Mobile Application Programming

CS 9033 FALL 2012

SignPass

Maneli Kadkhodazadeh
mkadkh01@students.poly.edu

Changpeng Li
cli12@students.poly.edu

Jiankai Dang (Team Leader)
jdang01@students.poly.edu

Yang Li
yli28@students.poly.edu

Contents

1. Objective

With a growing dependence on websites for our daily lives, we are required to create and memorize a growing number of username/password combinations. Each website has different requirements for password formats and minimum requirements are growing in complexity to better ensure our security. These different and complex passwords are not only hard to remember but also time consuming and annoying. In addition, even by choosing complex passwords; we cannot ensure our security.

Given this problem, we need a new solution which is simpler and more secure. A solution which better parallels real-world usage before websites took hold of our lives. In the 'real world', signatures in-person is used to ensure security. Signatures are a simple method for a person to rely on, since the same signature is used each of time with no requirement to memorize username/password combinations.

With our application, we seek to re-create the benefits of the in-person signature to replace complex and insecure username/password combinations. The application will allow users to login easily to their bank accounts simply and quickly using a touch-signature or touch-phrase.

Therefore in our project we will create an un-real website bank application to demonstrate the solution to this problem that we mentioned above, using real signature to authorize users account instead of using password.

2. Requirements

2.1 User Interface

a) The interface must be consistent in both iPhone and iPad which support iOS5.0 and later versions;
b) The interface must tell the end-user what the signature is being signed for, when the signature request is sent;
c) The interface must show the result of signature authentication;
d) The view should be maintained when the user switches the apps;
e) The interface should inform the user when the communication between iOS device and back-end server is interrupted or overtime;
f) The iOS device should provide a view for all possible coming information from service;
g) The view should provide sufficient prompts to let users understand the current situation and lead them accomplish the operation.

2.2 Server

a) The server should provide a capability of 20 connections for test version, and provide the possibility to extend to 100 connections;
b) The server should provide detailed errors descriptions;
c) The server should send text form message to the end users when it is necessary (such as verification request from third party, server maintenance notification);
d) The response time should be in an acceptable range with a max of 5 sec for signature verification and registration.

2.3 Security

a) The whole network communication must be encrypted;
b) There should be a time limit during the process of signature authentication;
c) The API libraries used for communication between iOS devices and back-end server are applicable to check incoming token which can prevent back-end server from potential injection attack;

2.4 Maintainability

a) All libraries must be well documented including at least library functionalities, API description, all variables, logic operations etc;
b) The naming rule should strictly follow the naming document;
c) The development over iOS should strictly follow MVC design pattern;
d) The server should reduce the coupling among the modules so that provide availability of extension of API.

2.5 Interoperability

The API for third party should be clear and straightforward, and well-performed over the given protocol when it communicates with the customer program.

2.6 Timeline

The project timeline should include at least iOS app finished, back-end server (signature authentication server) setup finished, third party simulation server finished and final delivery.

3. Overall Architecture

Our system has three major components: iOS Application, Back-End Server, and Bank Services.

3.1 iOS Application

We'll design a user friendly user interface of iOS App and provide the following basic functions:

- Training of user's signature
- Sign and Tap Login
- Access bank account through signature

3.2 Back-End Server

- Save the training signature data from iOS App to server database
- Recognize and match signature data between iOS App and server database
- Observe and handle the login event from bank service
- Return the user authentication results to bank server

3.3 Bank Services

- Provide login interface for user
- Dispatch login event to Back-End Server
- Receive the user authentication results from Back-End server and inform user about it

The implementation processes are shown in the following graph (Figure 3-1):

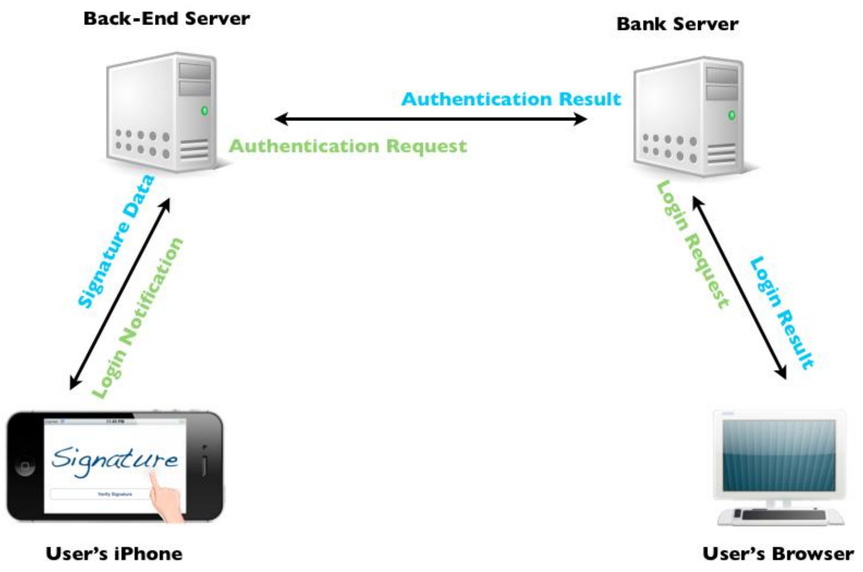


Figure 3-1

4. User Interface

The very first question that would come to mind after making decision to make an application is, in which operating system we want to create it, Android or iOS. iOS is more popular than Android, because of its ease of use and user-friendly environment. According to SignPass requirements, SignPass is focusing on ease of use and being user-friendly, Therefore SignPass is created on the iOS. SignPass user interface created with Xcode version 4.5, the newest version of the Xcode to make the application more compatible with older version of iOS.

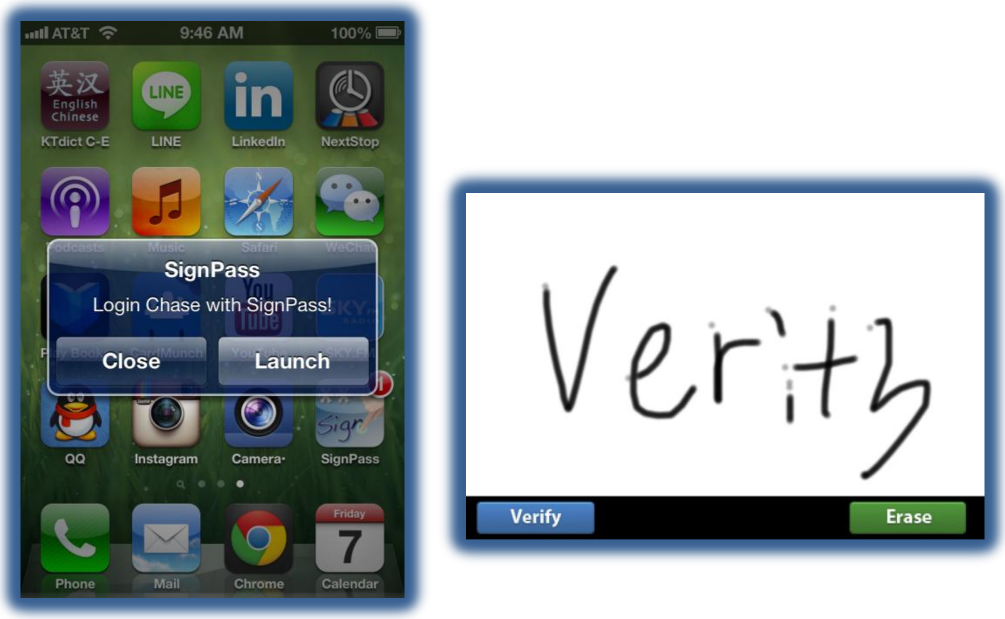
4.1 User Interface Design

There are two different scenarios in SignPass application. Receiving notification on the iPhone/iPad to sign in with SignPass

4.2 Starting the application from iPhone/iPad

4.2.1 Receiving notification on the iPhone/iPad to sign in with SignPass

This scenario starts when the user is trying to connect to the website's bank through the PC or laptop and he preferred to use his signature instead of password to sign in to the website. Therefore the bank server will send a notification with some user and bank information to the SignPass back-end. SignPass back-end will send the notification to the user's iPhone/iPad, and notify the user that he should sign his signature on the SignPass.



SignPass

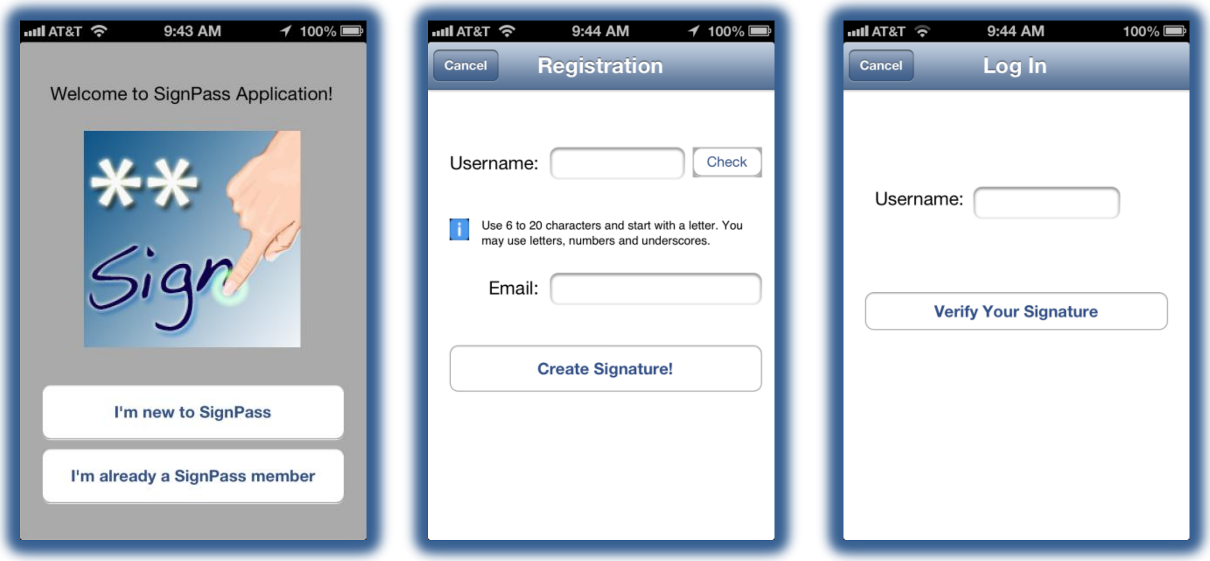
Login Chase with SignPass!

Figure 4-1 and 4-2

By pressing signature on the notification page, it will navigate the user to the signing page in the application. (Figure 4-2) User has 5 times limit to sign in his signature. If the signature verification succeeded, the user will be notified on the application that process is done and the user will log in to the bank web site. Otherwise user will be notified that the verification was not successful and the SignPass back-end will send unsuccessful verification to the web site bank server.

4.2.2 Starting the application from iPhone/iPad

This scenario starts when the user starting the application from main page of the iPhone/iPad. This scenario is also divided in two different scenarios, Figure 4-3



Use 6 to 20 characters and start with a letter. You may use letters, numbers and underscores.

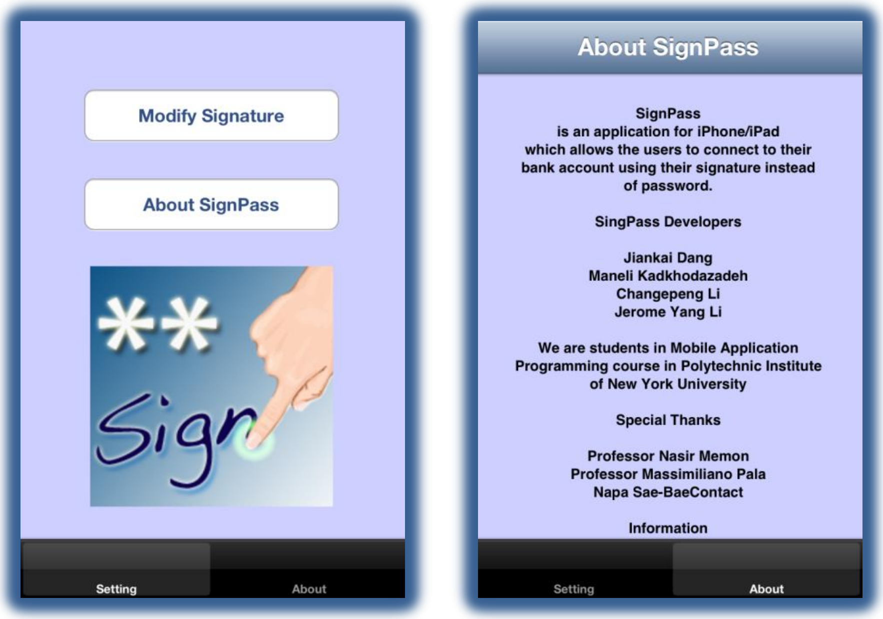
Figure 4-3 , 4-4 and 4-5

User is using SignPass for the first time will go Figure 4-4. User is already the SignPass member will go to Figure 4-5

4.2.2.1 User is using SignPass for the first time. By pressing the SignPass icon on the main menu of the iPhone/iPad, the user will be navigated to the first page, which has two different options. Which are, if the user is using SignPass for the first time and need to go through registration process. The registration view (Figure 4-4) asks the user a unique username and an email address followed by a button to create their own signature.

4.2.2.2 The option "I'm already a SignPass user" will navigate the user to the page which is asking the user's username (Figure 4-3) followed by the next view which the user is supposed to sign his signature (Figure 4-5).

If the signature verified successfully the user will navigate to another view, (Figure 4-7), the main view. The main menu of the SignPass application has 2 tabs in tab bar controller. The first tab is the page that shows the recent activities, Notifications, that has been sent to the SignPass from the bank server with the time and more information about the notification. The second tab in the tab bar controller is the setting view, (Figure 4-8), which allows the user to modify his signature, and also is a button, that navigates the user to the view that there is information about the SignPass.



SignPass
is an application for iPhone/iPad
which allows the users to connect to their
bank account using their signature instead
of password.

SingPass Developers

Jiankai Dang
Maneli Kadkhodazadeh
Changepeng Li
Jerome Yang Li

We are students in Mobile Application
Programming course in Polytechnic Institute
of New York University

Special Thanks

Professor Nasir Memon
Professor Massimiliano Pala
Napa Sae-BaeContact

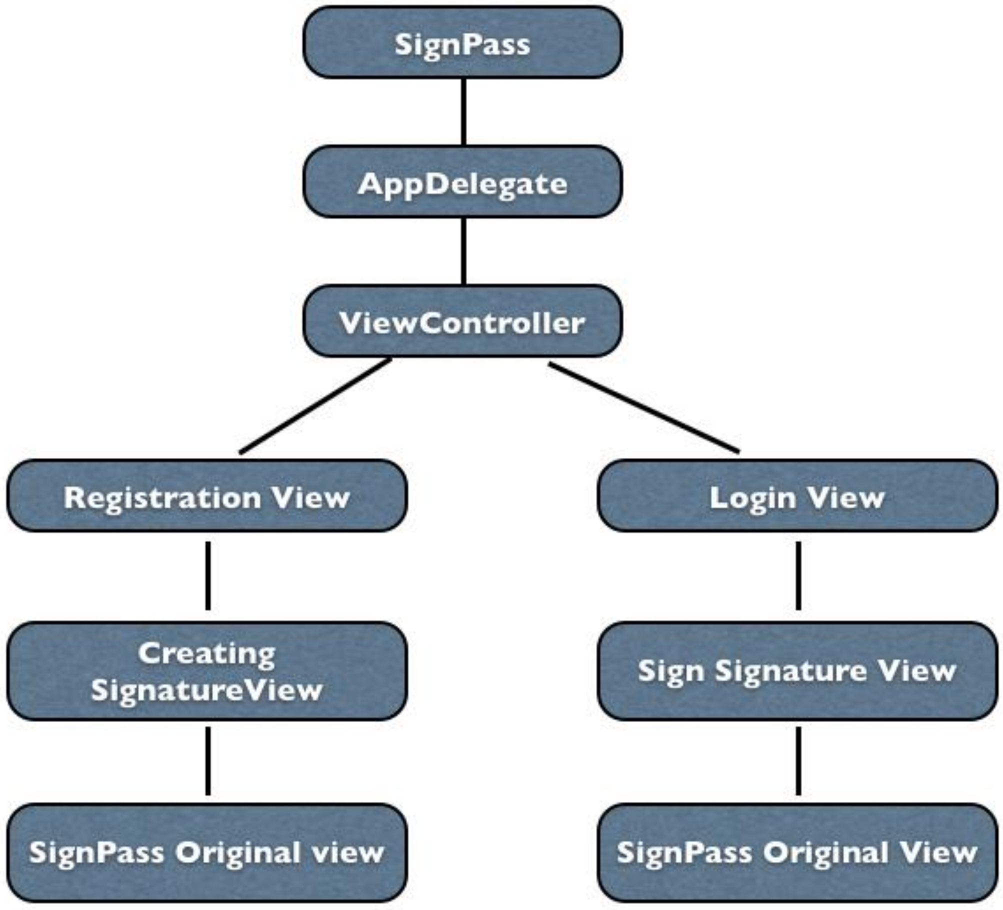
Information

Figure 4-6 and 4-7

4.3 User Interface Architechture

As it mentioned before, Xcode version 4.5 being used to implement the user interface for the SignPass application. The implementation of the user interface is pretty straight forward.

In the following, there is a class scheme (Flow Chart 4-1) that shows the overflow of the View Controllers. Each part of this scheme already explained in the user interface design part.



Flow Chart 4-1

5. Back-End

The Back End describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, processing logic, and external interfaces.

5.1 Architecture

To guarantee the security between the verification server and the end-user, we use HTTP as network communication protocol; we also use HTTP between the verification server and any third party.

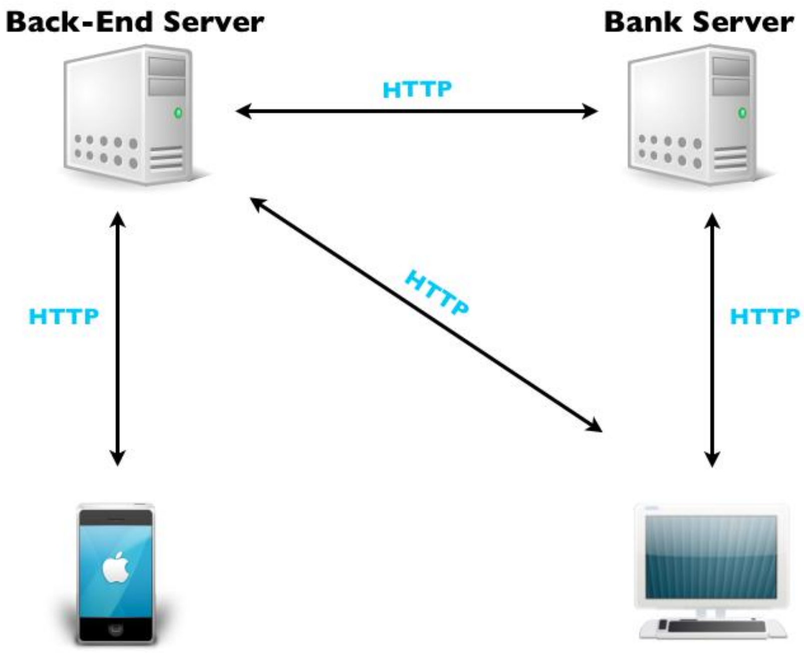


Figure 5 – 1

5.2 System Hardware Architecture

- Back-End Server
- Bank Server(for demonstration purposes, we’ll use the same server as Back-End Server)

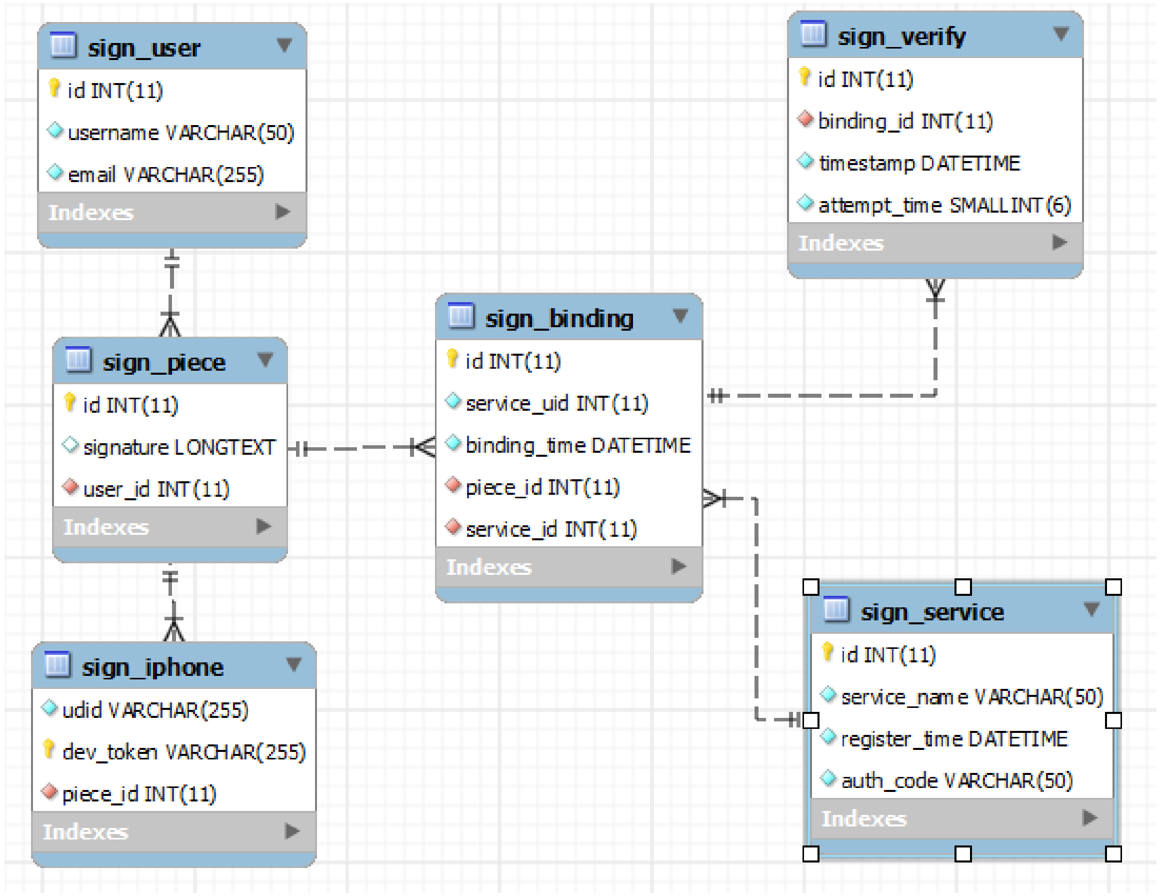
5.3 System Software Architecture

- iOS Application (Client Side) -- **Objective-C**
- Back-End Software Program (Server Side) -- **Python, SQL**
- Bank Services (third-party software, including Bank website and Bank Server software program) -- **Python, HTML, JavaScript, CSS**

5.4 File and Database Design

- Use Github to manage our source code and control software versions
- Choose MySQL as database

The database of SignPass server as below:



5.5 Communicational Design

The system includes more than one component, so there is a requirement for internal communications to exchange information, and support input/output functions.

The number of servers and clients to be included on each area network:

- 2 servers (Back-End Server and Bank Server)
- 2 clients (iOS client and Bank website)

Format(s) for data being exchanged between components:

- JSON and XML
- Encoded signature data

5.6 Work Flow

The figure 5 – 2 and 5 – 3 give the work flow of the Back End Server.

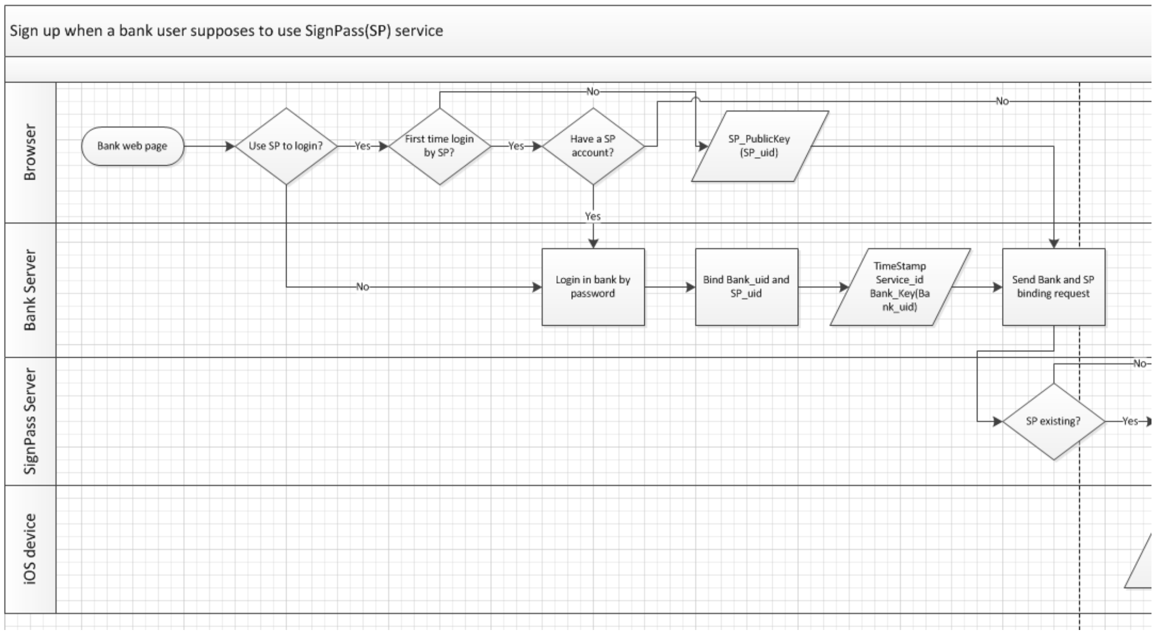


Figure 5 – 2

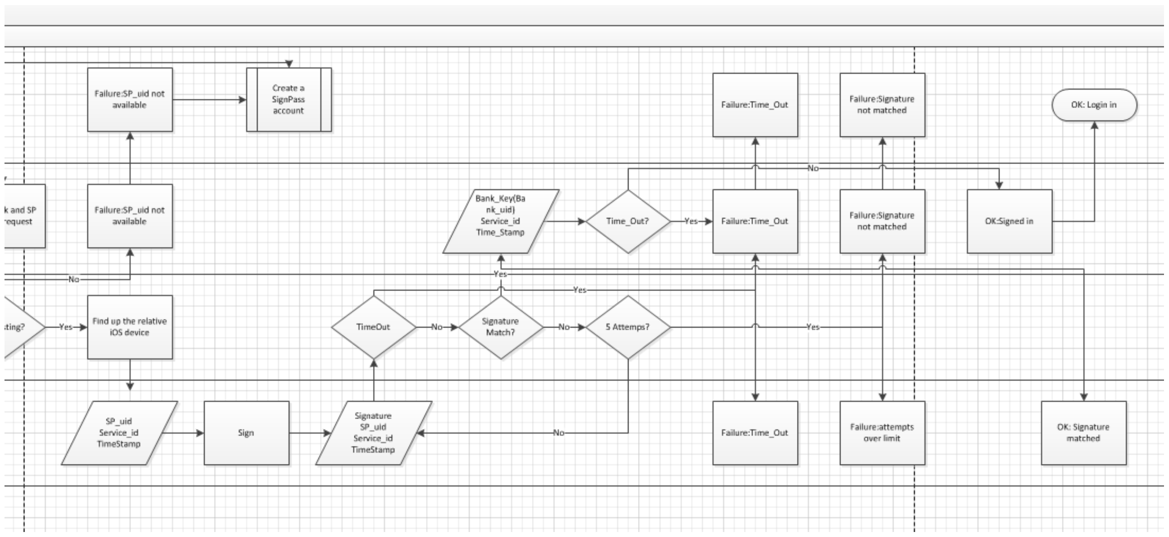

Failure:SP_uid not available
Create a SignPass account
Failure:SP_uid not available
Find up the relative iOS device
SP_uid Service_id TimeStamp
Sign
Signature SP_uid Service_id TimeStamp
TimeOut
Signature Match?
5 Attemps?
Bank_Key(Bank_uid) Service_id Time_Stamp
Time_Out?
Failure:Time_Out
Failure:Signature not matched
Failure:attempts over limit
OK:Signed in
OK: Login in
OK: Signature matched
Yes
No

Figure 5 – 3

6. Signature

The signature view can be used in difference cases. The button will be changed depends on the operation case, for example, “Verify” which is used to collect the signature information and send the information to the Back End Server. “Register” is used when user use SignPass in the first time. “Modify” means user wants to change his or her signature.

All cases have the button is “Erase” button, which help user to clear the signature on view.

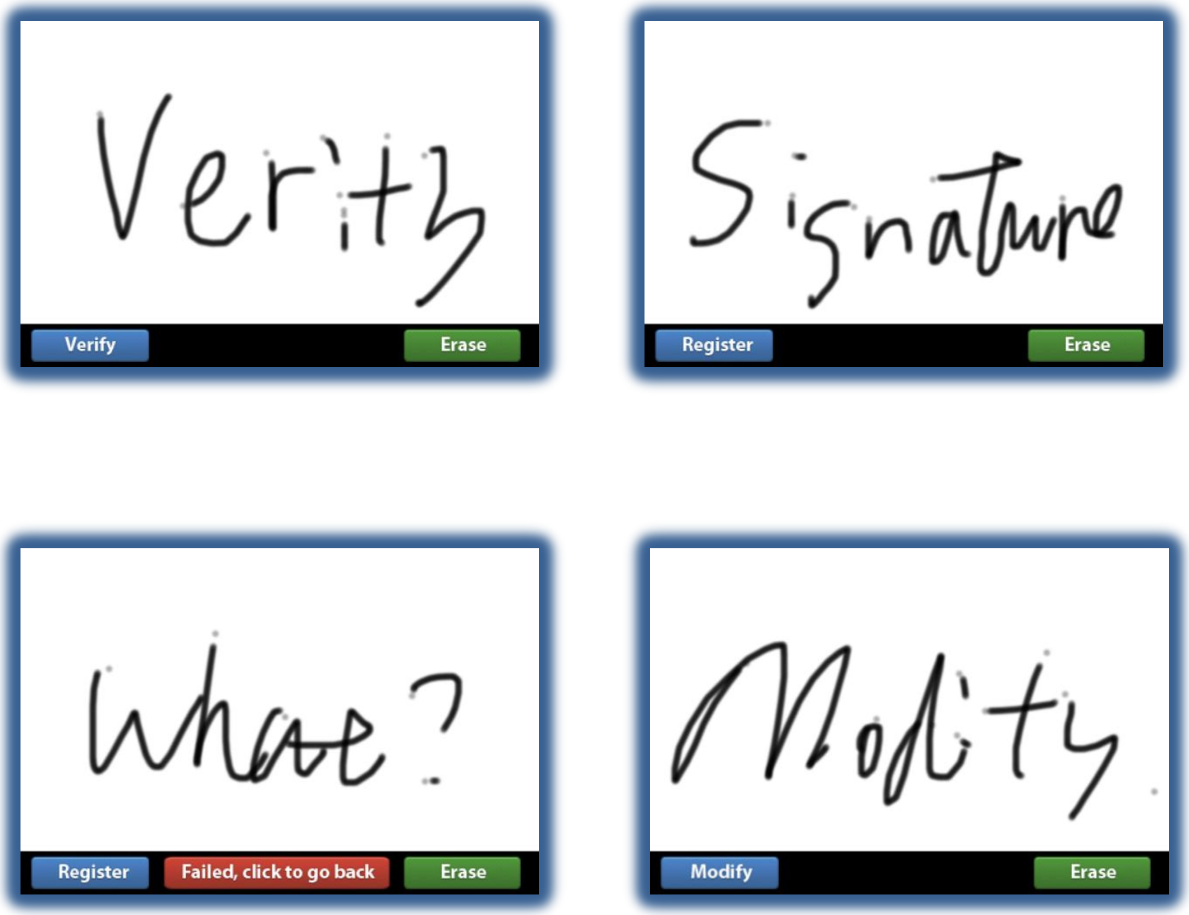
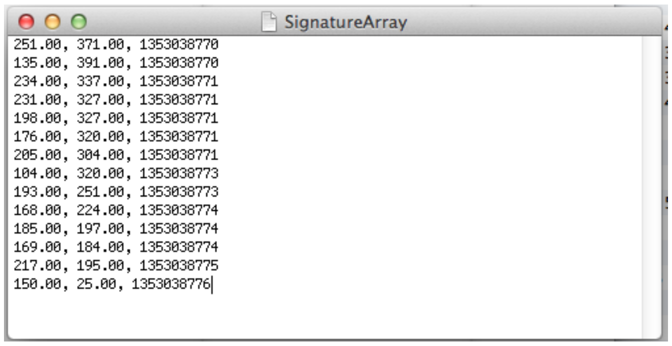


Figure 6 – 1



SignatureArray

```
251.00, 371.00, 1353038770
135.00, 391.00, 1353038770
234.00, 337.00, 1353038771
231.00, 327.00, 1353038771
198.00, 327.00, 1353038771
176.00, 320.00, 1353038771
205.00, 304.00, 1353038771
104.00, 320.00, 1353038773
193.00, 251.00, 1353038773
168.00, 224.00, 1353038774
185.00, 197.00, 1353038774
169.00, 184.00, 1353038774
217.00, 195.00, 1353038775
150.00, 25.00, 1353038776
```

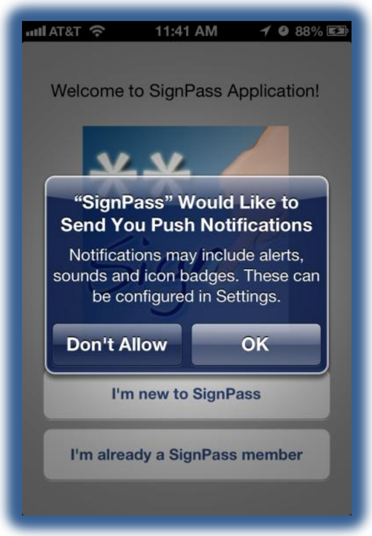
Figure 6 – 2

The Signature information array is defined as follow, the first column is X point, the second column is Y point, and the last point is time. The Figure 6-2 is the log file which collects from SignPass.

7. Integration

7.1 Push Notification

When the first time to start SignPass on iPhone, the iPhone will pop a dialog to ask user whether they want to use the SignPass notification. The dialog will like Figure 7-1.



AT&T 11:41 AM 88%

Welcome to SignPass Application!

"SignPass" Would Like to Send You Push Notifications

Notifications may include alerts, sounds and icon badges. These can be configured in Settings.

Figure 7-1

Apple Push Notification Service (APNS) is the centerpiece of the push notification feature of iOS. The flow below is the depiction of the virtual network APNS make possible among providers and devices. The device and provider sides of APNs both have multiple points of connection.

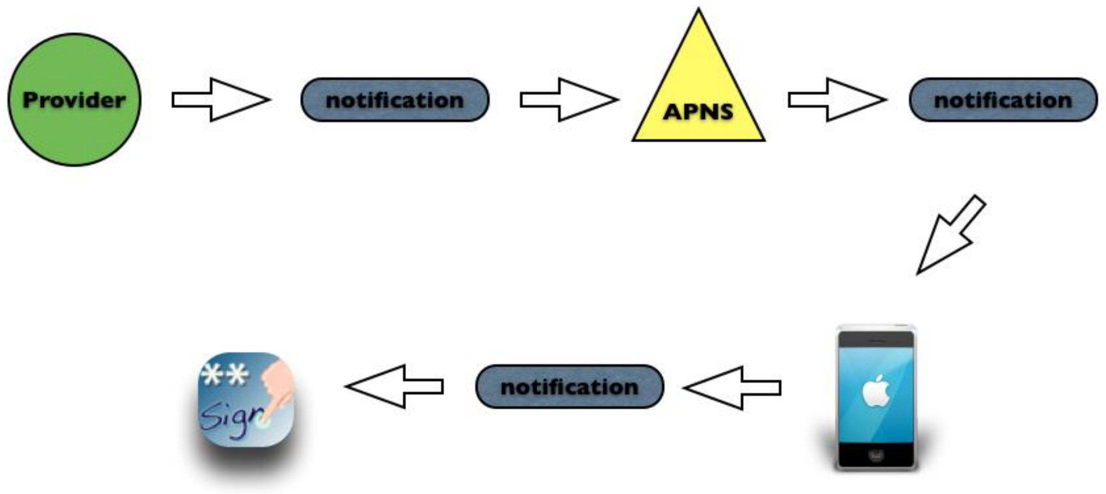


Figure 7-2

7.2 Picture Button

Since our signature view only support landscape rotation, so we create landscape direction button picture to implementation this. Each function has two buttons, which the normal style and press style.

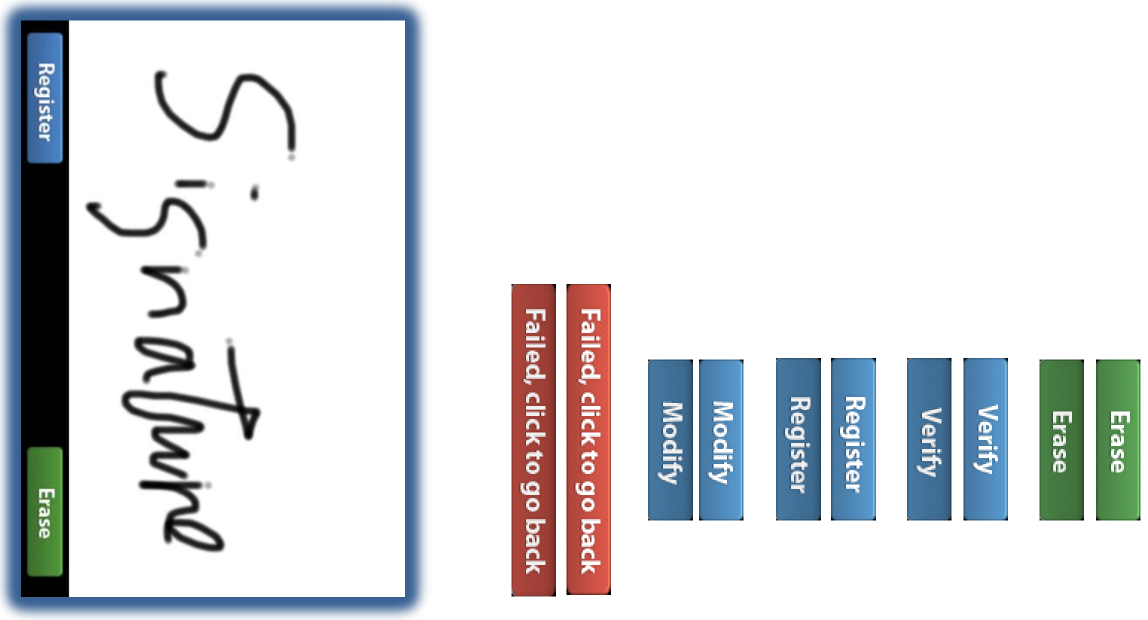


Figure 7-3

7.3 iOS & Django

SignPass send Http request to Back-End server at first, and then the Django platform which we deploy on Apache we take process the request. Django platform we send respond to Back-End Server, and then Back-End Server take the respond back to SignPass.

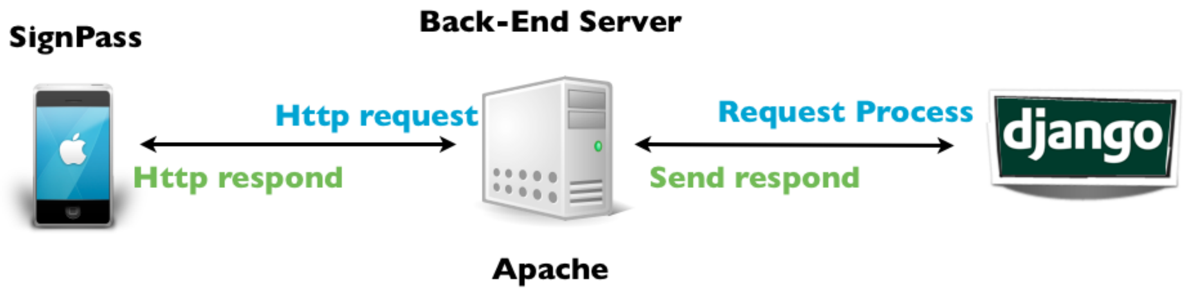


Figure 7-4

7.4 Verification (Non Implementation)

Since the iOS version verification library is on developing. SignPass cannot do verification with public verification library; we will combine the public signature verification library in the future implementation.

8. Business Model

8.1 Key Partners

Our key partners are all third-party services which require their customers to login with password. Among these third-party services, banks, including JPMorgan Chase, Bank of America, and Citibank, are our major target partners.

Motivations for partnerships:

- Optimization and economy
- Reduction of risk and uncertainty
- Acquisition of particular resources and activities

Our key partners are also our key suppliers. The key resources we acquire from partners are the necessary customer data, which will be encrypted and combined together with SignPass user data. The goal is to provide the customers with more convenient services.

The key activities our partners perform are providing their customers with SignPass login entrance, encrypting and sending necessary customer data to SignPass.

8.2 Value Propositions

The values we deliver to the customers are the convenience and security for login to third-party services. The biggest nuisance today is the 40 different passwords our customers have to create and change. We are helping to solve this problem for our customers.

Characteristics of SignPass:

- Convenience/Usability
- Performance
- Customization
- “Getting the Job Done”
- Design
- Cost Reduction
- Risk Reduction
- Accessibility

9. Future Work

9.1 The security check of geo location

SignPass will record the geo locations where users used their signatures to login. If a user uses SignPass in different places, especially when the distance between these two places is very long, SignPass will check the validation of user information to improve the data security of our user.

9.2 The ability to store various passwords

SignPass will provide our customers with “cheat sheet” to store various passwords for them. This will be a backup of the passwords for our customers. They could access the passwords via the signature they customized.

10. References

1. The Business Model Canvas:
http://www.businessmodelgeneration.com/downloads/business_model_canvas_poster.pdf

2. Core App Objects
http://developer.apple.com/library/ios/#documentation/iphone/conceptual/iphoneosprogrammingguide/AppArchitecture/AppArchitecture.html

3. GLPaint Sample Code
https://developer.apple.com/library/ios/#samplecode/GLPaint/Introduction/Intro.html#//apple_ref/doc/uid/DTS40007328-Intro-DontLinkElementID_2

4. django project reference
https://www.djangoproject.com/

5. MySQL reference
http://www.mysql.com/

6. Python reference
http://www.python.org/

7. Apple Developer Resources
https://developer.apple.com/resources/

8. GLPaint Dissected
http://www.visualnewt.com/OpenGL/learning_iphones_opengl_es/part_i_-_glpaint_dissected_.html

9 Push Notifications
http://developer.apple.com/library/mac/#documentation/NetworkingInternet/Conceptual/RemoteNotificationsPG/ApplePushService/ApplePushService.html