

# **Vote Application based on iOS**

Advanced Software Paradigms

Professor: Abdelghani Bellaachia

Author: Jianghua Kuai, Shuai Zhao

**Oct 11, 2013**

## **Included Document List**

- 1. Project Abstract**
- 2. Requirement Document**
- 3. Detail Design Document**
- 4. Workload and Milestone Sheet**
- 5. Test case Document**
- 6. Appendix**

# Term Project

## Vote Application for iOS

Jianghua Kuai, Shuai Zhao

### Abstract

During the past century, most of nations on this planet has entered the new era of democracy. As a result, people in these countries have rights to choose their preferred government by voting for themselves. Nowadays the spirit of the democracy seeps from everyone's pore. People desire equality no matter where they are, what they do. People want to vote for what plan should be chosen in their business group. People also hope to vote for what communal facilities they should own in their community. People with a big family even need to vote for where to go in the vacation. If we take traditional ways to solve these problems, it will take plenty of time and effort to collect every paper vote and to do statistic for a result. And we cannot make sure it's totally anonymous because of the unique handwriting. Now we can simplify all these intricate procedures by using this vote application.

This application can be used in many occasions such as meeting, classroom, party and so on so forth. Anyone could start a vote in this application. Then the vote event will be created with a password with which people are eligible to vote. Every people around this particular site can find this vote event on his or her application. If it doesn't work, this vote event can also be searched by its unique ID. Every participant's name will be showed in the application. And of course, the detailed choices of everyone will not be showed. The result will be pushed to everyone's application simultaneously after the last voter made his or her decision.

The whole project will use C/S architecture, a user can start a vote, then the client app will send the location(if available) and vote information to the server, users also can set a password for this vote. Server will open a new vote and send back a unique ID to client. After that other app client can search this vote by ID or the location infomation to join this vote, after the vote finish, the server will push the vote result to all clients which joined the vote by notification or can check the vote result by ID.

We will use iOS SDK to implement client-side application, which will involve the MapKit, CoreLocation, UIKit, and other general Cocoa framework. And we'll use Google Analytics to trace the application's quality and get user's preferences.

Server side will use *Plankton Server* framework, which designs for the iOS platform and integrates useful features for iOS platform such as notification push, request package encapsulation & encrypt and so on. This framework will works on Linux and uses MySQL as database. The information in network package will encode as JSON format.

As a result, we plan to implement these features for client as below. We want to keep our app's function as pure as possible, so we will only focus on vote relate feature.

- Start a vote
  - 1. Vote can be public or private; a public vote can start as anonymous.
  - 2. Vote can be permanent or timed.
- Search a vote
  - 1. Search vote can use current location and vote's unique ID.
- Join a vote
  - 1. Join a vote by password (private) or anonymous vote for a public vote around you.
- Check a vote's result.
  - 1. If user accepts a notification, they will be notifying after the vote finish (timed).
  - 2. If the vote is permanent type the user can check the result after vote.

# **Vote Application based on iOS**

## **Requirement Document**

Current Version: v1.0

Author: Jianghua Kuai, Shuai Zhao

Oct 11, 2013

## Maintenance History

Author	Date	Content	Version
Jianghua Kuai	Oct 11, 2013	Build the Document	V0.1
Shuai Zhao	Oct 12, 2013		V0.2
Jianghua Kuai	Oct 13, 2013	Add use case and finish the doc	V1.0
Shuai Zhao			

## **Table of Contents**

<b>1.</b>	<b>PURPOSE AND SCOPE .....</b>	<b>4</b>
1.1	PURPOSE.....	4
1.2	SCOPE .....	4
<b>2.</b>	<b>MARKET ASSESSMENT AND TARGET DEMOGRAPHICS .....</b>	<b>4</b>
<b>3.</b>	<b>PRODUCT OVERVIEW AND USE CASES.....</b>	<b>5</b>
<b>4.</b>	<b>REQUIREMENTS.....</b>	<b>5</b>
4.1	FUNCTIONAL REQUIREMENT.....	5
4.1.1	USER MANAGEMENT .....	5
4.1.2	VOTE INITIATION.....	6
4.1.3	VOTE SEARCHING.....	6
4.1.4	VOTE OPERATION.....	6
4.1.5	EXCEPTION HANDLING.....	7
4.1.6	NOTIFICATION PUSH.....	7
4.1.7	VIEW VOTE HISTORY.....	7
4.2	USABILITY REQUIREMENT.....	8
4.3	TECHNICAL REQUIREMENT.....	8
4.4	ENVIRONMENTAL REQUIREMENT.....	8
4.5	PERFORMANCE REQUIREMENT.....	8
<b>5.</b>	<b>PRIORITY AND RISKS .....</b>	<b>8</b>
<b>6.</b>	<b>APPENDIX .....</b>	<b>9</b>

## **1. Purpose and scope**

### **1.1 Purpose**

This application is designed to replace the regular time-consuming paper-based vote. People can efficiently initiate or participate in a particular vote in this application. And the result will be analyzed and be pushed back to every participant's iPhone instantly after everyone finishes his or her votes. Meanwhile, it evidently enhances the trustworthiness of the vote, since the possibilities of tellers' interference are gone. Furthermore, the vote will be totally anonymous because the encrypted message transferring from client to serve is more safety than the handwriting that can be differentiated by tellers.

### **1.2 Scope**

The app will release on Appstore without region restriction, and the language will be English only.

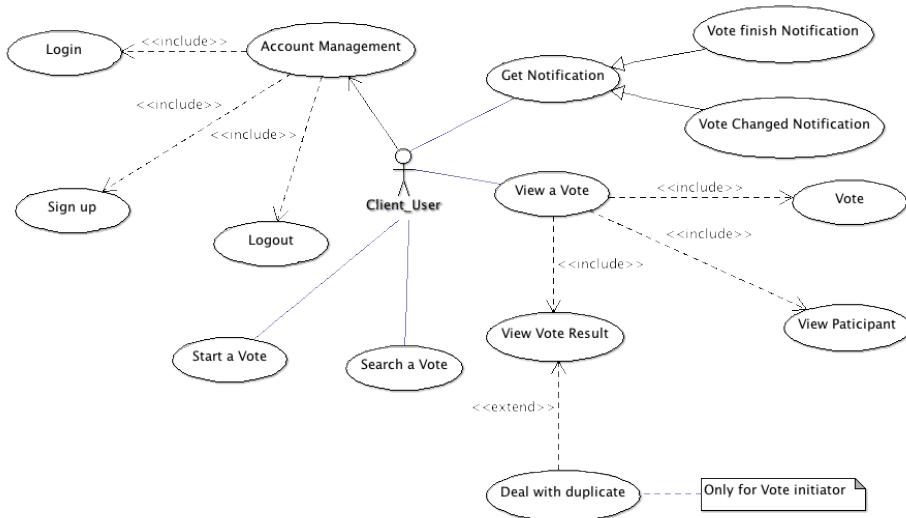
The application will only implement on the iOS platform on iPhone device and the server will run on CentOS 6.4 64bit. The prototype may not include the UI design.

For the UI design in release version, since the iOS7 is changed to the flat design style, the design of user interface of this application will also be flat to accommodate with the iOS7.

## **2. Market assessment and target demographics**

Although there are several applications existing in IOS, but the interesting fact is people tend to choose paper-based vote when they have to vote. A lot of them are designed for particular events such as US presidential election. And many others are too complicated to use. They are just like creating new social networks. However, we just want to use it to vote instead of joining another social network. Other voting application is offline. That is to say, bunches of people use only one device to complete voting, which is inconvenient and inefficient.

The client use case is as below.



### 3. Product overview and use cases

This application can be used in many occasions such as meeting, classroom, and party and so on so forth. Anyone could start a vote in this application. Then the vote event will be created with a password with which people are eligible to vote. Every people around this particular site can find this vote event on his or her application. If it doesn't work, its unique ID can also search this vote event. Every participant's name will be showed in the application. And of course, the detailed choices of everyone will not be showed. The result will be pushed to everyone's application simultaneously after the last voter made his or her decision.

## 4. Requirements

### 4.1 Functional requirement

#### 4.1.1 User Management

Requirement ID: F00001

Description: Every user should have a legal account to use this application. User's name, Email address and PIN are required when anyone sign up for an account. After entering Email authentication code, the user could successfully log in. If user already has an existing account, he or she could log in this application on anyone's iPhone after last user log out.

#### 4.1.2 Vote initiation

Requirement ID: F00002

Description: Anyone who wants to launch a new vote should follow the tips given by the application. The initiator should give the subject of the vote, every option of the vote, and the total number of the applicants, available time period and the password for everyone who is permitted to participate in this vote. When initiator submits this vote, the location will be recorded automatically.

#### 4.1.3 Vote searching

Requirement ID: F00003

Description: Every neighborhood vote event, based on the place where he or she is, within valid time will be shown in the searching result page in the application. And according to additional information such as the vote event ID, the initiator's name and the subject, the user could choose the particular one that he or she should attend. If the location-based search is failed, users could also choose to search the vote event by vote event ID, which will be created when the vote is initiated.

#### 4.1.4 Vote operation

Requirement ID: F00004

Description: Entering the correct password, users could vote in the application within the valid time period.

#### **4.1.5 Vote Exception handling**

Requirement ID: F00005

Description: There are some unexpected occasions we should take into consideration. If there are two or more participants with the same name in one vote event, these conflicts will be settled by the initiator. He should figure out if there are more than one people actually having the same name. If it is the case, he should valid all these votes. But if not, he also should differentiate the real one by checking his or her address. Then the votes with faked same name will be eliminated from this vote event.

#### **4.1.6 Notification push**

Requirement ID: F00006

Description: After every participant finish their vote or the valid vote time is over, if the exception happens, there will be a notification pushing to the initiator's iPhone and let him or her handle this problem. After that the end notification will be pushed to every participant's iPhone to inform that the vote's result could be checked in the application.

#### **4.1.7 View Vote History**

Requirement ID: F00007

Description: User could view the vote history list, which shows that the vote user is involved. User could check the result of vote that has been finished. Every vote result includes following message: 1.The verification that your vote is included in the final result in case the initiator make a mistake when handles the exception. 2. The total number of supporters of each option of this vote. 3.The entire number of all participants and everyone's name.

#### 4.2 Usability requirements

The application can be used in many occasions mostly when people are in the same area. Some examples are given as follows: students can use it to elect their class leader; people can choice their desired plan during business; family members can make the decision that where to go during the vacation.

#### 4.3 Technical requirements

The message transferring from device to the server will definitely be encrypted. And the exception handling is also taken into consideration.

The application will use *Google Analytics* to trace the exception of application, and get the active users number.

The server will use Plankton Server as a framework and will modify some features to meet the requirements.

#### 4.4 Environmental requirements

This application will be developed for iPhone4, iPhone4S, iPhone5, and iPhone5S. And the OS we will support is iOS5.0+.

The IDE for client app will be *Xcode5*, *eclipse CDT* for server app.

The database for server will be *MySQL*.

#### 4.5 Performance requirements

The user will not wait for more than 10 seconds with any operation.

The client should not use more than 256MB memory at peak.

### 5. Priority and Risks

Requirement ID	Priority	Risk
F00001	High	High

F00002	High	Low
F00003	Medium	Low
F00004	Medium	Low
F00005	Low	Medium
F00006	Low	High
F00007	Low	Medium

## 6. Appendix

# **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**

<b>1.INTRODUCTION .....</b>	<b>4</b>
1.1PURPOSE .....	4
1.2SCOPE .....	4
1.3REFERENCE MATERIAL.....	4
<b>2.SYSTEM OVERVIEW .....</b>	<b>4</b>
<b>3.SYSTEM ARCHITECTURE .....</b>	<b>5</b>
3.1 ARCHITECTURE DESIGN.....	5
3.2 DECOMPOSITION DESCRIPTION.....	6
<b>4.DATA DESIGN .....</b>	<b>7</b>
<b>5.HUMAN INTERFACE DESIGN.....</b>	<b>8</b>
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> .....	12
<i>5.3.2 Interaction of vote initiation .....</i>	19
<i>5.3.4 Interaction of attending a vote.....</i>	25
<i>5.3.5 Interaction of Vote History.....</i>	28
<i>5.3.6 Interaction of exception handling .....</i>	31
<b>6.APPENDICES .....</b>	<b>35</b>

# **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

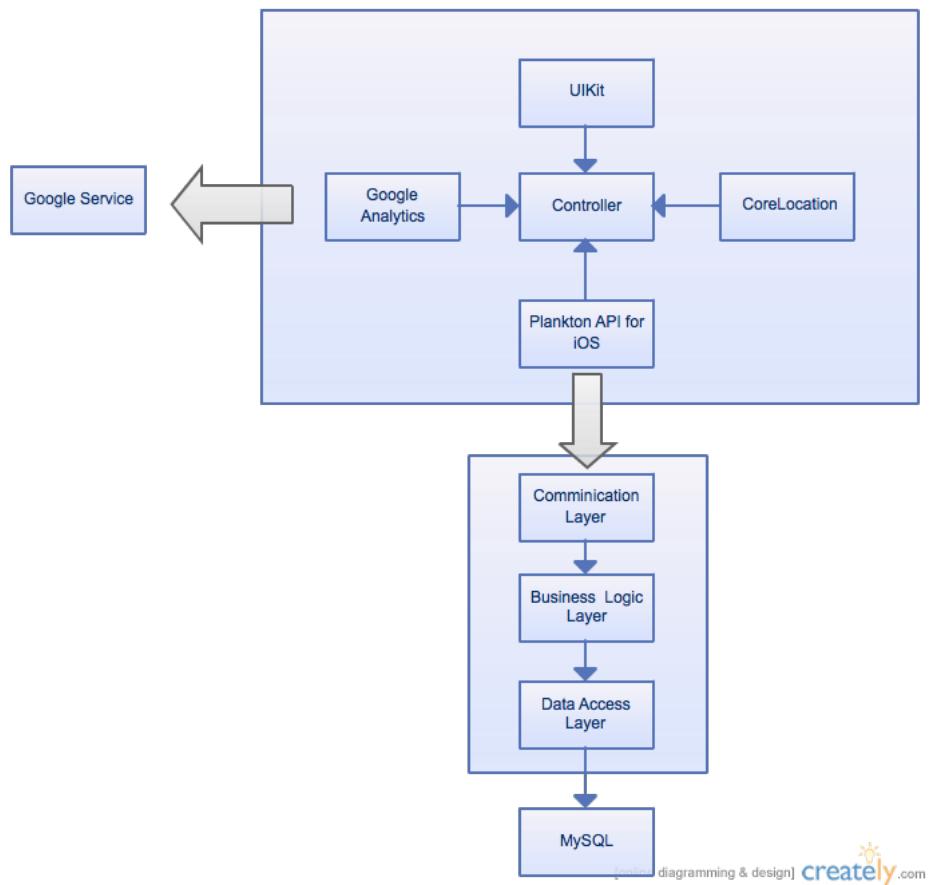
## 3.1 Architecture Design

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.

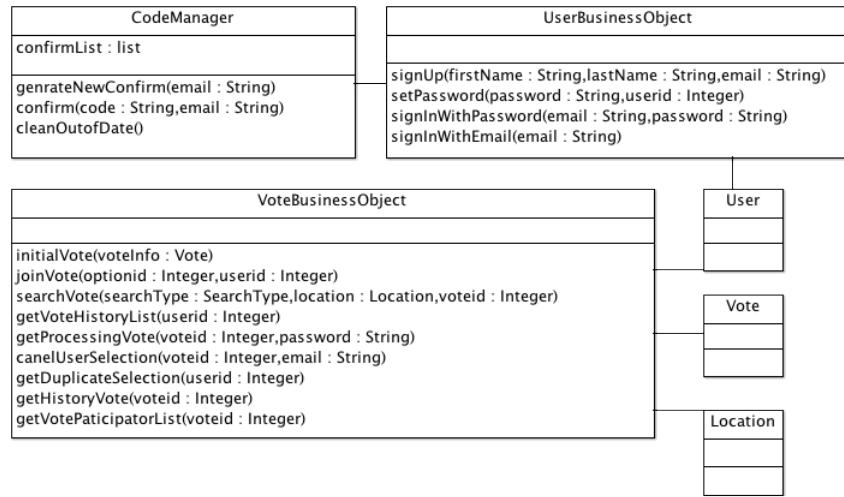
The diagram below shows the architecture of the project from a general view.



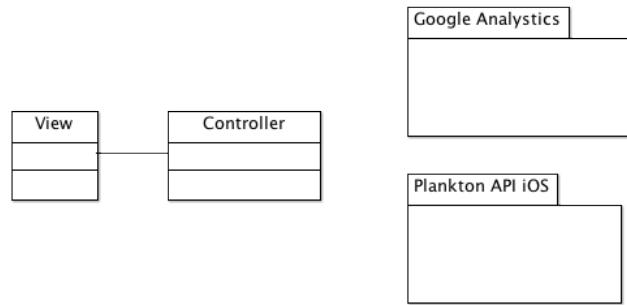
## 3.2 Decomposition Description

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.

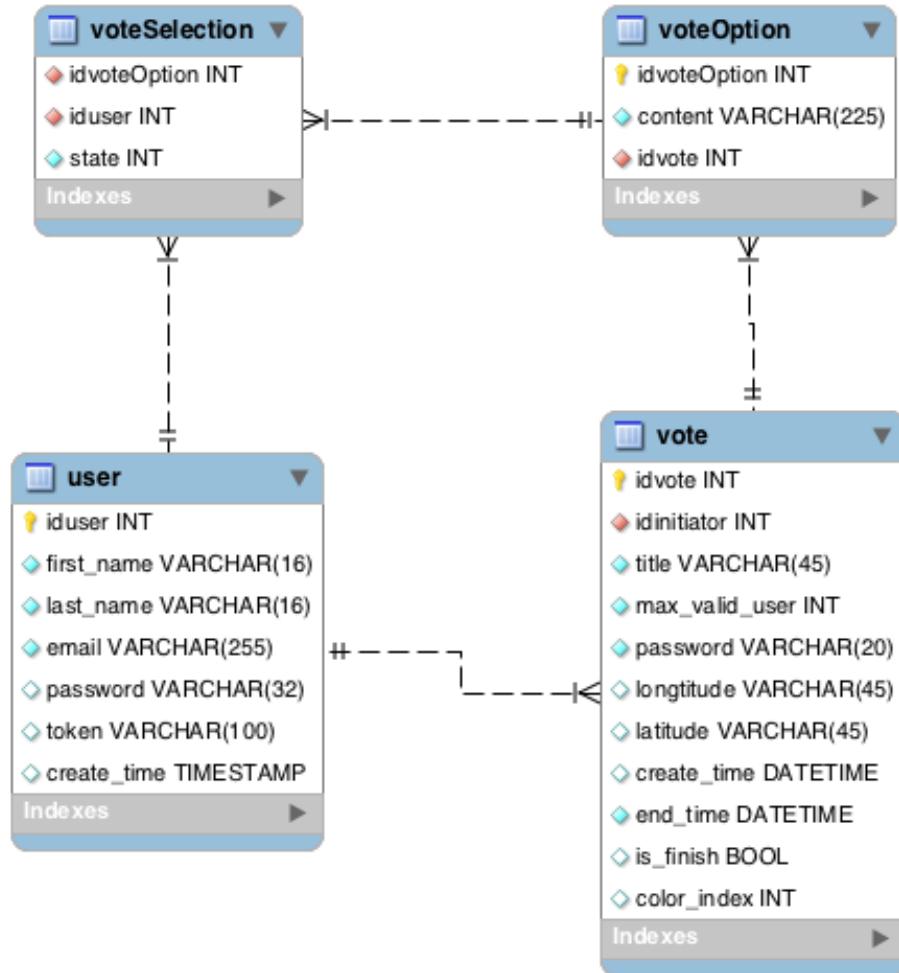


For client part, every controller will report the exception and user preference (if allowed) by using Google Analytics, and every controller will send request from Plankton API to server, so every single controller's class diagram should looks like the image below. The detail relationship of each controller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



## 4.Data Design

We use EER to describe our database design and structure.



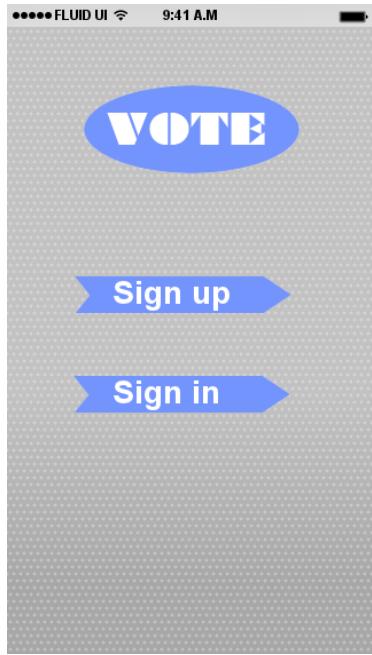
## 5.Human Interface Design

### 5.1 Overview of User Interface

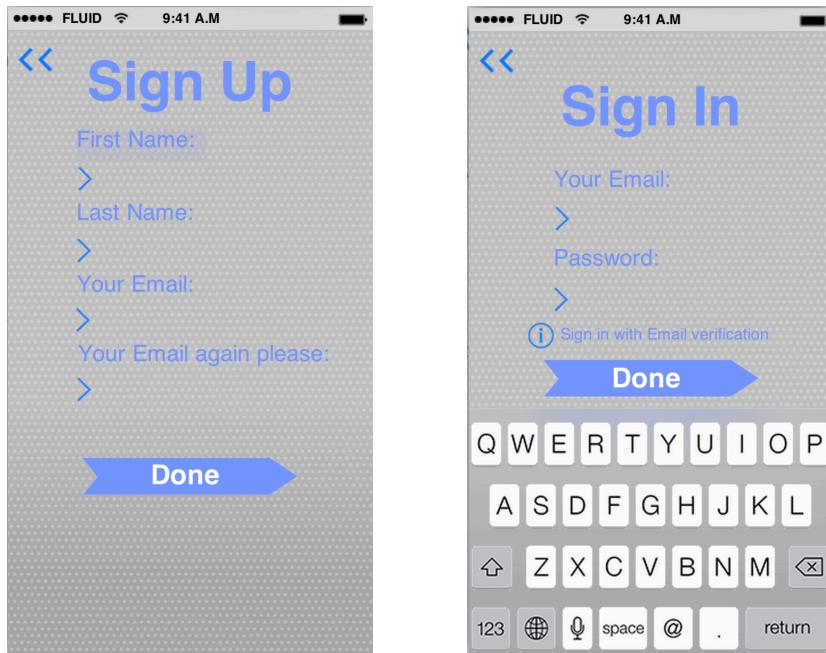
The whole human interface design is flat-design oriented. The main objective is giving users a clear and simple way to enjoy this application. We design this application on the web [fluidui.com](http://fluidui.com), which is a good place to design mobile application.

## 5.2 Main Screen Images

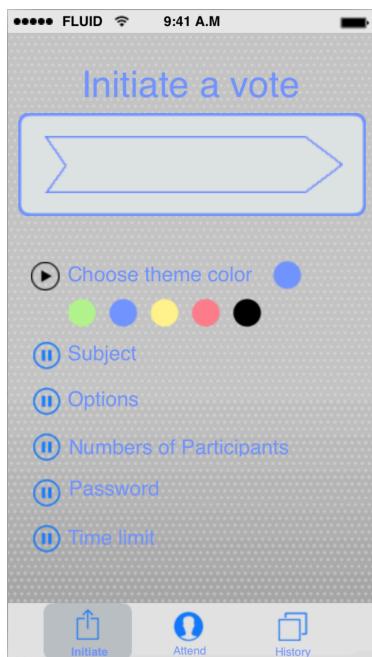
The launch screen image of this application:



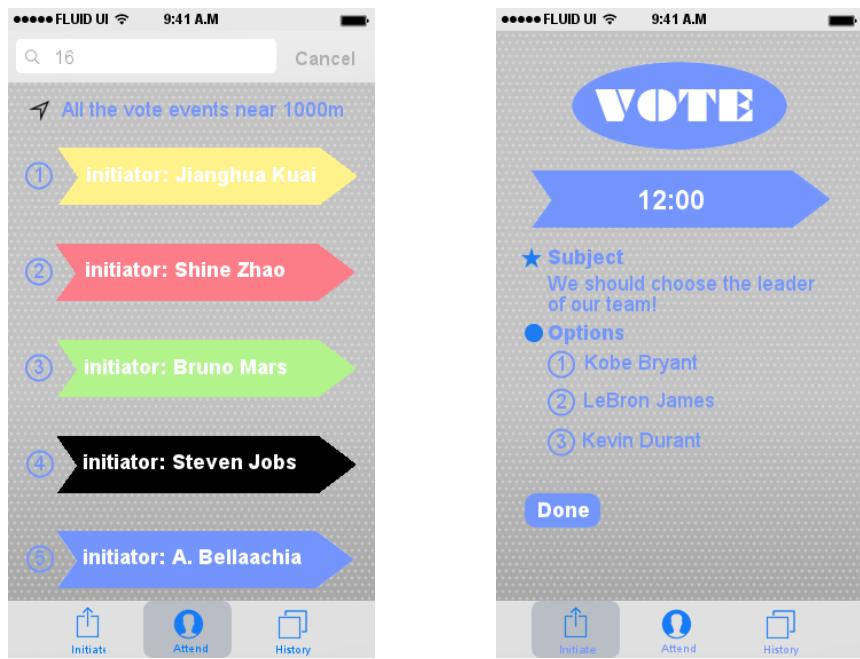
The sign-up and sign-in screen images:



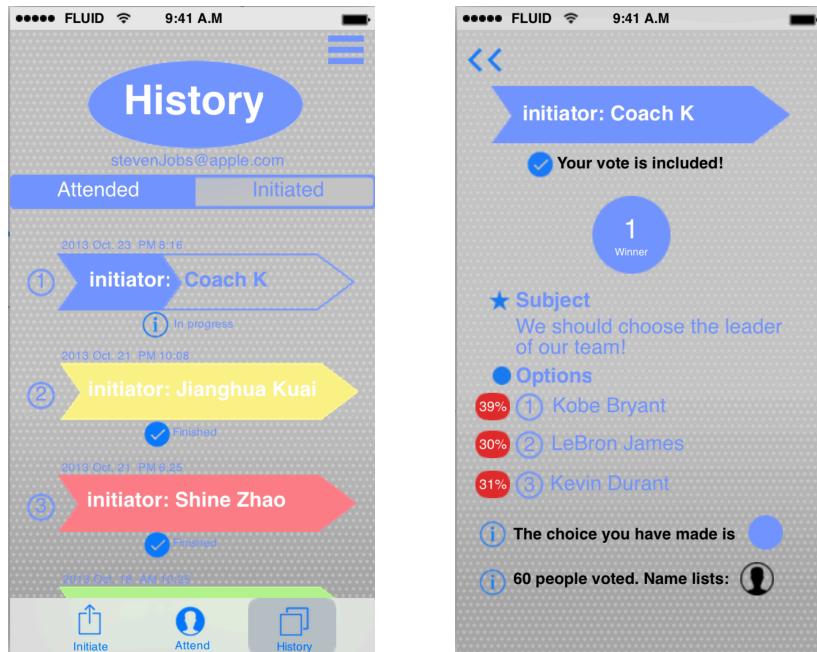
The main images of initiating a vote



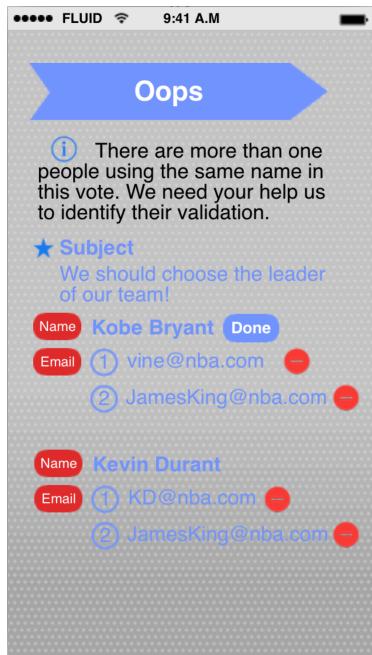
The main screen images of attending a vote event



The main screen images of History:



The main image of vote exception handling:



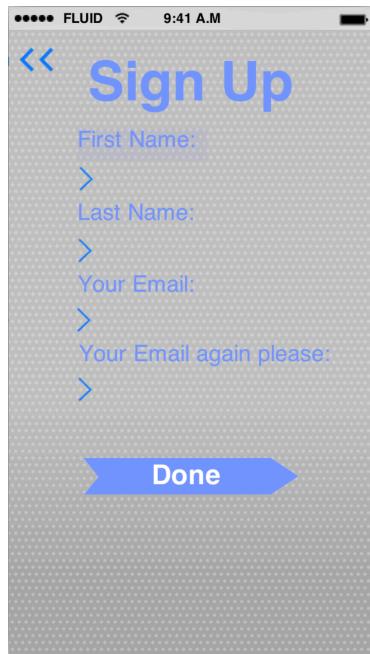
## 5.3 Screen Object and Actions

### 5.3.1 Interaction of User Management

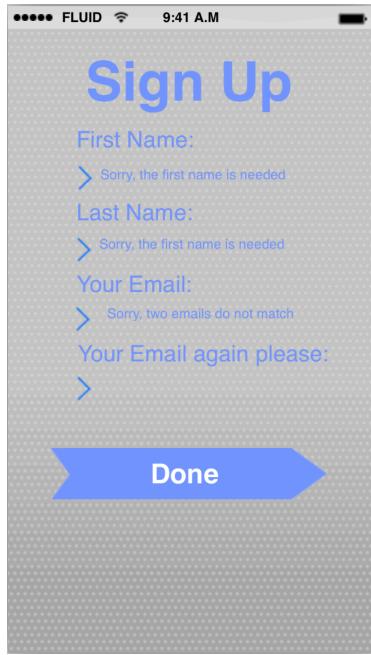
When you first launch this application in a particular iPhone, you would be asked to sign up or sign in with existing account.



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



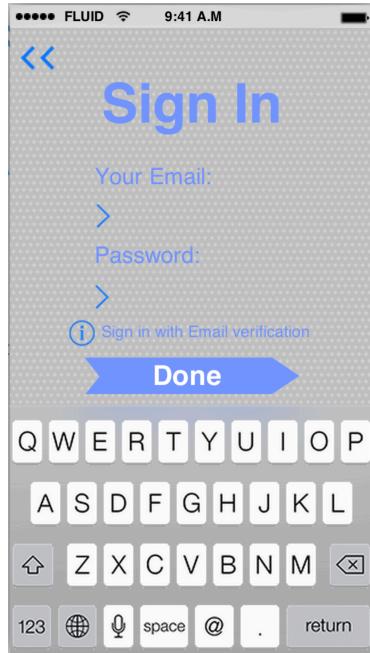
If any of these information are not given by the user or the two Email addresses cannot match, this step cannot be finished.



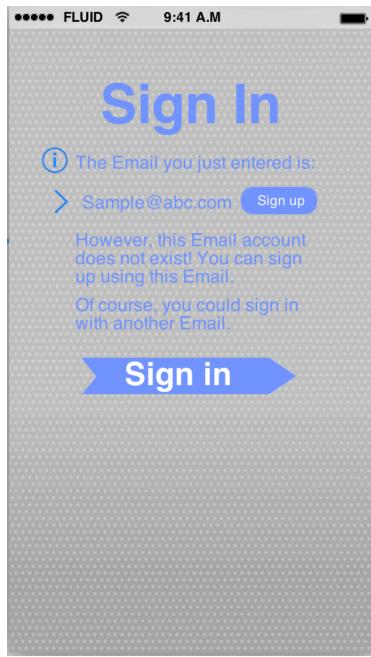
The next step is to validate your Email address, which assure that every participant of a particular vote event is valid. In this step, you can also change your email address if you find out that the Email address you just typed in is incorrect. In addition, if you do not receive your code by our Email 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.



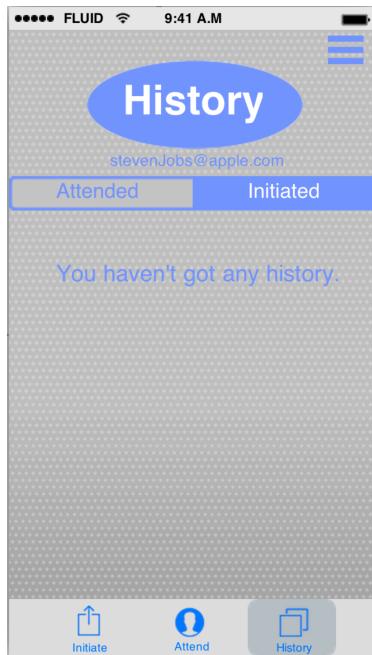
Instead of signing up, you can also sign in with your existing account in a particular iPhone in the first time. You can just type in your Email address and the password. If you accidentally forget your password, you could click on the “Sign in with Email verification”. By doing that, we would send you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



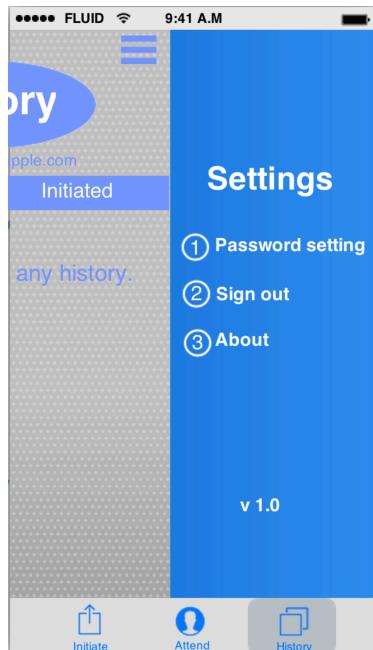
Of course, we take care of some special situations like that someone types in a non-existing account and try to sign in.



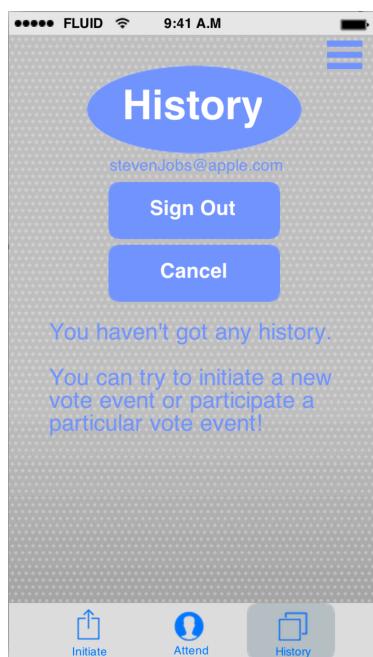
After you sign up or sign in successfully, you would enter into the “history” part of the application. Obviously, it does not have any records before you initiate a vote or attend any votes.



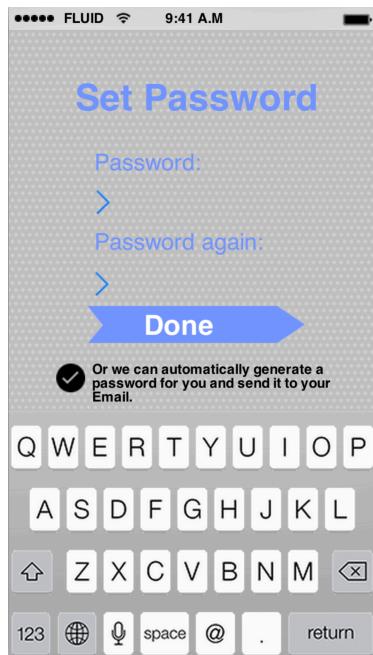
In the “History” part of this application, you could tap the fence-like button, which is in the right-up corner of the screen, to enter the setting menu. As I said, the setting menu would be slide from the right edge of the screen after the fence-like button is tapped. You could set the password of your 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.

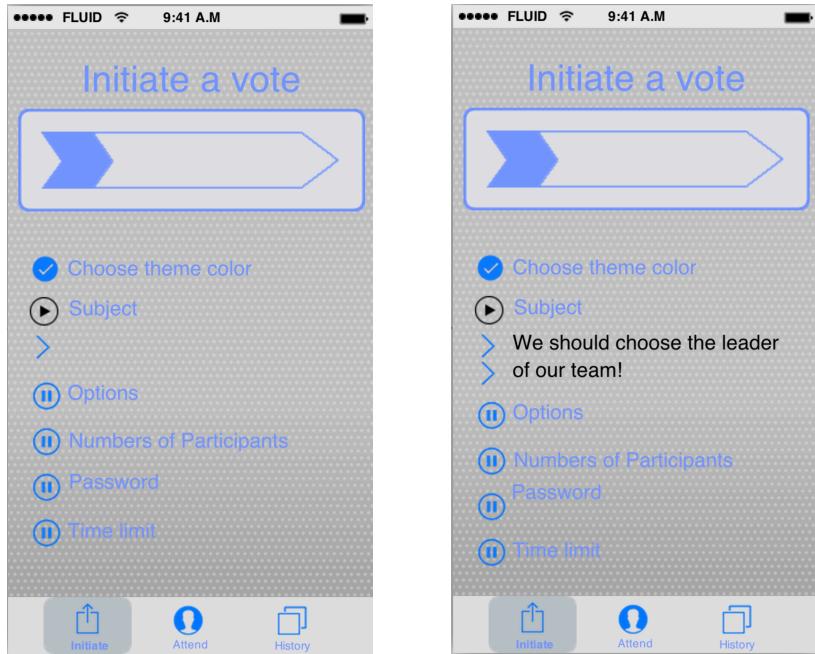


### 5.3.2 Interaction of vote initiation

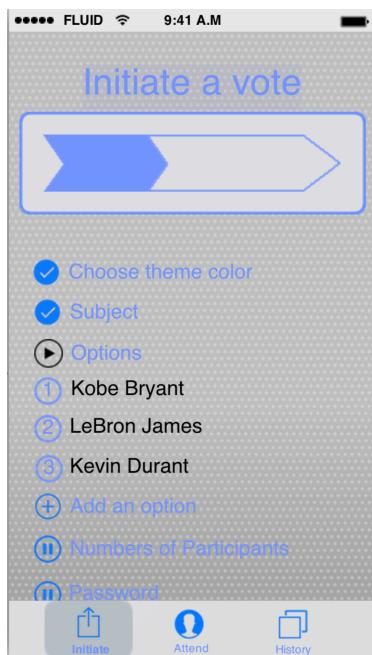
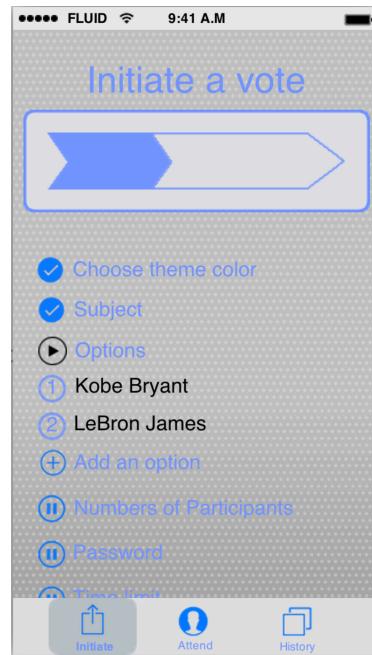
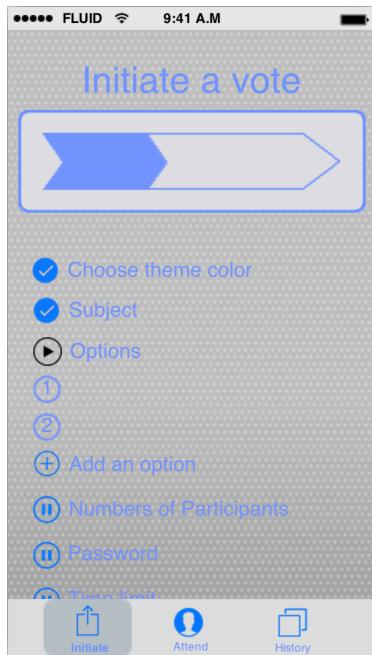
When you tap on the “Initiate” button on the left-bottom corner, you are about to initiate a vote in this application. The first step is setting the theme color of your vote. By the way, the color of a vote is a nice way to differentiate one from another. You could tap on any color circle to choose the 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.



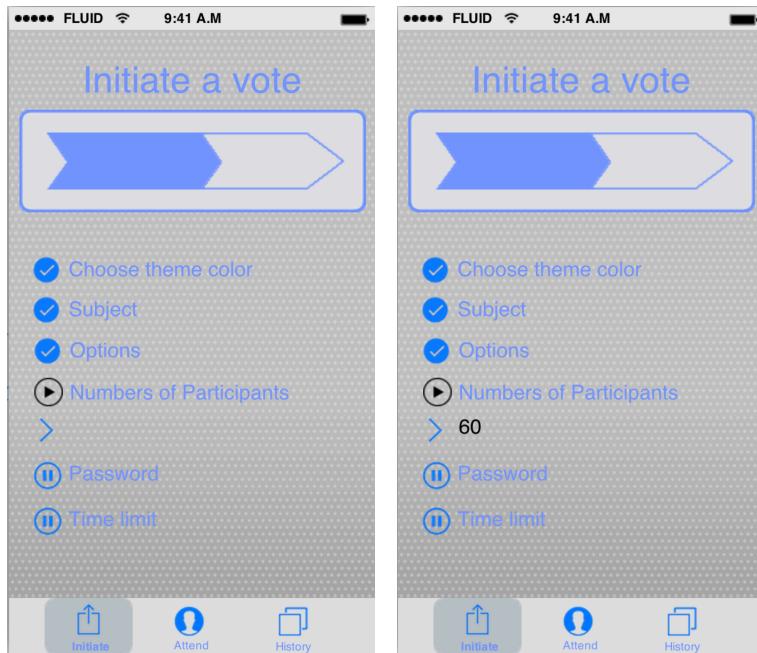
The second step is to give the subject of this vote event.



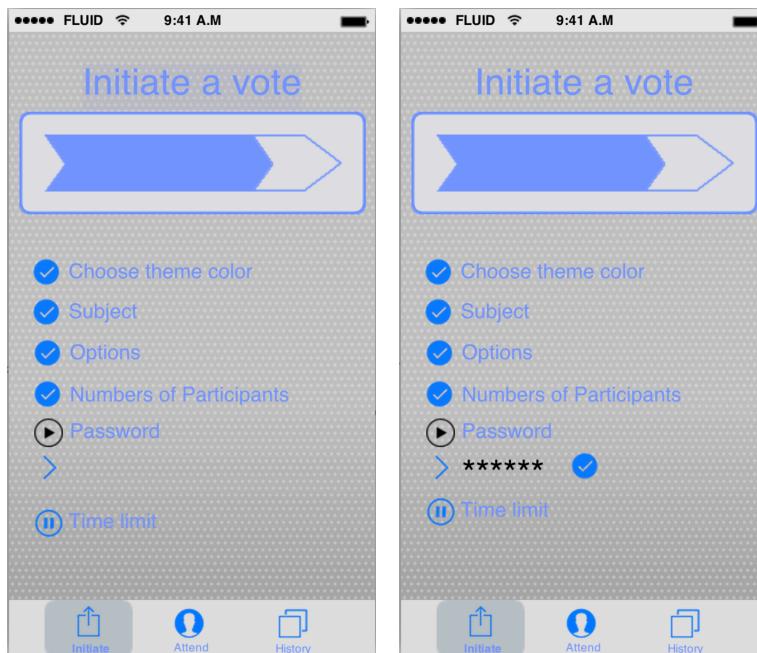
The third step is to offer the options of this vote event. The default number of options in one vote is two. You could add more option by tap on the “plus” button.



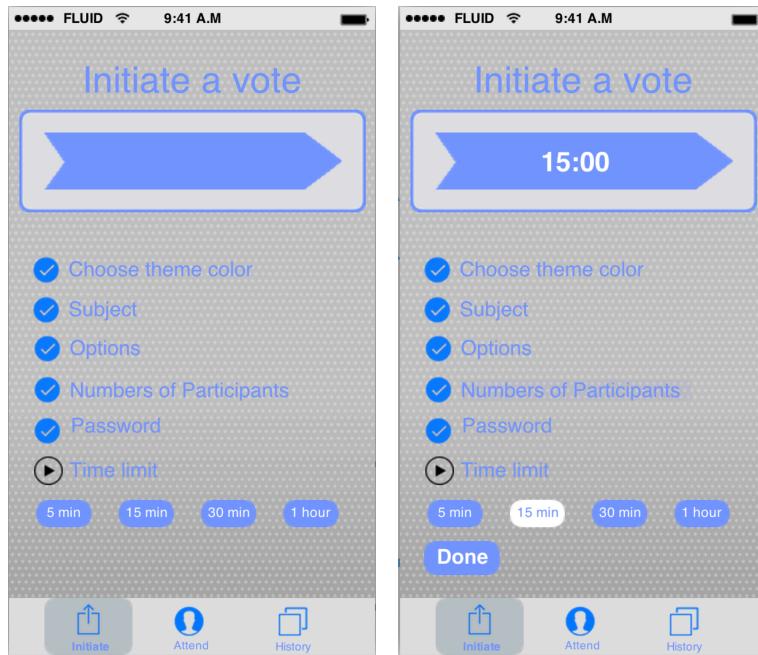
The fourth step is to set the numbers of all the participants.



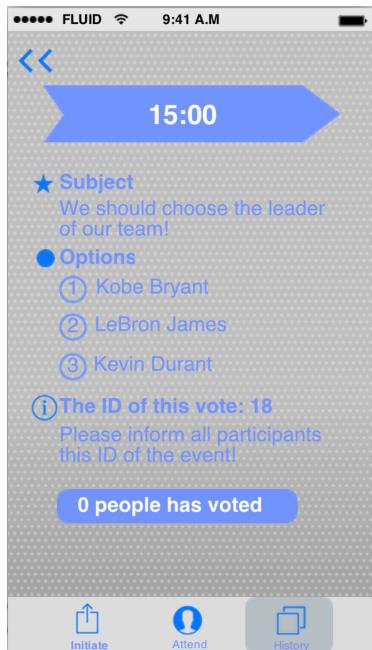
The fifth step is to set the password of this vote event so that only qualified people could attend in this vote. Because it is a simple password within 6 numbers, we won't ask you to enter the password twice.



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.



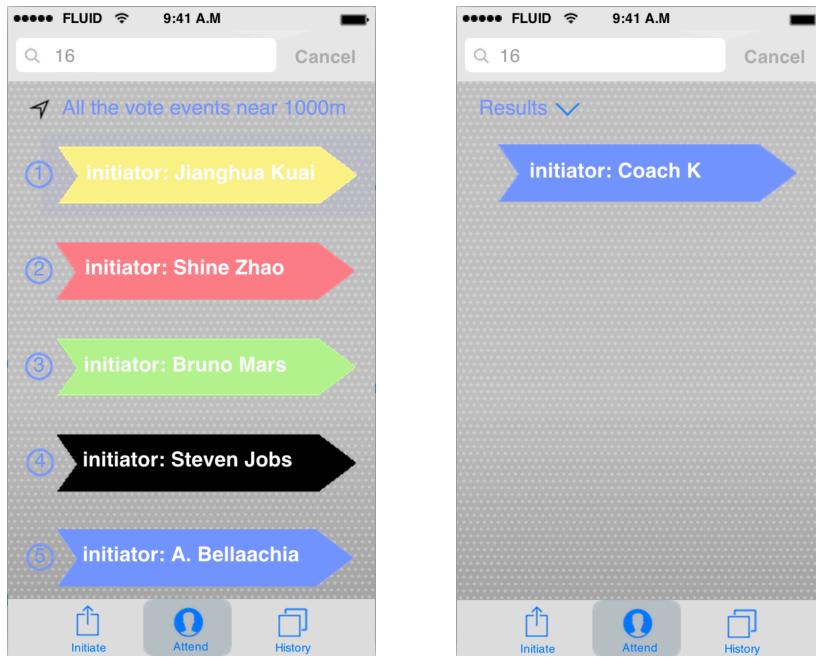
### 5.3.4 Interaction of attending a vote

This part is including the vote searching and vote operation.

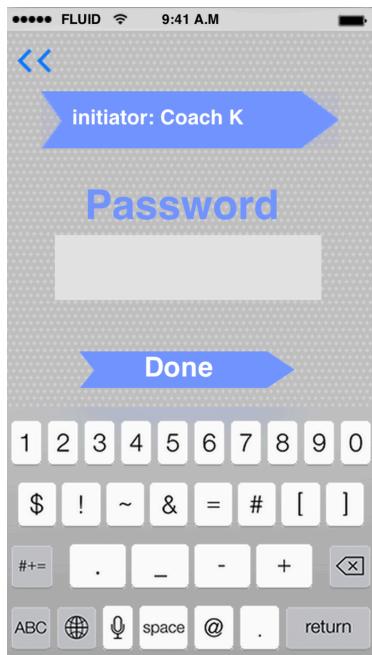
When you tap on the “attend” button on the middle of the toolbar, we would automatically list all the vote events on the screen, which are initiated within the scope of 1000 meters about your current position. You can use your finger scroll up or down on the screen to search the vote you should attend.



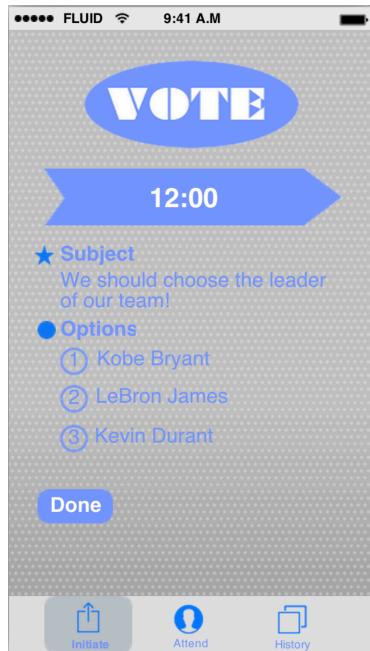
Of course, you could search the vote by specific vote id in the search bar on the top of the screen.



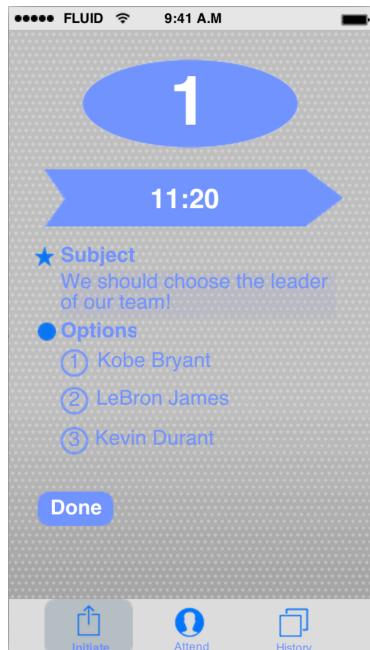
You could just tap on the specific arrow, which represents the vote you want to attend. And password would be needed before you start your vote. The initiator should tell you the password so that you can successfully enter in the vote.



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.

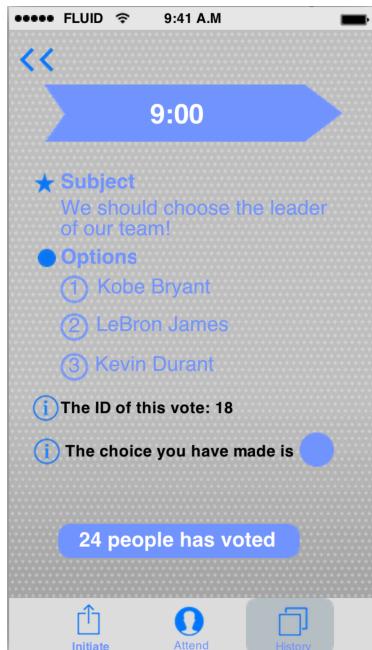


### 5.3.5 Interaction of Vote History

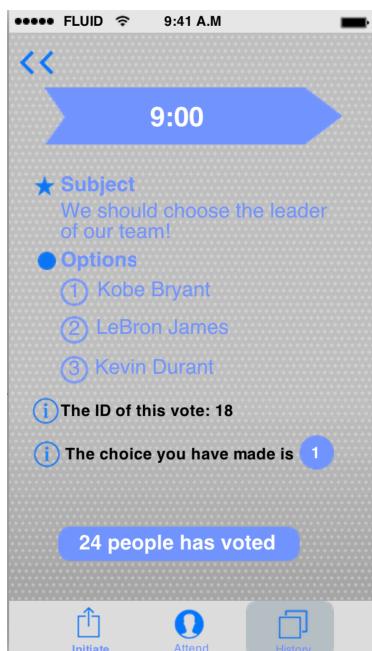
If you make your final decision and tap on “done” button, you vote would be finished and this vote event would appear on the “attended” part of the “History” in the application. Of course, you could scroll up and down to see other vote events you attended before. And the percentage of the refilling blue 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.



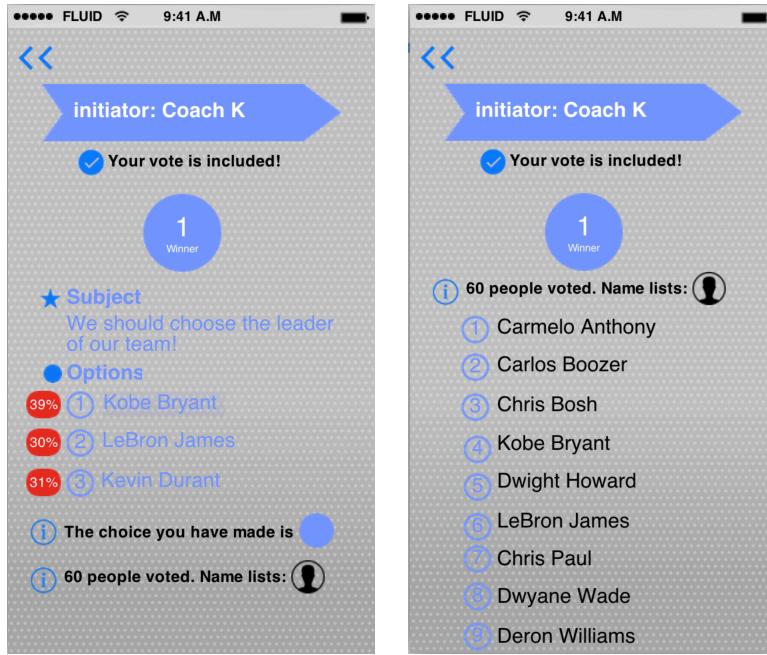
In consideration of the personal privacy, the choice you just made would not directly appear on the screen. You should tap on the little blue circle to see that.



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.

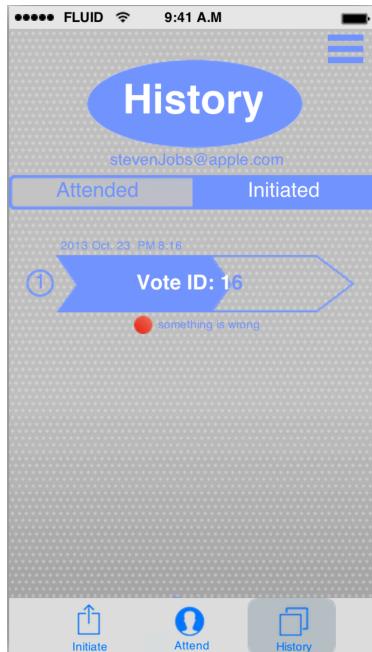


### 5.3.6 Interaction of exception handling

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”.

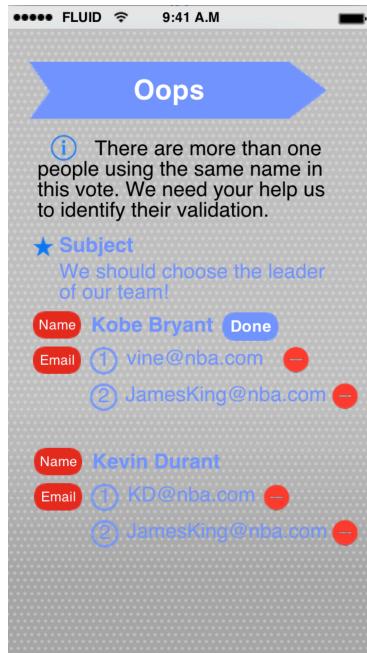


If there were more than one people using the same name in one vote event, there would be a warning for the initiator, which is a red circle under the arrow with the phrase “Something is wrong”.

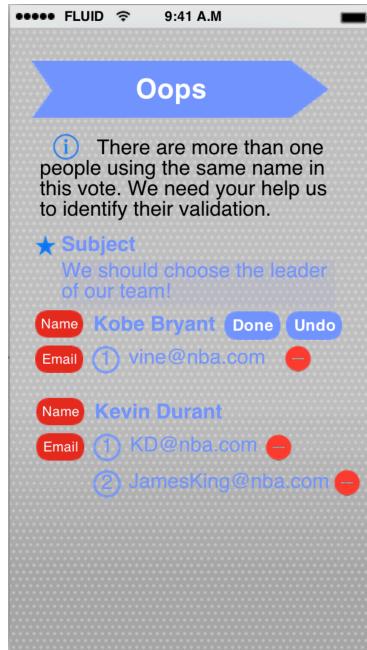


The initiator could tap on the arrow to deal with this unexpected situation. In the exception-handling page, the user would see the names that have more than one Email addresses binding with them. You

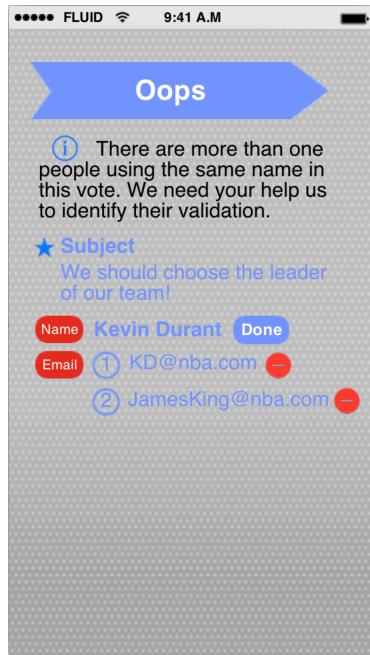
could tap on the deleting icon to delete the invalid Email address after you ask the particular person who has that name of his or her real Email address.



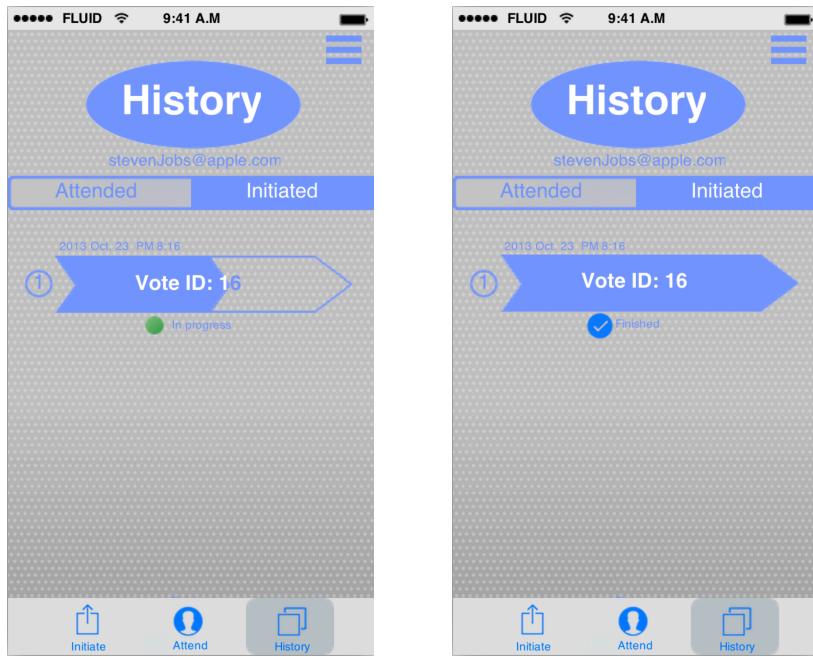
Say, the initiator deletes the second Email address, which is also binding with the name Kobe Bryant. Of course, if the initiator accidentally deletes the right one, he or she could also undo his or her action.



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)

Workload and Milestone Sheet					
Event Name	Start Date	End Date	Requirement ID	Assign To	State
Sprint One	Oct. 28 2013	Nov. 3 2013	F00001 Client&Server	Jianghua Kuai	Finished
			F00002 Server	Jianghua Kuai	Finished
			F00002 Client	Shuai Zhao	Finished
Sprint Two	Nov. 4 2013	Nov. 10 2013	F00004 Server	Jianghua Kuai	Finished
			F00003 Client&Server	Jianghua Kuai	Finished
			F00004 Client	Shuai Zhao	Finished
Sprint Three	No. 11 2013	Nov. 17 2013	F00005 Client&Server	Jianghua Kuai	Finished
			F00007 Server	Jianghua Kuai	Finished
			F00007 Client	Shuai Zhao	Finished
Sprint Four	Nov. 18 2013	Nov. 24 2013	F00006 Client&Server	Jianghua Kuai	Finished
			Test case Document	Shuai Zhao	Finished
			Manual Document	Shuai Zhao	Finished
Final Sprint	Nov. 25 2013	Nov. 30 2013	Make up all document	Jianghua Kuai&Shuai Zhao	Finished

---

## **VOTE APPLICATION ON IOS**

### **TEST CASE**

---

Version <0.8>  
<12/03/2013>

---

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>3</b>
1.1	Purpose of The Test Case Document .....	3
<b>2</b>	<b>TEST CASE SPECIFICATION .....</b>	<b>3</b>
2.1	Test case 1.....	3
2.2	Test case 2.....	4
2.3	Test case 3.....	5
2.4	Test case 4.....	6
2.5	Test case 5.....	7
2.6	Test case 6.....	8
2.7	Test case 7.....	9
2.8	Test case 8.....	10
2.9	Test case 9.....	10

## **1 INTRODUCTION**

### **1.1 PURPOSE OF THE TEST CASE DOCUMENT**

The Test Case document documents the functional requirements of the vote application test case. We have rookie iOS developer in our team so that there might be plenty of bugs existing in this application. And the design philosophy of our application is that we want to give the users of our application great experience interacting with this application. Therefore, any bugs that might affect user experience are not tolerable. We want to find out problems that existing in our application.

## **2 TEST CASE SPECIFICATION**

### **2.1 TEST CASE 1**

Test case ID:

V001

Test case Name:

Test Sign up page

Test Goal:

When tester type in disqualified information, the sign up request should be refused. While tester type in correct information, the sign up request should be accepted and the next page should be showed.

Test Data:

1. My first name and Last name
2. My Email address

Step description:

1. Launch on this application.
2. Tap on the “Sign Up” arrow shaped button
3. Just type in my first name while leave others blank and press “done” button
4. Just type in my last name while leave other blank and press “done” button
5. Just type in my Email address while leave other blank and press “done” button
6. Type in both my first name and last name as well as Email address, but on purposely type in the different Email address in the part of “Your Email again please” and finally press on “done” button.
7. Correctly type all the information as the requirement of “sign up” page and press “done” button.

**Expected Result:**

The page should give the tester notification that the information is incorrect when in third, fourth, fifth, sixth step. While in the seventh step, the sign up request should be accepted successfully.

**Actually result:**

In third, fourth, fifth step, the notification “Any information can’t be empty.” is showed. In sixth step, the notification “Email address is not the same, please check again.” is showed. In the last step, tester successfully enters into the email verification page.

## **2.2 TEST CASE 2**

Test case ID:

V002

Test case Name:

Email verification

Test Goal:

The tester could successfully create an account with typing correct code and enter into the main body of this vote application.

Test Data:

1. The received code in tester’s Email box

Step description:

1. Wait for Email from [bearworldbrave@gmail.com](mailto:bearworldbrave@gmail.com) to give the code
2. Type in the wrong code and press “OK” button
3. Type in the correct code and press “OK” button

**Expected Result:**

The tester should receive an Email that offers code. If the tester type in wrong code, he should be refused to get through the verification process. After tester type in the correct code he just received, he is supposed to enter the main body of this application.

Actually result:

After the tester tap on “done” button in the sign up page, the tester gets the Email with code “1518” within 10 seconds. And when tester tries to type in incorrect code, the notification “Wrong code” is showed. While the first time tester type in the correct code, the notification “ sorry, we have a technique problem” is showed. The second time the tester type in the correct code and press “ok” button, it finally success.

### **2.3 TEST CASE 3**

Test case ID:

V003

Test case Name:

Test initiating a vote

Test Goal:

The tester should initiate a vote successfully.

Test Data:

1. The theme color : green
2. The subject: Who is the best player in NBA?
3. The options: 1. Kobe; 2. James; 3.Durant; 4.Paul
4. The number of participants: 3
5. The password: 123456
6. The time limit: 1 hour

Step description:

1. Tap on “Choose theme color”
2. Tap on green circle
3. Tap on “Subject”
4. Type in “Who is the best player in NBA?”
5. Tap on options
6. Type “Kobe” and “James” on first blank and second blank
7. Tap on “Add an option”
8. Type in “Durant”
9. Tap on “Add an option”
10. Type in “Paul”

11. Tap on “number of participants”
12. Type in “3”
13. Tap on “Password”
14. Type in “123456”
15. Tap on “OK” button
16. Type in “123456” again
17. Tap on “OK” button
18. Tap on “Time limit”
19. Tap on “1 hour” button
20. Tap on “Done” button

Expected Result:

The animation relating these steps should be played correctly and the vote event should be created successfully.

Actually result:

The animation goes very well and the vote event is successfully.

#### **2.4 TEST CASE 4**

Test case ID:

V004

Test case Name:

Test the capability in text field of “subject”

Test Goal:

The height of text field should automatically modulate when the content changes. And the position of other titles would automatically change as well.

Test Data:

Repeat with following words 5 times: “we are testing the text field of subject”

Step description:

1. Tap on “Subject”

2. Type in repeat words “we are testing the text field of subject” 5 times

Expected Result:

The height of text field should automatically modulate when the content changes. And the position of other titles would automatically change as well.

Actually result:

The height modulation and the animation of other titles goes well.

## **2.5 TEST CASE 5**

Test case ID:

V005

Test case Name:

Test the capability of options

Test Goal:

The application should accept any number (at least 2) of options, and the animation related to this process should be OK.

Test Data:

Does need any data.

Step description:

Tap on “Add an option” about 13 times

Expected Result:

There should be 15 options existing on the screen and the layout of all the design is not in mess.

Actually result:

There are 15 options successfully being created and the layout in the page stays correct.

## **2.6 TEST CASE 6**

Test case ID:

V006

Test case Name:

Test the validation of Password

Test Goal:

The application should give the right notification when the tester enter qualified or disqualified password.

Test Data:

1. 123456
2. Any random numbers

Step description:

1. Tap on “password”
2. Just type in “1234”
3. Go on typing in “56” following the “1234”
4. Press “OK” button if it exists
5. Type in random numbers like “456938” and press “OK” button
6. Type in “123456” and press “OK” button
7. Type in “123456” again and press “OK” button

Expected Result:

After second step, the “OK” button would not show up and the “Tips” is still there tells the tester that needs six numbers. After third step, the “OK” button would come out and can be pressed. After fifth step, another tip would show up that tells “Two passwords that you entered are not matching. Please try again”. Then after seventh steps, the password should be created successfully and finally come out the tip that says “Your password is saved!”.

Actually result:

Everything goes quite well just as excepted result.

## **2.7 TEST CASE 7**

Test case ID:

V007

Test case Name:

Test the transition animation in vote initiation page

Test Goal:

The animation should go well and the layout should stay the same.

Test Data:

Do not need data

Step description:

1. Tap on “subject”
2. Tap on “Number of participants”
3. Tap on “password”
4. Tap on “Number of participants”
5. Tap on “option”
6. Tap on “choose the color”
7. Tap on “Time limit”
8. Tap on “Options”
9. Tap on “Add an option” 4 times
10. Tap on “Number of participants”
11. Tap on “Options”
12. Tap on “subject”

Expected Result:

The animation of transition from one title to another title should go as it is supposed to be. And the layout of this page stays the same as it is just started.

Actually result:

Everything goes just as expected.

## **2.8 TEST CASE 8**

Test case ID:

V008

Test case Name:

Test searching existing vote event

Test Goal:

Tester could successfully find the vote event that has just been created.

Test Data:

The vote event that has just been created

Step description:

Tap on “attend” on tap bar and look up for the vote event that we want to see.

Expected Result:

The vote event that tester has just created should be showed in the page with the initiator’s name on the arrow image.

Actually result:

The vote event is successfully load in attend page.

## **2.9 TEST CASE 9**

Test case ID:

V009

Test case Name:

Test attending a vote event

Test Goal:

Tester could successfully enter in the vote event and make the option.

Test Data:

Do not need any data

Step description:

1. Tap on the arrow image that we want
2. Type in correct password
3. Tap the first choice
4. Tap “Done” button

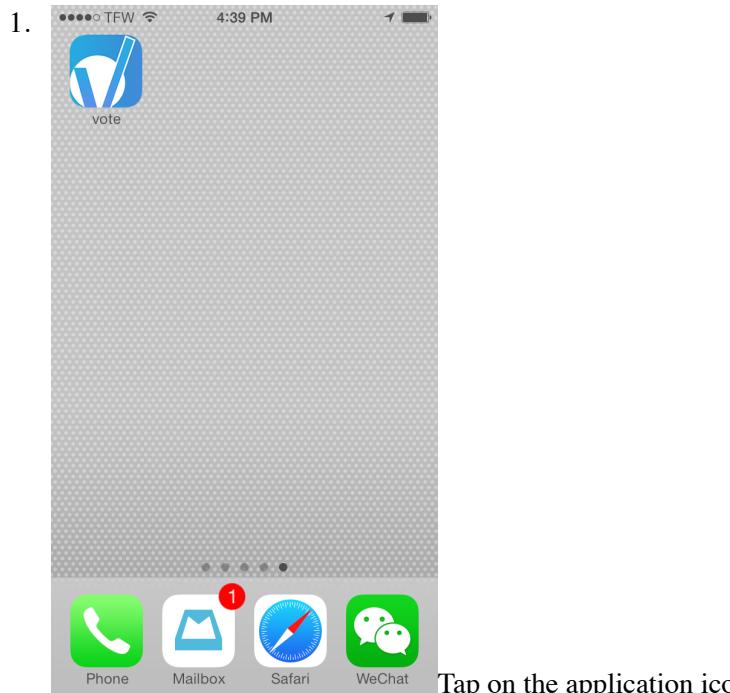
Expected Result:

After the second step, the tester should be entered into vote event successfully. And after third step, the tester could see the number “1” in the big oval image. And the “done” button should be showed up.

Actually result:

Everything just goes well as excepted.

## User manual



1. Tap on the application icon to enter into the application

2. If this application were first being launched, it would ask users to sign up or sign in with their existing account. Here, I will show the screen shots of steps that users should take when they sign up.

The image shows a mobile application's sign-up screen. At the top, there are two buttons: a left-pointing arrow labeled "Back" and a right-pointing arrow labeled "Sign Up". Below these are four input fields with labels: "First Name" (containing "Shuai"), "Last Name" (containing "Zhao"), "Email" (containing "shineamnys@hotmail.com"), and "Your Email again please" (containing "shineamnys@hotmail.com"). To the left of the input fields is a large blue arrow-shaped button with the text "Sign Up" in white. To the right of the input fields is a blue "Done" button. Below the "Sign Up" button is another blue arrow-shaped button with the text "Sign In" in white.

If you enter the correct information without leaving any blanks or entering different email addresses, users could see the next page that ask you to type in verification code, which would be sent to your email box that you just give.



## CODE

The code has sent to your email:

shineamnys@hotmail.com

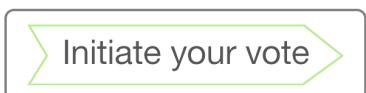


57

[Change my Email](#)



If users type in the correct code, they could successfully sign in. Of course, after that, users could initiate a vote. Users should take six steps to finish initiating a vote. First step: users could choose the color of this vote event, which would help other people to differentiate your vote from others.



► Choose theme color



II Subject

II Options

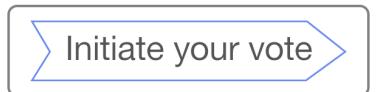
II Number of participants

II Password

II Time limit



Second step: users could enter the subject of this vote.

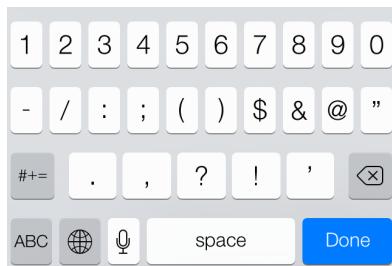


▶ Subject

› Who is the best player?

II Options

II Number of participants



Third step: users could type in the choices they want to have. If you want more choices, you could just tap on “add an option”.



checkmark Subject

▶ Options

① › Kobe

② › James

③ › Paul

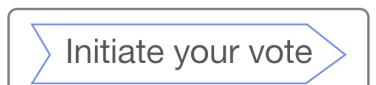
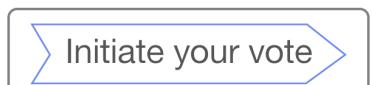
④ › Durante

plus Add an option

II Number of participants



When users have entered the choices, they could also delete any one if they want.



<input type="radio"/> Options <input checked="" type="radio"/> Kobe <input type="radio"/> James <input type="radio"/> Durante <input type="radio"/> Add an option  <input type="radio"/> Number of participants <input type="radio"/> Password <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <span style="font-size: 1em;">(i) Time limit</span> <span style="margin-left: 10px;"></span> </div>	<input type="radio"/> Options <input checked="" type="radio"/> Kobe <input type="radio"/> James <input type="radio"/> Durante <input type="radio"/> Add an option  <input type="radio"/> Number of participants <input type="radio"/> Password <div style="background-color: #f0f0f0; padding: 5px; margin-top: 10px;"> <span style="font-size: 1em;">(ii) Time limit</span> <span style="margin-left: 10px;"></span> <span style="margin-left: 10px;"></span> <span style="margin-left: 10px;"></span> </div>
--	--

Fourth step: users could type in the number of whole participants.

Options  
 Number of participants
 

minus add

Apply		
1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
	0	

Fifth step: Users should set up a password for this vote event so that people should have password to participate this vote event.

Number of participants  
 Password

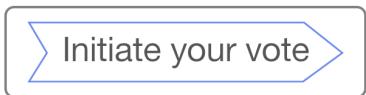
Number of participants  
 Password  
> Tip: Need Six Numbers

Number of participants  
 Password  
>  OK

Time limit

Time limit

1	2 ABC	3 DEF	1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO	4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ	7 PQRS	8 TUV	9 WXYZ
	0	☒		0	☒



Number of participants

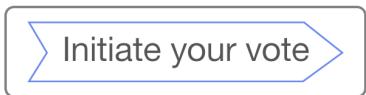
Password

Your password is saved! [reset](#)

Time limit

1	2 ABC	3 DEF
4 GHI	5 JKL	6 MNO
7 PQRS	8 TUV	9 WXYZ
	0	☒

Last step: Users should give a time limit to this vote. By doing so, every participant have to vote within limited time.



Subject

Options

Number of participants

Password

Time limit

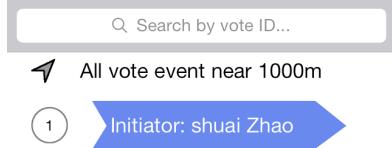
[5 min](#) [15 min](#) [30 min](#) [1 hour](#)

[Done](#)

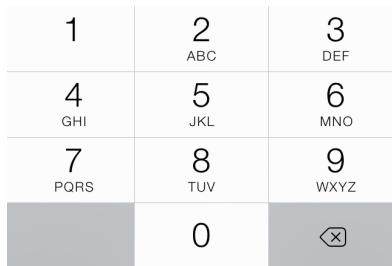


After all steps have finished, the “done” button would show up. Users could press this button to initiate this vote event. And the location information would be automatically stored on this vote event.

If users tap on the “attend” button on the tab bar that is on the bottom of the screen, they could find any vote event within 1000 meters.



When the participant gets the password of the vote event you are asked to attend, they could tap on the event they have found to enter in.



After participants successfully enter into the vote event, they could just tap on the option they want to choose.



✿ subject  
Who is the best player?

❖ Options

- (1) Kobe
- (2) James
- (3) Durante



If I type on the first choice, the page would be showed like the following screen shot.

1

Time left: about 59 minutes

✿ subject  
Who is the best player?

❖ Options

- (1) Kobe
- (2) James
- (3) Durante

Done



Finally, let me talk about the history part of this application. When you tap on the “History” button on the tab bar, you could see the “history” page. Of course, you could see the vote you attended or you initiated.

## History

Attended	Initiated
	2013-12-02 22:04:26
①	Initiator:shuai Zhao This vote is finished
	2013-12-03 21:38:44
②	Initiator:shuai Zhao This vote is in progress

