UCRChatline - iOS Mobile Application

Milestone 1 Documentation

November 2014

Computer Science 180, Fall 2014

Professor Iulian Neamtiu, Instructor
Aiping Shi, T.A

Team Members:
Gustavo Blanco
Hector Dominguez
Fernando Gonzalez
Sergio Morales

Table of Contents

.

1 INTRODUCTION

UCRChatline is a social-media mobile app that is designed to work off of hassle-free application programming interfaces, allowing us to shift the overhead of back-end support to front-end, delivering an easy to use interface and great platform for social media on the go. What makes UCRChatline even better is the fact that we deliver a social media network on one of the most popular platforms in mobile development.

1.1 PROJECT OVERVIEW

In this project, we introduce a social networking/communication mobile application. With this app, users can add other users as friends and then chat with each other as well as post text, image, video, link like a microblog. It also supports group chat and video calling. There is a timeline page for users to see what their friends have posted and write comment or like. They are also able to hide specific posts from specific groups of friends.

UCRChatline allows users to create their own accounts and log in, prompted when first opening the app. Users may have their own profile tab to display details, in addition to have the privacy of displaying it to their friends or not. Friends are added through basic friend requests after being approved or not. Just like all social networking apps, users may chat to each other using text, images, emojis, stickers, and even voice messaging. Finally, users may display statuses and posts to all friends, as a timeline.

Best of all, UCRChatline is a free, adware/spyware safe, and secure way of communication between friends, family, and other networks on Apple products.

1.2 PURPOSE AND SCOPE OF THIS SPECIFICATION

We will outline what UCRChatline is, and give an in-depth guide as to how to use the app, and what it does. Use cases will be provided with visuals will be used, followed by specific user and system requirements to use UCRChatline. Lastly, the testing phase of the design will be explained.

1.3 GLOSSARY

We assume the reader is familiar with various abbreviations and terminology for this Documentation.

App - An application, or program, usually ran on a mobile platform for a variety of uses.
API - Application Programming Interface, written set of functions calls written in order to make writing applications towards specific functionalities easier on the programmer.
Parse - The API that allows us to have backend support for data.
Storyboard - The Apple iOS term for representing a high level view of all possible views, or screens, in an iOS application.
Language - We refer to the programming language used in implementation of our application, specifically Objective C.
Tab - The menu bar located at the bottom of typical iOS applications

2 REQUIREMENTS

2.1 USER REQUIREMENTS

Where ‘5’ is of highest priority, and ‘1’ is lowest.

Req. #	Description	Priority
R1	The application must be available on Apple App Store or through Xcode 6.1.	5
R2	A network connection must be available to download the application.	5
R3	Users must be able to create an account using a name, email, and password	5
R4	All users are allowed to be able to connect to each other as ‘friends’	5
R5	Users must be able to chat with other users through messages	5
R6	Multiple users may chat as a group.	4
R7	Video calls must be available between users.	4
R8	Images can be uploaded onto messages and posts	4
R9	Every user must be provided a profile with their information	4
R10	There must be a timeline with posts from all users and their friends, from which they may comment or reply on.	4
R11	Everyone may be able to send a friend request to each other	4

2.2 SYSTEM REQUIREMENTS

Req. #	Description	Priority
S1	User must have an iPhone 4 or newer	5
S2	User's iPhone must have iOS 7 or later	4
S3	The user's iPhone must have a stable internet connection throughout the use of UCRChatline	4
S4	Loading the app prompts the user for a login, or uses previous login credentials	3
S5	Logging into the system must query all data to and from the Parse data API	4
S6	User object data must be made available through Parse query API calls	4
S7	Sending a message to another user must ensure that it will be saved after closing	3
S8	Messages sent to another user must be confirmed to be sent	4

2.3 STATUS

Currently, we have the following functionality:

Messaging: Able to create a message to anyone in your friend's list. Is is able to send Emojis and text and have a conversation with a single user or multiple user. Also, we can do video calling to a single.

Timeline: Users are able to view all moments posted by them and their friends. User can post moments in this page. A moment can include text, images, emoji, video, link. Users can like moments and write comment to moments. When users post a moment, They can make it visible to specific users(must be friends with) or groups of friends.

Profile: Users may edit several information about them such as name, password, and picture. Pictures can be uploaded using the phone's camera, and is stored in the database afterwards.

Friends: At this time, the user is only able to view current friends as a TableView, but may not add any friends by requesting them through the button at this time.

Data: All data stored and queried may easily be reachable using the Parse core API. This all accessible through one main account for the team.

3 DESIGN

The UCRChatline was designed in a way that allows the user to navigate through functionalities in a user-friendly manner. There are a few fundamental aspects of our application that makes this happen, separation of tabs, design, and interface.

3.1 DESIGN OVERVIEW & USE CASES

UCRChatline is built using a series of data structures stored on Parse. The following figure outlines the type of objects that we request and send to and from Parse using the iO

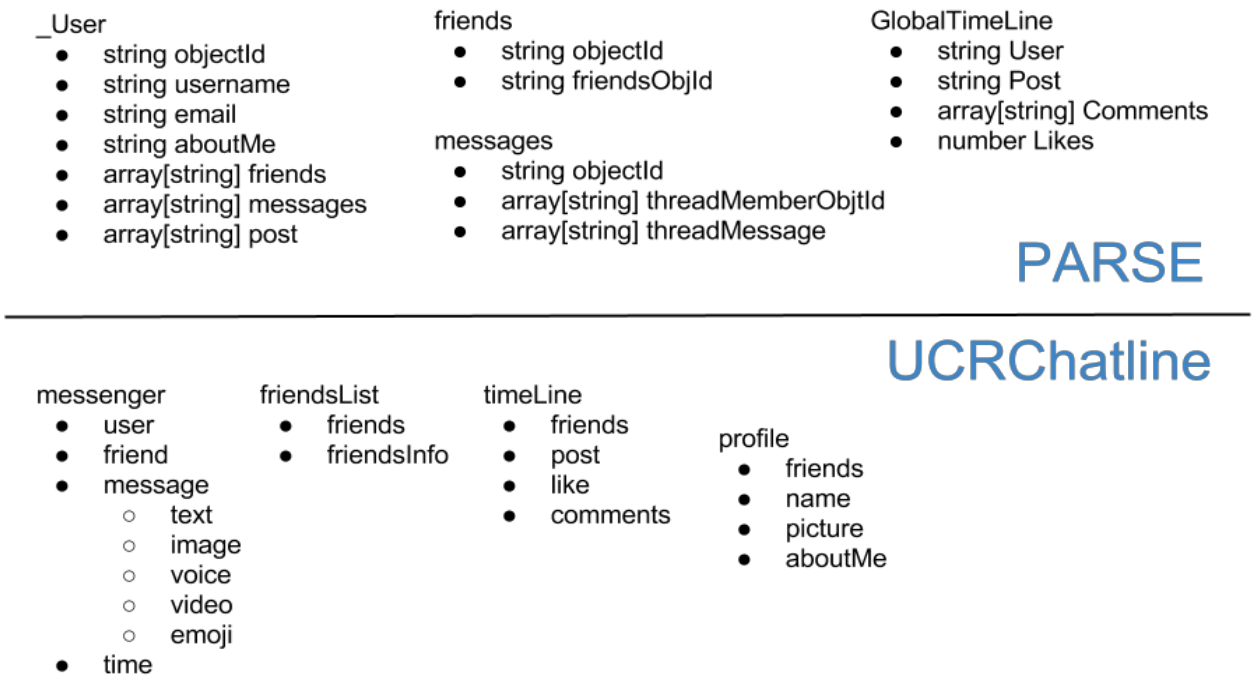


Figure 1: Data Structures stored on Parse.

Separation of Tabs

Since we modularized our project to allow for individual cooperation and easy merging, features were split up in terms of tabs in the application. Everyone contributed to their own tab in a way that wouldn't pose problems to the others. Once we had a stable build, we merged

our modules together. Modules avoided dependencies as much as possible to avoid breaking the whole app if one problem emerges.

Interface

The interface of UCRChatline is basic, yet powerful. With a touch of a button, a user may post a status to his/her timeline in a matter of seconds, to be saved just as easily thanks to the integration of Parse API. We aimed to keep the interface as simple as possible to make it rigid and easy to use for the user.

Walkthrough of Use Cases

- Sign up process
- Login
- Chatting with Friends
- Timeline
- Profiles
- Friends

Sign up Process

Users are able to sign up and create an account in UCRChatline by opening up the app without being logged in. If this is the first time using the app, the user will automatically have the option of signing up. The following figures demonstrate this:

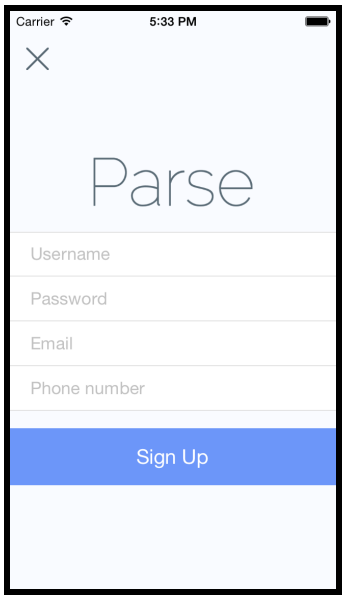


Figure 2. This image shows the required fields in order to create an account.

Login

In the case that the user already has an account, they may simply sign in using the credentials that they entered when they created the account before. Other features, such as password recovery and authentication are illustrated in the next figures:

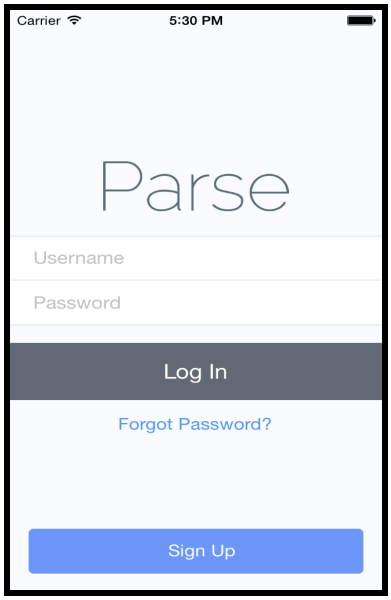


Figure 2. The login view for users who already have an account.

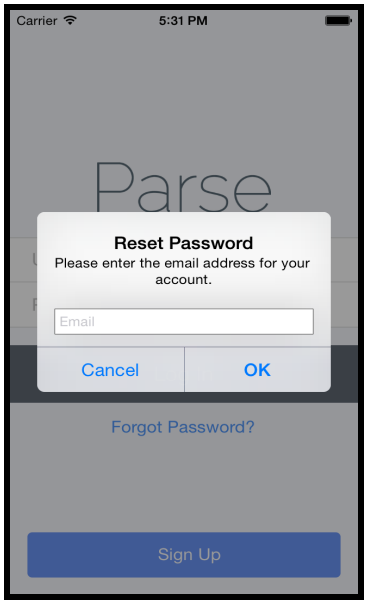


Figure 3. This image shows how to recover their password, which will be emailed to the user.

Chatting with Friends

Users are able to chat between a friend or group of friends by clicking on the bottom left tab. This displays a view of all current chat sessions happening between the user and friends. The user may create a new session by click on ‘New’, and selecting a friend. This will create a new session, where the user may send messages such as below:

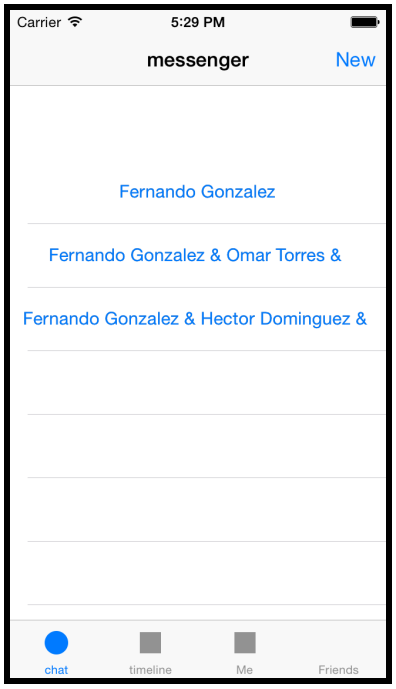
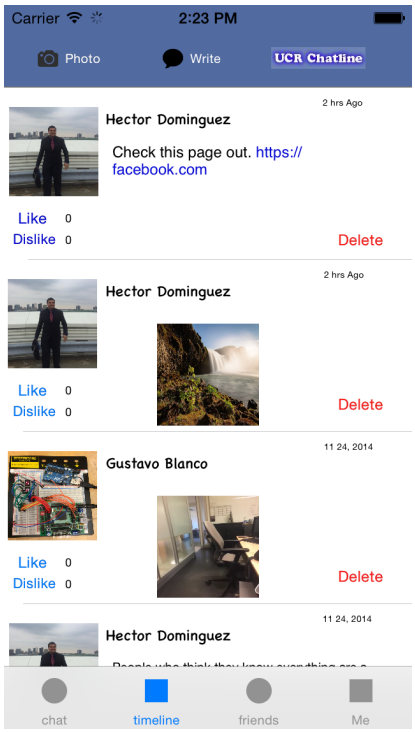


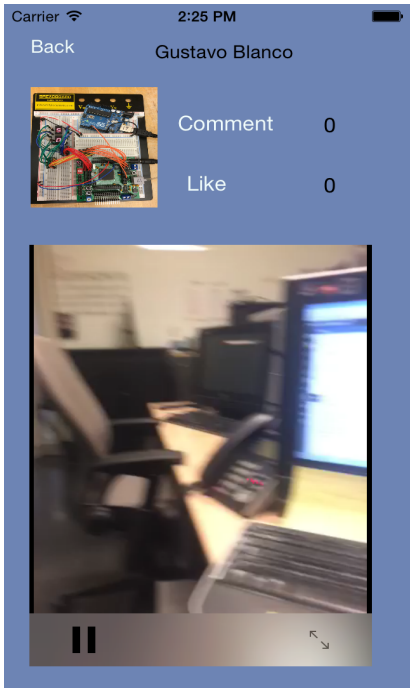
Figure 4: This image shows the message chat viewer, and all the users current conversations.

Figure 5. This image shows an example conversation and it’s corresponding view for it.

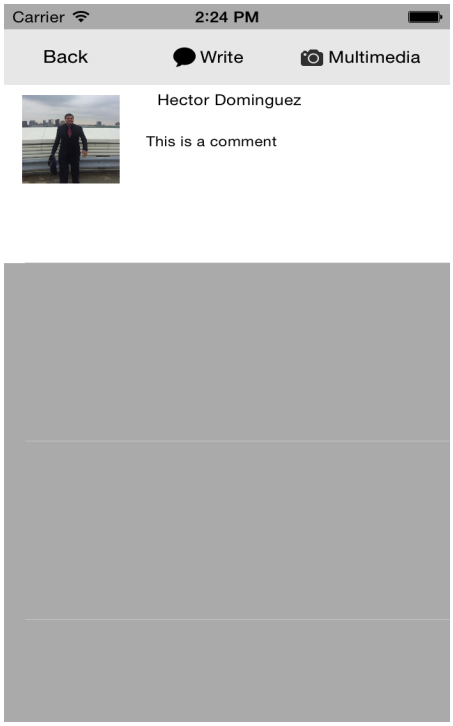
Timeline



This is the the Timeline view of the application. Here the user can post a text , Video, Photo, Link, or emoji moment.

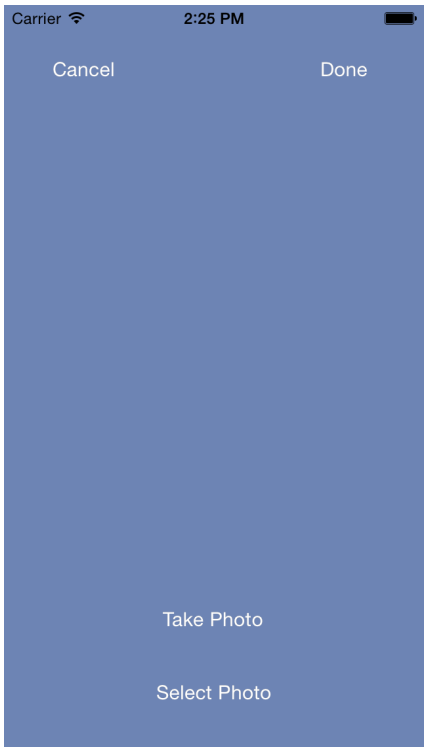


After the user clicks on moment posted on Timeline, the user will taken here where they can review the post and have the ability to comment on the specific post. If the post is a Video, then the video will play. If the post is a picture or a text then it will show here.

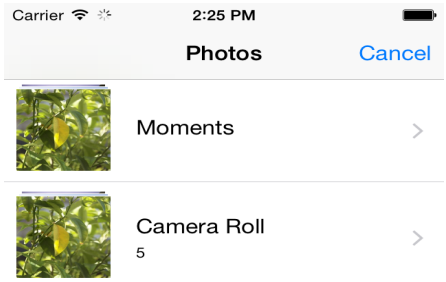


This is the view where the use will see all of the comments on the specific post and has the ability to post a comment if he likes. The user can comment a video, image, emoji or a link.

.



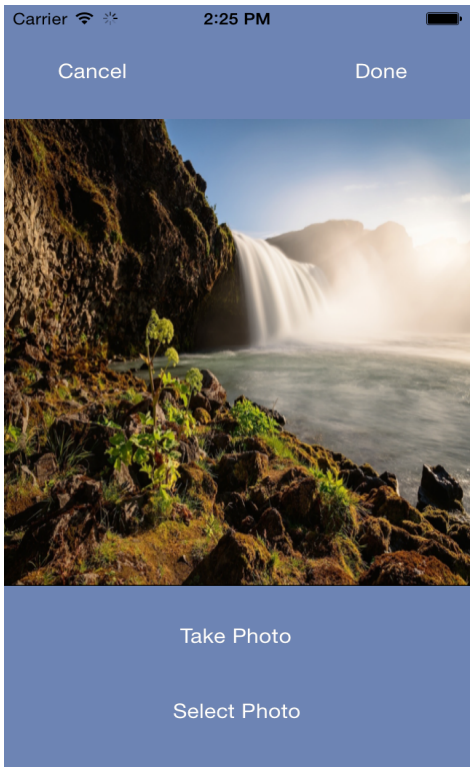
Here is the screen to post an Image or a Video onto the timeline. You can select a video/image stored on the phone or you can take one on the spot.



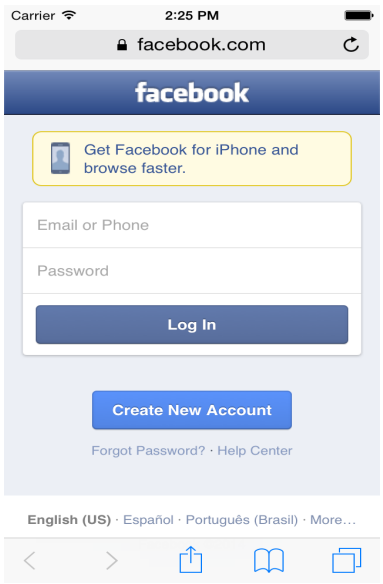
This is the view where the user seects a pcture stiored on the phone either for a comment post or a moment for timeline.

Here is where the user has selected the image and can choose what part of the image he desires to post.

.

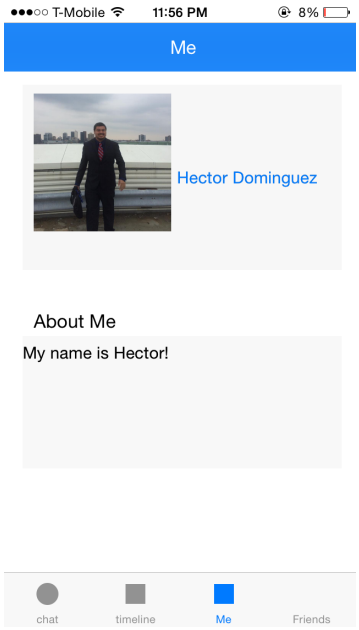


Here the image has been seected and resized and is ready to be posted



Shows that URL's are actually linking from Timeline to the desired website.

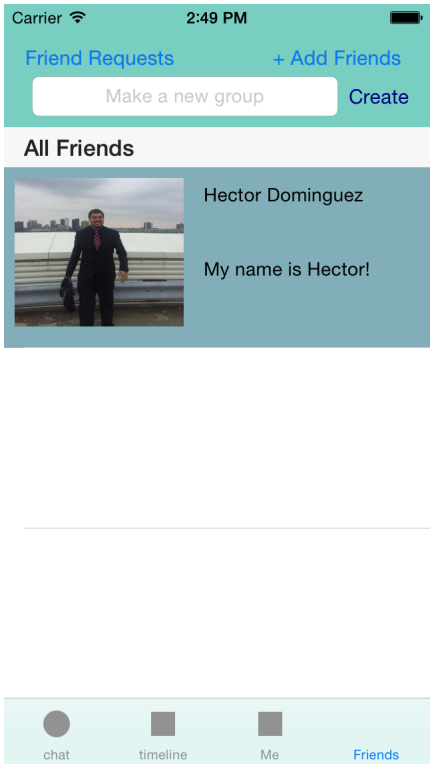
Profile



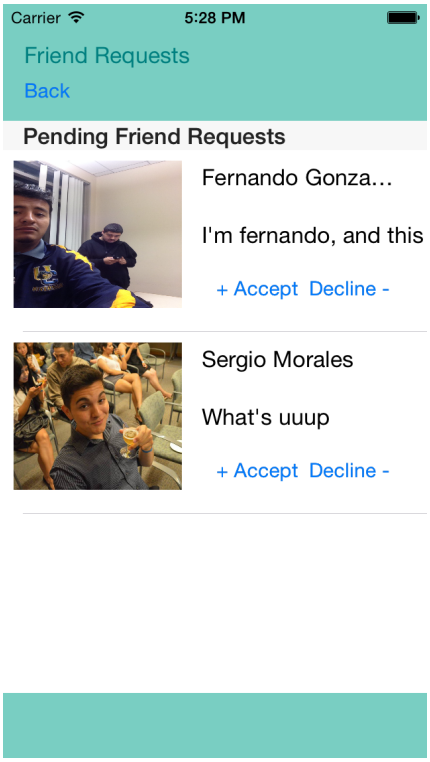
Users are able to edit their pictures, about me description, and name by selecting the ‘Me’ tab in the bottom bar. This will determine what information is available to the user.

Friends

In UCRChatline, users may add friends, delete them, and assign them to groups. Posts and messages in the app are only viewable by friends, so the user must have friends before being able to use the app's main features.



Clicking on the Friends tab displays a list of friends that the user has added. Groups are displayed after clicking on create groups button. They may assign a friend to the group after clicking on the friend cell and using the Picker view to move them to the group.



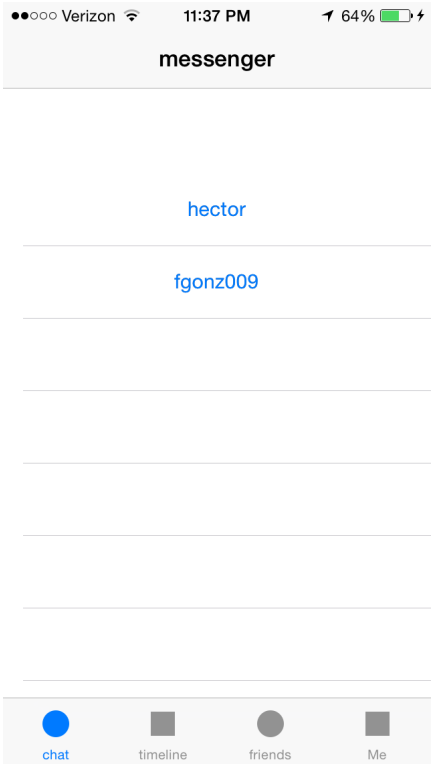
Clicking on Add Friends button displays a list of available friends to add. After sending a friend request, the friend will receive a request, as pictured.

3.2 INTEGRATION TESTING

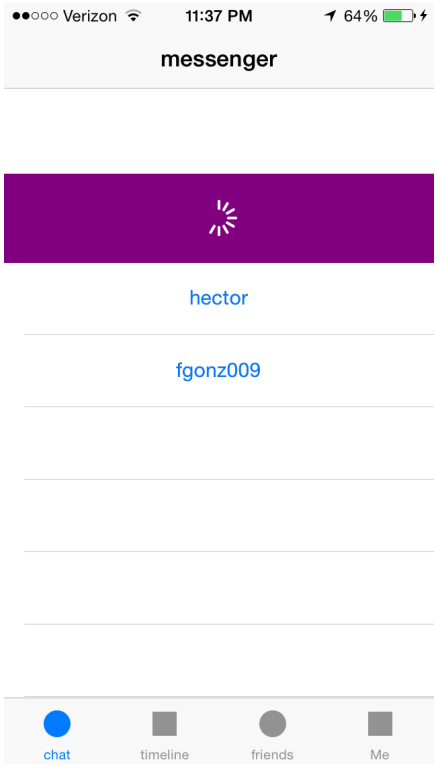
We implemented integration testing by breaking up tasks into groups using Git for version control.

3.3 SYSTEM TESTING

We tested the system using iOS emulator function on Xcode 6.1. The following diagrams represent them:



Testing the system for chat by sending messages to a friend



Testing that messages a refreshed