System Documentation

Red Eyes Black Dragons

December 4, 2021

Team Members

Blake Simpson
Renny Soto
Daanyaal Tariq
Jacob Anderson
Andrew Aeilts
Lila Shelton

Contents

Contents 2
1 Introduction - 4
2 Setup Proxy Server - 4
3 Access iOS Application Code - 4
4 Install on Simulator or Device - 4
5 System Maintenance - 5

Revision History

Version	Date	Name	Description
1	12/04/21	Red Eyes Black Dragons	Initial Document

1 Introduction -

The Commerce Banking application allows users to manage their Commerce Bank Account. The application gives users a way to login/signup, view/add transactions, as well as setting notification rules.

2 Netlify hosting -

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://134.193.128.7/svn/cs451/branches/team1/proxy_code/roobucks.pl

1 Obtain the current Perl distrubution

1.1 Install perl modules from CPAN

1.1.1 LWP::UserAgent

1.1.2 HTTP::Request::Common

2 Obtain the current Apache distribution

2.1 Configure apache with https support

2.1.1 Configure apache with cgi support

2.1.1.1 Place perl proxy software in the designated cgi-bin directory

3 Access Github code -

Access to the internet and the Red-Eyes Black Dragons Github revision control repository used to store the code is required. Administrator granted access privileges are required to access the repository.

All code for the project is found in the Github repository located at:
https://github.com/Software-Engineering-Capstone/Commerce-Banking-App

4 Install on Simulator or Device - Danny - Cloning from github and installing dependencies with NPM

4.1 Required Components

Access to a computer with Node.js and npm installed is required to simulate this application. The computer must also have Git installed and configured.

4.2 Install Code

1. Navigate to Github repository where Application is located.
2. Clone repository to local device
    2.1. Copy url from Github repository
    2.2. Navigate to desired directory on computer, and open a Git Bash terminal
    2.3. Enter in terminal: <<git clone [url]>> to clone to device
    2.4. Enter in terminal: <<npm install react>> to install dependencies
3. Open project in code editor
4. Run <<npm start>> in terminal to start a local host and view project in browser

5 System Maintenance -

5.1 Objective-C Code

The user interface consists of four tabs for viewing information, plus the login screen. Each of these is implemented in one XIB file and one controller class. There are two additional XIB files: the application root view ("MainView.xib") and tab bar navigation view ("RooBucksTabs.xib").

On the first run, the application displays the login screen. 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.

5.2 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.