System Documentation

Team 3

March 2, 2020

Team Members

Pavan Andrea

Micheloni Mirko

Vidrascu Andrea

Zuccolotto Devis

Contents

Revision History

Version	Date	Name	Description
1	12/03/20	Vidrascu Adrian, Zuccolotto Devis	Initial Document
2	28/03/20	Pavan Andrea	Added maintenance information on proxy code
3	18/04/20	Micheloni Mirko	Final measures

Introduction

The SF-Agricolture application allows users a simple way to manage their fields using an account. This document will provide instructions for accessing the application code.

Setup Proxy Server

A Unix/Linux based server with internet connectivity is required for the web proxy. Proxy server code can be obtained in the SVN repository located at:
http://134.133.118.8/svn/cs451/branches/team3/proxycode/roobucks.pl

1. Obtain the current Perl distrubution
    1. Install perl modules from CPAN
        1. LWP::UserAgent
            2. HTTP::Request::Common
2. Obtain the current Apache distribution
    1. Configure apache with https support
        1. Configure apache with cgi support
        2. Place perl proxy software in the designated cgi-bin directory

Access iOS Application Code

Access to the internet and the MARCONI SVN revision control repository used to store the code is required. MARCONI internet access privileges are required to access the repository.

All code for the iOS platform is found in the SVN repository located at:

http://134.133.118.8/svn/cs451/branches/team3/roobucks/

Install on Simulator or Device

Required Components

Access to a computer with the iOS SDK is required in order to use Xcode and the simulator.

An iOS-compatible device is required to run application outside of the simulator. Examples include the iPhone, iPod Touch, and iPad.

Install Code

1. Launch Xcode
2. Configure SCM to access the code in the repository.
    1. Click on SCM → Configure SCM
    2. Click '+' to add new repository
    3. Name the repository RooBucks
    4. Enter URL of repository given in section 3
    5. Apply changes
3. Load code for the project
    1. Click SCM-\> Repository

    2. Select RooBucks and Checkout
4. Open project "RooBucks.xcodeproj" from checked-out directory in Xcode.
5. Ensure that the drop-down menu in the top-left corner of the project window is set to the "Simulator" target OR that a suitable device and provisioning data are present.
6. Choose Build --\> Build and Go in Xcode.
7. App is now running and usable.

System Maintenance

Objective-C Code

This application is builted on a MVC template, so there are three main folder: Models, Controllers and Views. To documentation of this template can be found on: https://docs.microsoft.com/it-it/aspnet/core/mvc/overview?view=aspnetcore-3.1 .

The user interface consists of five tabs for viewing information. Each of these is implemented in one cshtml file and only one controller class.

On the first run, the application displays the Index page where the visitor can see some information about the application and his goals. Once the user has logged in, the view is switched to the tab bar, which is displayed at the bottom of the screen. The tab bar view enables the user to navigate between the tabs mentioned above, and hosts the four subviews for each type of information.

The logic in each view controller class mainly performs the task of pulling the appropriate data from XMLParser, formatting it, and displaying it in the correct fields. The login controller class also contains logic to store the user's credentials in the system Keychain.

Perl Proxy Code

The proxy server consists of a perl cgi, executed by apache when a specific POST is received via https. The perl code logs into managemyid using the user's credentials (provided in POST). From there, the code captures an https re-direct, and uses the unique URL in the re-direct to request account balance and account history. This is accomplished in three separate https transactions with managemyid.

As information is gleaned from the second two tranactions with managemyid, the perl script generates XML, and sends it to the objective-C parser on the iOS device through the (still open) tcp stream.

When the parsing and XML generation is complete, the tcp session is closed, and the perl script exits. No passwords or user data is retained on the proxy server.