

Vote Application based on iOS

Detail Design Document

Current Version: v1.0

Author: Jianghua Kuai, Shuai Zhao

Oct 26, 2013

Maintenance History

Author	Date	Content	Version
Jianghua Kuai	Oct 26, 2013	Build the Document	V0.1
Jianghua Kuai	Oct 27, 2013	Add architecture design, data design and upgrade the structure of document.	V0.2
Shuai Zhao	Oct 30, 2013	Human Interface Design and update the structure of document	V1.0

Tabel of Content

1.INTRODUCTION	4
1.1 PURPOSE.....	4
1.2 SCOPE.....	4
1.3 REFERENCE MATERIAL.....	4
2.SYSTEM OVERVIEW	4
3.SYSTEM ARCHITECTURE	5
3.1 ARCHITECTURE DESIGN.....	5
3.2 DECOMPOSITION DESCRIPTION.....	6
4.DATA DESIGN	7
5.HUMAN INTERFACE DESIGN.....	8
5.1 OVERVIEW OF USER INTERFACE.....	8
5.2 MAIN SCREEN IMAGES	9
5.3 SCREEN OBJECT AND ACTIONS	12
<i>5.3.1 Interaction of User Management.....</i>	<i>12</i>
<i>5.3.2 Interaction of vote initiation</i>	<i>19</i>
<i>5.3.4 Interaction of attending a vote.....</i>	<i>25</i>
<i>5.3.5 Interaction of Vote History</i>	<i>28</i>
<i>5.3.6 Interaction of exception handling</i>	<i>31</i>
6.APPENDICES	35

1.Introduction

1.1Purpose

This software design document describes the architecture and system design of Vote. The main audiences of this document are developers. And this document is also a summary of all other detail design documents.

1.2Scope

This document will mainly contain the database design, class design for server and client, interface for the communication of client and server, and human interface design.

Some contents may lead to other document because it will be easier to maintain.

1.3Reference Material

1. Class Diagram for client and server
2. C/S communication Interface Design Document
3. Vote Database Design
4. Human Interface Design (stored on fluidui.com)

2.System Overview

This project's client application will implement on iOS platform which will use Objective-C and cocoa Framework, at the same time, this project will integrate Google Analytics into it. For communication with server, since we use the Plankton Server as our framework for server, so we'll use Plankton API for iOS.

This project's server application will implement by using C++ on CentOS, and the database is MySQL.

3.System Architecture

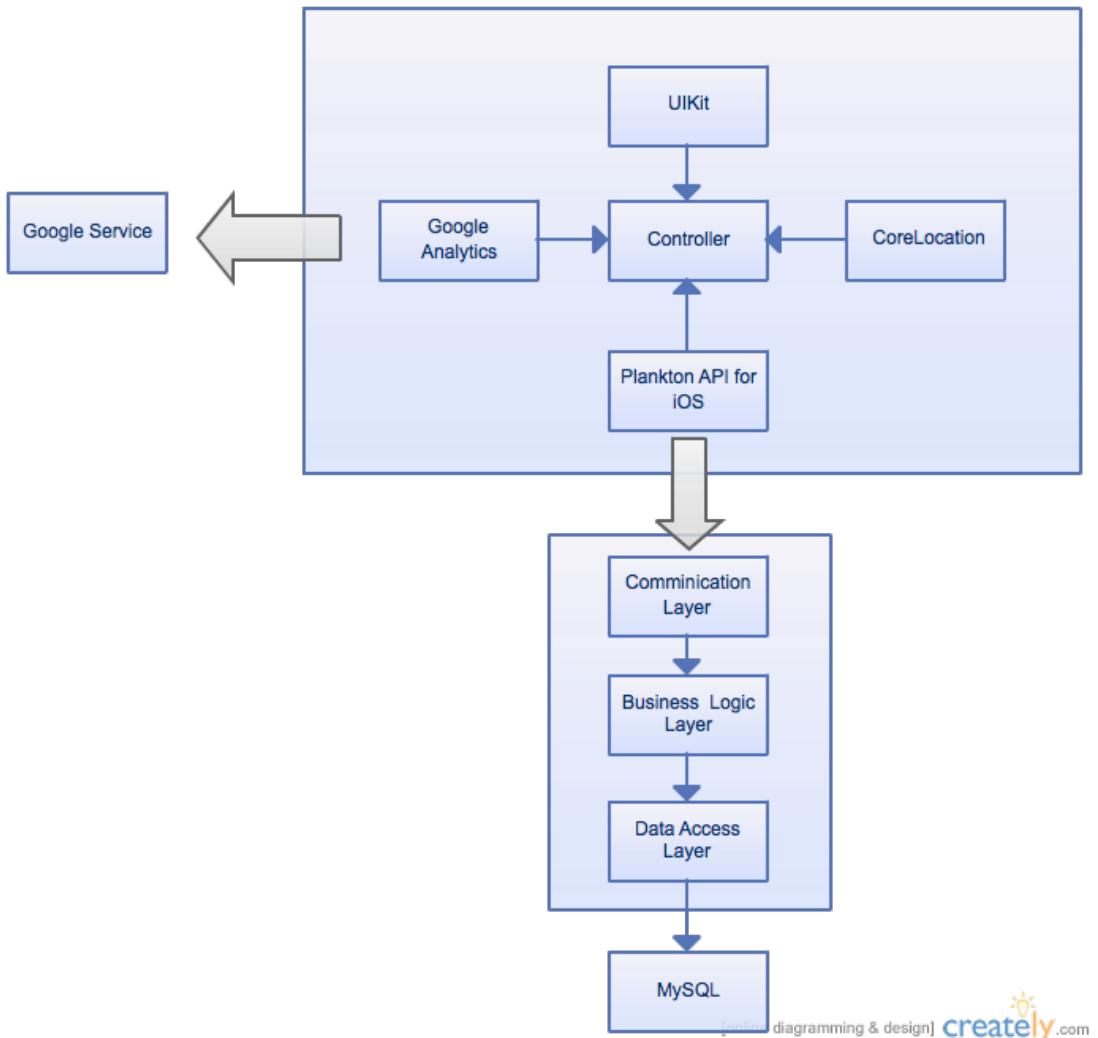
3.1 Architecture Design

This project is implemented by using C/S architecture.

For the server part, Plankton Server is a standard three-tier server framework. We'll focus on the business layer to build our business logics because the framework has already handled the communication and data access for us.

For the client part, cocoa framework is a very pure and standard MVC architecture so we can easily separate the data and operation.

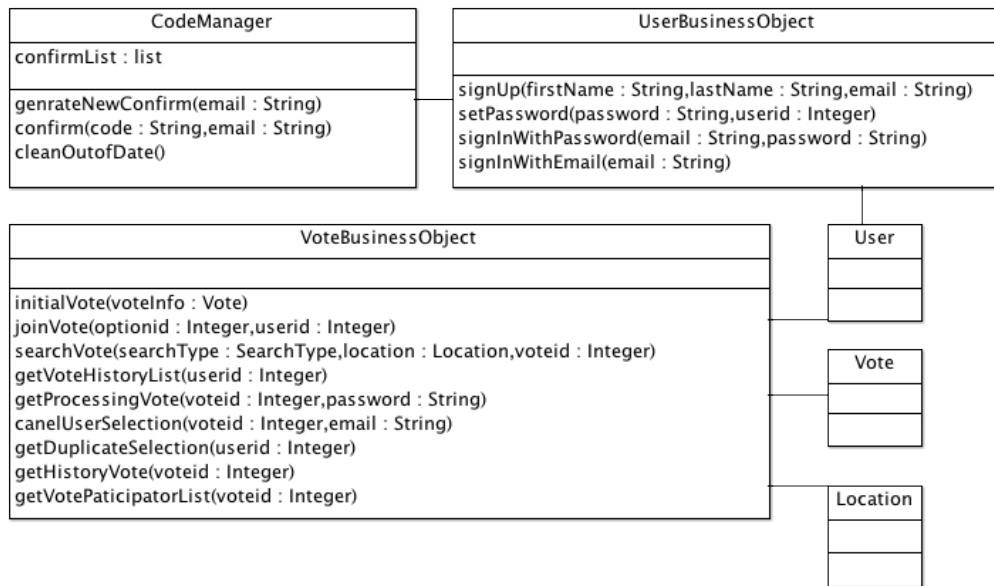
The diagram below shows the architecture of the project from a general view.



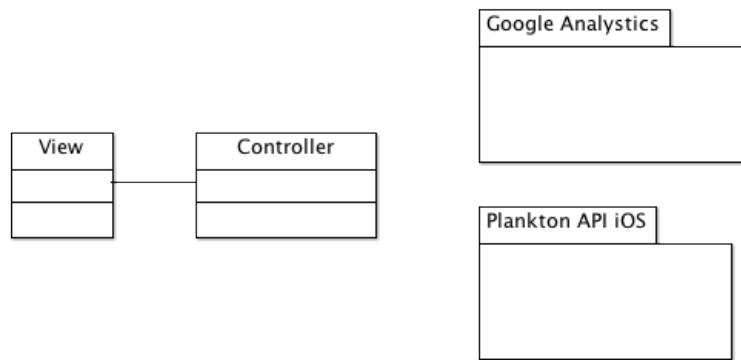
3.2 Decomposition Description

The communication between client and server will format the data as the formation of **Interface Document** described.

Since we will only design the Business Logic Layer for the server, this document will only shows the class diagram of that layer. The diagram below is server's Class Diagram.

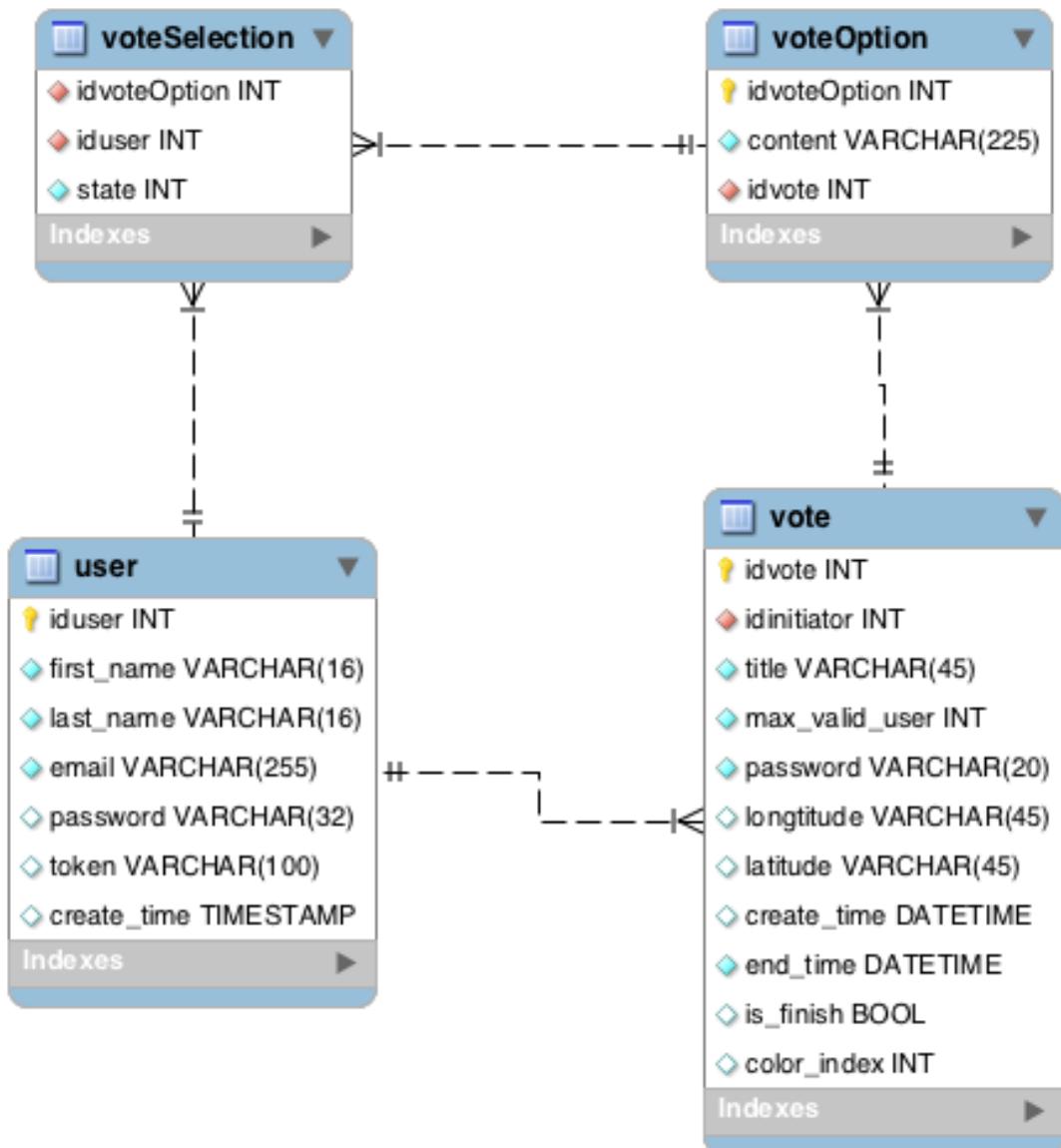


For client part, every controller will report the exception and user preference (if allowed) by using Google Analytics, and every controller will send request from Plankton API to server, so every single controller's class diagram should looks like the image below. The detail relationship of each controller and view will show in human interface design part.



4.Data Design

We use EER to describe our database design and structure.



5.Human Interface Design

5.1 Overview of User Interface

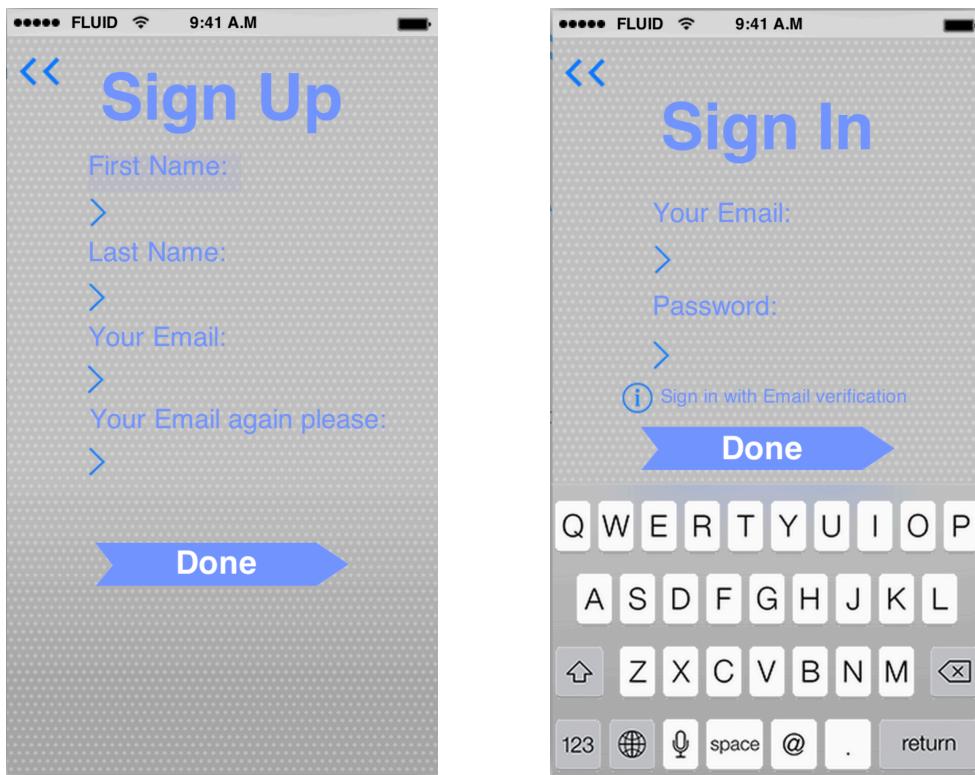
The whole human interface design is flat-design oriented. The main objective is giving users a clear and simple way to enjoy this application. We design this application on the web *fluidui.com*, which is a good place to design mobile application.

5.2 Main Screen Images

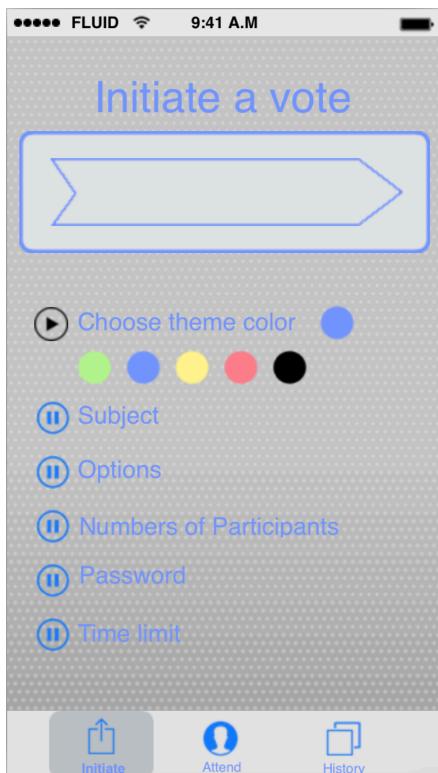
The launch screen image of this application:



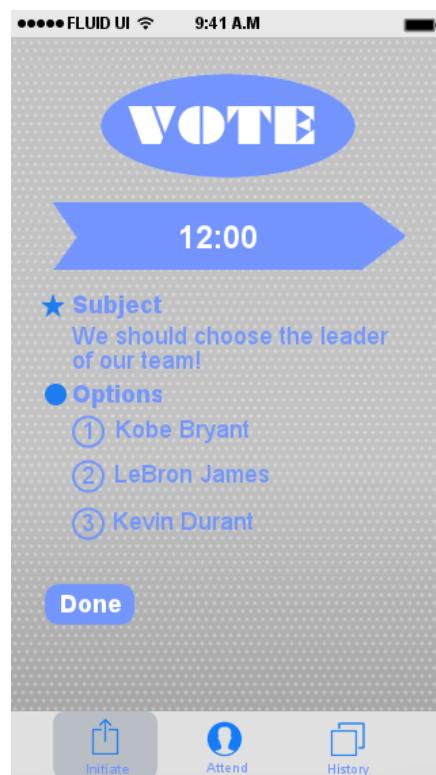
The sign-up and sign-in screen images:



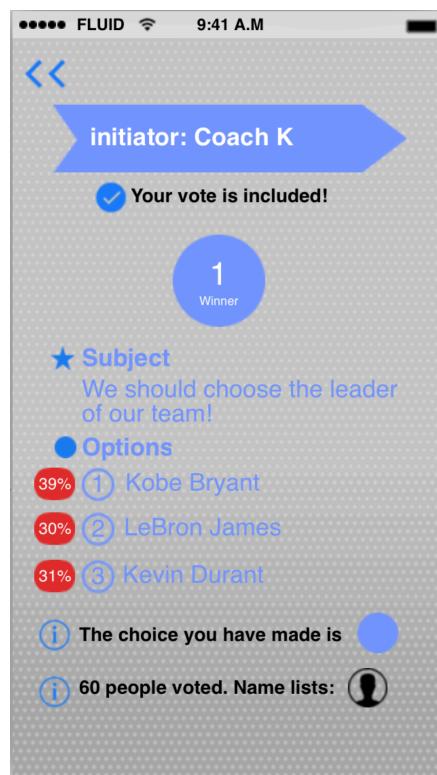
The main images of initiating a vote



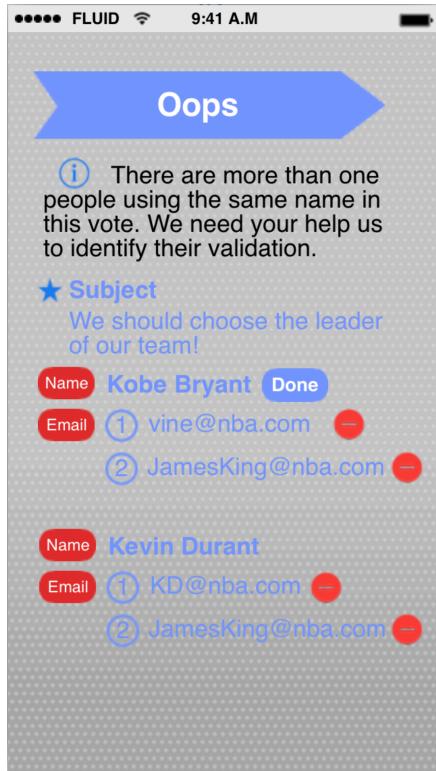
The main screen images of attending a vote event



The main screen images of History:



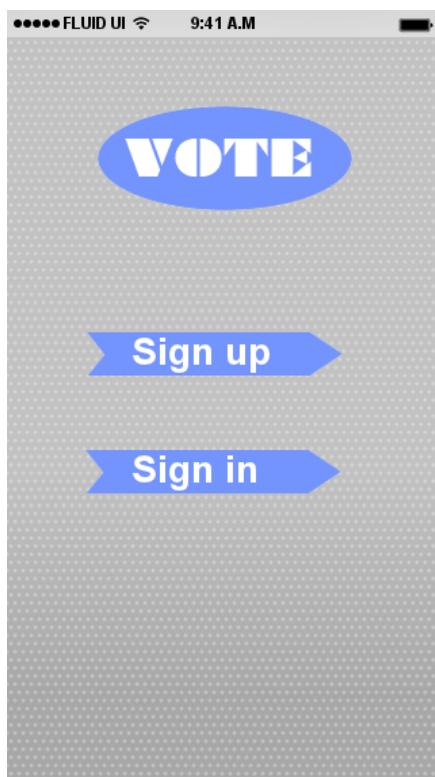
The main image of vote exception handling:



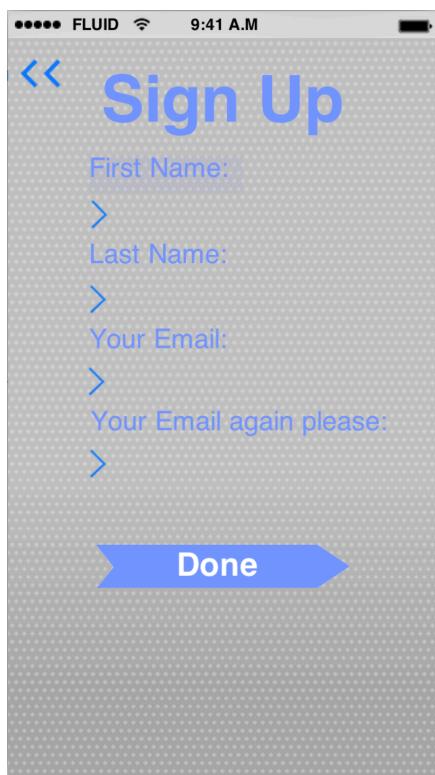
5.3 Screen Object and Actions

5.3.1 Interaction of User Management

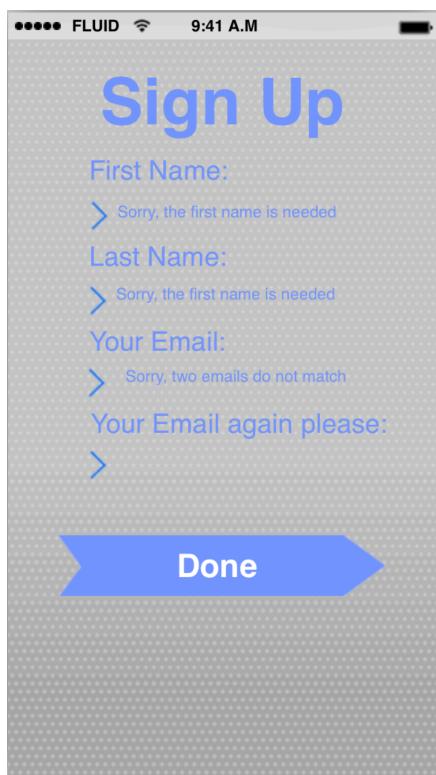
When you first launch this application in a particular iPhone, you would be asked to sign up or sign in with existing account.



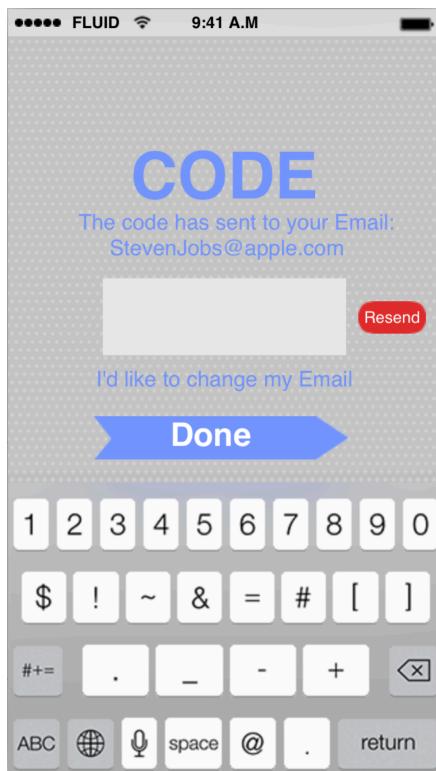
If you choose sign up, you should input your name and Email address in the next step.



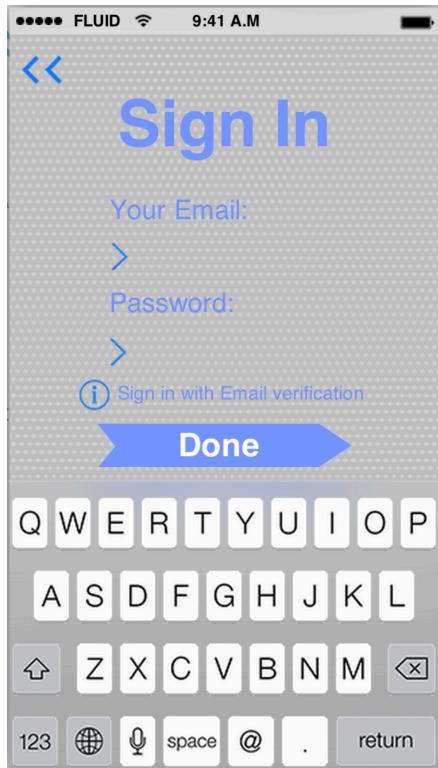
If any of these information are not given by the user or the two Email addresses cannot match, this step cannot be finished.



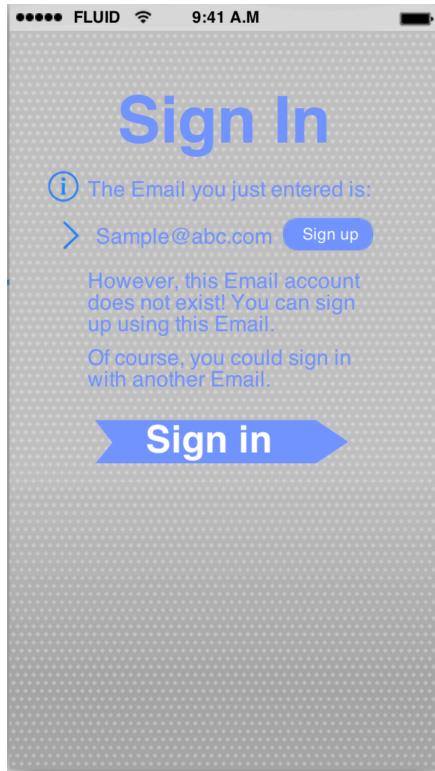
The next step is to validate your Email address, which assure that every participant of a particular vote event is valid. In this step, you can also change your email address if you find out that the Email address you just typed in is incorrect. In addition, if you do not receive your code by our Email after a while, you can try to click on the resend button. When you type in the correct code, you will be permitted to enter the main application.



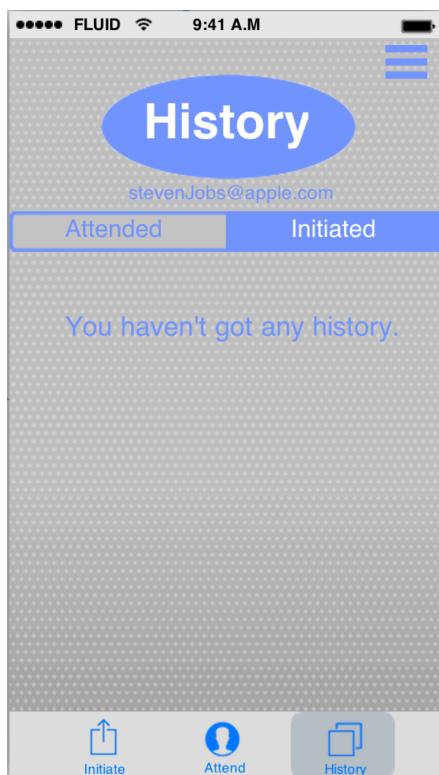
Instead of signing up, you can also sign in with your existing account in a particular iPhone in the first time. You can just type in your Email address and the password. If you accidentally forget your password, you could click on the “Sign in with Email verification”. By doing that, we would send you a code to your email.



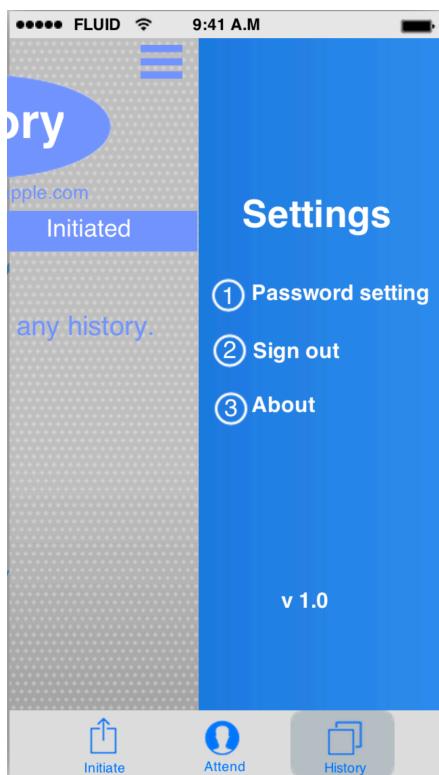
Of course, we take care of some special situations like that someone types in a non-existing account and try to sign in.



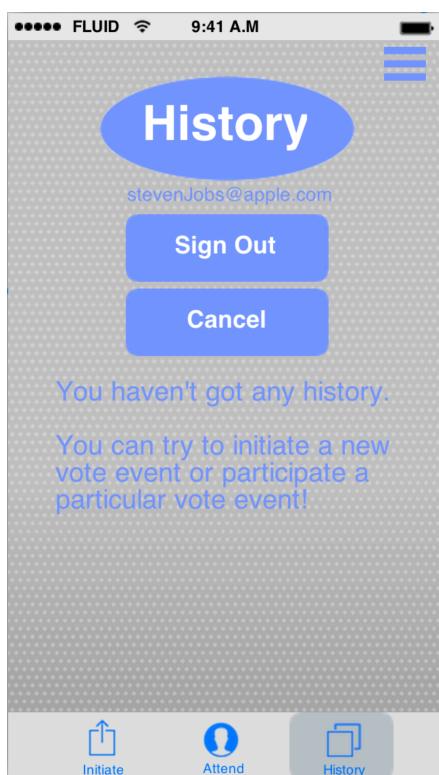
After you sign up or sign in successfully, you would enter into the “history” part of the application. Obviously, it does not have any records before you initiate a vote or attend any votes.



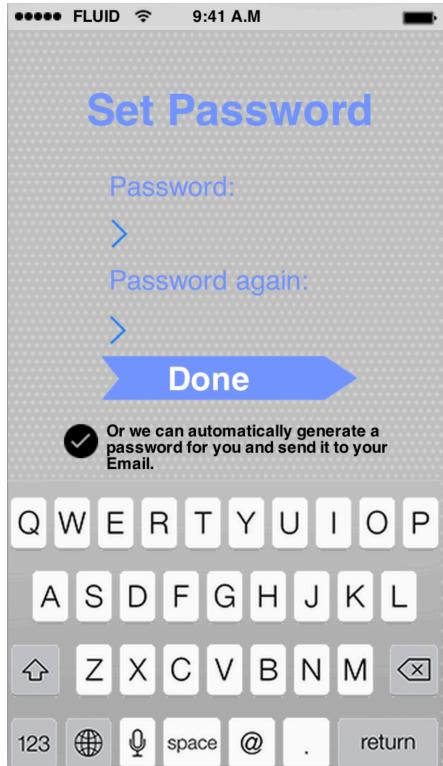
In the “History” part of this application, you could tap the fence-like button, which is in the right-up corner of the screen, to enter the setting menu. As I said, the setting menu would be slide from the right edge of the screen after the fence-like button is tapped. You could set the password of your account, sign out or see the information about this application. When you tap the “password setting” button, you would be asked to enter the current password so that you can set a new password, unless you have not set s password for you account.



In the “history” part of this application, you could also see your account name, which is your Email address, under the “History” icon. When you tap on your Email address, there will be two buttons sliding down from the Email address. Clearly, you could also sign out in here.

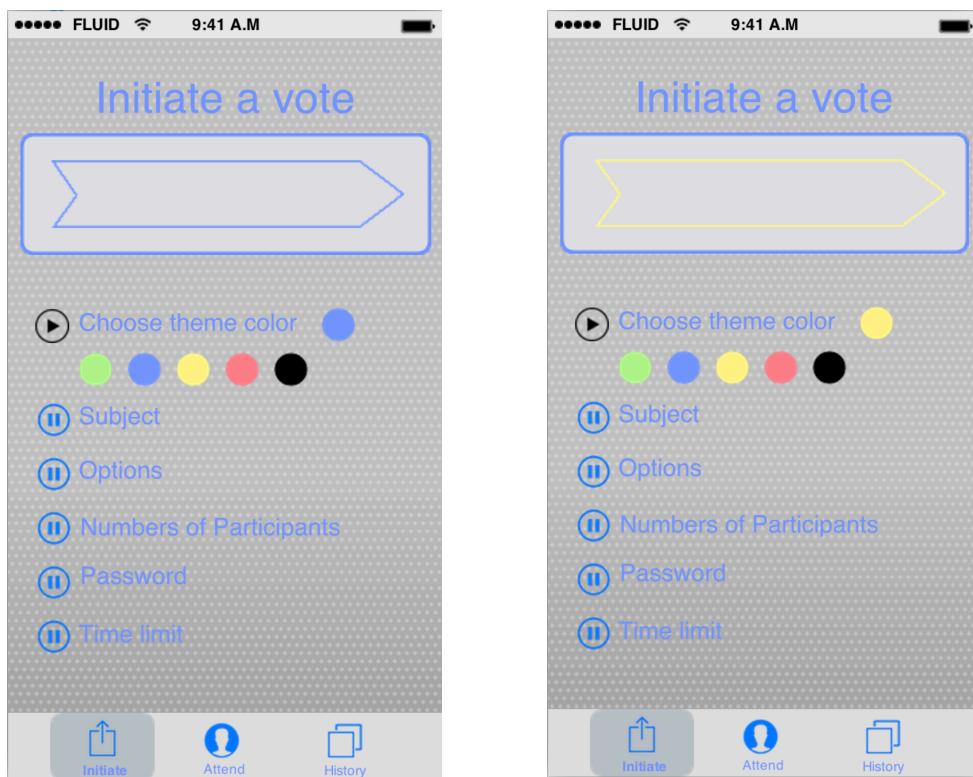


There is a highly possible situation that someone who never sets a password wants to sign out. So in such situation, we would ask the user to set a password before he or she actually sign out. What is more, he or she could also choose another option that we would automatically generate a password, which will be sent to the user's Email.

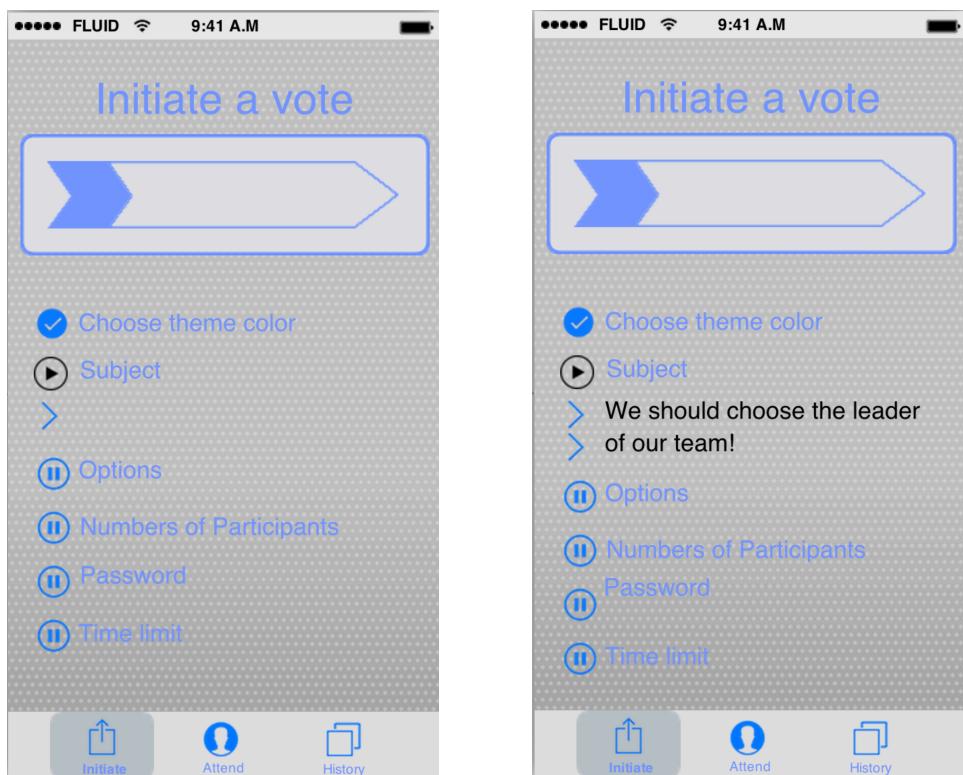


5.3.2 Interaction of vote initiation

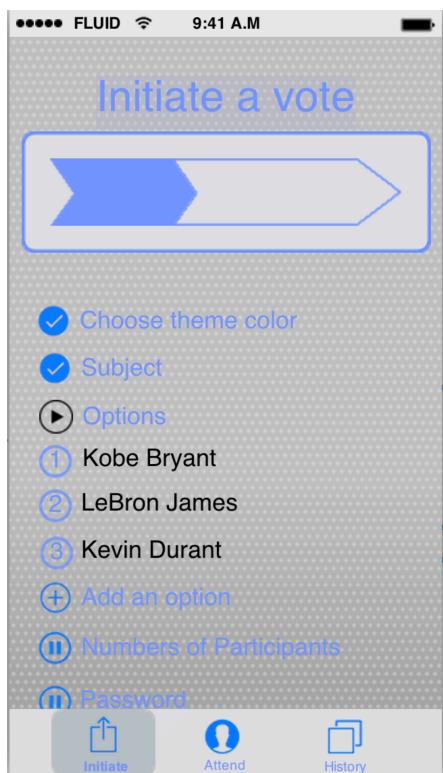
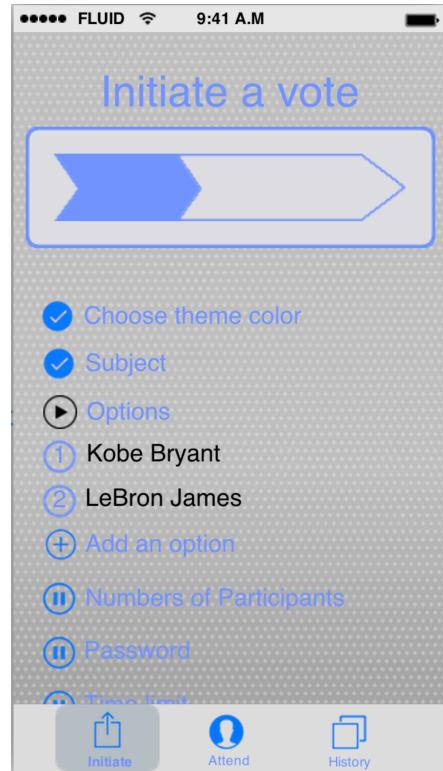
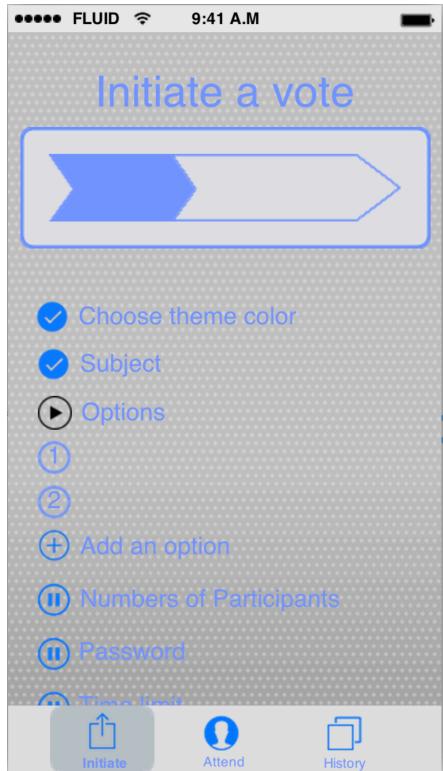
When you tap on the “Initiate” button on the left-bottom corner, you are about to initiate a vote in this application. The first step is setting the theme color of your vote. By the way, the color of a vote is a nice way to differentiate one from another. You could tap on any color circle to choose the color of your vote. And the big arrow in the blue-edge box will be filled with the color your chose step by step when you are finishing the vote initiation.



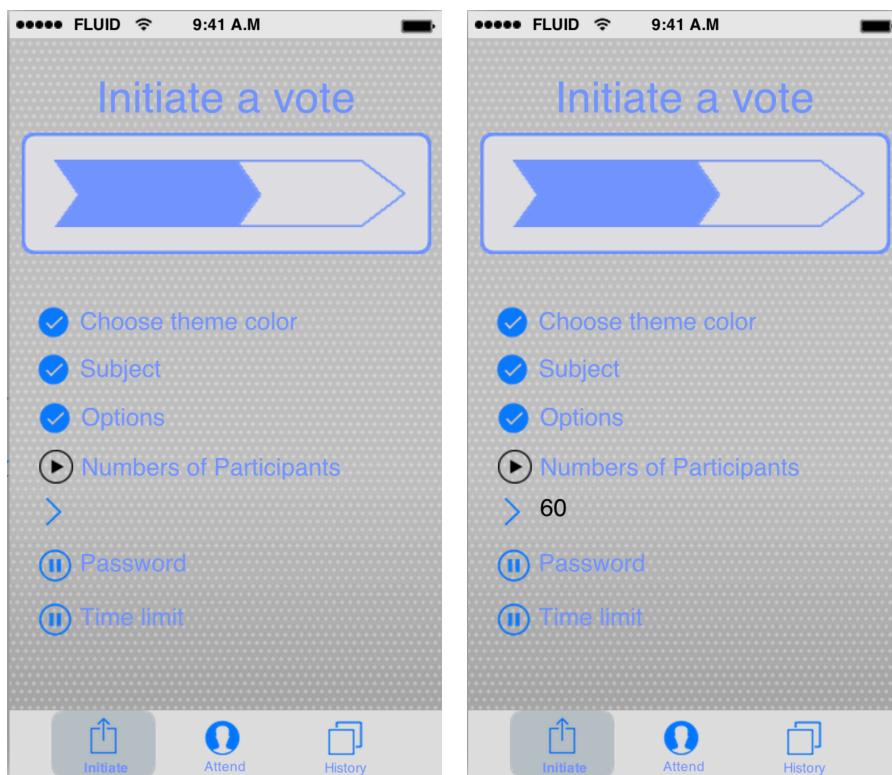
The second step is to give the subject of this vote event.



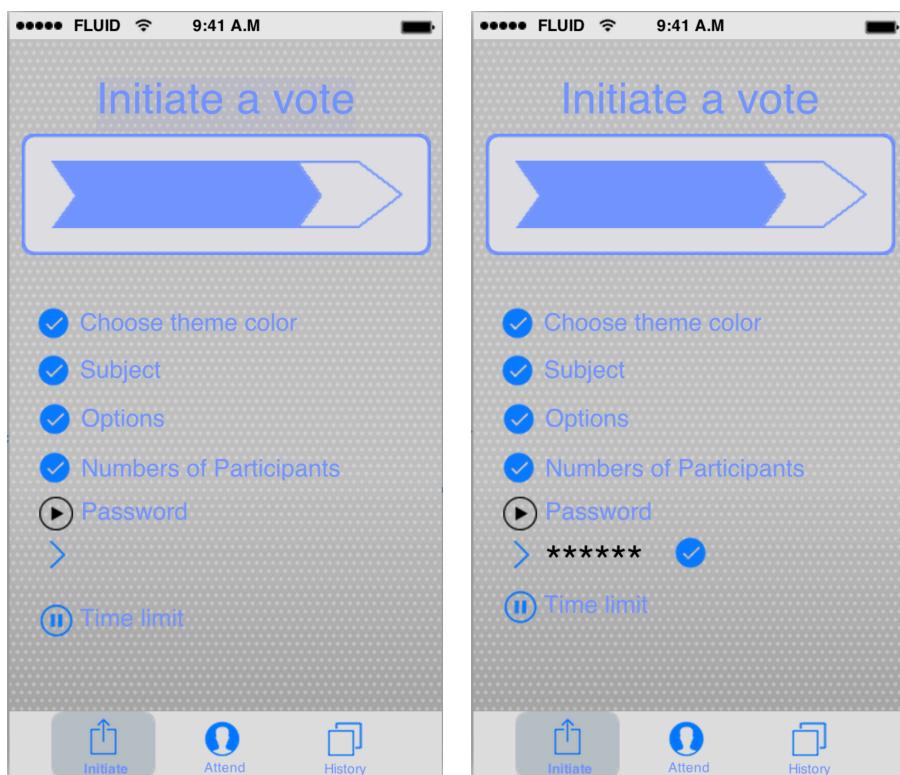
The third step is to offer the options of this vote event. The default number of options in one vote is two. You could add more option by tap on the “plus” button.



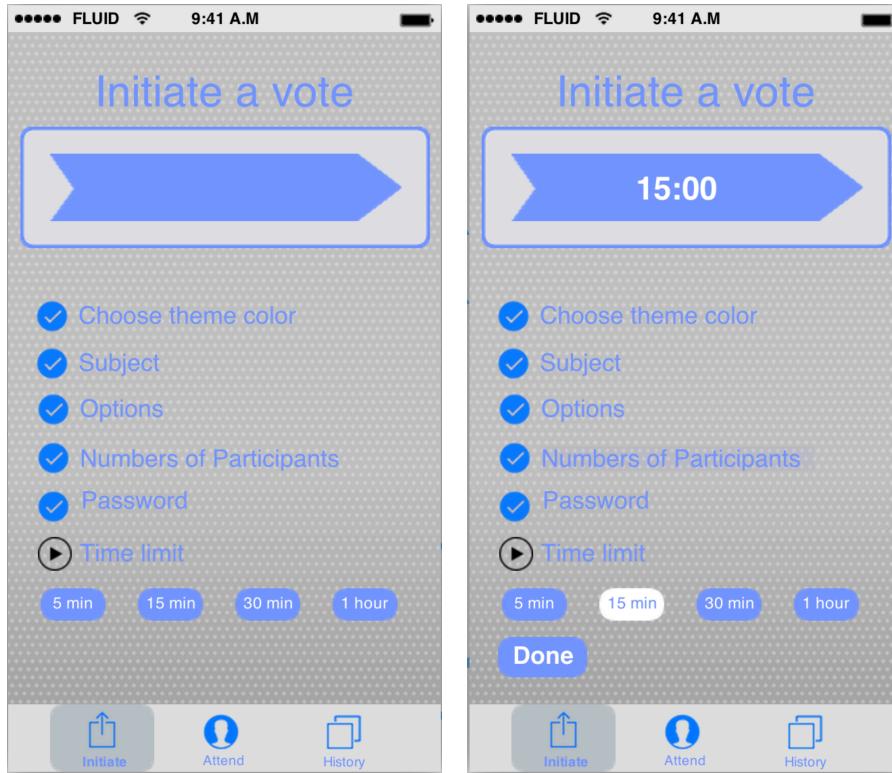
The fourth step is to set the numbers of all the participants.



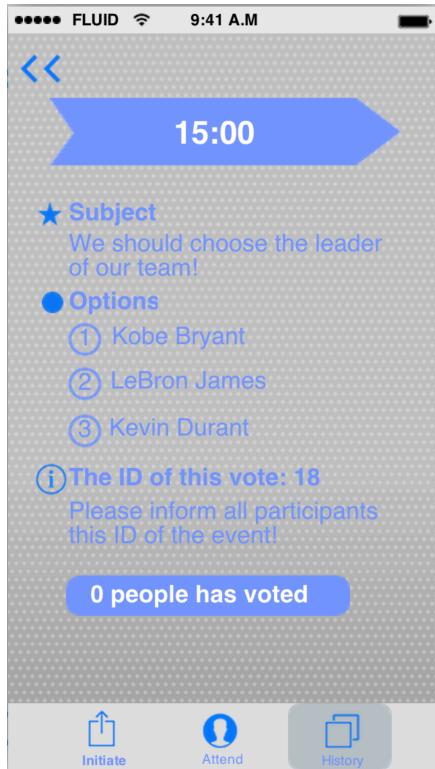
The fifth step is to set the password of this vote event so that only qualified people could attend in this vote. Because it is a simple password within 6 numbers, we won't ask you to enter the password twice.



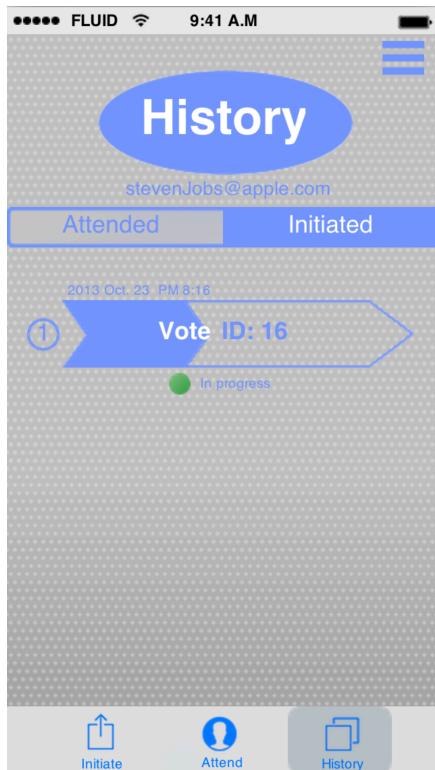
The last step is to set the time limit of this vote event. Instead of letting people entering a specific time period, we give 4 options of time period, which almost include most of the needs of time limit. We think it is a good way to improve the user experience.



After all the steps, your vote will be successfully created with your current location automatically. And the id of this vote event will be generated automatically and you can see it just after you finish initiating a vote. And you should tell every participant this id so that they can find this vote event much more easily. And in the detailed page of this vote, you could also see the time limit and how many people have voted in this vote event.



And you could always enter into this detailed page by search it on the “history” part. It would lie in the “initiated” part of the history.



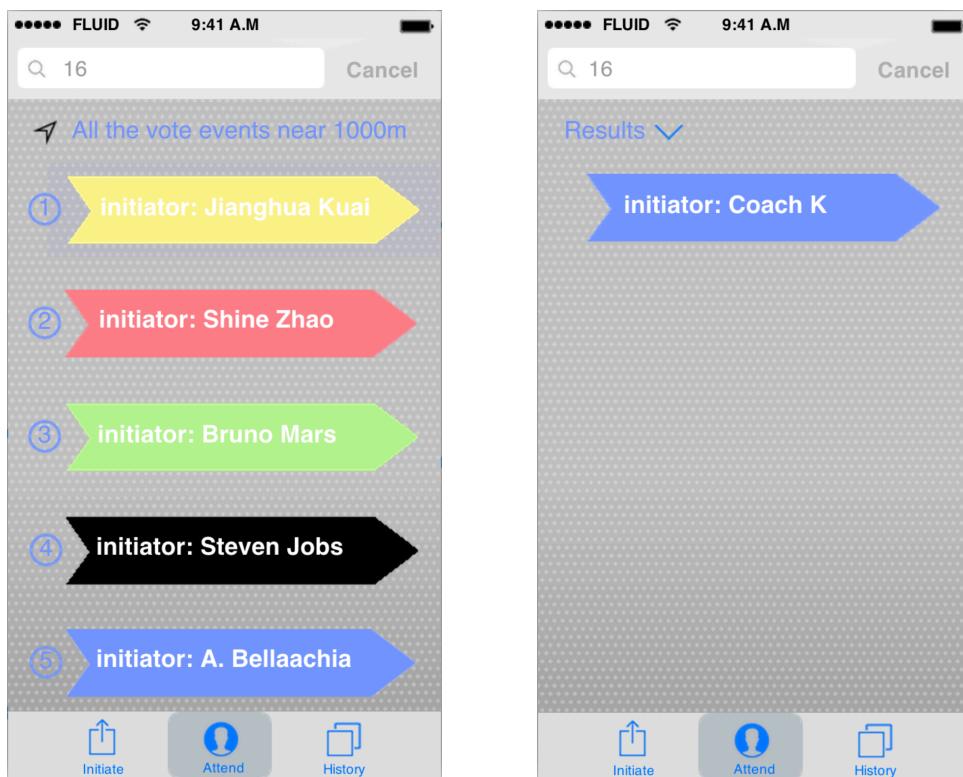
5.3.4 Interaction of attending a vote

This part is including the vote searching and vote operation.

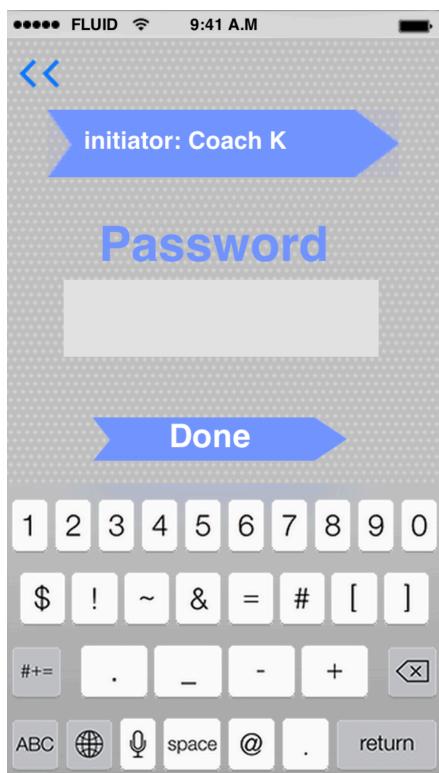
When you tap on the “attend” button on the middle of the toolbar, we would automatically list all the vote events on the screen, which are initiated within the scope of 1000 meters about your current position. You can use your finger scroll up or down on the screen to search the vote you should attend.



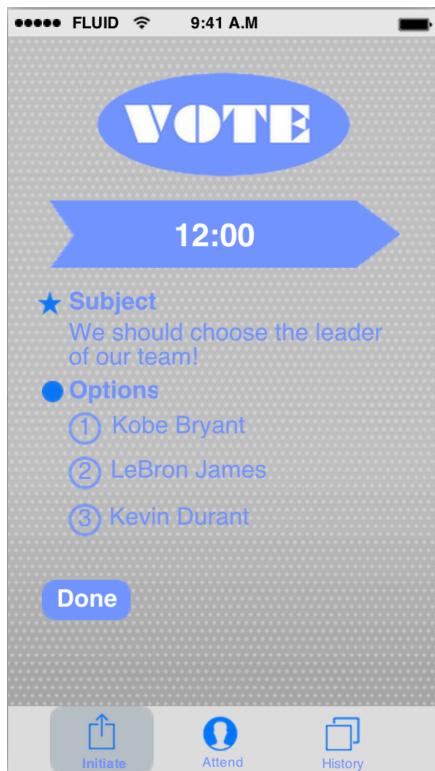
Of course, you could search the vote by specific vote id in the search bar on the top of the screen.



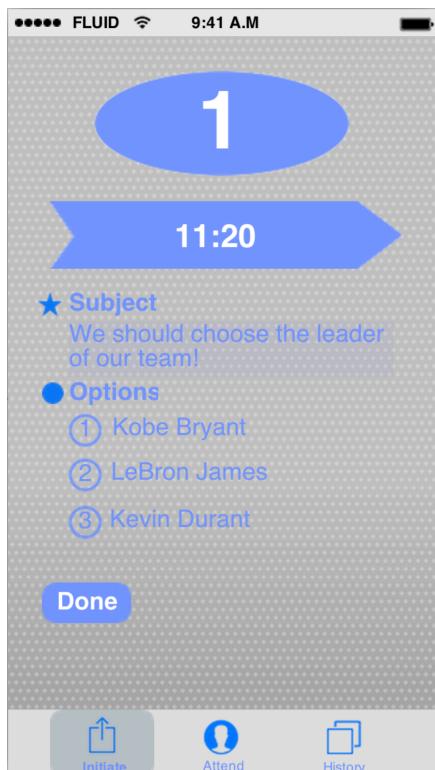
You could just tap on the specific arrow, which represents the vote you want to attend. And password would be needed before you start your vote. The initiator should tell you the password so that you can successfully enter in the vote.



Getting through the password verification, you could begin to vote by taping the option you want to choose.



Say, if you tap on the first option, the option number of your choice would appear in the blue oval area on the top of the screen.

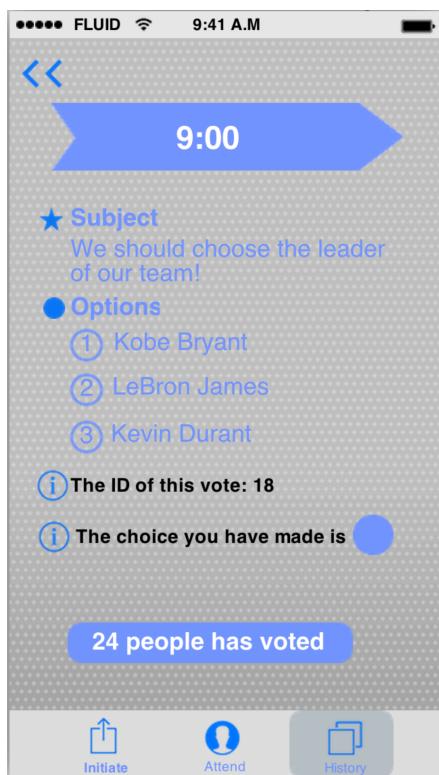


5.3.5 Interaction of Vote History

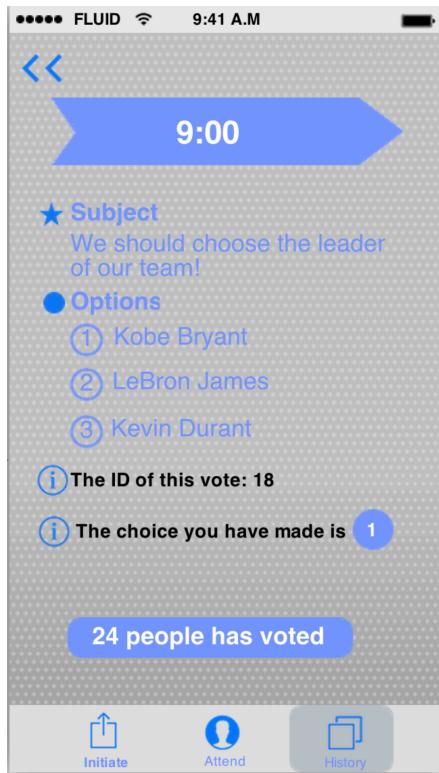
If you make your final decision and tap on “done” button, you vote would be finished and this vote event would appear on the “attended” part of the “History” in the application. Of course, you could scroll up and down to see other vote events you attended before. And the percentage of the refilling blue color in the arrow is actually a progress bar, which means the current state of this vote event. The date and time information on the top-left of every arrow tells about the initiating time of every vote event.



You could tap the vote event into the detailed page of this vote before it is finished. You could see the information as follows: subject, every option, the id of this vote, the choice you have made and how many people have attended until now.



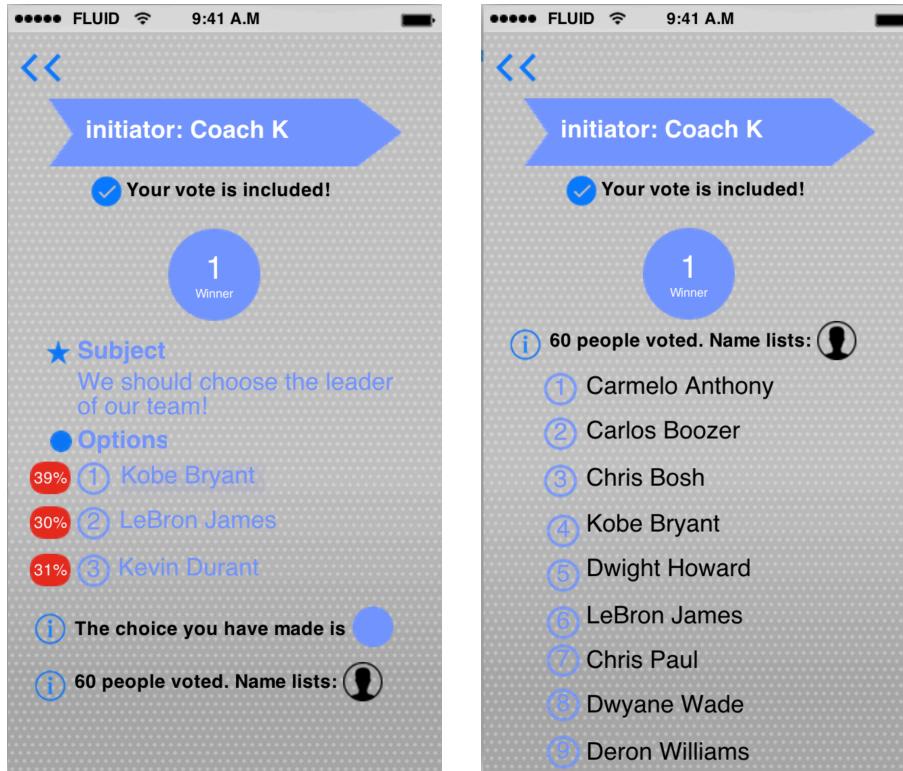
In consideration of the personal privacy, the choice you just made would not directly appear on the screen. You should tap on the little blue circle to see that.



When one specific vote event finally end, the area of the whole arrow would be filled with one particular color and the tips below the arrow would convert from “in progress” to “Finished”. It means you could know the final result of vote.

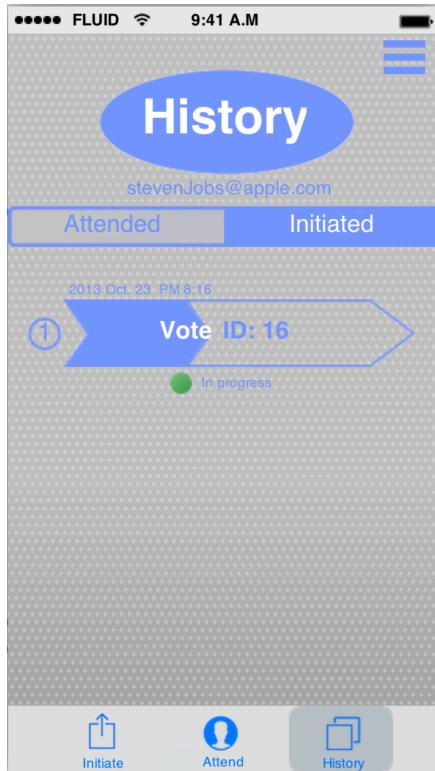


Tap on the blue arrow, you could see the detailed information about this vote. The most conspicuous information showed in this page is obviously the result of this vote, which is the winning option. In this case, the first option wins so that the number 1 is being put in the blue circle. And there is some other important information that every participant might be interested. It would tell you that if your vote is included in this final result, because the initiator could accidentally delete the valid vote while he or she is trying to eliminate some illegal votes. And you also could see what is the percentage of all the participants of every option. In addition, you could see your choice again by tapping on the blue circle and know the name of every participant of this vote event by tapping on the contacts icon.

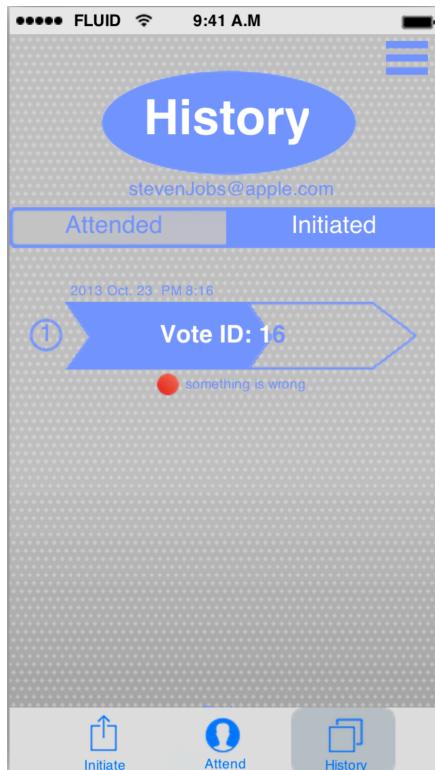


5.3.6 Interaction of exception handling

If you have initiated a vote, you could see it in the “initiated” part of the “History” in the application. The vote ID would be showed in the middle of the arrow, which represents this vote event. Thus, the initiator could easily find the vote ID if he or she happened to forget it. If everything is going on fine in one particular vote event, there will be a little green circle under the arrow with the phrase “In progress”.

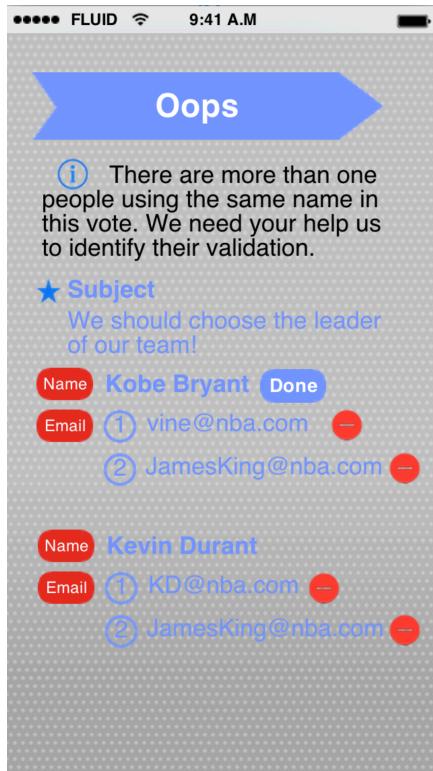


If there were more than one people using the same name in one vote event, there would be a warning for the initiator, which is a red circle under the arrow with the phrase “Something is wrong”.

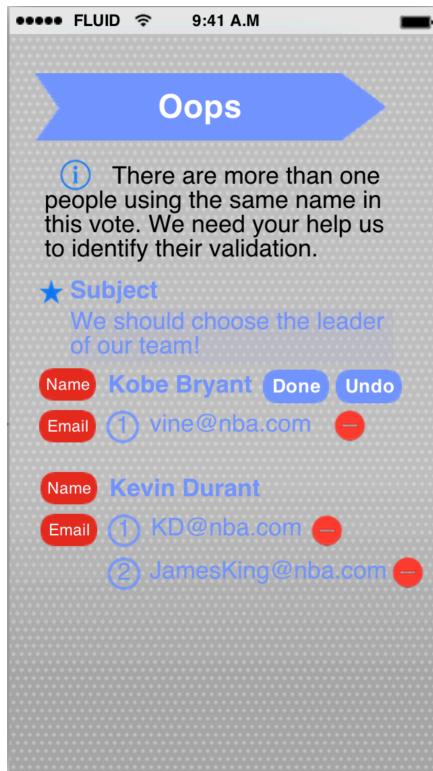


The initiator could tap on the arrow to deal with this unexpected situation. In the exception-handling page, the user would see the names that have more than one Email addresses binding with them. You

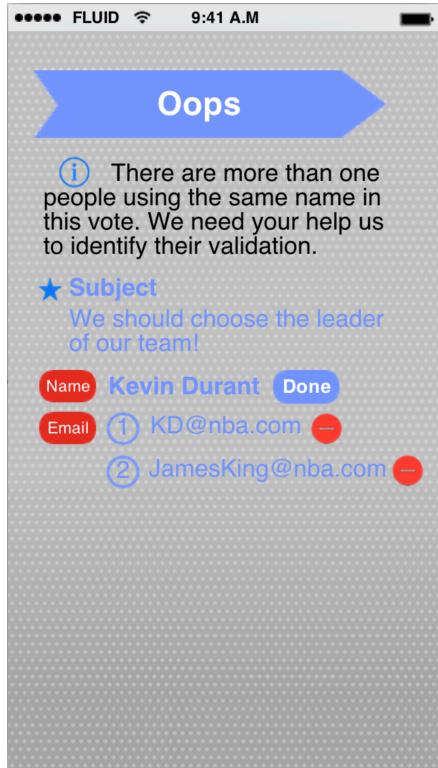
could tap on the deleting icon to delete the invalid Email address after you ask the particular person who has that name of his or her real Email address.



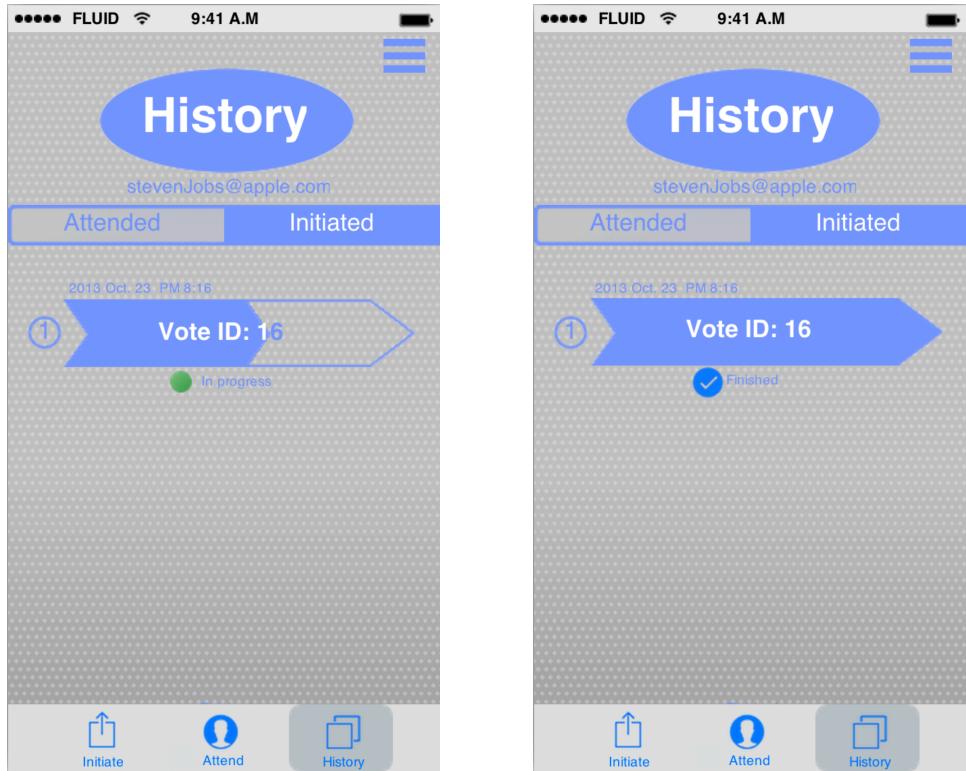
Say, the initiator deletes the second Email address, which is also binding with the name Kobe Bryant. Of course, if the initiator accidentally deletes the right one, he or she could also undo his or her action.



After deleting the invalid Email address, the initiator could tap on done if he or she is pretty sure that he or she makes the right decision. After the “done” button is pressed, this specific exception binding with this name would disappear on the screen. And the next exception would slide up to the first exception on the screen.



After you successfully deal with all the exceptions, the page would automatically direct to the “initiated” part of the “History” of the application. And you would see the green circle again until the vote is finished or another exception appears.



6.Appendices

DatabaseDesign.mwb (Works with MySQLWorkBench)

InterfaceDesign.xml (Works with Microsoft EXCEL)

Class Diagram (Works with ArgoUML)