



# FRIENSO APP

A  $\mu$ Social Network for Safety on College Campuses

Development Team

Tuesday 17<sup>th</sup> June, 2014

# CONTENTS

---

## CONTENTS    ii

1	INTRODUCTION	1
1.1	Minimum Viable Product (MVP)	1
1.1.1	Overview of the Frienso UI	1
1.2	Feature Set	6
1.2.1	Pending Requests	6
1.3	Data-Store Structures	7
1.3.1	NSUserDefaults	7
1.3.2	CoreData	7
1.3.3	Parse - Cloud Store	7
1.4	Team Collaboration	8
1.4.1	README iOS Code File	8
1.4.2	Team Collaboration on Github	8
1.5	Documenation Structure	8

# INTRODUCTION

---

## 1.1 MINIMUM VIABLE PRODUCT (MVP)

*Frienso's MVP describes the minimum feature set and functionality to frame the vision of what the mobile app can do to leverage help from trusted friends, encourage bystander interaction, and promote student safety on college campuses.*

### 1.1.1 Overview of the Frienso UI

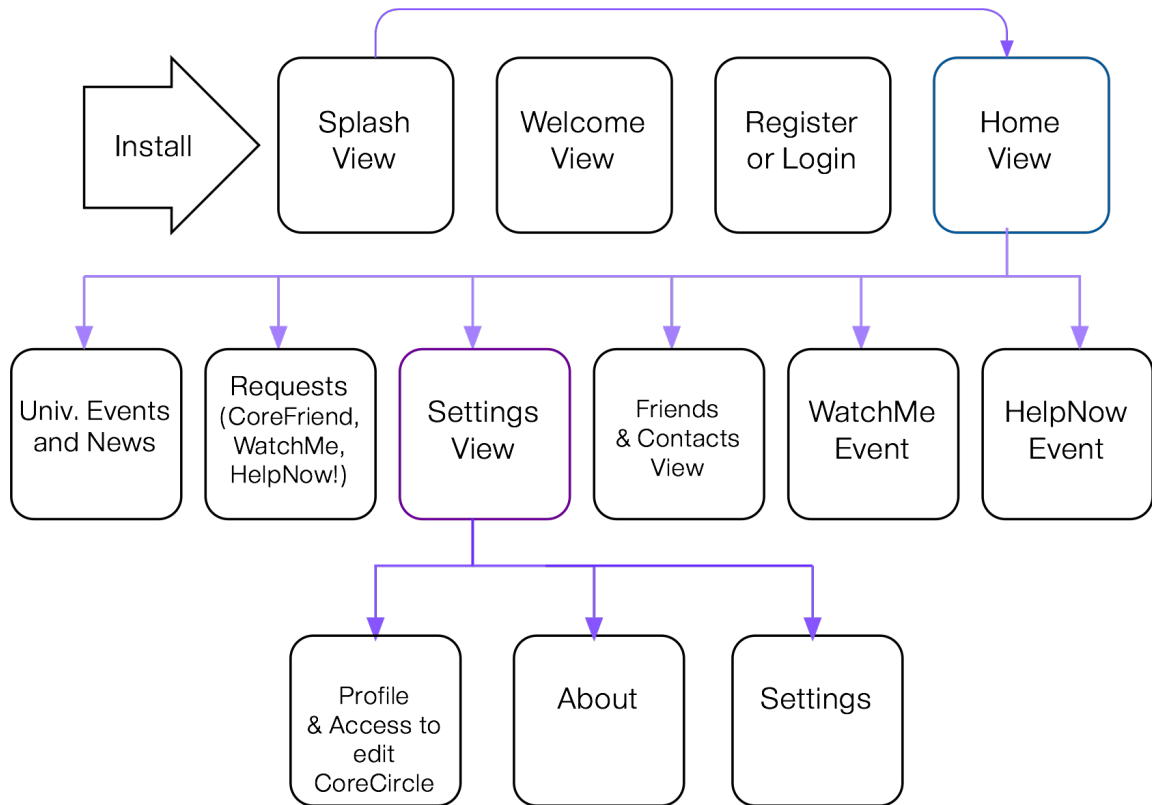


FIGURE 1.1 – An overview of the Frienso (alpha) showing the main blocks and views. .

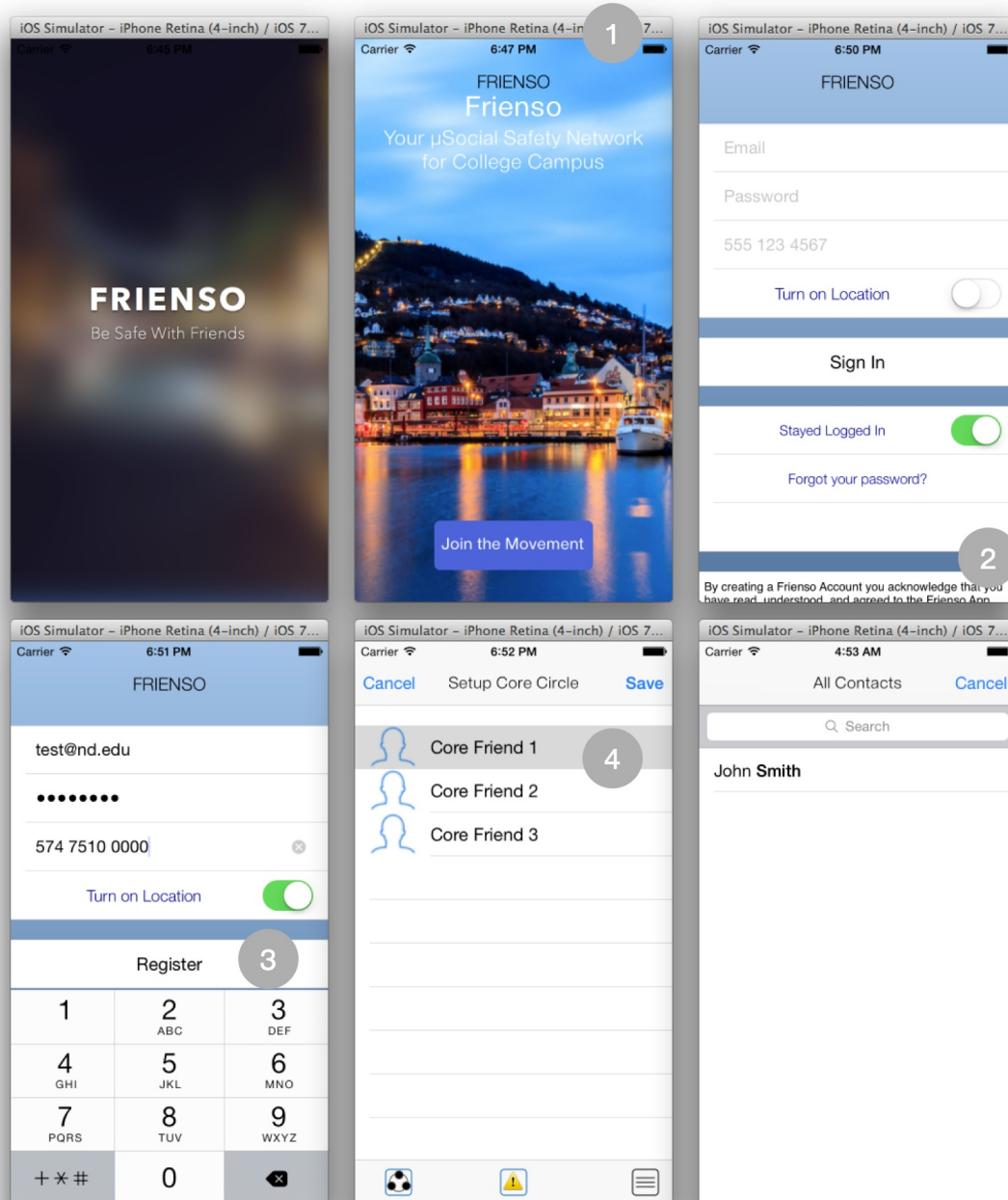


FIGURE 1.2 – Splash view, Welcome view, Login/Register view, and in the bottom center is the create your Core Circle view. (1) The wording between the splash view and the welcome should be more consistent (bug with 'FRIENSO' repeated on the Welcome view), (2) The disclaimer is near the bottom of the Login/Register view with a working link to a web-accessible page, (3) The Login/Register button change depending if the user is registering for the first time or if the user has already created an account, (4) Select the contacts from the iPhone Contacts app to be part of your CoreCircle of friends. .

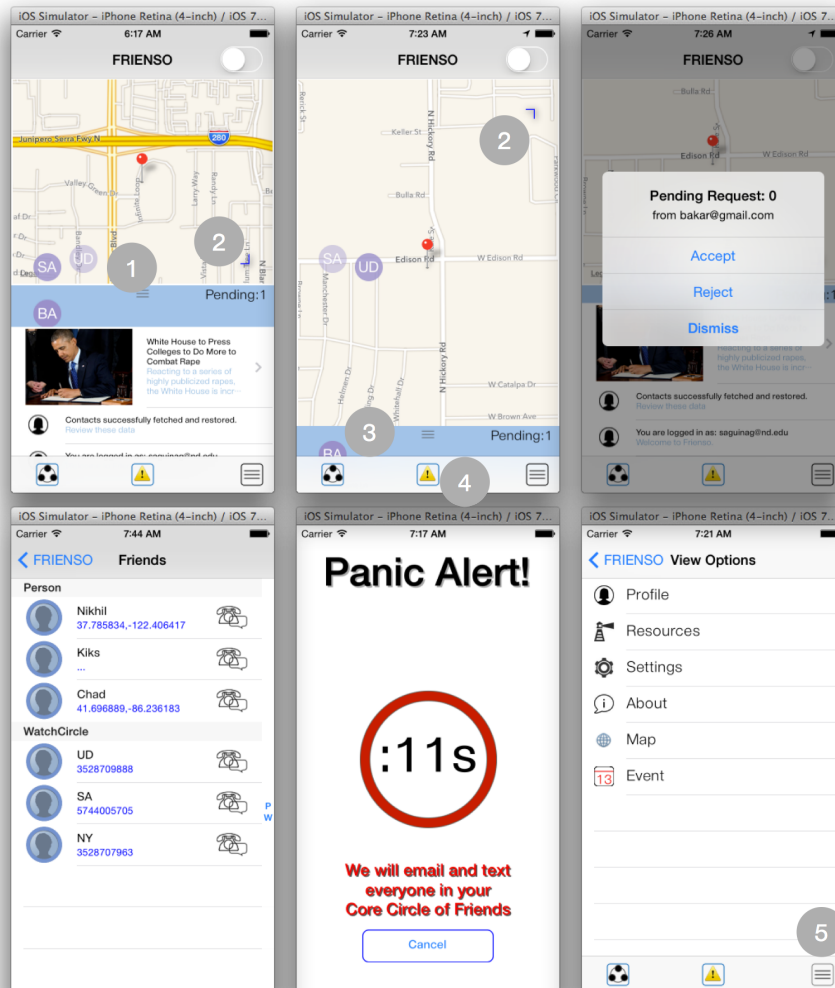


FIGURE 1.3 – Home view and toolbar features: Home view, Fullscreen mapview, Pending requests drawer, Friends view, HelpNow view, View options (top left to bottom right). (1) Pending requests drawer, (2) Notice the fullscreen and normal mode mapview buttons the user can use to toggle the mapview between fullscreen and normal mode. Pending requests enter into the drawer and when the user selects the bubble an alert-view gives the user options to accept or reject it. .

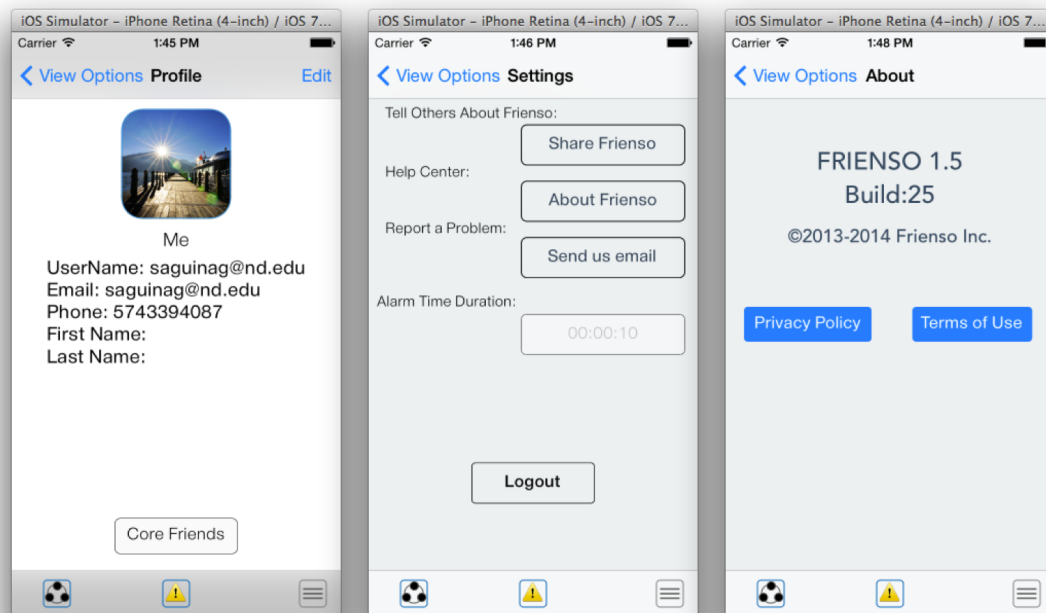


FIGURE 1.4 – test .

## 1.2 FEATURE SET

### 1.2.1 *Pending Requests*

Pending requests fall into four types: *watch*, *helpNow*, and *coreFriend*. Users have the following options: to accept, reject, or dismiss the request.

#### 1.2.1.1 Type: *watch*

Requests of type *watch* are accessed when a user that added you to their *coreFriends* circle has initiated a 'watchMe' alert.

**watch** alerts will share your location in near real time only with those in your *coreCircle* and only if they have accepted your 'watch' requests. Accept or reject allows for a feedback mechanism. Users that initiate a *watchMe* request will know who is watching them for the duration of the *watchMe* alert active period.

#### 1.2.1.2 Type: *helpNow*

Requests of type *helpNow* are accessed when a user has initiated an urgent help request. Initiating this request type triggers SMS texts to go to those in one's *coreFriends* circle (aka *coreCircle*) and push notifications.

**helpNow** alerts will share your location in near real time only with those in your *coreCircle* regardless if they've accepted your request. Users that initiate a *helpNow* request will also be prompted to dial campus police or 911 automatically.

#### 1.2.1.3 Type: *coreFriend*

Requests of type *coreFriend* are accessed when i) a user creates its *μsocial* network. The user selects from its contacts up to three friends, and ii) when the user changes or edits its *coreCircle*. The actions after the user hits 'save' are to update the local data-store (*NSUserDefaults*, and *CoreData*) and update the cloud-store (*Parse*).

**coreFriend** requests notify contacts that they have been chosen to be a trusted friend on the *Frienso* app. The notification will appear on their phone's *Frienso* app or as a SMS text.

These requests have bidirectional properties, you might ask user X to be in your *coreCircle* and that same user might ask you to be in their *coreCircle*. In this case, the user should only appear once in your *Frienso* friends list.



## 1.3 DATA-STORE STRUCTURES

### 1.3.1 *NSUserDefaults*

Key-value standard defaults:

Key	Value
<i>adminID</i>	username, email
<i>adminPass</i>	password
<i>userName</i>	user's email
<i>userEmail</i>	user's email
<i>adminInParse</i>	flag, o = Not in parse
<i>userPhone</i>	user's phone number
<i>CoreFriendsContactInfoDicKey</i>	NSDictionary of user's coreCircle

### 1.3.2 *CoreData*

The Frienso projects has the following NSManagedObject subclass files and entity attributes:

- **CoreFriends[.h,.m]**
  - property (nonatomic, retain) NSString \* coreEmail;
  - property (nonatomic, retain) NSString \* coreFirstName;
  - property (nonatomic, retain) NSString \* coreLastName;
  - ... see the CoreDataFiles for more details
- **FrienEvent[.h,.m]**
  - property (nonatomic, retain) NSString \* eventTitle;
  - property (nonatomic, retain) NSString \* eventSubtitle;
  - property (nonatomic, retain) NSString \* eventObjId;
  - ... see the CoreDataFiles for more details
- **FriensoEvent[.h,.m]**
  - property (nonatomic, retain) NSString \* resTitle;
  - property (nonatomic, retain) NSString \* resDetail;
  - property (nonatomic, retain) NSString \* resUrlLink;
  - ... see the CoreDataFiles for more details

### 1.3.3 *Parse - Cloud Store*

This projects uses Parse as the cloud-store back-end and can accessed at here: [~ Parse/Dashboard/Data Browser link.](#)

† these classes are under construction.

TABLE 1.1 – Parse Frienso Classes .

Classes	Description
<i>User</i>	The standard user object
<i>UserCoreFriends</i>	Dictionary data structure holds each user's coreCircle
<i>Resources</i>	College/University resources
<i>UserConnections</i>	Linking a phone number to a user (class to be deprecated, because the phone number has been added to the User class).
<i>CoreFriendRequest</i>	†
<i>TrackRequest</i>	†
<i>UserEvent</i>	†

## 1.4 TEAM COLLABORATION

### 1.4.1 README iOS Code File

The iOS project folder contains the file README.md to be used to maintain brief descriptions of the app's version and build number. The build number is a number generated from the git commits. The build version number is updated each time a new executable is generated for ad-hoc distribution.

In addition to the app's version, the README file maintains a list of TODO items and descriptions of how they were completed.

### 1.4.2 Team Collaboration on Github

[Resolving merge conflicts](#)

## 1.5 DOCUMENTATION STRUCTURE