##### USER GUIDE

##### Florida International University

Senior Project

CIS 4911

12/8/2014

Mentor:

Tracy Beeson

Development Team:

Ariel Diaz

Yesenia Yser

Copyright © 2014 by Ariel Diaz and Yesenia Yser

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed “Attention: Permissions Coordinator,” at the address below.

Florida International University

College of Computer Science

11200 SW 8th Street

Miami, Fl. 33199

Table of Contents

[1.0 INTRODUCTION 3](#_Toc405725842)

[2.0 HARDWARE AND SOFTWARE REQUIREMENTS 4](#_Toc405725843)

[2.1 Software 4](#_Toc405725844)

[2.2 Hardware 5](#_Toc405725845)

[3.0 INSTALLATION AND SETUP 6](#_Toc405725846)

[3.1 Android Setup 6](#_Toc405725847)

[3.2 Web Application Setup 7](#_Toc405725848)

[4.0 GETTING STARTED 8](#_Toc405725849)

[4.1 Android Setup 8](#_Toc405725850)

[*4.1.1 How to run the application* 8](#_Toc405725851)

[*4.1.2 Use Case 1* 9](#_Toc405725852)

[*4.1.3 Use Case 2* 9](#_Toc405725853)

[4.2 Web Application Setup 9](#_Toc405725854)

[*4.2.1 How to run the application* 9](#_Toc405725855)

[*4.2.2 Use Case 1* 9](#_Toc405725856)

[*4.2.3 Use Case 2* 9](#_Toc405725857)

[5.0 QUICK REFERENCE 10](#_Toc405725858)

[6.0 ACCESS ONLINE REFERENCES 11](#_Toc405725859)

[8.0 REFERENCES 12](#_Toc405725860)

# INTRODUCTION

The User Guide contains all essential information for the user to make full use of History Explorer. This guide includes Environment setup for Android Development and the Web Application Development.

# HARDWARE AND SOFTWARE REQUIREMENTS

In this chapter, software and hardware used during the overall process of History Explorer are shown. Each tool has a brief description of how it was used and includes a version number is available.

## 2.1 Software

**Trello**: Each individual is granted access to the Senior Project’s Trello account to organize the software development process and to identify the projects current state.

**Github**: Each individual is granted access to the Senior Project’s project repository to have one centralized location to coordinate on the development.

**StarUML 2 v2.0.0-beta12**: For the creation of UML diagrams, use cases, and sequence diagrams.

**Android Studio Beta v0.8.14**: For the development of the Android application and as well as testing and emulation of the device

**Google drive**: Each individual is granted access to a shared Senior Project folder, which holds documentation, presentations, UML diagrams, images, and project associated data is kept.

**Google hangouts**: a video chat method to communicate ideas and discuss current project status and assign task.

**Webcam and microphone**: Used during the google hangouts video chats and provides ease of communication

**Dropbox**: Each individual is granted access to a shared Senior Project folder, which holds the final draft of the documents and presentations due to formatting issues with Google Drive.

**Microsoft Office Suite 2010/2013**: For the creation, modification, and reviewing documents and presentations.

**Microsoft Project 2013**: For the creation, modification and reviewing Project scheduling documents, such as GANTT Charts.

**IE/Firefox/Chrome**: Used to access the course, and testing the web application

**Gmail**: For the creation of GitHub, Trello, google drive, and drop box account, as well as communication between partners, mentor and professor.

**Appium v1.3.4**: Mobile Device Test Automation; used during subsystem and system testing of Android application

**Selenium v2.8.0**: Browser Automation used during subsystem and system testing of Web Application

**NodeJS v0.10.33:** Used by appium to run the automated test

**Java SE 8:** Android application is written in Java

**Phpmyadmin v4.3.0**: Administration of MySQL over the web.

**Google Maps V2 API:** API used for the map section of each application

**PHP 5.3.0:** Scripting language used to create web services

**JSON 3:** Content-type of the web services

**Putty**: Used to ssh into the server where the web application is hosted

**FileZilla 3.9.0**: File transfer GUI used to insert media files quickly and easily

**Photoshop 1.14 CS6**: Used to mock up the original design of the application

## 2.2 Hardware

**PC**: Windows 7 or 8, Mac OSX, minimum of 4GB of RAM, 100GB of hard drive memory space, and minimum of 1.86GHz pc spec

**Android device**: Meant for testing the app in a real device. Device: Samsung Galaxy S4; Minimum SDK: Android 4.3

# INSTALLATION AND SETUP

Below are the tools used during our development process of History Explorer. Using other tools or simply the command line is your choice. The latest version of each tool was used during our installation and is labeled so the same version can be used; attempting to use another version may cause side-effects.

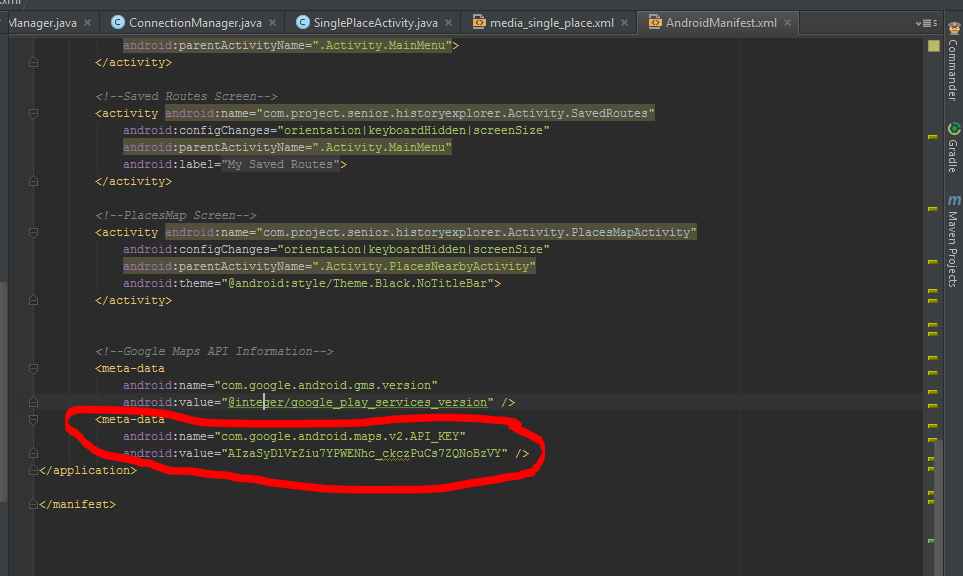
1. Download a telnet/ssh client to remote into the server: (optional) <http://www.putty.org/>
2. Download a FTP client: (optional) <https://filezilla-project.org/>
3. Install Github GUI: <https://windows.github.com/>
4. Install Git Bash: (optional) http://git-scm.com/downloads

## 3.1 Android Setup

* Install NodeJS v0.10.33: <http://nodejs.org/>
* Install Android Studio v0.8.14: <https://developer.android.com/sdk/installing/studio.html>
* Follow the instructions on Android Studio to setup the environment.
* Open a command prompt/git bash/terminal and clone the github directory into your preferred directory. All production was done under the Dev branch. ‘git clone –b Dev <https://github.com/FIU-SCIS-Senior-Project-2014-Fall/History-App.git>’
* Start Android Studio and open the following directory Code/Source Files/Android app/History Explorer
* Select Tools/Android/SDK Manager and install all contents of the following SDK Packages:
  + Tools
  + API 17-20
  + Extra
* On your Android Device, make sure you have Developer Options in the Settings with USB Debug enabled. For more info, <http://developer.android.com/tools/device.html>
* In git bash/command prompt/ terminal, you are able to see which devices are connecting but typing ‘adb devices’ this starts the adb server and will show UDID and Availability of the devices (Offline, Connected, etc). For more info, <http://developer.android.com/tools/help/adb.html>
* Retrieve a Google Maps API Key. FIU email address was used as the account for the API Key

<https://developers.google.com/maps/documentation/android/start#get_an_android_certificate_and_the_google_maps_api_key>

* Site to enable API Key. Make sure you are logged on first: <https://console.developers.google.com/project/395015670207/apiui/api?authuser=0>
* Create an OAuth 2.0 API Key. <https://console.developers.google.com/project/395015670207/apiui/credential?authuser=0>
* Once you receive the API Key, replace the key within the Android Solution. This can be found in “HistoryExplore/app/src/AndroidManifest.xml” in the highlighted area shown below



* If the API Key is not changed, there will be an issue trying to connect to the Google Maps on the Android Application.

## 3.2 Web Application Setup

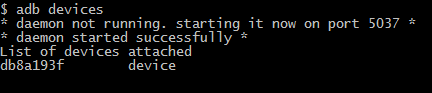
# GETTING STARTED

In this chapter, we were show how to run the application and give two specific use case scenarios as examples. Any installation or environment configuration will be discussed as well. This chapter is divided into each individual system since they both require a different setup.

## 4.1 Android Setup

To run the Android application there are a few pre-conditions that need to be satisfied before installing the apk.

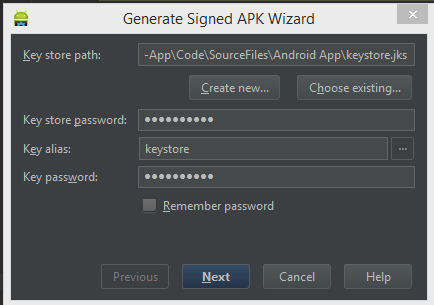
1. Make sure your Android device has Android OS 4.3 or higher
2. Make sure your Android Device is connected via USB to the main computer.
3. Open Android Studio with the History Explorer project open.
4. Open a terminal/command prompt/git bash, and run ‘adb devices’, you should see the following, but with a different UDID. This means your device is connected properly and ready for commands.



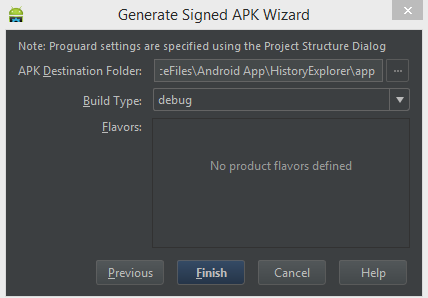
### *4.1.1 How to run the application*

Due to Google Map being used on the application, the Emulator cannot be used to test the Google Map portion of the application. Therefore, an apk must be generated for use. Below are instructions on how to generate the apk and install it on your Android Device.

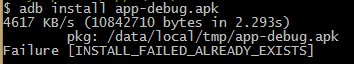
1. In Android Studio, select Build and ‘Generate signed APK’
2. Go to the keystory directory, ‘../Code/Source Files/Android App/keystory.jks’
3. Both key store password and key password are historyapp
4. Key alias is keystore



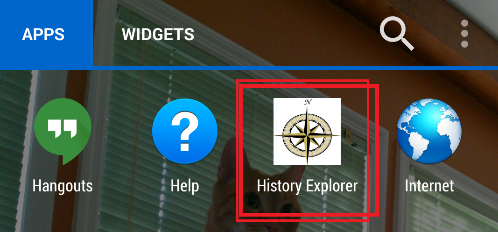
1. Select next. Build type should be debug, APK Destination Folder should be ‘../Android App/HistoryExplorer/app’. Click Finish.



1. In your command prompt/terminal/ git bash, cd to the APK Destination folder
2. Type ‘adb install app-debug.apk’ and enter. The application will install onto your Android Device and return the statement Success.
3. If you receive the following error, make sure to delete the application from your device before reinstalling. There is a command for removing before installing but I had less issues if I just removed it manually.



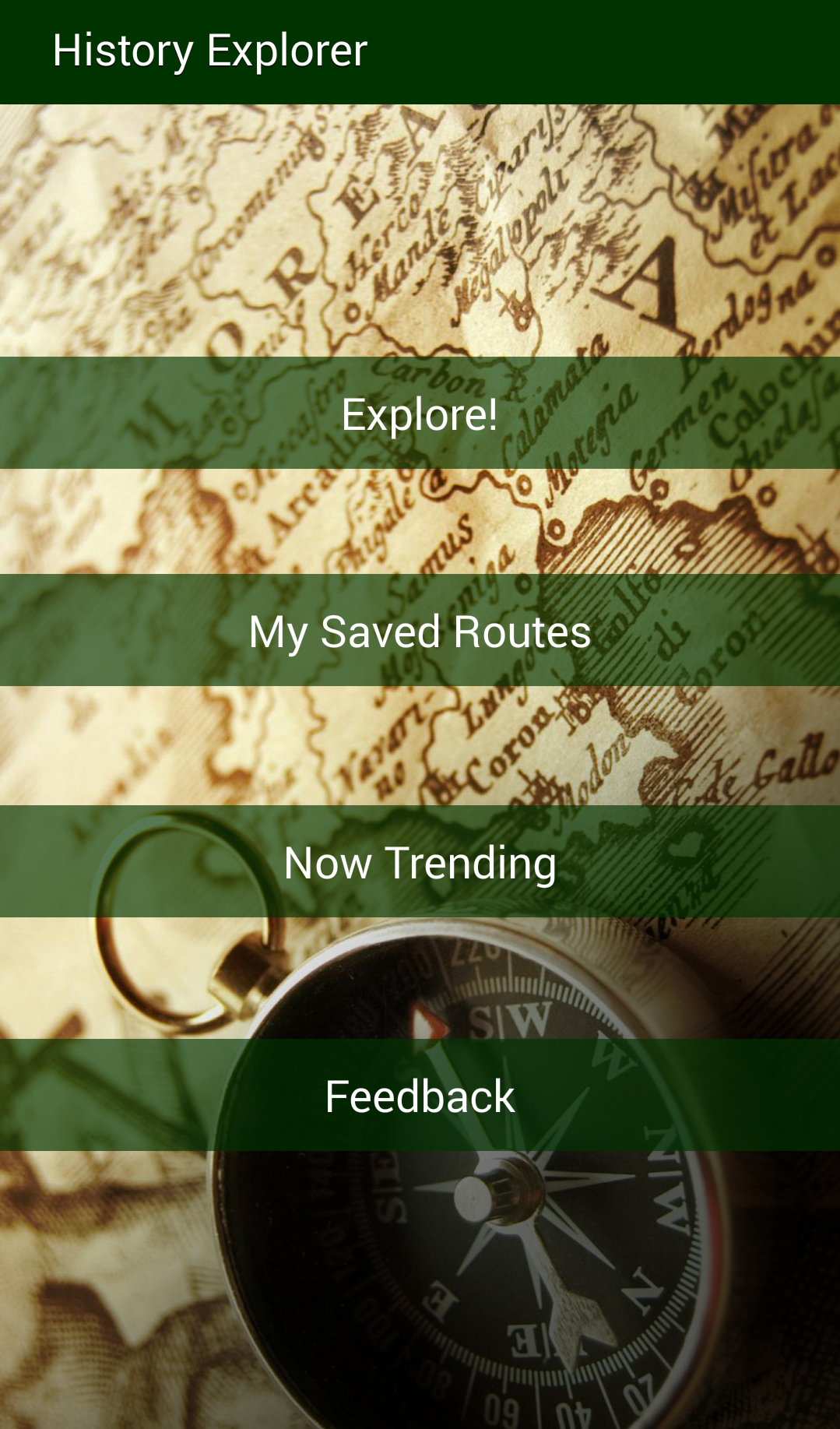
1. You should now see the application on your android device as below.



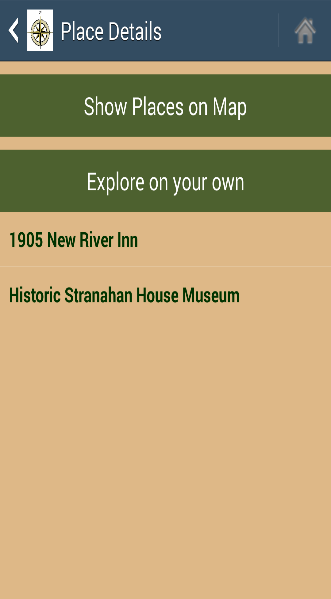
### *4.1.2 Use Case 1: AccessPlaceDetails*

Before being able to run the following use case, section 4.1.1 must be completed. Use Case HEA004: AccessPlaceDetails is performed the following way.

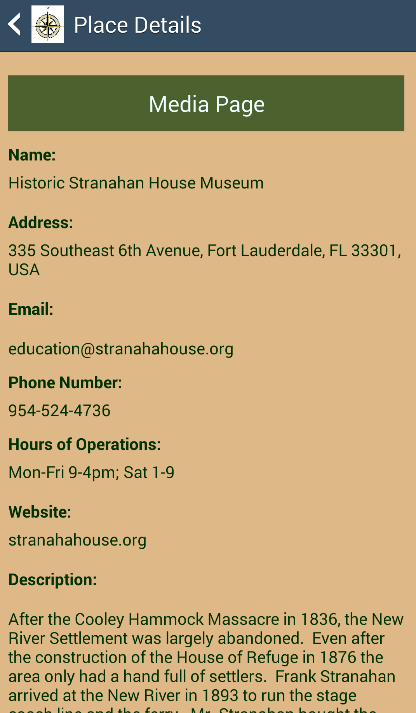
1. Start open History Explorer app by tabbing on the icon.
2. Splash screen will show up and then you will be transferred to the Main Screen.
3. Within the main screen, select “Explore”



1. You will then to transition to the PlacesNearbyActivity and will see all the places within the database.
2. Select a location. For this example, I will select “Historic Stranahan House Museum”



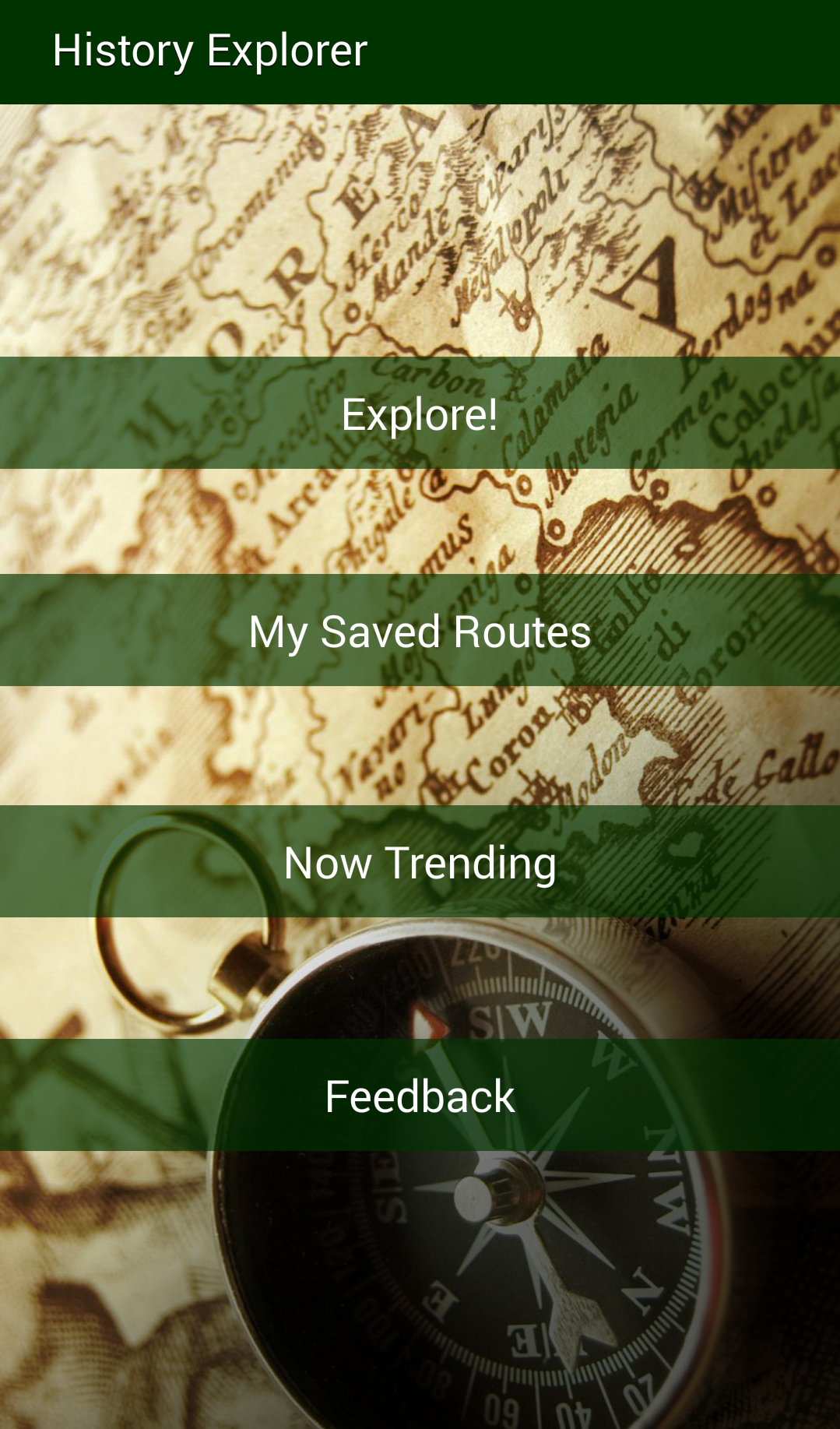
1. This will take me to the SinglePlaceActivity screen and I will be able to see all the details about this site on one screen.



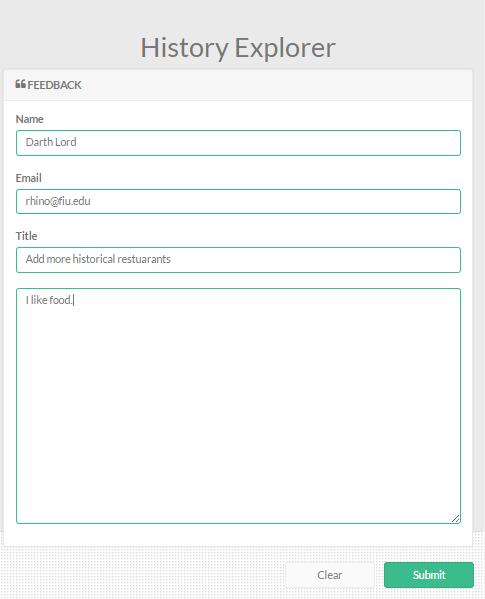
### *4.1.3 Use Case 2: AndroidSubmitFeedback*

Before being able to run the following use case, section 4.1.1 must be completed. Use Case HEA006: AndroidSubmitFeedback is performed the following way.

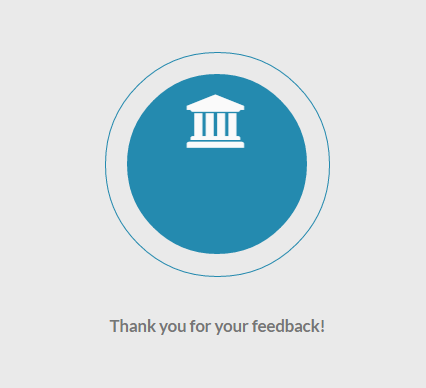
1. Start open History Explorer app by tabbing on the icon.
2. Splash screen will show up and then you will be transferred to the Main Screen.
3. Within the main screen, select “Explore”



1. Your default browser will open to “http://ha-dev.cis.fiu.edu/WebApp/#/layout/feedback
2. Fill in the following details, as such.



1. Click Submit
2. You will be redirected to the Thank page.



## 4.2 Web Application Setup

### *4.2.1 How to run the application*

### *4.2.2 Use Case 1*

### *4.2.3 Use Case 2*

# QUICK REFERENCE

All references are online sources. Refer to 6.0 Access Online Reference.

# ACCESS ONLINE REFERENCES

* Android API Developer Google Services: <https://developer.android.com/google/index.html>
* Android API Reference of Packages: <https://developer.android.com/reference/packages.html>
* Android API Guide: <https://developer.android.com/guide/index.html>
* Appium Documentation: <http://appium.io/slate/en/master/>
* Appium Tutorial: <http://appium.io/slate/en/tutorial/android.html>
* Appium Discussion Forum: <https://discuss.appium.io/>
* Selenium Documentation: <http://www.seleniumhq.org/docs/>
* Android Google Maps V2 API: <https://developers.google.com/maps/documentation/android/>
* Creating a RESTful API with PHP: <http://coreymaynard.com/blog/creating-a-restful-api-with-php/>
* Github Learning Resources: <https://help.github.com/articles/good-resources-for-learning-git-and-github/>
* Git Documentation: <https://help.github.com/>
* PHPMyAdmin: <http://docs.phpmyadmin.net/en/latest/>

# REFERENCES