**Name**: Project completion report

**Version**: 1.0

**Date**: 15 July 2011

**Author**: Dr. Ambikesh Jayal, University of Gloucestershire

**Reviewers:** Dr. Kevin Hapeshi, Dr Vicky Bush, Dr. Godfried Williams

**Contact**: ajayal@glos.ac.uk

**Project title**: i-phone low level analysis

**Knowledge-base provider**:The University of Gloucestershire

**Knowledge-base academic** contact Dr. Kevin Hapeshi, Head of Computer Department

**Knowledge-base Business Development contact**:Rod Pugh

**Recipient**: Rod Pugh, Managing Director, First Cyber Security Limited

**Comments**: None

Table of Contents

[1. About](#_xqj7ynn14z8f)

[2. Objectives](#_vfhmrrkv2amg)

[3. Deliverable](#_ztmyvd902qg)

[3.1 Approaches considered](#_39x05fijdfti)

[Approach 1: Write a Background app to intercept all TCP traffic](#_cydnm43en1ks)

[Approach 2: Use javascript and frame/iframe](#_70qxbic3fy17)

[Approach 3: Use a proxy server](#_7hiqnt8axunc)

[Approach 4: Create an app with a custom browser](#_u7x967k671gi)

[3.2. iPhone App](#_n99bd68i5gz5)

[4. Appendix](#_xo8n0rfxfpx8)

[4.1 Useful Links](#_8qzr5aqncfi6)

[4.2 Messages posted in forums](#_c09j515hyh1h)

[Message 1 : iPhone app to intercept TCP traffic](#_z8u39bx6dto0)

[Link:](#_nzgojda2m1zk)

[Message:](#_g2u3jx84xph3)

[Response](#_tx6ito9etl15)

[Message 2 : iPhone app to intercept TCP traffic](#_z8u39bx6dto0)

[Link](#_g4fjholkqtwh)

[Message](#_8eeb7ej7kzrq)

[Response](#_imcohvp4nptp)

[Message 3 : iPhone app to intercept TCP traffic](#_z8u39bx6dto0)

[Link](#_ozswaoxeyiq)

[Message 06-29-2011](#_l2ylz2k61zmw)

[Response 2](#_irnma39hj44e)

[Response 3](#_ou5zrtmjfx7l)

[4.3 Code](#_7yi1cfotwy92)

[firstAppViewController.m](#_puyjdjfw9n1)

[firstAppViewController.h](#_bue2zny4pfdu)

[firstAppAppDelegate.m](#_fsp8lvvpkzvw)

[firstAppAppDelegate.h](#_n25y9qvuteof)

[firstAppViewController.xib](#_tmx8lmbvzal6)

# **1. About**

This document details the work done by University of Gloucestershire for First Cyber Security Limited towards the project titled "i-phone low level analysis" which is a prototype iPhone app.

First Cyber Security Limited has developed patent pending S.O.L.I.D. Authentication® technology which incorporates innovative processes for authenticating web sites, validating logo use, and proactively discovering, warning and protecting users from phishing sites, logo abuse, and counterfeit web sites. This technology currently runs on Microsoft Windows based operating systems, but the company would like to extend it to other platforms and mobile devices such as the iPhone.

The aim of this work is to exploring various approaches to implement the S.O.L.I.D. Authentication® technology on iPhones and demonstrating the technical feasibility of the recommended approach by developing a prototype iPhone app.

# **2. Objectives**

1. Ability for the i-phone browser to send to an application shell information on what

URL is being visited.

This objective has been achieved. The following information is being printed on the console.

BEFORE LOAD: This is the URL entered by the User in the browser URL bar

DOCUMENT LOADED: This is where the browser ended up.

IP Addresses

IFRAME SOURCE

IMAGE SOURCE

2. The ability to know when the browser is the foreground/active application.

Since we have created our own app with an inbuilt browser, so there is not need to know if the browser is .foreground/active application.

3. Ability to send information on links within a web page.

This objective has been achieved. The information on links within a webpage is being printed on console if the user clicks on the link.

4. Information on how to create indicators and display logos and move them on the

screen.

This objective has been achieved. A horizontal text bar has been created in the bottom of the screen. This text bar is being used to display messages to the user. Although it is not possible to move this text bar on the screen but it can be used to display logos.

# **3. Deliverable**

The deliverable of this project is an iPhone app that demonstrates the technical feasibility of implementing the S.O.L.I.D. Authentication® technology on iPhones. We have considered the following four approaches and decided to use the fourth approach which is to create our own browser app with a custom browser.

## **3.1 Approaches considered**

### **Approach 1: Write a Background app to intercept all TCP traffic**

This approach involves writing an app for that will run in background and intercept all TCP traffic on iPhone generated by Safari or other browser. The iPhone app store will not allow such an app and so we have decided against using this approach. We have posted some messages in the official apple developer forums and two other forums to confirm this. See the appendix for these messages.

### **Approach 2: Use javascript and frame/iframe**

This approach involves forcing the user to browse the internet from within a webpage created by us and using Java script to collect and process the information entered by the user. The webpage will open the URL entered by the user within a frame/iframe. The main disadvantage of this approach is that javascript is not secure and so we have decided against using this approach. The other difficulty in this approach is in making sure that user always stays within the frame/iframe of our web page because otherwise we lose control over URL visited by the user.

### **Approach 3: Use a proxy server**

This approach involves creating a proxy server and changing the setting of iPhone browser to always go through this proxy server. This approach is technically not feasible as it is not possible to change the proxy server settings of the iPhone browser. So we have decided against this approach.

### **Approach 4: Create an app with a custom browser**

This approach involves creating an app with an inbuilt custom browser. We recommend that this approach should be used. This approach is technically feasible as we have created a prototype of such an app. This would be a secure approach as it uses Apple Cocoa framework which is implemented in Objective-C. It should be noted that according to the current iPhone policy, this app will be rated 17+ because it contains an embedded web browser.

## **3.2. iPhone App**

This iPhone app consists of an inbuilt web browser with address bar, back and forward buttons and a horizontal text bar at the bottom of the screen to display messages to the user. The screenshot of the app have been shown in Figure 1 through Figure 4. The code of this app is provided in Appendix 4.3.

This app has two main features. Firstly it intercepts the URL entered by the user and prints the shell information about this URL on console. The shell information includes the Host, IP Addresses, IFRAME SOURCE and IMAGE SOURCE. Table 1 shows the data generated when user enters “www.o2.co.uk”. Table 2 shows the data generated when user enters “[www.o2.co.uk/shop](http://www.o2.co.uk/shop)” which is equivalent to user clicking the “Shop” link on home page “www.o2.co.uk”. Table 3 shows the data generated when user enters “www.cnn.com”.

Secondly it intercepts the URL within a web page that the user has clicked and prints this URL in the bottom horizontal text bar.

|  |
| --- |
| BEFORE LOAD www.o2.co.uk  DOCUMENT LOADED [http://www.o2.co.uk/](https://webmail.glos.ac.uk/exchweb/bin/redir.asp?URL=http://www.o2.co.uk/)  IP Addresses 82.132.141.84  IFRAME SOURCE: <http://view.atdmt.com/iaction/repo2s_o2home_1> |

Table 1: OUTPUT when user enters www.o2.co.uk

|  |
| --- |
| BEFORE LOAD www.o2.co.uk/shop  DOCUMENT LOADED [http://shop.o2.co.uk/home](https://webmail.glos.ac.uk/exchweb/bin/redir.asp?URL=http://shop.o2.co.uk/home)  IP Addresses 2.18.178.128  IFRAME SOURCE: [http://switch.atdmt.com/iaction/repo2s\_warmlead1o2shop\_4](https://webmail.glos.ac.uk/exchweb/bin/redir.asp?URL=http://switch.atdmt.com/iaction/repo2s_warmlead1o2shop_4)  IFRAME SOURCE: [http://switch.atdmt.com/iaction/dnaoo2\_shophomepage\_1](https://webmail.glos.ac.uk/exchweb/bin/redir.asp?URL=http://switch.atdmt.com/iaction/dnaoo2_shophomepage_1) |

Table 2: OUTPUT when user enters www.o2.co.uk/shop

|  |
| --- |
| BEFORE LOAD www.cnn.com  DOCUMENT LOADED [http://edition.cnn.com/](https://webmail.glos.ac.uk/exchweb/bin/redir.asp?URL=http://edition.cnn.com/)  IP Addresses 157.166.226.45  IP Addresses 157.166.226.46  IP Addresses 157.166.255.31  IP Addresses 157.166.255.32 |

Table 3: Output when user enters www.cnn.com



Figure 1: Welcome screen



Figure 2: User enters <http://www.o2.co.uk> and clicks double arrow button

## 

Figure 3: User clicks url within a webpage

## 

Figure 4: URL clicked by user is printed at bottom bar

# **4. Appendix**

## **4.1 Useful Links**

Cocoa framework: <http://developer.apple.com/technologies/mac/cocoa.html>

Apple Developer Forum: <https://devforums.apple.com/community/ios/devtools>

iOS Developer Library

http://developer.apple.com/library/iOS/navigation/#

## **4.2 Messages posted in forums**

### **Message 1 : iPhone app to intercept TCP traffic**

|  |
| --- |
| **Link:**  https://devforums.apple.com/thread/108470 **Message:**  iPhone app to intercept TCP traffic  29-Jun-2011 03:26  Hi,    Subject: iPhone app to intercept TCP traffic    I would like to write an app for iPhone that will run in background and intercept all TCP traffic on iPhone generated by Safari browser.    Is it possible to write such an app? Any relevant links or articles would be much appreciated.    Anyone aware of similar app that runs on normal iPhone (not jailbreak)?      Thanks.  Ambi. **Response** Such an application will not make it to the app store.    You will need a jailbroken phone to do this, and there certainly is no official documentation on it, either.    Alex |

### **Message 2 : iPhone app to intercept TCP traffic**

|  |
| --- |
| **Link** https://discussions.apple.com/thread/3148599 **Message** 29-Jun-2011 04:18  Hi,  I would like to write an app for iPhone that will run in background and intercept all TCP traffic on iPhone generated by Safari browser.    Is it possible to write such an app? Any relevant links or articles would be much appreciated.    Anyone aware of similar app that runs on normal iPhone (not jailbreak)?    Thanks.  Ambi. **Response** >intercept all TCP traffic on iPhone generated by Safari browser.    Your app does not have access outside of it's own sandbox....it's a privacy thing and a good one too. |

### **Message 3 : iPhone app to intercept TCP traffic**

|  |
| --- |
| **Link** http://www.iphonedevsdk.com/forum/iphone-sdk-development/84140-iphone-app-intercept-tcp-traffic.html **Message** 06-29-2011 Hi,  Subject: iPhone app to intercept TCP traffic  I would like to write an app for iPhone that will run in background and intercept all TCP traffic on iPhone generated by Safari browser.  Is it possible to write such an app? Any relevant links or articles would be much appreciated.  Anyone aware of similar app that runs on normal iPhone (not jailbreak)?  Thanks.  Ambi.  Response 1  no, can't be done with public api.... **Response 2** depending on your exact needs you might be able to write/compile an HTTP proxy which runs on the iPhone and logs traffic, then configure the iPhone to access the net through your local proxy.  I'm not sure if you can accept sockets in a background app however.  Whatever the case there's probably no easy out! **Response 3** yes, can be done but can't go to appstore |

## **4.3 Code**

This section contains code for the following five files.

firstAppViewController.m : This is the main file for business logix.

firstAppViewController.h

firstAppAppDelegate.m

firstAppAppDelegate.h

firstAppViewController.xib: This is the main file for GUI.

### **firstAppViewController.m**

### 

|  |
| --- |
| // firstAppViewController.m  // firstApp  //  // Created by Ambikesh Jayal on 25/05/2011.  // ajayal@glos.ac.uk  // ambi1999@gmail.com  // Copyright 2011 University of Gloucestershire. All rights reserved.  //  #import "firstAppViewController.h"  @implementation firstAppViewController  @synthesize webView, addressBar, activityIndicator;  -(void) printTime{  NSDate\* date=[NSDate date];  NSDateFormatter\* formatter=[[[NSDateFormatter alloc] init] autorelease];  [formatter setDateFormat:@"HH:MM:SS"];    NSString\* str=[formatter stringFromDate:date];    valueOfURLByUser.text=str;  }  -(void) loadView {  valueOfURLByUser.text=@"loading start ";  [super loadView];  valueOfURLByUser.text=@"loading finished ";  }  - (void)viewDidLoad {    [super viewDidLoad];    webView.delegate=self;      /\*  NSString \*urlAddress = @"http://www.o2.co.uk";  NSURL \*url = [NSURL URLWithString:urlAddress];  NSURLRequest \*requestObj = [NSURLRequest requestWithURL:url];    [webView loadRequest:requestObj];  [addressBar setText:urlAddress];  \*/    NSString \*html = @"<html><head><title>Test html page></head><body><h1> Welcome to University of Gloucestershire.</h1> <h1>Please enter the URL in the address bar and click the double arrow button. </h1></body></html>";  [webView loadHTMLString:html baseURL:[NSURL URLWithString:@"http://www.glos.ac.uk"]];  [addressBar setText:@"http://www.glos.ac.uk"];  }  -(void)getAllFramesSrcFromHTMLPageContent:(NSString \*) strAllHTMLContent{      NSError \*error = NULL;    NSRegularExpression \*regex = [NSRegularExpression regularExpressionWithPattern:@"<iframe[^>]\*src=\"(.\*?)\""  options:NSRegularExpressionCaseInsensitive  error:&error];      NSArray \*matches = [regex matchesInString:strAllHTMLContent  options:0  range:NSMakeRange(0, [strAllHTMLContent length])];    NSString \*myString = [NSString stringWithFormat:@"%d", [matches count]];    int count=[matches count];    if(count>0){    //NSLog(@"Iframe count: %@",myString);  //NSLog(@"\n\*\*\*\*\*\*\*\*\*\*\*\* IFrame SOURCES START \n");  for (NSTextCheckingResult \*match in matches) {  NSRange matchRange = [match range];  NSString\* s1=[strAllHTMLContent substringWithRange:matchRange];  NSString \*search = @"src=\"";  NSString \*s2 = [s1 substringFromIndex:NSMaxRange([s1 rangeOfString:search])];  s2=[s2 stringByReplacingOccurrencesOfString:@"\"" withString:@""];  NSLog (@"IFRAME SOURCE: %@", s2);      }    //[regex release];    //NSLog(@"\n \*\*\*\*\*\*\*\*\*\*\*\* IFrame SOURCES END\n ");  }    }  -(void)getAllImageSrcFromHTMLPageContent:(NSString \*) strAllHTMLContent{      NSError \*error = NULL;    NSRegularExpression \*regex = [NSRegularExpression regularExpressionWithPattern:@"<img[^>]\*src=\"(.\*?)\""  options:NSRegularExpressionCaseInsensitive  error:&error];      NSArray \*matches = [regex matchesInString:strAllHTMLContent  options:0  range:NSMakeRange(0, [strAllHTMLContent length])];    NSString \*myString = [NSString stringWithFormat:@"%d", [matches count]];  int count=[matches count];    if(count>0){  //NSLog(@"Image count: %@",myString);  //NSLog(@"\n\*\*\*\*\*\*\*\*\*\*\*\* Image SOURCES START \n ");  for (NSTextCheckingResult \*match in matches) {  NSRange matchRange = [match range];  //NSRange firstHalfRange = [match rangeAtIndex:1];  //NSRange secondHalfRange = [match rangeAtIndex:2];  //NSLog (@"%@", [strAllHTMLContent substringWithRange:matchRange]);    NSString\* s1=[strAllHTMLContent substringWithRange:matchRange];    NSString \*search = @"src=\"";  NSString \*s2 = [s1 substringFromIndex:NSMaxRange([s1 rangeOfString:search])];  s2=[s2 stringByReplacingOccurrencesOfString:@"\"" withString:@""];  NSLog (@"IMAGE SOURCE: %@", s2);      }    //[regex release];    //NSLog(@"\n\*\*\*\*\*\*\*\*\*\*\*\* Image SOURCES END \n ");  }    }  -(IBAction)gotoAddress:(id) sender {      NSLog(@"\n\n DATA Generated by Department of Computing, University of Gloucestershire \n\n");    NSString\* urlByUser=[addressBar text];  NSString\* formattedURLByUser=urlByUser;  if(![urlByUser hasPrefix:@"http"]){  formattedURLByUser=[NSString stringWithFormat:@"%@%@", @"http://", urlByUser];  }    //NSURL \*url = [NSURL URLWithString:[addressBar text]];  NSURL \*url = [NSURL URLWithString:formattedURLByUser];    NSURLRequest \*request = [NSURLRequest requestWithURL:url];    //int validationCode=[self getValidationCodeFromCyberSecurity: request];  //[self processAccordingToValidationCode:validationCode:request];    //Load the request in the UIWebView.  [webView loadRequest:request];  [addressBar resignFirstResponder];    // Create the request.  //NSURLRequest \*theRequest=[NSURLRequest requestWithURL:[NSURL URLWithString:@"http://www.o2.co.uk"]  // cachePolicy:NSURLRequestUseProtocolCachePolicy  // timeoutInterval:60.0];    //NSURLRequest \*theRequest=[NSURLRequest requestWithURL:[NSURL URLWithString:@"http://www.bbcnews.com/"]  // cachePolicy:NSURLRequestUseProtocolCachePolicy  // timeoutInterval:60.0];      //NSURLRequest \*theRequest=[NSURLRequest requestWithURL:[NSURL URLWithString:@"http://www.angelfire.com/super/badwebs/"]  // cachePolicy:NSURLRequestUseProtocolCachePolicy  // timeoutInterval:60.0];  NSURLRequest \*theRequest=[NSURLRequest requestWithURL:url  cachePolicy:NSURLRequestUseProtocolCachePolicy  timeoutInterval:60.0];    //NSHost \*host=[NSHost hostWithName:@"www.cnn.com"];  //NSHost \*host=[NSHost hostWithName:@"www.o2.co.uk"];  NSHost \*host=[NSHost hostWithName:[url host]];  NSArray \*nsArrayIPAddresses =[host addresses];  //NSLog(@"BEFORE LOAD %@", [url absoluteString]);  NSLog(@"BEFORE LOAD %@", urlByUser);    NSDictionary \*nsDictionary=[theRequest allHTTPHeaderFields];  NSArray \*nsArray =[nsDictionary allValues];    valueOfURLByUser.text=@"REQUEST url= {";  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: [[theRequest URL] absoluteString ]];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"}"];    valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @" host= {"];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: [[theRequest URL] host ]];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"}"];      valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @" Header Info ["];      for (id value in nsArray){  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"{"];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: value];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"}"];  }    valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"]"];    // create the connection with the request  // and start loading the data  NSURLConnection \*theConnection=[[NSURLConnection alloc] initWithRequest:theRequest delegate:self];  if (theConnection) {  // Create the NSMutableData to hold the received data.  // receivedData is an instance variable declared elsewhere.  //receivedData = [[NSMutableData data] retain];  } else {  // Inform the user that the connection failed.  }    //valueOfURLByUser.text =[addressBar text];    //NSLog(@"\*\*\*\*\*\*\* End of gotoAddress");      }  -(void) connection:(NSURLConnection \*)connection didReceiveResponse:(NSURLResponse \*)response {    // cast the response to NSHTTPURLResponse so we can look for 404 etc  NSHTTPURLResponse \*httpResponse = (NSHTTPURLResponse \*)response;    //valueOfURLByUser.text =[[httpResponse URL] absoluteString ];  NSDictionary \*nsDictionary=[httpResponse allHeaderFields];  NSArray \*nsArray =[nsDictionary allValues];      //valueOfURLByUser.text=@"url= {";    valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"\n RESPONSE url = {"];      valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: [[httpResponse URL] absoluteString ]];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"}"];    valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @" host= {"];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: [[httpResponse URL] host ]];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"}"];    NSLog(@"DOCUMENT LOADED %@", [[httpResponse URL] absoluteString ]);  NSHost \*host=[NSHost hostWithName:[[httpResponse URL] host ]];  NSArray \*nsArrayIPAddresses =[host addresses];    for (id ipValue in nsArrayIPAddresses){  NSLog(@"IP Addresses %@",ipValue);  }      valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @" Header Info ["];    for (id value in nsArray){  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"{"];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: value];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"}"];  }    valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"]"];    if ([httpResponse statusCode] >= 400) {  // do error handling here  NSLog(@"remote url returned error %d %@",[httpResponse statusCode],[NSHTTPURLResponse localizedStringForStatusCode:[httpResponse statusCode]]);  } else {  // start recieving data  }      //NSLog(@"\*\*\*\*\*\*\* End of didReceiveResponse");  }  - (void)connection:(NSURLConnection \*)connection didReceiveData:(NSData \*)theData{    //[self getAllFrameSrc: [[NSString alloc] initWithData:theData encoding:NSUTF8StringEncoding]];  //[self getAllImageSrc: [[NSString alloc] initWithData:theData encoding:NSUTF8StringEncoding]];  //NSLog(@"Ambi String sent from server %@",[[NSString alloc] initWithData:theData encoding:NSUTF8StringEncoding]);    NSString \*str = [[NSString alloc] initWithData:theData encoding:NSUTF8StringEncoding];  //NSLog(@"Before: %@", str);  //str=[str lowercaseString];  NSString \*str1=[str lowercaseString];  [self getAllFramesSrcFromHTMLPageContent: str1];  //[self getAllImageSrcFromHTMLPageContent: str1];    [str release];    }  -(IBAction) goBack:(id)sender {  [webView goBack];  }  -(IBAction) goForward:(id)sender {  //valueOfURLByUser.text= addressBar.text;  [webView goForward];  //valueOfURLByUser.text=webView.request.URL.absoluteString;  }  int buttonIndexClickedByUser=-9999;  -(int) getValidationCodeFromCyberSecurity: (NSURLRequest\*)request{  [self printTime];  //[self showConfirmAlert];  NSURL \*URL = [request URL];  int cyberSecurityValidationCode=-9;    //code by ambi    NSString \*urlUserWantsToGo=[URL absoluteString];  NSString \*allowedURL1 = @"bbc";  NSString \*allowedURL2 = @"glos";    NSString \*badURL1 = @"rediff";  NSString \*badURL2 = @"dellfake";    if (([urlUserWantsToGo rangeOfString:allowedURL1].location != NSNotFound) || ([urlUserWantsToGo rangeOfString:allowedURL2].location != NSNotFound) )  {  cyberSecurityValidationCode= 0;    }else if (([urlUserWantsToGo rangeOfString:badURL1].location != NSNotFound) || ([urlUserWantsToGo rangeOfString:badURL2].location != NSNotFound) )  {  cyberSecurityValidationCode= 1;    }else {  //not sure about the url  cyberSecurityValidationCode= 2;    }  return cyberSecurityValidationCode;  }  -(void)loadURLAndUpdateAddressBar: (NSURLRequest\*)request{  NSURL \*URL = [request URL];  [addressBar setText:[URL absoluteString]];  [webView loadRequest:request];  [addressBar resignFirstResponder];  }  - (void) processAccordingToValidationCode: (int) cyberSecurityValidationCode: (NSURLRequest\*)request{  NSURL \*URL = [request URL];  valueOfURLByUser.text=[@"URL user wants to go: " stringByAppendingString: [URL absoluteString]];  if(cyberSecurityValidationCode==0){  [self loadURLAndUpdateAddressBar:request];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"\n URL is GOOD"];    }else if(cyberSecurityValidationCode==1){  NSString \*message=@"URL is BAD. Do you still want to proceed Yes or No?";  [self showCyberSecurityAlert: message];    if(buttonIndexClickedByUser==0){  //yes, proceed  [self loadURLAndUpdateAddressBar:request];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"\n URL is BAD \n BUT USER WANTS TO PROCEED"];  }else{  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"\n URL is BAD \n USER DOES NOT WANT TO PROCEED"];  }  }else if(cyberSecurityValidationCode==2){  NSString \*message=@"URL is NOT AUTHENTICATED . Do you still want to proceed Yes or No?";  [self showCyberSecurityAlert: message];  if(buttonIndexClickedByUser==0){  //yes, proceed  [webView loadRequest:request];  [self loadURLAndUpdateAddressBar:request];  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"\n URL is NOT AUTHENTICATED \n USER WANTS TO PROCEED"];  }else{  valueOfURLByUser.text=[valueOfURLByUser.text stringByAppendingString: @"\n URL is NOT AUTHENTICATED \n USER DOES NOT WANT TO PROCEED"];  }    }else{  //do nothing  }  }  - (void)showCyberSecurityAlert: (NSString\* ) message  {  //buttonIndexClickedByUser=-9999;  UIAlertView \*alert = [[UIAlertView alloc] init];  [alert setTitle:@"First CyberSecurity Alert"];  //[alert setMessage:@"URL is INVALID. Do you still want to proceed Yes or No?"];  [alert setMessage:message];  [alert setDelegate:self];  [alert addButtonWithTitle:@"Yes"];  [alert addButtonWithTitle:@"No"];  [alert show];  [alert release];  //valueOfURLByUser.text= [alert delegate];    }  - (void)alertView:(UIAlertView \*)alertView clickedButtonAtIndex:(NSInteger)buttonIndex  {  buttonIndexClickedByUser=buttonIndex;    }  //by ambi  - (BOOL)webView:(UIWebView\*)webView shouldStartLoadWithRequest:(NSURLRequest\*)request navigationType:(UIWebViewNavigationType)navigationType {      //valueOfURLByUser.text=@"1111";          //CAPTURE USER LINK-CLICK.  if (navigationType == UIWebViewNavigationTypeLinkClicked) {  int validationCode=[self getValidationCodeFromCyberSecurity: request];  [self processAccordingToValidationCode:validationCode:request];    //[self printTime];  //valueOfURLByUser.text=@"1111";  //NSURL \*url = [NSURL URLWithString:[addressBar text]];  //NSURLRequest \*requestObj = [NSURLRequest requestWithURL:url];    //Load the request in the UIWebView.  //NSURL \*URL = [request URL];  //[addressBar setText:[URL absoluteString]];  //[webView loadRequest:request];  //[addressBar resignFirstResponder];  return NO;  //7196  /\*  valueOfURLByUser.text=@"1111";  [self gotoAddress:nil];  return NO;    NSURL \*URL = [request URL];    //[self cyberSecurityValidation: request];    //code by ambi  valueOfURLByUser.text=[URL absoluteString];    if ([[URL scheme] isEqualToString:@"http"]) {  [addressBar setText:[URL absoluteString]];    //code by ambi  valueOfURLByUser.text=[URL absoluteString];    [self gotoAddress:nil];  }  \*/  }  return YES;    }  - (void)webViewDidStartLoad:(UIWebView \*)webView {    //[self printTime];  [activityIndicator startAnimating];  //valueOfURLByUser.text=@"1111";  }  - (void)webViewDidFinishLoad:(UIWebView \*)webView {  [activityIndicator stopAnimating];  //[self printTime];  //valueOfURLByUser.text=@"2222";  }  /\*  // Implement viewDidLoad to do additional setup after loading the view, typically from a nib.  - (void)viewDidLoad {  [super viewDidLoad];  }  \*/  /\*  // Override to allow orientations other than the default portrait orientation.  - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {  // Return YES for supported orientations  return (interfaceOrientation == UIInterfaceOrientationPortrait);  }  \*/  - (void)didReceiveMemoryWarning {  [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview  // Release anything that's not essential, such as cached data      }  - (void)dealloc {    [super dealloc];    }  @end |

### **firstAppViewController.h**

|  |
| --- |
| //  // firstAppViewController.h  // firstApp  //  // Created by Ambikesh Jayal on 25/05/2011.  // ajayal@glos.ac.uk  // ambi1999@gmail.com  // Copyright 2011 University of Gloucestershire. All rights reserved.  //  #import <UIKit/UIKit.h>  @interface firstAppViewController : UIViewController<UIWebViewDelegate, UIAlertViewDelegate> {    IBOutlet UIWebView \*webView;  IBOutlet UITextField \*addressBar;  IBOutlet UIActivityIndicatorView \*activityIndicator;  IBOutlet UILabel \*myaddresslable;  IBOutlet UILabel \*valueOfURLByUser;  }  @property(nonatomic,retain) UIWebView \*webView;  @property(nonatomic,retain) UITextField \*addressBar;  @property(nonatomic,retain) UIActivityIndicatorView \*activityIndicator;  -(IBAction) gotoAddress:(id)sender;  -(IBAction) goBack:(id)sender;  -(IBAction) goForward:(id)sender;  -(void) printTime;  -(NSArray\*)getAllFrameSrc:(NSString \*) htmlPageSource;  -(NSArray\*)getAllImageSrc:(NSString \*) htmlPageSource;  @end |

### **firstAppAppDelegate.m**

|  |
| --- |
| //  // firstAppAppDelegate.m  // firstApp  //  // Created by Ambikesh Jayal on 25/05/2011.  // ajayal@glos.ac.uk  // ambi1999@gmail.com  // Copyright 2011 University of Gloucestershire. All rights reserved.  //  #import "firstAppAppDelegate.h"  #import "firstAppViewController.h"  @implementation firstAppAppDelegate  @synthesize window;  @synthesize viewController;  #pragma mark -  #pragma mark Application lifecycle  - (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions {    // Override point for customization after application launch.  // Set the view controller as the window's root view controller and display.  self.window.rootViewController = self.viewController;  //[self.window addSubView:viewController.view];  [self.window makeKeyAndVisible];    return YES;  }  - (void)applicationWillResignActive:(UIApplication \*)application {  /\*  Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application and it begins the transition to the background state.  Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES frame rates. Games should use this method to pause the game.  \*/  }  - (void)applicationDidEnterBackground:(UIApplication \*)application {  /\*  Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its current state in case it is terminated later.  If your application supports background execution, called instead of applicationWillTerminate: when the user quits.  \*/  }  - (void)applicationWillEnterForeground:(UIApplication \*)application {  /\*  Called as part of transition from the background to the inactive state: here you can undo many of the changes made on entering the background.  \*/  }  - (void)applicationDidBecomeActive:(UIApplication \*)application {  /\*  Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh the user interface.  \*/  }  - (void)applicationWillTerminate:(UIApplication \*)application {  /\*  Called when the application is about to terminate.  See also applicationDidEnterBackground:.  \*/  }  #pragma mark -  #pragma mark Memory management  - (void)applicationDidReceiveMemoryWarning:(UIApplication \*)application {  /\*  Free up as much memory as possible by purging cached data objects that can be recreated (or reloaded from disk) later.  \*/  }  - (void)dealloc {  [viewController release];  [window release];  [super dealloc];  }  @end |

### **firstAppAppDelegate.h**

|  |
| --- |
| //  // firstAppAppDelegate.h  // firstApp  //  // Created by Ambikesh Jayal on 25/05/2011.  // ajayal@glos.ac.uk  // ambi1999@gmail.com  // Copyright 2011 University of Gloucestershire. All rights reserved.  //  #import <UIKit/UIKit.h>  @class firstAppViewController;  @interface firstAppAppDelegate : NSObject <UIApplicationDelegate> {  UIWindow \*window;  firstAppViewController \*viewController;    }  @property (nonatomic, retain) IBOutlet UIWindow \*window;  @property (nonatomic, retain) IBOutlet firstAppViewController \*viewController;  @end |

### **firstAppViewController.xib**

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <archive type="com.apple.InterfaceBuilder3.CocoaTouch.XIB" version="7.10">  <data>  <int key="IBDocument.SystemTarget">1056</int>  <string key="IBDocument.SystemVersion">10J869</string>  <string key="IBDocument.InterfaceBuilderVersion">851</string>  <string key="IBDocument.AppKitVersion">1038.35</string>  <string key="IBDocument.HIToolboxVersion">461.00</string>  <object class="NSMutableDictionary" key="IBDocument.PluginVersions">  <string key="NS.key.0">com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>  <string key="NS.object.0">141</string>  </object>  <object class="NSMutableArray" key="IBDocument.EditedObjectIDs">  <bool key="EncodedWithXMLCoder">YES</bool>  <integer value="6"/>  </object>  <object class="NSArray" key="IBDocument.PluginDependencies">  <bool key="EncodedWithXMLCoder">YES</bool>  <string>com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>  </object>  <object class="NSMutableDictionary" key="IBDocument.Metadata">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="NSArray" key="dict.sortedKeys" id="0">  <bool key="EncodedWithXMLCoder">YES</bool>  </object>  <object class="NSMutableArray" key="dict.values">  <bool key="EncodedWithXMLCoder">YES</bool>  </object>  </object>  <object class="NSMutableArray" key="IBDocument.RootObjects" id="1000">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="IBProxyObject" id="372490531">  <string key="IBProxiedObjectIdentifier">IBFilesOwner</string>  <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>  </object>  <object class="IBProxyObject" id="843779117">  <string key="IBProxiedObjectIdentifier">IBFirstResponder</string>  <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>  </object>  <object class="IBUIView" id="774585933">  <reference key="NSNextResponder"/>  <int key="NSvFlags">274</int>  <object class="NSMutableArray" key="NSSubviews">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="IBUIWebView" id="263013782">  <reference key="NSNextResponder" ref="774585933"/>  <int key="NSvFlags">274</int>  <string key="NSFrame">{{0, 30}, {320, 351}}</string>  <reference key="NSSuperview" ref="774585933"/>  <object class="NSColor" key="IBUIBackgroundColor">  <int key="NSColorSpace">1</int>  <bytes key="NSRGB">MSAxIDEAA</bytes>  </object>  <int key="IBUIContentMode">4</int>  <bool key="IBUIMultipleTouchEnabled">YES</bool>  <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>  <bool key="IBUIScalesPageToFit">YES</bool>  <int key="IBUIDataDetectorTypes">1</int>  <bool key="IBUIDetectsPhoneNumbers">YES</bool>  </object>  <object class="IBUIButton" id="964909738">  <reference key="NSNextResponder" ref="774585933"/>  <int key="NSvFlags">292</int>  <string key="NSFrame">{{-2, -2}, {30, 29}}</string>  <reference key="NSSuperview" ref="774585933"/>  <bool key="IBUIOpaque">NO</bool>  <object class="IBUIAccessibilityConfiguration" key="IBUIAccessibilityConfiguration">  <integer value="5" key="IBUIAccessibilityTraits"/>  </object>  <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>  <int key="IBUIContentHorizontalAlignment">0</int>  <int key="IBUIContentVerticalAlignment">0</int>  <object class="NSFont" key="IBUIFont" id="646309731">  <string key="NSName">Helvetica-Bold</string>  <double key="NSSize">10</double>  <int key="NSfFlags">16</int>  </object>  <int key="IBUIButtonType">1</int>  <string key="IBUINormalTitle">&lt;</string>  <object class="NSColor" key="IBUIHighlightedTitleColor" id="201440552">  <int key="NSColorSpace">3</int>  <bytes key="NSWhite">MQA</bytes>  </object>  <object class="NSColor" key="IBUINormalTitleColor">  <int key="NSColorSpace">1</int>  <bytes key="NSRGB">MC4xOTYwNzg0MzQ2IDAuMzA5ODAzOTMyOSAwLjUyMTU2ODY1NgA</bytes>  </object>  <object class="NSColor" key="IBUINormalTitleShadowColor" id="229994396">  <int key="NSColorSpace">3</int>  <bytes key="NSWhite">MC41AA</bytes>  </object>  </object>  <object class="IBUIButton" id="414693114">  <reference key="NSNextResponder" ref="774585933"/>  <int key="NSvFlags">292</int>  <string key="NSFrame">{{27, -2}, {28, 29}}</string>  <reference key="NSSuperview" ref="774585933"/>  <bool key="IBUIOpaque">NO</bool>  <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>  <int key="IBUIContentHorizontalAlignment">0</int>  <int key="IBUIContentVerticalAlignment">0</int>  <reference key="IBUIFont" ref="646309731"/>  <int key="IBUIButtonType">1</int>  <string key="IBUINormalTitle">&gt;</string>  <reference key="IBUIHighlightedTitleColor" ref="201440552"/>  <object class="NSColor" key="IBUINormalTitleColor">  <int key="NSColorSpace">1</int>  <bytes key="NSRGB">MC4xOTYwNzg0MzQ2IDAuMzA5ODAzOTMyOSAwLjUyMTU2ODY1NgA</bytes>  </object>  <reference key="IBUINormalTitleShadowColor" ref="229994396"/>  </object>  <object class="IBUITextField" id="215964141">  <reference key="NSNextResponder" ref="774585933"/>  <int key="NSvFlags">292</int>  <string key="NSFrame">{{59, -3}, {221, 31}}</string>  <reference key="NSSuperview" ref="774585933"/>  <bool key="IBUIOpaque">NO</bool>  <bool key="IBUIClipsSubviews">YES</bool>  <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>  <int key="IBUIContentVerticalAlignment">0</int>  <string key="IBUIText"/>  <int key="IBUIBorderStyle">3</int>  <object class="NSColor" key="IBUITextColor">  <int key="NSColorSpace">3</int>  <bytes key="NSWhite">MAA</bytes>  <object class="NSColorSpace" key="NSCustomColorSpace" id="505753881">  <int key="NSID">2</int>  </object>  </object>  <object class="NSFont" key="IBUIFont">  <string key="NSName">Helvetica</string>  <double key="NSSize">10</double>  <int key="NSfFlags">16</int>  </object>  <bool key="IBUIAdjustsFontSizeToFit">YES</bool>  <float key="IBUIMinimumFontSize">17</float>  <object class="IBUITextInputTraits" key="IBUITextInputTraits">  <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>  </object>  </object>  <object class="IBUILabel" id="300530103">  <reference key="NSNextResponder" ref="774585933"/>  <int key="NSvFlags">292</int>  <string key="NSFrame">{{0, 379}, {320, 81}}</string>  <reference key="NSSuperview" ref="774585933"/>  <bool key="IBUIOpaque">NO</bool>  <bool key="IBUIClipsSubviews">YES</bool>  <int key="IBUIContentMode">7</int>  <bool key="IBUIUserInteractionEnabled">NO</bool>  <object class="IBUIAccessibilityConfiguration" key="IBUIAccessibilityConfiguration">  <integer value="0" key="IBUIAccessibilityTraits"/>  </object>  <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>  <string key="IBUIText">Label</string>  <object class="NSFont" key="IBUIFont">  <string key="NSName">Helvetica</string>  <double key="NSSize">12</double>  <int key="NSfFlags">16</int>  </object>  <object class="NSColor" key="IBUITextColor">  <int key="NSColorSpace">1</int>  <bytes key="NSRGB">MCAwIDAAA</bytes>  </object>  <reference key="IBUIHighlightedColor" ref="201440552"/>  <int key="IBUIBaselineAdjustment">1</int>  <float key="IBUIMinimumFontSize">10</float>  <int key="IBUINumberOfLines">5</int>  <int key="IBUILineBreakMode">1</int>  </object>  <object class="IBUIButton" id="937651243">  <reference key="NSNextResponder" ref="774585933"/>  <int key="NSvFlags">292</int>  <string key="NSFrame">{{281, 0}, {39, 28}}</string>  <reference key="NSSuperview" ref="774585933"/>  <bool key="IBUIOpaque">NO</bool>  <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>  <int key="IBUIContentHorizontalAlignment">0</int>  <int key="IBUIContentVerticalAlignment">0</int>  <reference key="IBUIFont" ref="646309731"/>  <int key="IBUIButtonType">1</int>  <string key="IBUINormalTitle">&gt;&gt;</string>  <reference key="IBUIHighlightedTitleColor" ref="201440552"/>  <object class="NSColor" key="IBUINormalTitleColor">  <int key="NSColorSpace">1</int>  <bytes key="NSRGB">MC4xOTYwNzg0MzQ2IDAuMzA5ODAzOTMyOSAwLjUyMTU2ODY1NgA</bytes>  </object>  <reference key="IBUINormalTitleShadowColor" ref="229994396"/>  </object>  </object>  <string key="NSFrameSize">{320, 460}</string>  <reference key="NSSuperview"/>  <object class="NSColor" key="IBUIBackgroundColor">  <int key="NSColorSpace">3</int>  <bytes key="NSWhite">MC43NQA</bytes>  <reference key="NSCustomColorSpace" ref="505753881"/>  </object>  <bool key="IBUIClearsContextBeforeDrawing">NO</bool>  <object class="IBUISimulatedStatusBarMetrics" key="IBUISimulatedStatusBarMetrics"/>  <string key="targetRuntimeIdentifier">IBCocoaTouchFramework</string>  </object>  </object>  <object class="IBObjectContainer" key="IBDocument.Objects">  <object class="NSMutableArray" key="connectionRecords">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="IBConnectionRecord">  <object class="IBCocoaTouchOutletConnection" key="connection">  <string key="label">view</string>  <reference key="source" ref="372490531"/>  <reference key="destination" ref="774585933"/>  </object>  <int key="connectionID">7</int>  </object>  <object class="IBConnectionRecord">  <object class="IBCocoaTouchOutletConnection" key="connection">  <string key="label">webView</string>  <reference key="source" ref="372490531"/>  <reference key="destination" ref="263013782"/>  </object>  <int key="connectionID">12</int>  </object>  <object class="IBConnectionRecord">  <object class="IBCocoaTouchEventConnection" key="connection">  <string key="label">goBack:</string>  <reference key="source" ref="964909738"/>  <reference key="destination" ref="372490531"/>  <int key="IBEventType">1</int>  </object>  <int key="connectionID">17</int>  </object>  <object class="IBConnectionRecord">  <object class="IBCocoaTouchEventConnection" key="connection">  <string key="label">goForward:</string>  <reference key="source" ref="414693114"/>  <reference key="destination" ref="372490531"/>  <int key="IBEventType">1</int>  </object>  <int key="connectionID">18</int>  </object>  <object class="IBConnectionRecord">  <object class="IBCocoaTouchOutletConnection" key="connection">  <string key="label">addressBar</string>  <reference key="source" ref="372490531"/>  <reference key="destination" ref="215964141"/>  </object>  <int key="connectionID">22</int>  </object>  <object class="IBConnectionRecord">  <object class="IBCocoaTouchOutletConnection" key="connection">  <string key="label">valueOfURLByUser</string>  <reference key="source" ref="372490531"/>  <reference key="destination" ref="300530103"/>  </object>  <int key="connectionID">26</int>  </object>  <object class="IBConnectionRecord">  <object class="IBCocoaTouchEventConnection" key="connection">  <string key="label">gotoAddress:</string>  <reference key="source" ref="937651243"/>  <reference key="destination" ref="372490531"/>  <int key="IBEventType">1</int>  </object>  <int key="connectionID">29</int>  </object>  <object class="IBConnectionRecord">  <object class="IBCocoaTouchOutletConnection" key="connection">  <string key="label">delegate</string>  <reference key="source" ref="263013782"/>  <reference key="destination" ref="774585933"/>  </object>  <int key="connectionID">32</int>  </object>  </object>  <object class="IBMutableOrderedSet" key="objectRecords">  <object class="NSArray" key="orderedObjects">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="IBObjectRecord">  <int key="objectID">0</int>  <reference key="object" ref="0"/>  <reference key="children" ref="1000"/>  <nil key="parent"/>  </object>  <object class="IBObjectRecord">  <int key="objectID">-1</int>  <reference key="object" ref="372490531"/>  <reference key="parent" ref="0"/>  <string key="objectName">File's Owner</string>  </object>  <object class="IBObjectRecord">  <int key="objectID">-2</int>  <reference key="object" ref="843779117"/>  <reference key="parent" ref="0"/>  </object>  <object class="IBObjectRecord">  <int key="objectID">6</int>  <reference key="object" ref="774585933"/>  <object class="NSMutableArray" key="children">  <bool key="EncodedWithXMLCoder">YES</bool>  <reference ref="964909738"/>  <reference ref="414693114"/>  <reference ref="215964141"/>  <reference ref="937651243"/>  <reference ref="263013782"/>  <reference ref="300530103"/>  </object>  <reference key="parent" ref="0"/>  </object>  <object class="IBObjectRecord">  <int key="objectID">11</int>  <reference key="object" ref="263013782"/>  <reference key="parent" ref="774585933"/>  </object>  <object class="IBObjectRecord">  <int key="objectID">15</int>  <reference key="object" ref="964909738"/>  <reference key="parent" ref="774585933"/>  </object>  <object class="IBObjectRecord">  <int key="objectID">16</int>  <reference key="object" ref="414693114"/>  <reference key="parent" ref="774585933"/>  </object>  <object class="IBObjectRecord">  <int key="objectID">21</int>  <reference key="object" ref="215964141"/>  <reference key="parent" ref="774585933"/>  </object>  <object class="IBObjectRecord">  <int key="objectID">25</int>  <reference key="object" ref="300530103"/>  <reference key="parent" ref="774585933"/>  </object>  <object class="IBObjectRecord">  <int key="objectID">27</int>  <reference key="object" ref="937651243"/>  <reference key="parent" ref="774585933"/>  </object>  </object>  </object>  <object class="NSMutableDictionary" key="flattenedProperties">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="NSArray" key="dict.sortedKeys">  <bool key="EncodedWithXMLCoder">YES</bool>  <string>-1.CustomClassName</string>  <string>-2.CustomClassName</string>  <string>11.IBPluginDependency</string>  <string>11.IBPropertyAccessControl</string>  <string>11.IBViewBoundsToFrameTransform</string>  <string>15.IBPluginDependency</string>  <string>15.IBViewBoundsToFrameTransform</string>  <string>16.IBPluginDependency</string>  <string>16.IBViewBoundsToFrameTransform</string>  <string>21.IBPluginDependency</string>  <string>21.IBViewBoundsToFrameTransform</string>  <string>25.IBPluginDependency</string>  <string>25.IBViewBoundsToFrameTransform</string>  <string>27.IBPluginDependency</string>  <string>27.IBViewBoundsToFrameTransform</string>  <string>6.IBEditorWindowLastContentRect</string>  <string>6.IBPluginDependency</string>  </object>  <object class="NSMutableArray" key="dict.values">  <bool key="EncodedWithXMLCoder">YES</bool>  <string>firstAppViewController</string>  <string>UIResponder</string>  <string>com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>  <integer value="3"/>  <object class="NSAffineTransform">  <bytes key="NSTransformStruct">P4AAAL+AAAAAAAAAw7wAAA</bytes>  </object>  <string>com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>  <object class="NSAffineTransform">  <bytes key="NSTransformStruct">P4AAAL+AAAAAAAAAwjwAAA</bytes>  </object>  <string>com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>  <object class="NSAffineTransform">  <bytes key="NSTransformStruct">P4AAAL+AAABCEAAAwdgAAA</bytes>  </object>  <string>com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>  <object class="NSAffineTransform">  <bytes key="NSTransformStruct">P4AAAL+AAABCbAAAwngAAA</bytes>  </object>  <string>com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>  <object class="NSAffineTransform">  <bytes key="NSTransformStruct">P4AAAL+AAAAAAAAAw+UAAA</bytes>  </object>  <string>com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>  <object class="NSAffineTransform">  <bytes key="NSTransformStruct">P4AAAL+AAABDgoAAwdAAAA</bytes>  </object>  <string>{{494, 174}, {320, 480}}</string>  <string>com.apple.InterfaceBuilder.IBCocoaTouchPlugin</string>  </object>  </object>  <object class="NSMutableDictionary" key="unlocalizedProperties">  <bool key="EncodedWithXMLCoder">YES</bool>  <reference key="dict.sortedKeys" ref="0"/>  <object class="NSMutableArray" key="dict.values">  <bool key="EncodedWithXMLCoder">YES</bool>  </object>  </object>  <nil key="activeLocalization"/>  <object class="NSMutableDictionary" key="localizations">  <bool key="EncodedWithXMLCoder">YES</bool>  <reference key="dict.sortedKeys" ref="0"/>  <object class="NSMutableArray" key="dict.values">  <bool key="EncodedWithXMLCoder">YES</bool>  </object>  </object>  <nil key="sourceID"/>  <int key="maxID">32</int>  </object>  <object class="IBClassDescriber" key="IBDocument.Classes">  <object class="NSMutableArray" key="referencedPartialClassDescriptions">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="IBPartialClassDescription">  <string key="className">firstAppViewController</string>  <string key="superclassName">UIViewController</string>  <object class="NSMutableDictionary" key="actions">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="NSArray" key="dict.sortedKeys">  <bool key="EncodedWithXMLCoder">YES</bool>  <string>goBack:</string>  <string>goForward:</string>  <string>gotoAddress:</string>  </object>  <object class="NSMutableArray" key="dict.values">  <bool key="EncodedWithXMLCoder">YES</bool>  <string>id</string>  <string>id</string>  <string>id</string>  </object>  </object>  <object class="NSMutableDictionary" key="actionInfosByName">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="NSArray" key="dict.sortedKeys">  <bool key="EncodedWithXMLCoder">YES</bool>  <string>goBack:</string>  <string>goForward:</string>  <string>gotoAddress:</string>  </object>  <object class="NSMutableArray" key="dict.values">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="IBActionInfo">  <string key="name">goBack:</string>  <string key="candidateClassName">id</string>  </object>  <object class="IBActionInfo">  <string key="name">goForward:</string>  <string key="candidateClassName">id</string>  </object>  <object class="IBActionInfo">  <string key="name">gotoAddress:</string>  <string key="candidateClassName">id</string>  </object>  </object>  </object>  <object class="NSMutableDictionary" key="outlets">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="NSArray" key="dict.sortedKeys">  <bool key="EncodedWithXMLCoder">YES</bool>  <string>activityIndicator</string>  <string>addressBar</string>  <string>myaddresslable</string>  <string>valueOfURLByUser</string>  <string>webView</string>  </object>  <object class="NSMutableArray" key="dict.values">  <bool key="EncodedWithXMLCoder">YES</bool>  <string>UIActivityIndicatorView</string>  <string>UITextField</string>  <string>UILabel</string>  <string>UILabel</string>  <string>UIWebView</string>  </object>  </object>  <object class="NSMutableDictionary" key="toOneOutletInfosByName">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="NSArray" key="dict.sortedKeys">  <bool key="EncodedWithXMLCoder">YES</bool>  <string>activityIndicator</string>  <string>addressBar</string>  <string>myaddresslable</string>  <string>valueOfURLByUser</string>  <string>webView</string>  </object>  <object class="NSMutableArray" key="dict.values">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="IBToOneOutletInfo">  <string key="name">activityIndicator</string>  <string key="candidateClassName">UIActivityIndicatorView</string>  </object>  <object class="IBToOneOutletInfo">  <string key="name">addressBar</string>  <string key="candidateClassName">UITextField</string>  </object>  <object class="IBToOneOutletInfo">  <string key="name">myaddresslable</string>  <string key="candidateClassName">UILabel</string>  </object>  <object class="IBToOneOutletInfo">  <string key="name">valueOfURLByUser</string>  <string key="candidateClassName">UILabel</string>  </object>  <object class="IBToOneOutletInfo">  <string key="name">webView</string>  <string key="candidateClassName">UIWebView</string>  </object>  </object>  </object>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBProjectSource</string>  <string key="minorKey">Classes/firstAppViewController.h</string>  </object>  </object>  </object>  <object class="NSMutableArray" key="referencedPartialClassDescriptionsV3.2+">  <bool key="EncodedWithXMLCoder">YES</bool>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">Foundation.framework/Headers/NSError.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">Foundation.framework/Headers/NSFileManager.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">Foundation.framework/Headers/NSKeyValueCoding.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">Foundation.framework/Headers/NSKeyValueObserving.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">Foundation.framework/Headers/NSKeyedArchiver.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">Foundation.framework/Headers/NSObject.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">Foundation.framework/Headers/NSRunLoop.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">Foundation.framework/Headers/NSThread.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">Foundation.framework/Headers/NSURL.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">Foundation.framework/Headers/NSURLConnection.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UIAccessibility.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UINibLoading.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier" id="757709063">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UIResponder.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIActivityIndicatorView</string>  <string key="superclassName">UIView</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UIActivityIndicatorView.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIButton</string>  <string key="superclassName">UIControl</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UIButton.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIControl</string>  <string key="superclassName">UIView</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UIControl.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UILabel</string>  <string key="superclassName">UIView</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UILabel.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIResponder</string>  <string key="superclassName">NSObject</string>  <reference key="sourceIdentifier" ref="757709063"/>  </object>  <object class="IBPartialClassDescription">  <string key="className">UISearchBar</string>  <string key="superclassName">UIView</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UISearchBar.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UISearchDisplayController</string>  <string key="superclassName">NSObject</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UISearchDisplayController.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UITextField</string>  <string key="superclassName">UIControl</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier" id="113000848">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UITextField.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIView</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UIPrintFormatter.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIView</string>  <reference key="sourceIdentifier" ref="113000848"/>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIView</string>  <string key="superclassName">UIResponder</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UIView.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIViewController</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UINavigationController.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIViewController</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UIPopoverController.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIViewController</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UISplitViewController.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIViewController</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UITabBarController.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIViewController</string>  <string key="superclassName">UIResponder</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UIViewController.h</string>  </object>  </object>  <object class="IBPartialClassDescription">  <string key="className">UIWebView</string>  <string key="superclassName">UIView</string>  <object class="IBClassDescriptionSource" key="sourceIdentifier">  <string key="majorKey">IBFrameworkSource</string>  <string key="minorKey">UIKit.framework/Headers/UIWebView.h</string>  </object>  </object>  </object>  </object>  <int key="IBDocument.localizationMode">0</int>  <string key="IBDocument.TargetRuntimeIdentifier">IBCocoaTouchFramework</string>  <object class="NSMutableDictionary" key="IBDocument.PluginDeclaredDependencyDefaults">  <string key="NS.key.0">com.apple.InterfaceBuilder.CocoaTouchPlugin.iPhoneOS</string>  <integer value="1056" key="NS.object.0"/>  </object>  <object class="NSMutableDictionary" key="IBDocument.PluginDeclaredDevelopmentDependencies">  <string key="NS.key.0">com.apple.InterfaceBuilder.CocoaTouchPlugin.InterfaceBuilder3</string>  <integer value="3100" key="NS.object.0"/>  </object>  <bool key="IBDocument.PluginDeclaredDependenciesTrackSystemTargetVersion">YES</bool>  <string key="IBDocument.LastKnownRelativeProjectPath">firstApp.xcodeproj</string>  <int key="IBDocument.defaultPropertyAccessControl">3</int>  <string key="IBCocoaTouchPluginVersion">141</string>  </data>  </archive> |