Miscellaneous Reports

# Test Report

The testing for this project was conducted on the web browser and on mobile phones. The Apache Cordova framework facilitates testing and deployment of applications because of the fact that it is written in HTML, JavaScript and CSS. For my testing phase, I used Google Chrome as my default web browser and a few android phones and tablets as my deployment device.

Based on the fact that the application is written in HTML, JavaScript and CSS, you can easily test a function without having to deploy the application to a virtual or physical device. For plain JavaScript functions, such as getting location, displaying a map and testing layouts, I simply used my web browser to test it. For the native mobile phone functions, such as accessing camera, displaying messages, accessing calendar, I had to deploy the app to a physical, or virtual, device in order to access the test results.

This approach was appropriate for this project. The reason why it was appropriate is due to the fact that I was using one of the best IDEs for developing Apache Cordova applications. The IDE is called JetBrains WebStorm. On this IDE, you can have direct access to virtual devices and also deployment options, making it easy to build, compile and run your code across several devices. The best useful feature about JetBrains WebStorm is the ability to create a Local Host in order to test your applications. This is how I was able to conduct testing on my Server – Client connections without always having to deploy it on a physical device.

For my PHP/MySQL scripts, I was using an IDE called JetBrains PhpStorm. This IDE is also great for writing PHP scripts, it provides code completion, refactoring and more. I chose this IDE because of its Version Control Systems integration, which enables the IDE to support remote deployment, databases/SQL, REST Clients and other tools. It also takes care of the Code Quality Analysis to help the developer write neat code that’s easy to support.

This approach was simpler, because I only had to test connections and queries to my remote Database, which is stored in a web server called 24Hosting. Once I knew the script was working, I deployed it to my web server and then accessed it from the Church Finder Ireland application.

The most interesting test happened when I sent the application to a friend over in Brazil. I needed to test the speed of the application when accessed from another country. Since my web server is located in England, the connections took longer than expected when compared to connections made from Ireland. This phase of the testing only used the *Date().getTime()* method to get the milliseconds that a function takes to execute.

The table below shows the difference between accessing my web server from two different locations:

|  |  |  |
| --- | --- | --- |
| **Name of function** | **Brazil (in seconds…)** | **Ireland (in seconds…)** |
| Accessing list of Churches | 3.1 | 1.2 |
| Getting details of a Church | 1.9 | 0.8 |
| Accessing map | 1.7 | 1.8 |
| Uploading a picture | 2.1 | 0.6 |
| Viewing an existing Picture | 2.6 | 1.3 |

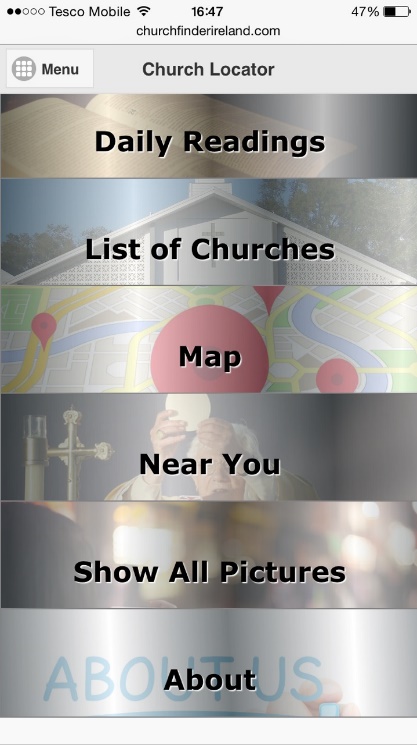
When the application was finished with the development, I initialized the system testing. This was an enjoyable approach because I was able to identify the main problem when developing Apache Cordova applications.

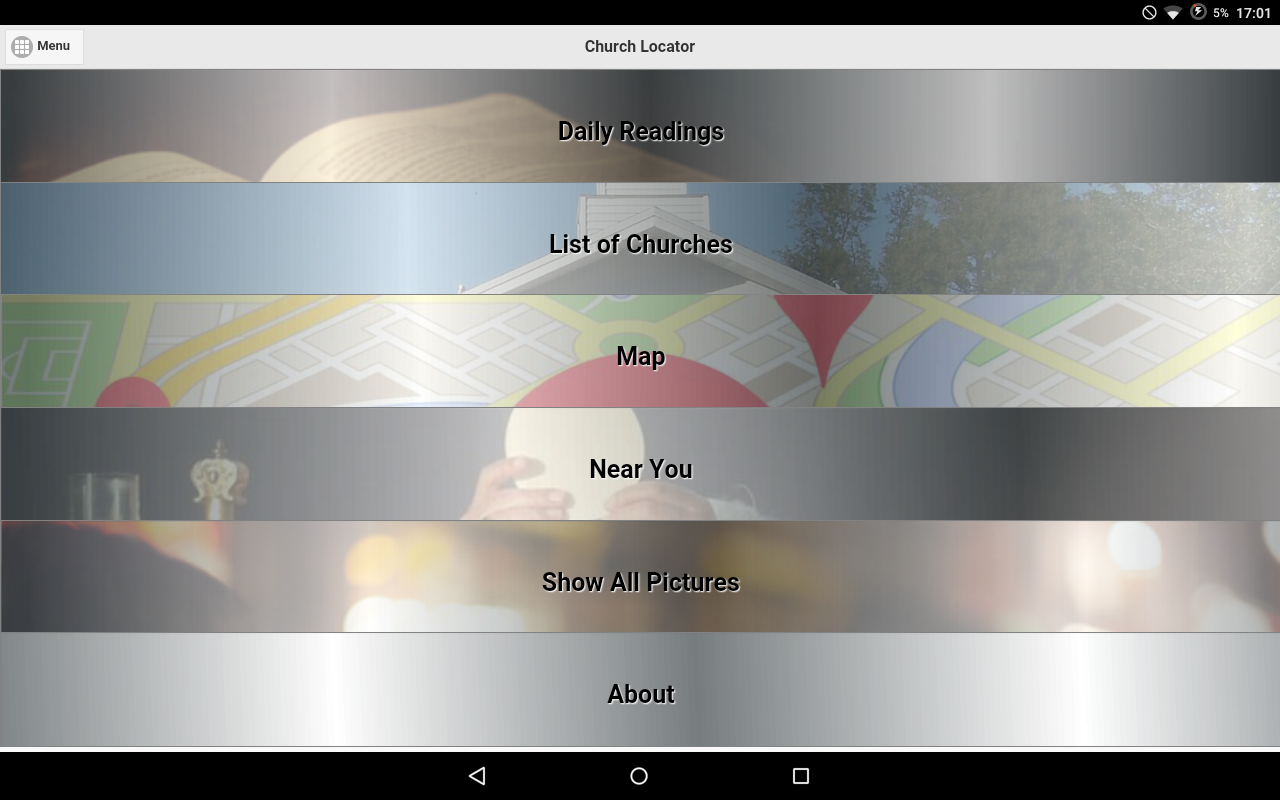
The table below shows a list of devices, and an indication of whether the application was successfully tested on the device:

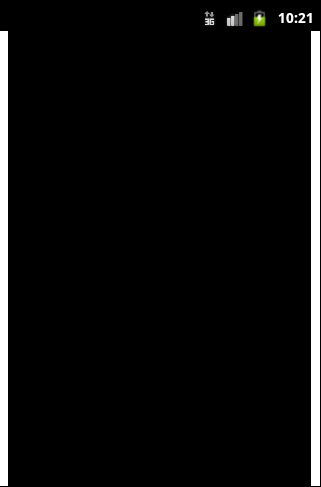
|  |  |  |  |
| --- | --- | --- | --- |
| **Device Name** | **OS Version** | **Successful** | **Comments** |
| Samsung Galaxy S5 | Android 5.1 | Yes |  |
| Google Nexus 5 | Android 5.1 | Yes |  |
| Motorola Moto G | Android 5.1 | Yes |  |
| HTC One | Android 5 | Yes with errors | CSS content not being correctly rendered, making the layout of the application really messy |
| Sony Xperia SP C5303 | Android 4.2 | No | The application was unable to start. Unknown problem |
| Samsung Galaxy S4 | Android 4.4 | Yes with errors | CSS content not being correctly rendered, making the layout of the application really messy |
| Samsung Galaxy S3 | Android 4.4 | No | The application was unable to start. Unknown problem |
| Samsung Galaxy Tab 2 10.1 | Android 4.4 | Yes with errors | CSS content not being correctly rendered, making the layout of the application really messy |
| Samsung Galaxy Pro 4 10.1 | Android 5.1 | Yes |  |
| iPhone 6 | iOS 8 | Yes |  |
| iPhone 5S | iOS 8 | Yes |  |
| Desktop | Windows 8.1 | Yes with errors | Mobile native functions do not work |

The images below are an illustration of the system testing made on physical devices:

**Samsung Galaxy S5 iPhone 6**

****

**Sony Xperia SP C5303 Galaxy Tab Pro 4 10.1**

****

I was able to identify two major problems with my application by conducting the system testing.

The first problem is with the Android version. Whenever I tried to run my application on devices that had Android 4.4 or lower, the application either displayed incorrectly or did not even start. The reason is still unknown, as I haven’t been able to identify the problem yet.

The second problem is with the screen size. This application was tested mostly on 5’ inches devices. The CSS rendered properly on 5’ inches devices and also on devices running Android 5 or 5.1, but it did not run properly on lower API’s or on smaller screen sizes. From research, I found out that the developer has to specify CSS-media-queries to resize the application according to the actual resolution of the device. This will be implemented in future versions of the application.

The most difficult part of the testing phase occurred when I was testing the GPS functions in order to get the routes between two points (the user location and Church location). In order to achieve this, I had to deploy the application to the phone every time I changed a line of code, and this was difficult to handle. After several tries, I was able to get the function working properly on the testing device.

Unfortunately, I did not have time to use testing frameworks, such as QUnit (for JavaScript unit testing) and PHPUnit (for PHP unit testing).

# Installation Manual

This installation manual will cover everything that is needed for a developer to compile, run and deploy this application on a mobile device.  
 **Operating System:** Microsoft Windows 8.1 / Windows 8 / Windows 7  
**Programs needed:** Github / JetBrains WebStorm / Notepad++  
**Plugins and SDKs:** JDK / Android SDK / Apache Cordova / Node.js

The first step is to clone my application source code folder on Github to your desktop. Open your web browser, navigate to *https://windows.github.com* and download Github for Windows. Once the download is finished and you have installed the program, you will get 2 new icons on your desktop: Github and Git Shell. For this tutorial, we will be using the Git Shell.   
Once the Shell is opened and loaded, type “**git clone**”and then enter the project’s repository page, which is [**https://github.com/x00093357/4th\_Year\_Project.git**](https://github.com/x00093357/4th_Year_Project.git). The complete command should look like this:   
**>** git clone <https://github.com/x00093357/4th_Year_Project.git>  
Once this task is finished, open the Windows Explorer and go to your Documents – GitHub. The clone of the repository should be there.

Now that we have a copy of the repository, the next step is to get the IDEs and frameworks needed to compile, run and deploy the application. We will begin by installing Apache Cordova on our computer.  
To install Apache Cordova on your computer, follow these steps:

1. Download and install Node.js (<https://nodejs.org>). Following installation, you should be able to invoke **node** and **npm** on your command line.
2. Download and install a Git Client (<http://git-scm.com>). Following installation, you should be able to invoke **git** on your command line. We won’t be using GIT directly, but Apache Cordova uses it behind-the-scenes to download plugins.
3. Install the **cordova** module using **npm** utility of Node.js. On your command line, enter: **npm install –g cordova**.

After successfully following these instructions, you should have the Apache Cordova framework installed on your computer. The next step is to download the Android SDK and the JDK (Java Development Kit).

You can download the Android SDK by navigating to the Developer Android website (<https://developer.android.com/sdk/index.html>). You do not need Android Studio for this tutorial, but the installation should take a long time to be completed.

You can download the JDK by navigating to the Oracle website (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>). Make sure to select the right version of your Operating System to prevent any errors.

Once you have download and install the Android SDK and the JDK, you will be able to compile, run and deploy Apache Cordova applications.

This project does not require MySQL or PHP to be installed on our local machines, since these files are located on the remote web server. But if you want to access the files locally instead of remotely, you can download MySQL by going to this website:   
- http://www.mysql.com/downloads/  
and you can download PHP by going to this website:   
- <http://php.net/downloads.php>

Now that we have all the plugins, frameworks and APIs, the last thing we need is an IDE. For this tutorial, I will be using JetBrains WebStorm. The program has a 30-day free trial and if you are a student, you can have a student license for 365 days.

You can download JetBrains WebStorm by going to this website:   
- <https://www.jetbrains.com/webstorm/download/>

Once you have installed JetBrains WebStorm, you can open this project by going to File – Open Project and selecting the cloned GitHub folder (Documents – GitHub).

The HTML, JavaScript and CSS files will be located on the “www” folder (ChurchFinderIreland – www).

Next step is to select a physical or virtual device. Go to the “Run” menu, and select “Edit configurations”. If you want to run the application on a virtual device, change the “Command” option from “run” to “emulate”. Also make sure you have installed an Android Virtual Device (you can accomplish this while installing the Android SDK).

If you want to deploy the project on a physical device, the “Command” option should be “run”. Also make sure you have installed the appropriate drivers for your device on your computer and that you have enabled “Developer Options” and “USB Debugging Mode” on your Android device.

To compile and run this application, select “Run” on the navigation menu, and then select “Run” or press Shift + F10. If you correctly installed all the plugins and SDKs, the project should compile and get deployed on your selected device.

# User Manual

This user manual will cover every aspect of the Church Finder Ireland application. In order to use this application and get the best experience, the user has to be connected to a Wi-Fi or a 3G data connection. Also make sure that you have your Location/GPS services enabled. The application will not work without an Internet connection, and it’s limited without the Location/GPS services enabled.

### How to turn on your Wi-Fi / 3G (Android)

1. Head into the Settings menu of your device
2. Select the “Wireless & Networks” option
3. Select Wi-Fi (Select More for 3G) and turn the toggle or button to “ON”
4. Select your network and hit Connect

### How to turn on your Wi-Fi / 3G (iOS)

1. Head into the Settings menu of your device
2. Select the “Wi-Fi” option
3. Toggle the Wi-Fi button to “ON”

### How to turn on your Location/GPS (Android)

1. Head into the Settings menu of your device
2. Select the “Locations” option
3. Select an accuracy setting

### How to turn on your Location/GPS (iOS)

1. Head into the Settings menu of your device
2. Select the “Privacy” option
3. Select the “Location services” option

If you have an Internet connection and the Location services enabled, the application should work without any problems. The next part of the manual will cover a “How-to” for the functions within the application.

### How to Register

1. Press the Register button once you enter the app
2. Enter the details for all fields ( Must have a valid email / Password is case-sensitive / Password must have at least four characters)
3. Press the Submit button

### How to Log In

1. Enter your username and password
2. Press the Sign In button

### How to Log Out

1. On the main screen, press the Menu button
2. Press the Log Out button

Once you are logged in, this is what you should see on your screen:



### How to read the Mass Daily Reading

1. Press the Daily Readings button
2. Navigate through the readings by selecting an option (1st reading/Psalms/2nd reading/Gospel)
3. Press the Home button to go back to the app’s main screen

### How to access the List of Churches and get the details of a Church

1. Press the List of Churches button
2. Press the Show All Churches button
3. You can search for a Church, City/Town or County by entering text in the Search box field
4. Press into a Church to get the details of it
5. Press the Home button to go back to the app’s main screen

### How to get directions from your location to a selected Church

1. On the details page, press the Options button
2. Press the Show on Map button
3. Press Home to go back to the app’s main screen

### How to view existing pictures of a selected Church

1. On the details page, press the Options button
2. Press the Pictures button
3. Press the X button to exit or swipe down

### How to upload a picture from your camera/gallery

1. On the details page, press the Options button
2. Press the From Camera/From Gallery button
3. Select a picture and press the OK button to upload the picture

### How to see all the Churches in the map

1. Press the Map button
2. Tap on a marker to see details of the Church
3. Press Home to go back to the app’s main screen

### How to get a list of Churches near your location

1. Press the Near you button
2. Select a distance (in kilometres) on the slider
3. Press the Get Near Masses button
4. Press Home to go back to the app’s main screen

### How to update the list of Churches near your location

1. On the Near You page, select a new value in the slider
2. Press the Update Results button
3. Press Home to go back to the app’s main screen

### How to set notifications for a upcoming Mass

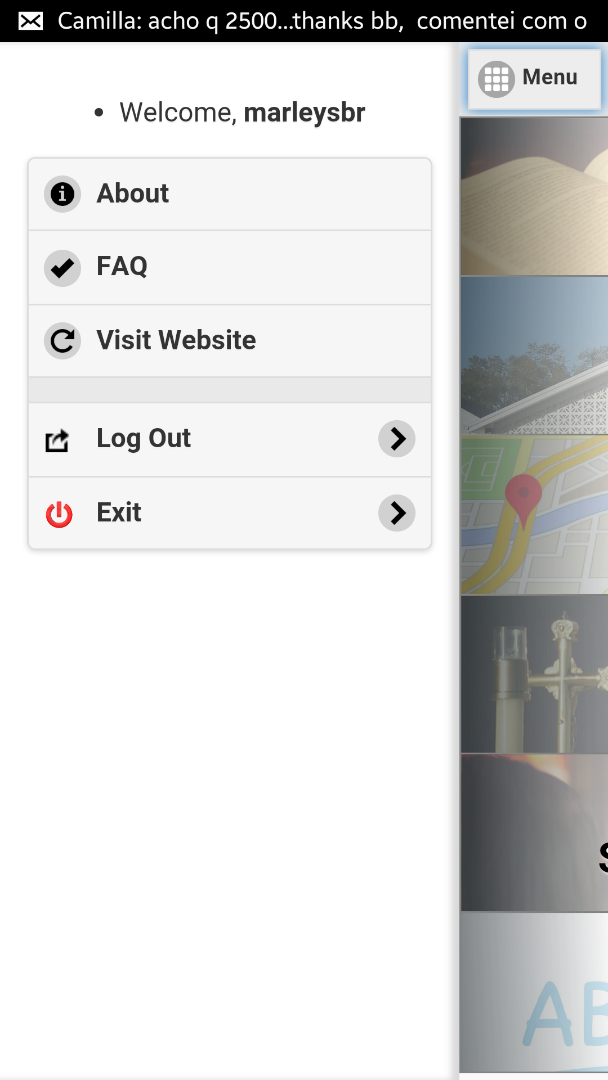
1. On the Near You page, select a Church to see more details
2. When the panel appears, press the Set Notification button
3. This should bring you to your Calendar function

### How to view all existing pictures

1. On the main screen, press the Show All Pictures button
2. Press the X button or swipe down to go back to the app’s main screen

### How to exit the application

1. On the main screen, press the Menu button
2. Press the Exit button



# Post-Project Review

### Review of the project

This project was completed on time and I learned a lot about Server – Client communications. That is what my project is all about, developing one application to be used across several devices, from Desktop to mobile devices. Learning about the different APIs to develop HTML 5 applications was certainly a good experience. I was able to sharpen my skills in HTML, JavaScript, CSS, JQuery and also Java. Learning about deploying a finished product to a physical device was also an enjoyable experience, this enabled me to learn about different devices, different specifications and more.

The original idea was to develop a native Android application. I decided to change the scope of the project and use Apache Cordova. Apache Cordova is a set of device APIs that allow a mobile app developer to access native device function such as the camera, or accelerometer from JavaScript. When you combine this set of device APIs with a UI framework, in my case I used JQuery Mobile, it allows a smartphone to be developed with just HTML, CSS and JavaScript. When you use the Cordova APIs, the application can be built without any native code. Instead, web technologies are used. And because these JavaScript APIs are the same across multiple device platforms, the application should be portable to other device platforms with minimal to no changes.

The proposed functionality was developed and implemented on time. Thanks to the extensibility of web UI frameworks, such as JQuery Mobile, I was able to develop a simple and easy-to-use user interface. The user experience while using the application should be satisfactory. The effectiveness of the UI relies on simple buttons with an image to illustrate the option selected by the user and the efficiency relies on the speed of JQuery Mobile and HTML 5 to execute the code and display the page to the user in almost no time.

This project was aimed to help mainly tourists, and also locals, to find Churches, Mass times and also interact in a unique social network, created by the user, by uploading and viewing pictures of existing Churches. The application is focused on the user, and the UI is focused on providing the user a good experience with the service provided. On the application the user will be able to log in/log out, register a new account, read the Mass Daily Readings of the day, get a list of all Churches in Ireland, view a Map with their location and also the Churches’ locations, get Churches by distance, view and upload pictures, get the route between their location and a selected Church and also set notifications for upcoming Masses. The application will provide a simple how-to in the FAQ section, in case the user has problems while using it.

In general, the finished product looks simple, beautiful and easy-to-use. That was one of the main goals set for this project. Having something innovative, that grabs the attention of the user and makes them use it again. The Church Details page really grabs the user’s attention by providing a picture of the selected Church on the background. This is a unique feature.

Apache Cordova was possibly the best decision I had to make for this project. It gave me experience with something new, interesting and fun. I would recommend Apache Cordova for simple to medium-high type of applications. Depending on how deep you want native mobile functions, it may get tricky to use Cordova, because you will end up having to use somebody’s plugin or having to develop your own. That is one weak point about Cordova. Because it is not native code, developing big applications with a really deep functionality may be hard and tedious.

In a general context, the part that I most enjoyed while developing this application was seeing how fascinating it is to develop a multi-platform mobile application, making connections to a remote server, pulling data from it simultaneously and getting the results in no time from another device.

In a summary, I can say that this project gave me some personal and technical skills that will be really useful for my professional career, for example, having to administrate a big project on your own and also learning how to connect your application to remote web servers and remote databases.

### Possible extensions of the project

###### User adding new Churches

This function will enable the user to add new Churches to our Database. Currently, only the admin has the rights to add new Churches through our Database Management Portal. I plan to make this application as interactive as possible with the user by letting them add new Churches that may not be yet stored in our Database.

###### Desktop version

By creating a Desktop version of this application, we will extend our limits. I plan to create a Desktop version of the application, with the same features, so that users can have more options to use our services.

###### User adding videos

This function will enable the user to also add videos to our Database. Sometimes a picture is not enough, and a video could describe a Church in more details. This section would be monitored and the video would only be posted prior to an admin reviewing it and approving it. This will be done to prevent junk data from being added to our Database.

###### Admin application

This admin application will contain all admin features. Adding new Churches, reviewing pictures and videos, responding user’s queries and much more.

###### Reduce the internet’s usage

So far, our application is really dependent on an Internet connection. My plan is to try and store the data on the application, so that the user can have an offline access to our services. This would reduce the internet’s usage and it would increase the user’s experience with this application.

###### Increase validation

So far, no validation is done on uploading a picture. This function relies on admins reviewing and approving new pictures and videos before they go online on our application.

###### Push notifications

Push notifications sometimes help users not forgetting about a function or feature of the application. We would also be using this service as a way to providing news about the Catholic Church to the user.

###### Adding friends

This feature would increase the idea of a social network within the Church Finder Ireland application. The user would be able to add friends and also send messages to existing users.

###### Commenting on existing pictures

This feature would enable the user to comment on existing pictures. The owner of the picture would have controls over the comments, meaning that they can delete or edit comments made by other users.

###### Adding reviews of Churches

This feature would enable the user to give a review about a Church they visited or know. It would be helpful for people looking for tips or any type of advice about the Church they’re about to visit.

###### Adding a section for the Holy Rosary

This feature would enable the user to pray the Holy Rosary of the day.

### Review of Research, Analysis and Design