

## **Group C15**

### **Deliverable H – Prototype 3 and Customer Feedback**

#### **Engineering Design – GNG 1103 – Section C**

##### **Group Members:**

- **Eleftheria Sarsaroudi: 300189060**
- **Boyu Zhao: 300069815**
- **Adrian Perras: 8231683**
- **Illia Negovora: 300070880**
- **Rui Pang: 300118019**

## Table of Contents

<b>1. Introduction.....</b>	<b>5</b>
<b>2. Prototype 3.....</b>	<b>5</b>
<b>2.1. Find location.....</b>	<b>5</b>
<b>2.2. Full screen: .....</b>	<b>7</b>
<b>2.3. Return to home: .....</b>	<b>8</b>
<b>3. Test Plan .....</b>	<b>9</b>
<b>3.1. Why are we doing these tests? .....</b>	<b>9</b>
3.1.1. Test objectives .....	9
3.1.2. Possible types of results .....	10
3.1.3. How results will be used to make decisions or select concepts .....	10
3.1.4. Criteria for test success or failure .....	10
<b>3.2. What is the prototype and what is the test? .....</b>	<b>11</b>
3.2.1. Type of prototype and reason of selection.....	11
3.2.2. Description of testing process .....	11
3.2.3. Required materials.....	11
3.2.4. Approximation of estimated cost of the prototype.....	11
<b>3.2.5 Work to be done (Testing, construction, modeling work, research) .....</b>	<b>11</b>
<b>3.3. How is the prototype used? .....</b>	<b>11</b>
3.3.1. Recorded information.....	11
3.3.2. How the results will be recorded.....	12
3.3.3. Is this important data for the project?.....	12
<b>3.4. When is the testing happening and how long will it take? .....</b>	<b>12</b>
3.4.1. Duration of test and dependencies .....	12
3.4.2. When are the results required.....	12
3.4.3. Test planning Gantt chart .....	12
<b>4. User Feedback .....</b>	<b>13</b>
<b>5. Conclusion .....</b>	<b>13</b>

## Table of Figures

Figure 1 Invalid Input Terminal Example .....	5
Figure 2 Empty Data Example .....	6
Figure 3 Searched Location Example .....	6
Figure 4 Data Collection Example .....	7
Figure 5 Add Bin Example .....	7
Figure 6 Full-Screen Example .....	8
Figure 7 Return to Home Example 1 .....	8
Figure 8 Return to Home Example 2 .....	9
Figure 9 Unit test performance example .....	9
Figure 10 Gantt Chart Sample for Prototyping Stages .....	12

## **Abstract**

This report lays out the details of development and testing of the third and final prototype. A relay of information will be given on how testing is done and of any refinements for this final stage. User feedback on the prototype was obtained and will be included. While major features were not changed, code has been further optimized / cleaned up for ease of use.

## 1. Introduction

In the previous report, the team was committed to producing products to meet customer needs within the budget, and tested the products again and again, and improved the products. In this report, we will briefly introduce the theoretical knowledge of the project prototype and describe in detail the results after testing the final prototype. The team will continue to collect feedback from customers and users and continue to improve the products.

## 2. Prototype 3

### 2.1. Find location

In this prototype, the range of the latitude is set from  $-90$  to  $90$  and of longitude from  $-180$  to  $180$ . The following image shows an example of what happens when invalid values of latitude and longitude are inserted (view of terminal). The inputted value of latitude is  $100$ , which is invalid because it exceeds the range of latitude in of the prototype, and the inputted value of longitude is  $-200$ , which is also invalid because it exceeds the range of longitude as well. Therefore, the image shows the view of the terminal and the error message at the end.

```
illian@illiaLaptop:~$ curl -i -X GET http://localhost:3000/public/bins?lat=100&long=-200
[1] 167520
illian@illiaLaptop:~$ HTTP/1.1 400 Bad Request
X-Powered-By: Express
Access-Control-Allow-Origin: *
Content-Type: application/json; charset=utf-8
Content-Length: 80
ETag: W/"50-ZjSncGly0dmRwB1vXvB40q1hT6M"
Date: Sun, 07 Mar 2021 23:58:11 GMT
Connection: keep-alive
Keep-Alive: timeout=5

"STATUS (400): insufficient query parameters ||| RESOURCE: /public/bins?lat=100"
```

Figure 1 Invalid Input Terminal Example

The following image shows what happens when valid values of latitude and longitude are inserted (view of browser).

The inputted value of latitude is  $90$ , which is valid because it is in the range of latitude in the prototype, and the inputted value of longitude is  $180$ , which is also valid because it is in the range of longitude in the prototype. Therefore, the image shows the view of the browser and there is empty data because there are no registered bins in this location at the moment.

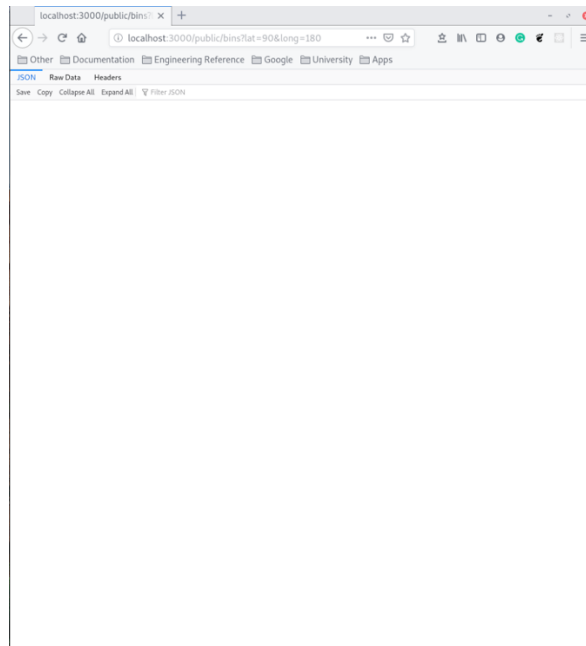


Figure 2 Empty Data Example

In order to add a bin on the server, first the appropriate information must be inserted in the system: the location (longitude and latitude), the waste type, and the colour of bin. In the following image, the orange location icon represents the searched location, and the black location icon represents a recycling bin that matches the inserted information.

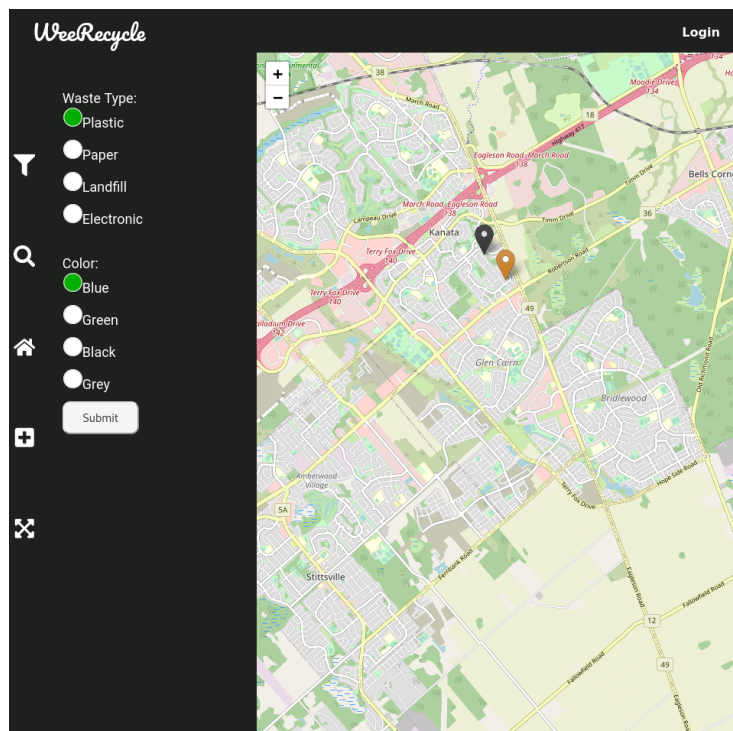


Figure 3 Searched Location Example

The system collects the data inputted (the following image is an example of the data):

```
{
  latitude: 45.30524722258941,
  longitude: -75.88034453938353,
  type: 'plastic',
  color: 'blue'
}
```

Figure 4 Data Collection Example

Then, in the current searched location, a bin may be added by clicking on the desired area in the map and is therefore added in the map/system.

In the following image, the green location icon represents the added bin, and the black location icon represents a recycling bin that matches the inserted information as well.

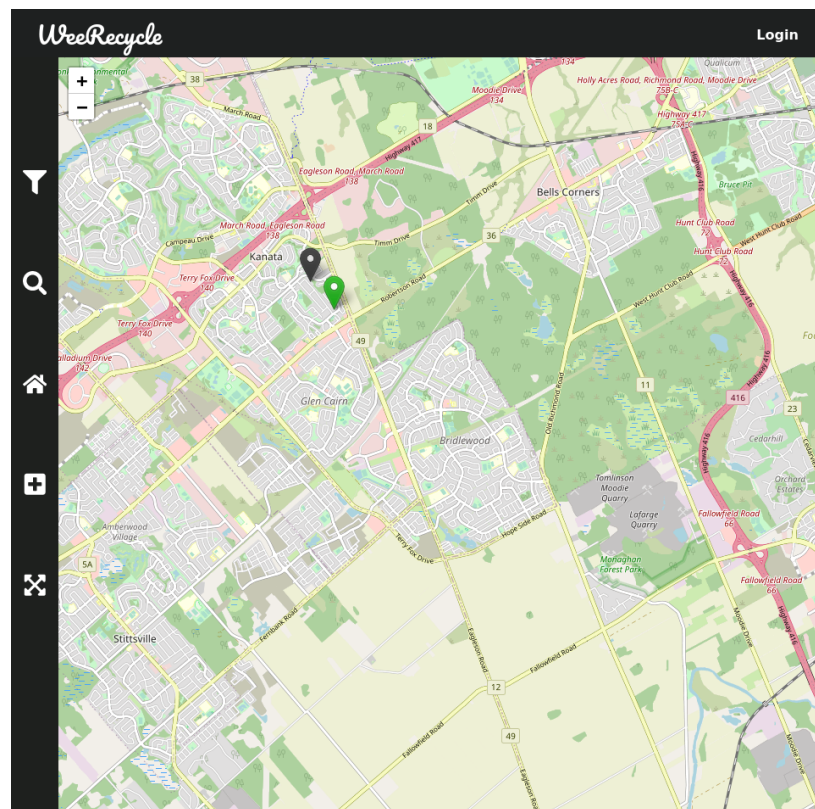


Figure 5 Add Bin Example

## 2.2. Full screen:

When the user wants the map to go into full screen, they just need to click on the “full screen” icon that is located on the side bar. After doing that, the top bar disappears, and the map is in full screen. The following images show how the full screen looks like:

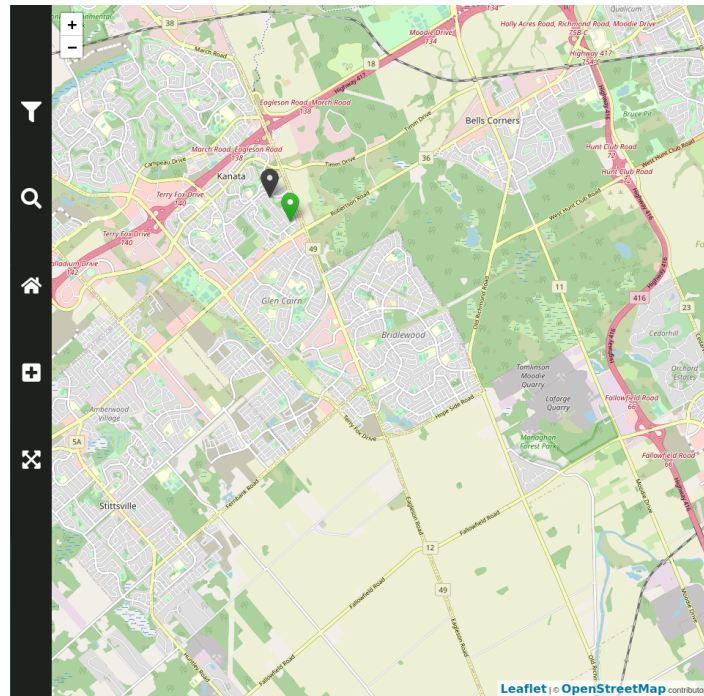


Figure 6 Full-Screen Example

### 2.3. Return to home:

When the user wants to return to the home screen, they just need to click on the “home” icon that is located on the side bar.

The following images represent the “return to home” function. The following image shows the user’s current map view after searching a recycling location, and the next image shows the home view after the user has clicked on the “home” icon.

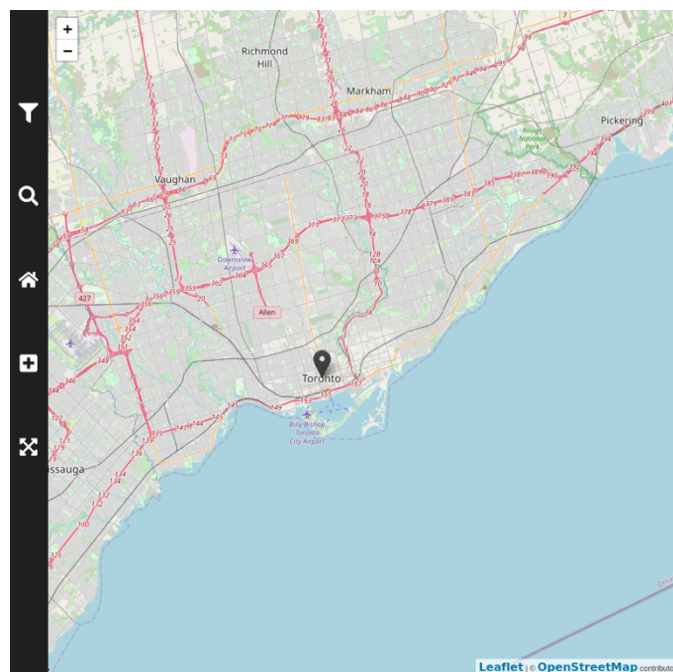


Figure 7 Return to Home Example 1



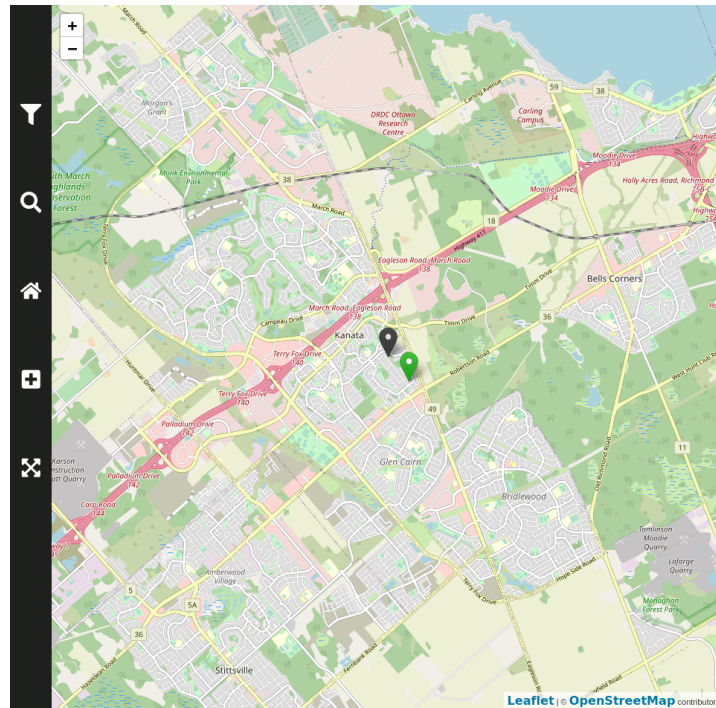


Figure 8 Return to Home Example 2

### 3. Test Plan

#### 3.1. Why are we doing these tests?

##### 3.1.1. Test objectives

The purpose is to be able to automatically test the integrity of the system, after any change made to the code base. In addition, manual testing will be used to test the overall integrity of the system. In general, the types of tests that will be performed for the prototype are unit testing (using Jest Framework) and manual testing. Unit testing is used to assert expected output under defined input of isolated functions. Manual testing is used to test the overall system (from request to respond).

```
const sum = require('./sum');

test('adds 1 + 2 to equal 3', () => {
  expect(sum(1, 2)).toBe(3);
});
```

Figure 9 Unit test performance example

In Prototype 1, we tested if the radius function works properly (find location). In Prototype 2 we tested out the functions of adding a recycling bin, putting the map into full screen, and returning to the home screen as well as the location function just in case an error occurs

because of the added functions. In Prototype 3, we want to test out all the functions that were tested in the previous prototypes.

### **3.1.2. Possible types of results**

#### **3.1.2.1. Find location:**

The user needs to insert valid coordinates to find a location. Coordinates have boundaries. So, if the function is developed correctly, then no errors will occur when acceptable latitude and longitude is passed to the function. When invalid latitude and longitude is passed, we expect the server to respond with an error message.

#### **3.1.2.2. Adding recycling bin:**

When this function works, a bin can be added to the desired location on the system's map with no glitches. In case if bin cannot be added to the database by the server, browser console will print the error. Currently there is no error message on the UI.

#### **3.1.2.3. Map into full screen:**

When the function works, the user will click on the "full screen" icon and the top bar will vanish and therefore, the map will enter full screen. The feature does not rely on database, and therefore no error can occur during hiding and displaying process.

#### **3.1.2.4. Return to home screen:**

When the function works, no errors will occur. The user will be able to return to the home location (current geographical location) by simply clicking on the "home" icon. If this function fails (geolocation of person cannot be obtained), location will not be changed.

### **3.1.3. How results will be used to make decisions or select concepts**

Positive results will certainly increase the functionality of the prototype, but even if the results are errors, we will know what the system can handle and can therefore take appropriate actions. Since this is the last prototype, it is very important that all errors are corrected.

### **3.1.4. Criteria for test success or failure**

Whenever unit test or manual test fails, this indicates that the system has a serious bug which has to be fixed. Also, the feedback from the client is critical. If the client does not like the final prototype, then we are not meeting the expectations. In the recent client meet, the client liked what we presented and did not suggest anything to add to the prototype. The feedback from users is also very helpful to identify if the prototype is successful since ease of use is a priority in the prototype.

## **3.2. What is the prototype and what is the test?**

### **3.2.1. Type of prototype and reason of selection**

This prototype is *comprehensive* because all 4 functions were tested out and for the first time all together: finding location, adding bin, entering full screen, and returning to home page. The map is considered the “heart” of the prototype since the goal is to help users identify their recycling waste and find recycling bins near them. All the current functions are focused on the map.

Because we have progressed with the prototype (there is more than one function), we have a *comprehensive* prototype.

### **3.2.2. Description of testing process**

For the testing of the radius (find location), manual system testing was done. Http request with invalid parameters was sent to the server and expected error was return. However, when http request with valid permeameters was sent, the expected result was received (bins within the radius were fetched). For the rest of the functions (adding bin, full screen, and return to home screen) manual system testing was done as well. By applying manual testing, it is possible to determine immediately if the functions successfully work.

### **3.2.3. Required materials**

Node.js, Express Framework, ReactJS, and other open-source libraries.

### **3.2.4. Approximation of estimated cost of the prototype**

As mentioned in Deliverable E and the previous prototypes, any material used for the prototypes is free because this is a student project and all the remaining technologies involved are free and open-source software.

### **3.2.5 Work to be done (Testing, construction, modeling work, research)**

All the required technologies are researched, and the software architecture is modelled. API subsystem is capable of fetching bins within the radius and adding new bins to the system. UI subsystem can display the map, overall user interface and add fetched bins on the map.

## **3.3. How is the prototype used?**

### **3.3.1. Recorded information**

The behavior of the system, and failure of isolated function, or complete features.

### 3.3.2. How the results will be recorded

The result will be logged in console and it will be clearly seen when the system fails.

### 3.3.3. Is this important data for the project?

The radius is not functional part of the project, but it indicates the overall quality of the software and functionality of certain features.

Fetch and add bins are functional components of the system, and therefore testing these features will represent the overall quality.

While UI is an important part of the system, the entire UI is only a gateway to API. Therefore, failures of the UI do not indicate overall failures of the system.

## 3.4. When is the testing happening and how long will it take?

### 3.4.1. Duration of test and dependencies

The unit tests must be developed before they can run. In this case, test cases will be developed using jest framework before after or during the functional elements of the system are developed. The manual testing will be developed after every new feature is added.

### 3.4.2. When are the results required

The results from the testing are required before the due date of this deliverable, which is March 28<sup>th</sup> 11:59 pm. First, the results need to be gathered and then are added in this deliverable.

### 3.4.3. Test planning Gantt chart

For the protype stages, the team is still following the same Wrike Gantt chart that was created in Deliverable E and used for the previous prototypes.

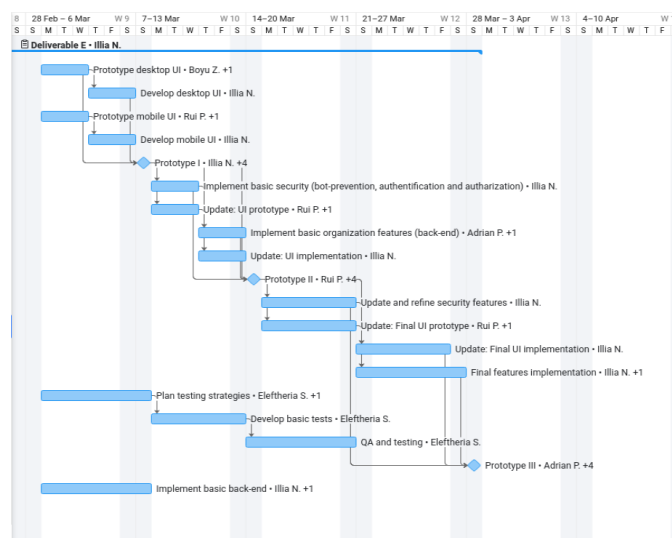


Figure 10 Gantt Chart Sample for Prototyping Stages

A different Wrike chart though has been created to finish writing this specific deliverable (submitted in separate PDF document).

#### **4. User Feedback**

Feedback from users indicated that the application had the level of polish that the team was hoping for, but that there was still some room to improve. Positive comments mentioned that the application was “easy to use”, “very intuitive” and “fast to use”. The general opinion on the user interface was that it was simple and user friendly. In terms of performance, it was also agreed on that the web app responded quickly, and that the overall experience when interacting with the app was fluid. Some room for improvement was in minor tweaks to the UI, as some UI elements were clipped off slightly on mobile devices. There were occasional query errors as well. In terms of actual content, it was mentioned by most users that the system simply lacked bins to be able to search for. While this is an issue at the moment, it is only the case as the application is being tested, and therefore the database of test bins is small due to lack of user entered bins. This would not be a problem in a real-world deployment of this solution.

#### **5. Conclusion**

For the final prototype, the team has taken prior user and client feedback and used it to polish previously developed features. Over the several iterations of the project, the design has approached the desired level of polish and functionality that fits the previously defined requirements and problem statement. The initial prototype showed explosive growth, with a focus on developing all the core features in one go. As seen in both the previous and current iteration, the focus has instead switched to refining the features already added. This ensures a polished final product, that is not spread too thin between an overwhelming amount of features and has let the team focus on making each component of the product not rely on other paywalled work, making this solution the easiest to scale, rework, and deploy for any company desiring a similar solution.