

Where2P4Free

Bachelor of Science (Honours) Software Development

Name: Patrick Browne

ID: C00190601

Year: 4th Year

Supervisor: Paul Barry

Due date: 12-04-2019

Final Project Report

Table of Contents

Table of Contents	2
Introduction	3
Project Description	4
Mobile App Screens (Map Screen)	5
Add Restroom	5
Update Restroom	6
Remove Restroom	7
Restroom List	8
Conformance to Specification & Design	9
Description of Learning	10
Technical	11
Personal	12
Review of Project	13
What went right?	13
What went wrong?	14
What is missing?	14
Future Improvements	15
Acknowledgements	16

1. Introduction

The purpose of this report is to document the development plan of the project. The project is based upon the development of a mobile application for the Android market. It is intended to be a guide for locating nearby free public restrooms. Firstly, to assist tourists, businesspeople or locals in locating restrooms in various urban areas, and secondly, to allow users to directly add, remove or update locations. For example, if a tourist were looking for a nearby restroom to relieve themselves quickly - as a user you could quickly find this location quickly through the app, helping tourists enjoy the various tourist spots better.

2. Project Description

Where2P4Free is a mobile application for the Android platform. The purpose of this app is to allow users to locate nearby free public restrooms. Whether the user is a tourist visiting a city, or a business person is attending a conference, this app provides peace of mind to the traveller. Nearby locations are shown using the Google Maps using marker. Users can click on these markers to get the location and directions. Directions are provided using the Google Maps App. The map is displayed on the first page for quick access if needed in a hurry.

A unique point of this app is the crowdsourcing functionality. The app consumes data generated directly from the users themselves. If the users wish to add locations, they can enter in information such as location, description, and type. Restrooms can also be updated or removed by the users to improve the accuracy of the app. In order to reduce the likelihood of wrong data being submitted by malicious users, a threshold system is used. When a request is made to add, update, or delete a count is incremented. In order for the request for either to be submitted, the count needs to be over the threshold. When the threshold is met, the request is submitted to the database.

2.1 Mobile App Screens (Map Screen)

2.1.1 Add Restroom

To access this form the 'add' button is initial clicked to open the form once a marker has been added. The user enters in the required information into this form. Once all fields have been filled out, they tap 'add', and the data is sent to the server.

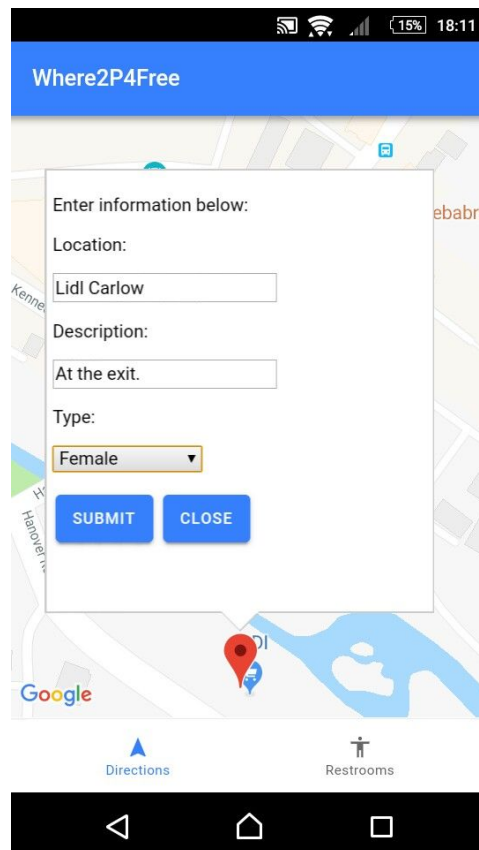


Fig 1.1 Add Restroom

2.1.2 Update Restroom

The user enters in the required information into this form. Once all fields have been filled out, they tap 'update', and the data is sent to the server. For security purposes, changes made to an existing restroom are not implemented until a third request.

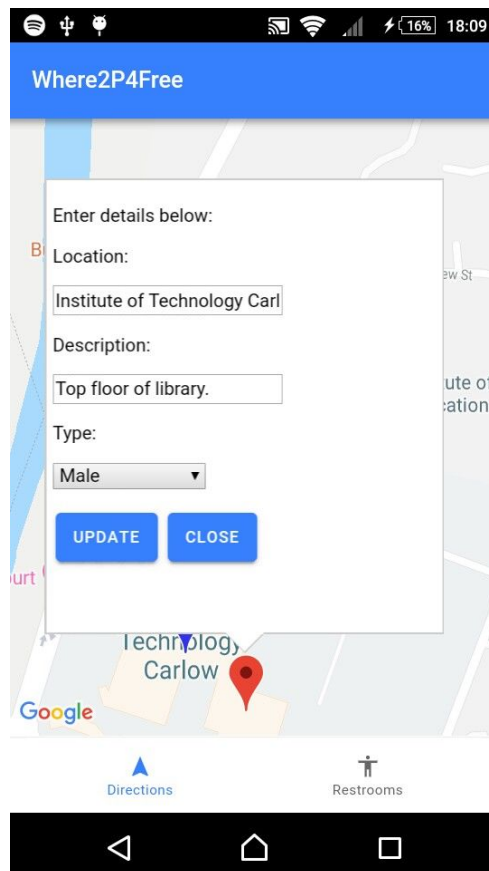


Fig 1.2 Update Restroom

2.1.3 Remove Restroom

This option is displayed on either existing or newly added markers. If the user clicks on an existing marker, an option to remove appears. Upon clicking 'remove', the marker is removed from the map. However, for security purposes the location still remains in the database. It is removed for the user.

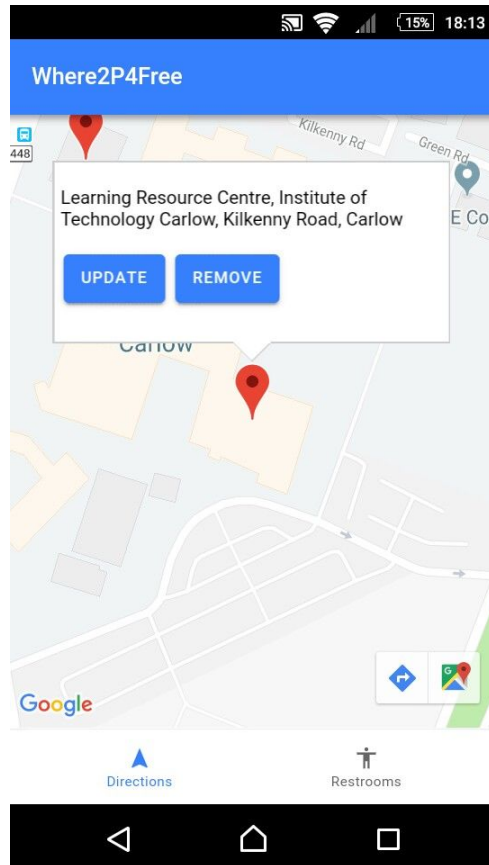


Fig 1.3 Remove Restroom

2.1.4 Restroom List

This screen displays a list of restrooms previously added by the user or other users. Additional information is listed. Includes a brief description of said restroom, and type, e.g. Male, Female, etc.

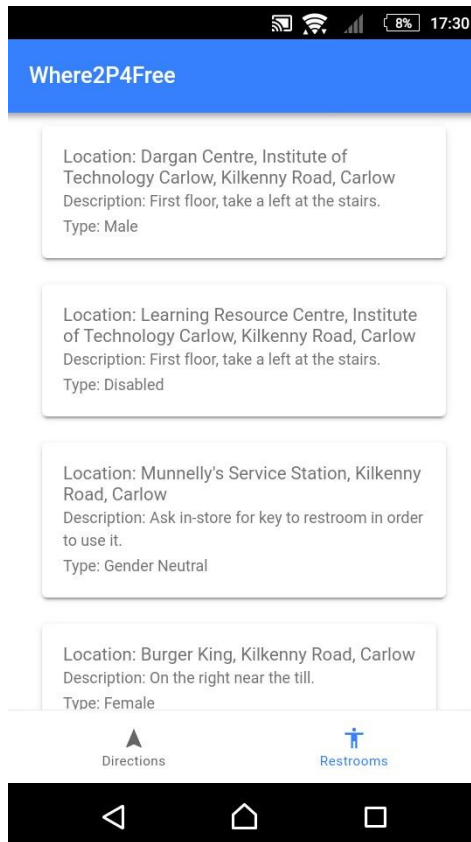


Fig 1.4 Restroom List

3. Conformance to Specification & Design

The majority of the functionality was implemented. The one functionality that wasn't implemented was the user accounts. One reason why it wasn't implemented was during development, the scope of the project changed to a degree. Originally the idea was to have the users make accounts in order to use certain functionality. In this case, the user would need to create an account to add, update or remove locations. The reason behind this was to provide a degree of oversight over user activity. If a malicious user attempted to input incorrect data, their account could be blocked. However, this was eventually dropped due to the fact that the app was not handling any user-sensitive data. The second reason why it wasn't implemented was due to time constraints. The core functionality was only finally implemented at the very end of the project. Hence, the user accounts were dropped.

Another functionality which was not implemented was threshold for adding restrooms. This was to conform to the crowdsourcing aspect of the project. In order to add a restroom, a threshold needs to be reached before it is saved to the database. Updating, and deleting restrooms follows this approach. Threshold level is three, anything below that is not taken into account. Anything above that is. Again time constraints prevented the implementation of this feature in the API.

The implemented functionalities are as follows:

- Create new restroom.
- Delete new restroom.
- Update existing restroom.
- Delete existing restroom.
- Display list of restrooms.
- Threshold for request from app.
- API with endpoints.

4. Description of Learning

4.1 Technical

This project proved to be quite the technical challenge for me. One of the reasons being the scale of the project. I had never attempted to develop something of this scale before, let alone design. However, when choosing technologies in which to develop the app with, I was careful with what was needed. Choosing Ionic as the mobile framework was perhaps the best decision I made for this project. Ionic proved to be the best very thing needed to get an app quickly off the ground and running locally. With extensive documentation, a very helpful and informed community, and a wealth of online technical resources, the app was built quickly and easily. The framework itself is simply HTML, CSS, and TypeScript working together to create superior UI components which are easier to implement versus other frameworks. Also, using Visual Studio Code as my text-editor of choice allowed me to run Ionic apps locally directly from the editor. When it came to scaling the app with more functionality, generating additional pages, services, components, proved very easy. Simply entering in a command such as “ionic g page pages/restroom-tab” created the necessary page with all the Angular Routing built in.

In order for the app to consume data from the users, an API was needed. This API was developed using Python in a Flask web application hosted on PythonAnywhere. It allows for separate technologies working server-side and client-side to work together. The API being the connection between the two. Python was the chosen language for the API as I had previous knowledge with Python. It is a powerful web programming language ideally suited for creating custom API quickly.

Flask was the web application used to create the API endpoints to which requests would be sent to and sent from. Flask comes with many Python libraries, including a RESTful API library offering advanced functionality such as custom error-handling and OOP API scalability. Although for the purposes of this project these advanced functionalities were not needed.

The host chosen to host remotely the API was PythonAnywhere. Like Python, I had previously experience working with PythonAnywhere in hosting earlier projects. It made sense to use this host as PythonAnywhere is specifically built to work with web applications written in Python. It's free plan was also attractive from the point of view of the processing quotas offered daily for requests from the app. All the libraries that could have possibly be needed were pre-installed and setting up the web application initially was straightforward and painless.

4.2 Personal

My biggest personal achievement with this project was learning multiple technologies at once. In a preferred scenario I would solely be doing this and nothing else. However having to juggle not just this project but other coursework at the same time proved challenging. Luckily I had chosen the technologies specifically for their easy implementation. This meant that instead of spending too much time trying to get things working from the start, it was done in a matter of days. With this done I made great progress with the project.

However it wasn't entirely straightforward from there. Even with the progress I was making it proved a challenge to work with AngularJS. Angular is extremely powerful in what it is capable of doing. But this was an entirely new technology I had never seen, let alone heard of before. At the beginning, an uphill struggle was the norm for the first few months of development. Although the powerful features of Angular were right there for me to implement, actually understanding how to implement them was not entirely possible for me. However, with perseverance and a strong work ethic I nailed down the core concepts of Angular.

5. Review of Project

5.1 What went right?

The core functionality of the project was achieved. This included the following:

1. Find the user's current location - The app can locate the user's location using a combination of GPS and Cellular networks. Both are used to achieve high-accuracy geolocation finding. From there the user can get directions to nearby restrooms.
2. Restrooms - The app can display markers on the ap which indicate the exact location of free public restrooms. When each marker is clicked, a small info window is displayed showing the location. There are also a couple of options available such as Add, Remove, Update.
3. Input form - An input form allows the user to submit data on a particular restroom if they are adding or updating one. This form appears when an existing restroom is selected and the 'update' button is clicked. It also appears when the a new marker is added to the map and the 'add' button is clicked. For simplicity, the input field appears inside the InfoWindow above the marker. Allowing the user to input the data whilst being on the same page.
4. List - The second page of the app allows users to view more information regarding nearby restrooms. Such information listed includes; Location, a brief description of the restroom, and the type. This can prove useful for the user to make a decision which location to use at their leisure.
5. Crowdsourcing - The app consumes data generated by the users. The API which handles the data implements a threshold approach. Any requests to add, update or remove locations must meet a certain threshold before action is taken. Ensures that the data available to the app is reliable and accurate. Also reduces the likely of abuse of the system by malicious users.

5.2 What went wrong?

Issues arose during the development of the project. The issues that came up were as follows:

1. Broken UI - During the design and implementation of the UI an issue arose with some of the pages of the app. As the functionality of the app changed somewhat, some of the existing layouts were no longer needed. When some of the pages were removed, the app ceased to function properly. This resulted in the working code being extracted from the original app and placed into a newly built app.
2. Dynamically adding markers - The biggest issue for this project was attempting to allow users to add markers to the map anywhere they wished. The logic behind the implementation proved difficult to develop. Eventually after much time test logic was implemented which worked as required. Having successfully achieved this, the project progressed quickly.
3. Form submit data - A tedious issue which plagued the project during its final iteration. A form for both adding and updated locations had been created. However, an issue arose with how to retrieve the data from the form.

5.3 What is missing?

Although the core functionality was achieved, there were some were not implemented:

1. User accounts - After the crowdsourcing the use of user accounts was the next most important functionality. The idea of user accounts was to provide a degree of oversight for the app owner. Even though the users themselves are responsible for the data, user accounts would prevent malicious users from using the main functionalities of the app. For instance, blocking a user via a registered email address from adding, updating, and removing locations. User activity could allow be monitored using said accounts to view previous activity on the app.
2. Live database - Although the app interacts successfully with an API hosted via PythonAnywhere, the host those not offer live database. This is provided through Google Firebase. JSON data is used to construct the structure of said databases and any changes to the data is sent to all devices registered with the platform. This would have proved exceptionally useful for the Google Map API. Removing a location on one device would result in the location being removed from all other devices in real time.

5.4 Future Improvements

Potential improvements for the application in the future:

1. Implement user accounts using Firebase.
2. Use real time database synchronisation through Firebase.
3. Track the user's location in real time.
4. Provide information on distance between locations.
5. Only show nearby within locations within a certain radius.
6. Reject incorrect information submitted by users.

6. Acknowledgements

I would like to thank my project supervisor, Paul Barry, lecturer at IT Carlow, who introduced me to the concept of crowdsourcing. His wealth of knowledge in the field of Python helped me develop the crowdsourcing aspect of the project. Anytime an issue arose which I had no knowledge of, my first port of call was Paul. Without him, the project would not have reached the state it currently is in now.

I would also like to thank my friends and family. My friends for the advice they gave me throughout the year, as well as any insightful opinions they had to assist me with pushing the development forward. And family, who provided support for me during the whole project, especially the past few months. But more importantly, for helping me believe in myself to develop the project.