CLLocationCoordinate2d
- listofmoments : NSMutableArray
Subscribed User
- Followers


Moment
- title : NSString
- user : User
- content : Content
- date : NSDate
- coordinates : CLLocationCoordinate2d
- comments : NSArray
- tripname : NSString
initMomentWithTitle:(NSString)title
andUser:(User)user
andContent:(Content)content
andDate:(NSDate)date
andLocation:(CLLocationCoordinate2d)coords
andTrip:(NSString)trip
MomentDataController
- moments : NSMutableArray
- addMoment
- removeMomentAtIndex:(int)index
- removeAllMoments
Comment
- commentbody : NSString
- postedby : NSString
- posteddatetime : NSDate
initCommentWIthBody:(NSString)body
andUser:(User)user->username
andDate:(NSDate)date
Note
- note : NSString
Content
- type : NSString
- tags : NSArray
- icon : UIImage
Photo/Video
- filename : NSString
- image : UIImage
- video : MPMoviePlayerController
Audio
- filename : NSString
- audio : AVAudioPlayer
- playAudio

## System Sequence Diagram


iOS MyM System Sequence Diagram
CSC470 - Team iOS
Home Screen Login
Register Account
Main Screen Map
Share Moment
View Moment
Camera
Menu
Settings
Friends
Trips
Login()
addMoment()
getPicture()
Shared
selectMoment()
MomentDetailView
viewMenu()
viewSettings()
editSettings()
Back
viewFriends()
addFriend()
deleteFriend()
Back
Back
viewTrips()
Back
viewTripMap()
regAccnt()
AccntInfo
Login()
Validation Failed
Logout()
Logout Failed