

Movement Code Overview

Parse Database Structure:

Automoose1

Analytics

Data Browser

Cloud Code

Push Notifications

Settings

Classes

+ Row

- Row

+ Col

More

Installation

14

Role

0

User

14

Activity

25

Event

0

EventActi...

27

Location...

15

objectId

Stri...

appIdIdentifier

String

appName

String

appVersion

String

badge

Number

channels

Array

deviceToken

Strin

KroCmIXY9W

com.tofustan.movement

Movement

1.0

0

["user_0kZZjXNQsA"]

2c1b6eb43295df3d

R6i73pDVmJ

com.tofustan.movement

Movement

1.0

0

["user_mJzww05Z5g"]

45ce884ebfdc0869e

4ABRpa5OGz

com.tofustan.movement

Movement

1.0

0

["user_LUW2gx0ES5"]

113752ee82270094

Qv1rdlCalY

com.tofustan.movement

Movement

1.0

0

9731866cfc6134fac

E6MziJD08U

com.tofustan.movement

Movement

1.0

0

["user_3HwmAbcmyM"]

(undefined)

hlbzDj4doA

com.tofustan.movement

Movement

1.0

0

["user_3HwmAbcmyM"]

(undefined)

N1sUPugbYw

com.tofustan.movement

Movement

1.0

0

["user_3HwmAbcmyM"]

(undefined)

WoSGV0LQhW

com.tofustan.movement

Movement

1.0

0

["user_3HwmAbcmyM"]

(undefined)

3ohYreVAvX

com.frank.Automoose

Movement

1.0

0

(undefined)

(undefined)

O6kqhaURaU

com.frank.Automoose

Movement

1.0

0

["user_3HwmAbcmyM"]

(undefined)

lbo32MtV3o

com.frank.Automoose

Movement

1.0

0

(undefined)

(undefined)

9Lbjy474vc

com.frank.Automoose

Movement

1.0

0

44176a158c174501

tcYmrOIZk2

com.frank.Automoose

Movement

1.0

0

["user_2C6K58jgof"]

(undefined)

klwRzSIIF

com.frank.Automoose

Movement

1.0

0

(undefined)

Above image shows the parse data base structure for Movement. Various classes in the database are seen on the left side.

Installation: This table shows the statistics about which device(+details) is installed and which device has subscribed for push notifications.

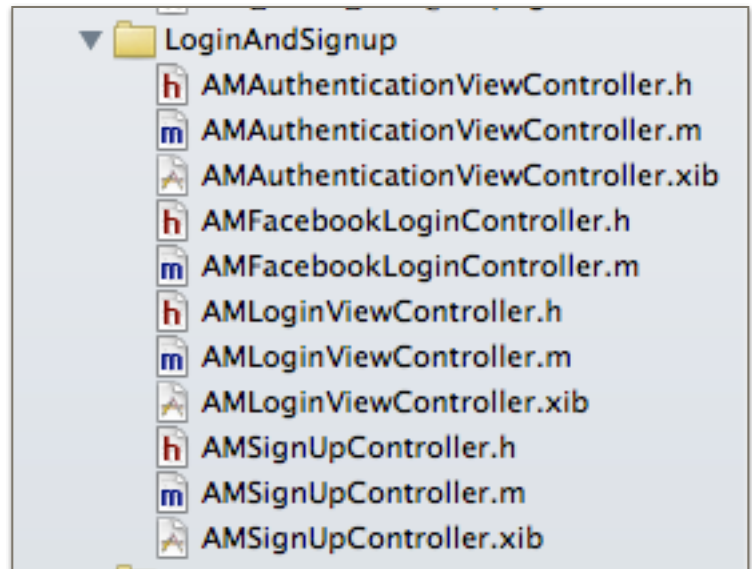
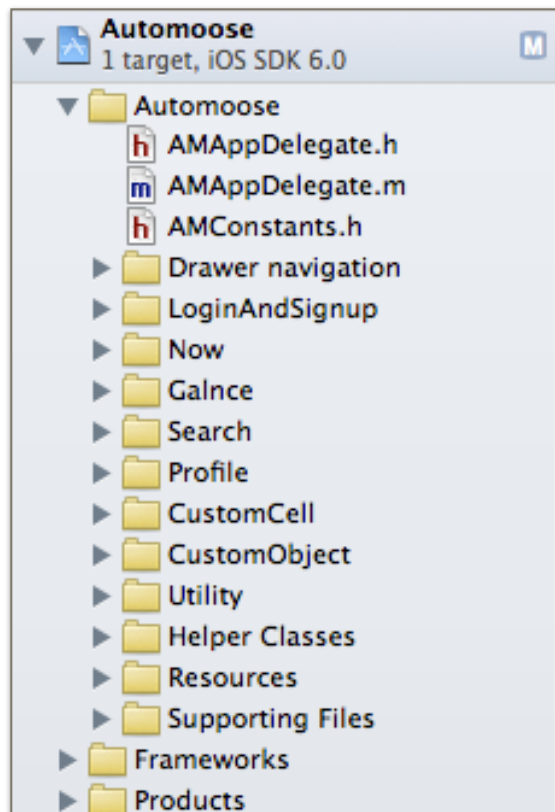
User: This table has all the user related stuff like display name, profile pic, joined date, email id and other trivial stuff.

Activity: This is the first table which the app actually interacts after initial signup/sign in. This table has data about who us follows any user and who this user follows. We differentiate between these using activity type and object id's of the users.

Event Activity: This table has all the data corresponding to events. Details like event details, number of events created or attended by any user are all saved and fetched from this table.

Location: This table has data of the user's location when any event is being tracked in 'Now' screen.

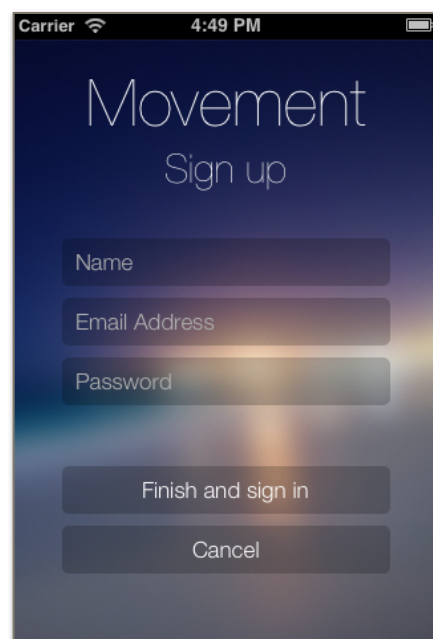
Code folder layout:



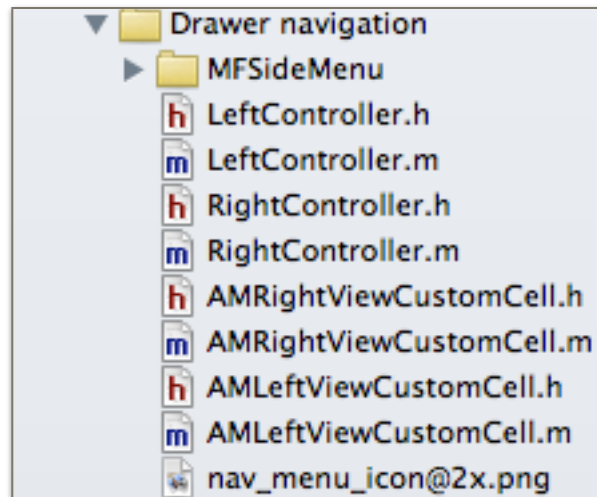
This is the code structure of entire classes divided between different groups.

AM is the abbreviation of "AutoMoose". When we started the app development this was the name, so, I followed this convention to create classes.

AMAppDelegate - This contains application life cycle. I used this to check if user has already logged in or not and correspondingly show login or main screen home screen. This class also has logout functionality code.

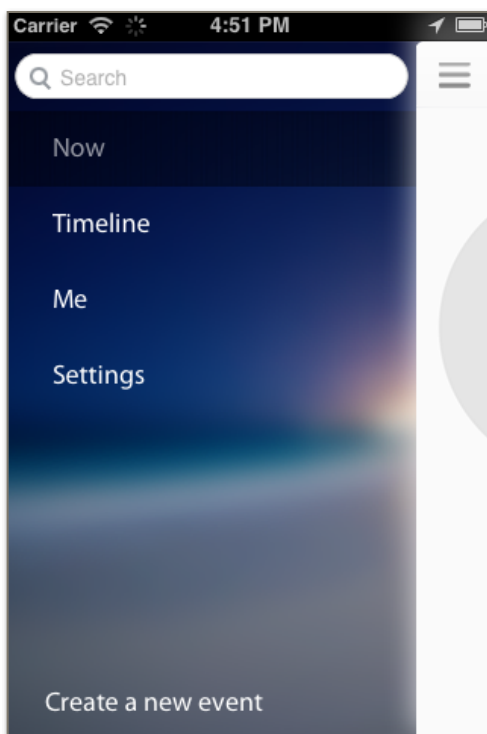


Drawer Navigation-

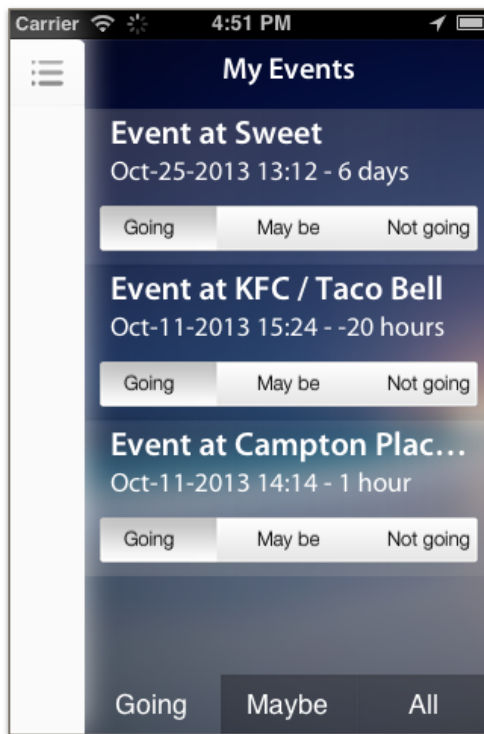


MFSideMenu is an open source library we used for side menu navigation. LeftController and Right Controller has corresponding view functionalities. Search controller is also embedded in left controller.

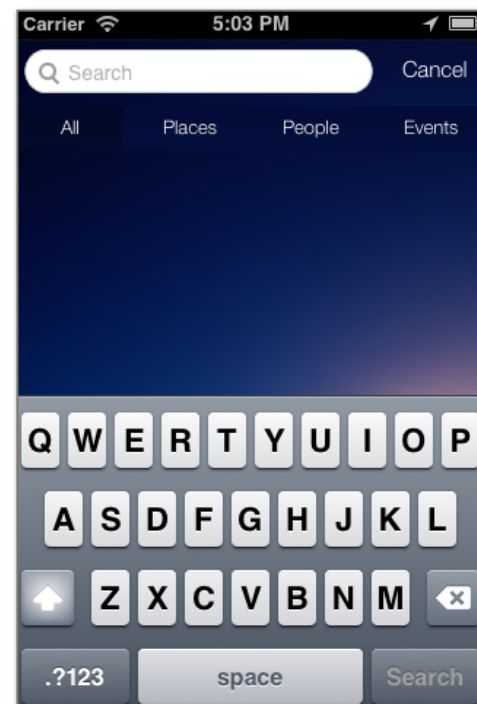
Left Controller



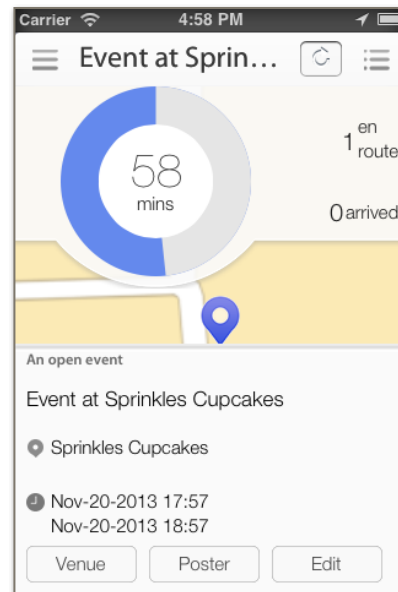
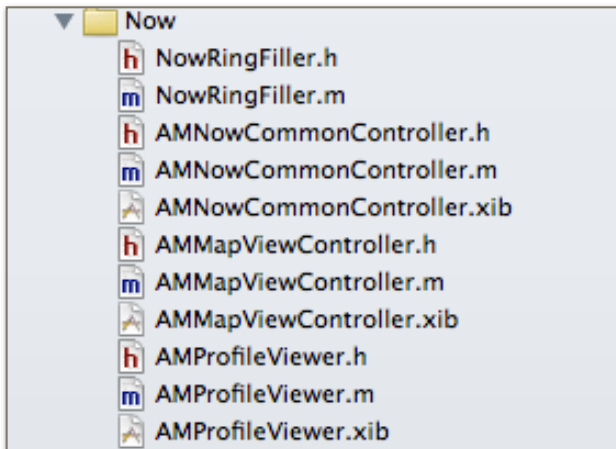
Right Controller



Search Controller



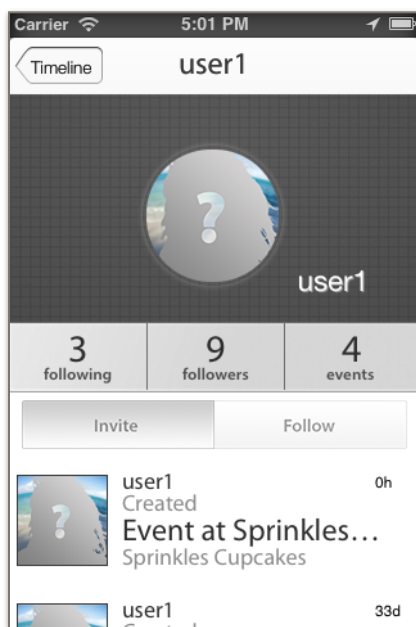
Now



AMNowCommonController - This class housed the count down view at top , event details at the bottom and map view in the background.

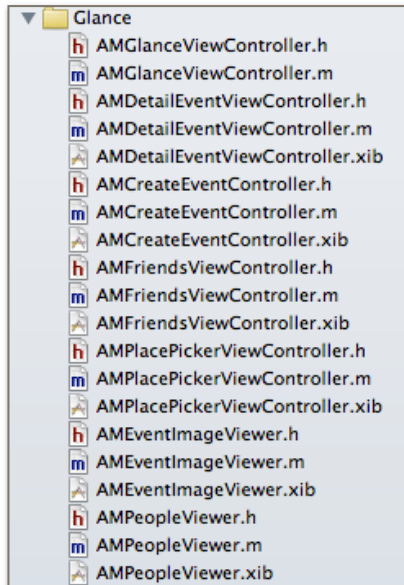
AMMapViewController - Background mapview in AMNowCommonController is instance of this class. Delegation is used to communicate between these two classes.

AMProfileViewer -



This class is used to show the profile for any user profile tap actions - from timeline, search screen, followers, following(Re-usable class).

Glance



This group has all the major classes required to create an event (includes invitee selection and place selection).

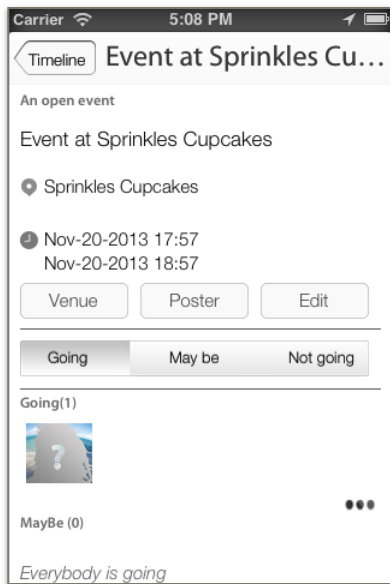


AMGlanceViewController - This class has the functionality to show time line of user. All of the required UI is written via code(not xib).

Tapping on profile image takes user to corresponding user profile (AMProfileViewer).

Tapping event description will take user to AMDetailEventViewController.

Tapping on poster opens the poster in full screen(AMEventImageViewer).



AMDetailEventViewController - This class shows the event in detail.

Tapping on venue takes user to AMPlaceViewController with location details.

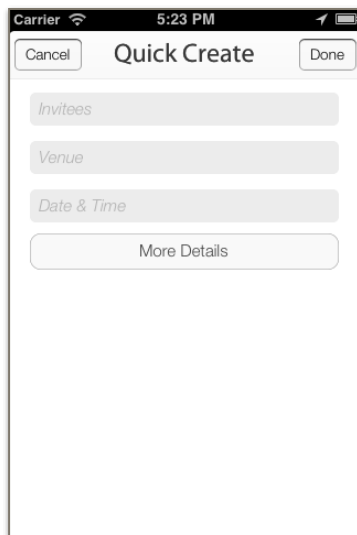
Tapping poster takes user to AMEventImageViewer.

Tapping on edit takes user to AMCreateEventController for updating the event.

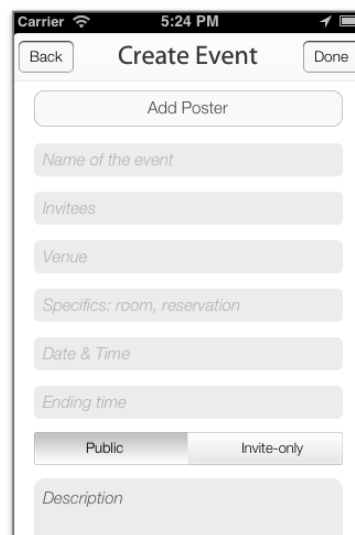
List of all people either going/may go to the event are shown to user when he/she taps on more button on the right bottom of each category. This is done using AMPeopleViewer.

Same functionality is present AMNowCommonController event details.

AMCreateEventController :



Quick create

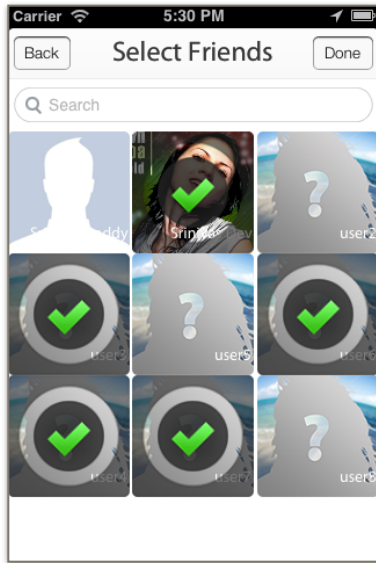


Detailed Create

This class is used to create an event and is shown to user when user taps on create event from left controller. User can create an event illetllow selection of followers.

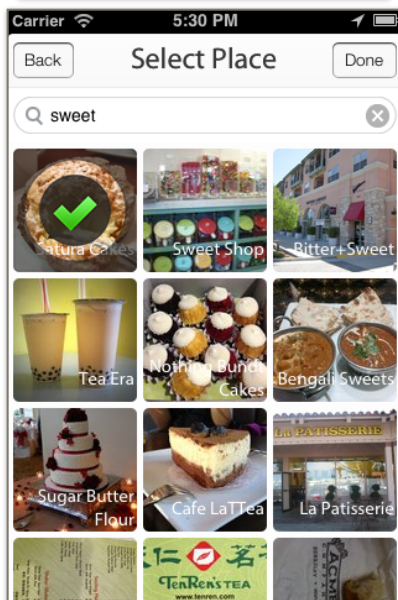
Tapping on venue takes user to AMPlacePickerController for place selection for the event.

Other details are self - explanatory in the screens.



AMFriendsViewController - This class lets user invite people for the event. This is called from AMCreateEventController.

Data for this screen is cached and refreshed for every refresh call from timeline to get the latest people following the user.



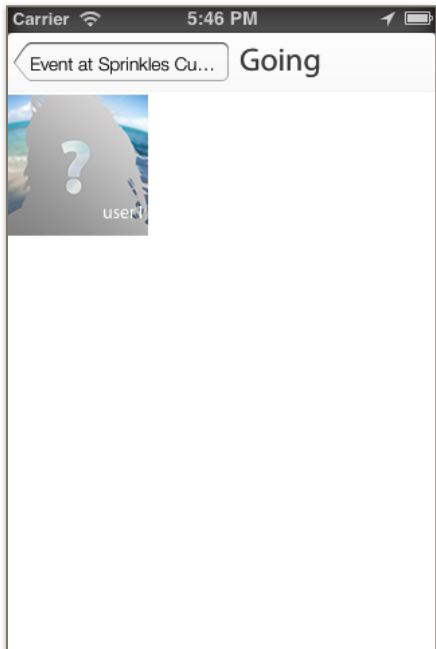
AMPlacePickerController - This class is used to select location for the event. This is also called from **AMCreateEventController**.

Basing on user's input string the search bar, relevant details are obtained from Yelp and presented on the screen.

Delegation is used in both AMFriendsViewController and AMPlacePickerController to communicate to AMCreateEventController.

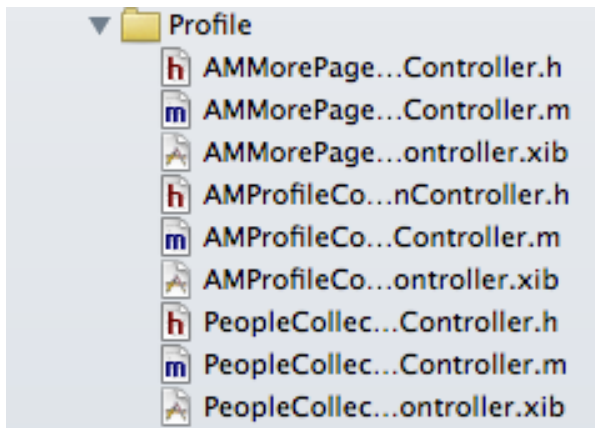
AMEventImageViewer - This class is used to show poster selected for event in full screen. This can be done from now screen(event details), time line screen and event details screen.

AMPeopleViewer- This class is called when user taps on more going/more may be going button in event detail screen.

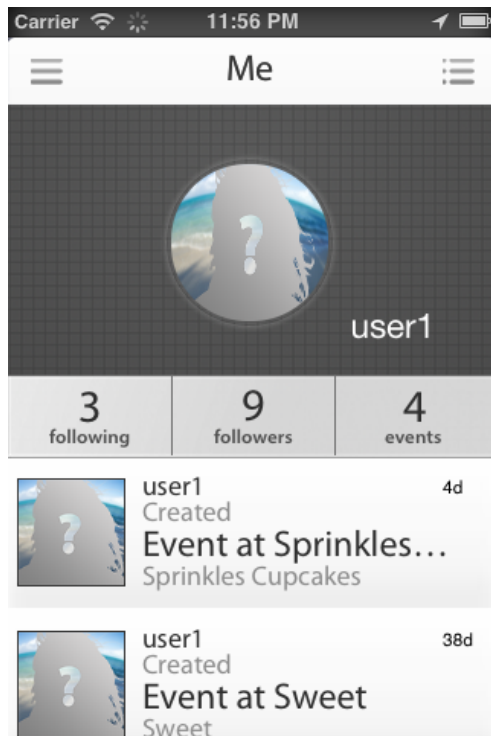


Tapping on any user's profile picture opens user's profile in AMProfileViewer.

Profile and Settings:



AMMorePageViewController

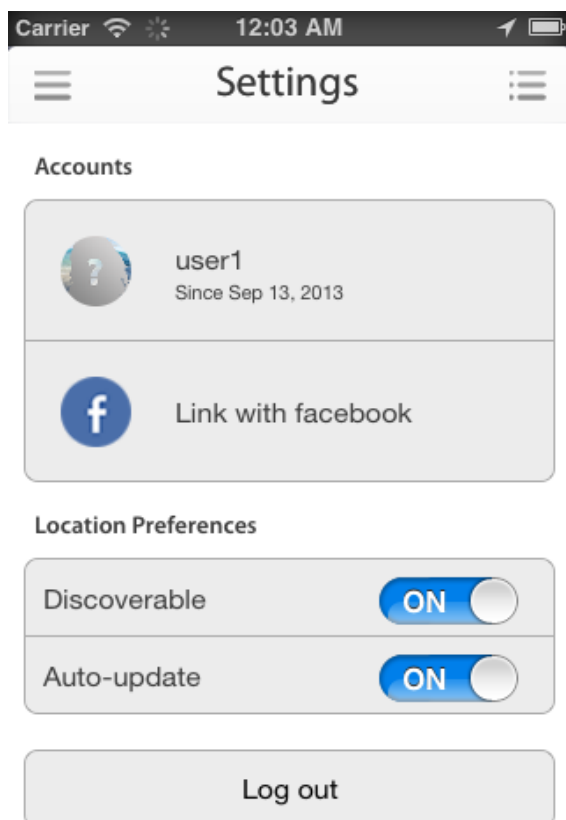


This controller is the profile of user. Name is different because this controller was used for more page when tab bar controller was used in the app. When side bar approach is used, i converted existing controller to profile controller.

PeopleCollectionViewController is used to show grid of people following and followers. Tapping on any event opens AMDetailEventViewController with event details.

Tapping on profile image opens camera album to pick image and PECropViewController(open source) is used to implement crop functionality.

AMProfileCommonController



This is the settings controller of the application(opened from side bar). Functionalities in this controller are self explanatory.

Utility Classes:

AMDownloadMngr: This class is used to download images in place selection controller in gridview.

AMUtility: This class is basically a singleton helper class which has various methods as which we might need to run in background or more often.

AMGridView: As name suggests, this class is used to arrange users or places on a gridview. Delegation approach is used to notify when user taps on any grid view component.