Group C15

Deliverable D – Conceptual Design

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

List of Figures

List of Tables

Abstract

This report establishes 3 global concepts, meant to solve the issues highlighted in the problem statement. From the 3 global concepts, further benchmarking is done to better understand how well each concept solves the problem, based on our four chosen criteria of what is most important: Having a simple user interface, improving the recycling sortation process, being easy to scale, and having good documentation. Each team member has presented ideas and from them we identified 3 unified main concepts. From this benchmarking, Solution 1 (a public web API with a provided UI) best solves the problem at hand.

1. Introduction

In this report, our group proposes conceptual designs based on the problem statement, benchmarking, and the list of prioritized design criteria. Our five group members put forward five different plans according to the four concepts that are *User interface, Identification of recyclable material and where it should go, Scalability, and quality Documentation*. In addition, each of our solutions focus on fulfilling customer service needs and solving customer problems. After group discussion, by learning each other's strengths and weaknesses, the rudiments of project design are formed.

The team's conceptual design is analyzed and evaluated, and the solutions are compared in the technical benchmarking process. To facilitate understanding and lay the foundation for subsequent design, our conceptual design proposal is described through pictures and text, and the technical benchmarking process is presented as a table. Our group decided to use a web API to make our service more accessible to more platforms.

2. Individual Ideas

2.1. Boyu's Idea

2.1.1. User interface

The user interface should be brief and intuitive and should directly show that the Web page's function is to help people sort garbage and find garbage bins. In addition, in the page design, green highlights the concept of sustainable development of environmental protection.

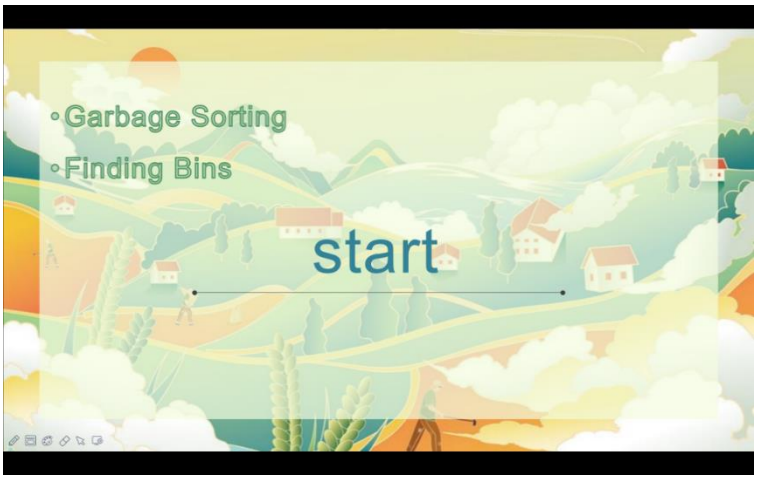

Figure 1 created by Boyu Zhao

```
#include <stdio.h>

int main()
{
	char x[30];
	char flag;


	int y;

	do
	{
		printf("What garbage do you want to drop?\n");
		scanf("%s",x);
		if (x=="magazine"||x=="book"||x=="can")
			{y=1;}
		else if (x=="fish"||x=="fruit"||x=="rice")
			{y=2;}
		else if (x=="woodchip"||x=="plastic"||x=="wax")
			{y=3;}
		else
			{printf("invilid garbage\n");}

		printf("this garbage goes to");
		switch(y)
		{
			case 1:printf("blue");break;
			case 2:printf("green");break;
			case 3:printf("black");break;
		}
		printf("garbage bin\n");
		printf("Would like to go again?(y/n)");
		scanf("%s",flag);
	}
	while (flag=='y');
		return 0;
}
```

2.1.2. Identification of recyclable material and where it should go

Users directly input the garbage they want to classify. If the garbage is in the database, the interface will directly display the colour of the garbage bin corresponding to the garbage. In addition, users need to input their approximate position, and the interface will display the location of the garbage nearest to the user.

Users typically want to sort multiple types of garbage, so the garbage sorting interface will loop through to make sure the user has sorted all the garbage.

Figure 2 created by Boyu Zhao

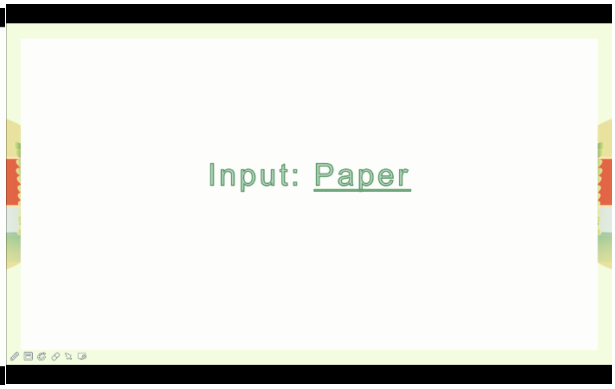

Figure 3 created by Boyu Zhao

Figure 4 created by Boyu Zhao

Figure 5 created by Boyu Zhao

Figure 6 created by Boyu Zhao

Figure 7 created by Boyu Zhao

2.1.3. Scalability

We cannot guarantee that the database contains all kinds of garbage, so user feedback is very important to us. Therefore, if the user cannot find the corresponding garbage can, we hope that the user can contact the developer to improve the database.

Figure 8 created by Boyu Zhao

2.1.4. Documentation

When the user does not know how to do it, the user can click “Help”, and then a detailed explanation will appear to help the user use it. When the user wants to end the use, all the user must do is click the “×”.

Figure 9 created by Boyu Zhao

2.2. Eleftheria's Idea

2.2.1. User Interface

The user will access a web service and the main function of this will be to search and locate recycling bins. More specifically, on the main page, the user will enter a type of recyclable material in the search tool and when they click on the search icon or they press "Enter," a map will be loaded which will automatically show the user's nearest recycling bin locations. Also, at the lower left corner of the map, there would be a button named "Directions" and when clicked, it will show the directions to those nearest bin locations.

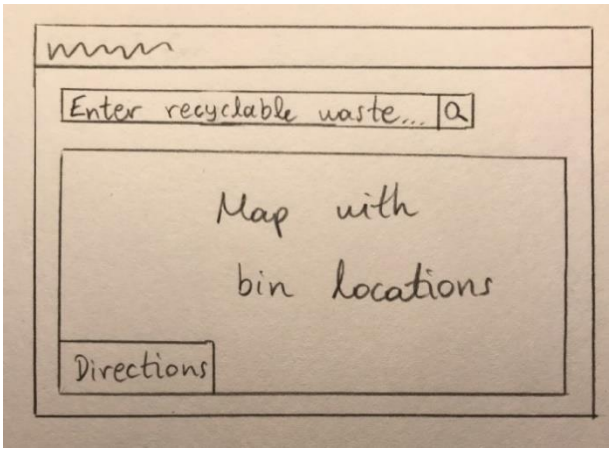

Figure 10 created by Eleftheria Sarsaroudi

Similarly, to the previous concept, the user will access a web service, but the only difference is that on the main page of this, there will be two search tools: a search tool for entering a type of recyclable material and another search tool for entering a specific location.

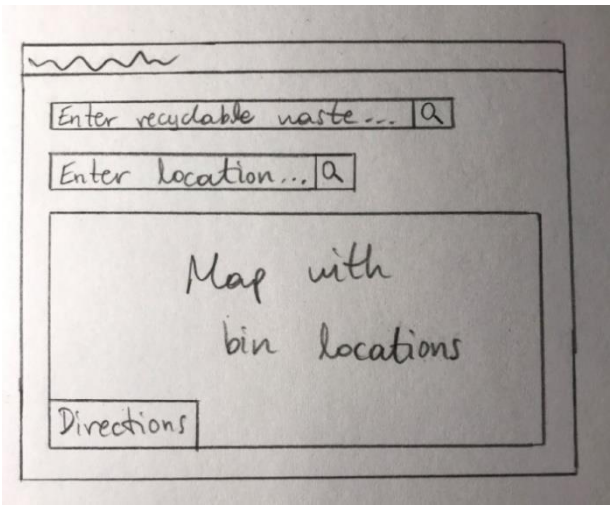

Figure 11 created by Eleftheria Sarsaroudi

2.2.2. Identification of recyclable material and where it should go

When the user accesses the web service, further down on the main page there will be a document which lists which materials are recyclable (so whatever is not on the list is simple waste) as well as another document indicating in which coloured-bins the recyclable materials belong in (currently for coloured-bins in Ottawa). Another alternate is to click a button on the main page and a different page will open showing the two documents. Also, the user could have the option to download the documents, if they want to, by clicking the download icon on the top right corner of each document.

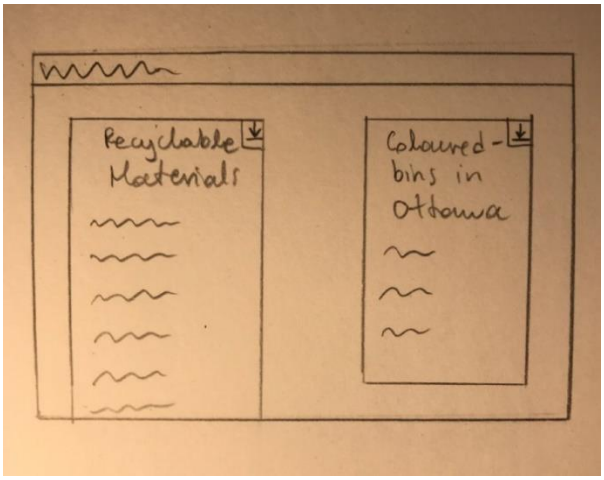

Figure 12 created by Eleftheria Sarsaroudi

Another concept for the Identification Subsystem is to have a comment section further down on the main page of the web service or on a different page by clicking on a button. In the comment section users can help each other by asking questions and/or providing answers regarding recycling. Therefore, users can help by giving tips on where bins are exactly located in an area (i.e., the bin at Marbel St is behind the big tree next to the store).

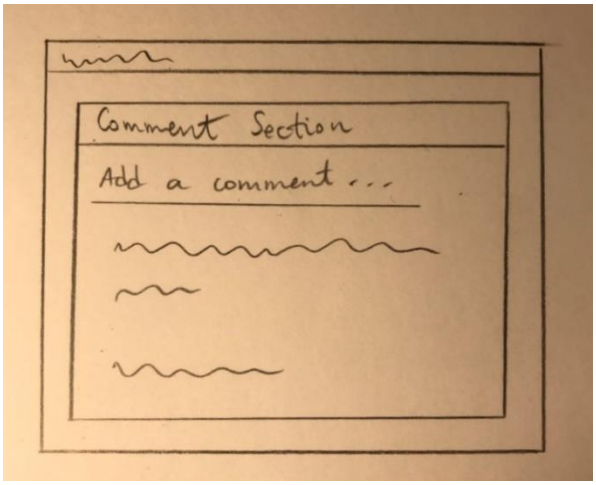

Figure 13 created by Eleftheria Sarsaroudi

2.2.3. Scalability

To attract more people to the web service, there could be a contest where a winner is randomly selected, for example, every 2 months. To enter the contest, the user will have to sign in with their email address and password, so an email can be sent to the user if they are the winner.

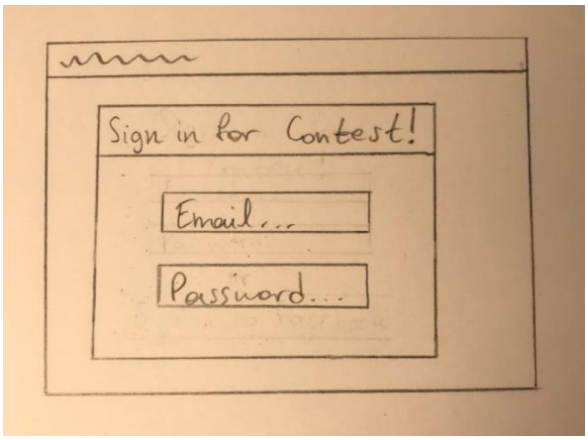

Figure 14 created by Eleftheria Sarsaroudi

2.2.4. Documentation

As soon as the user visits the web service, a message pops up which includes brief instructions on how to use the web service. The user can then click on the “x” on the top right corner to exit the pop-up message. These instructions can be accessed again by clicking on “Instructions” on the top right corner of the page and then the same pop-up message will appear.

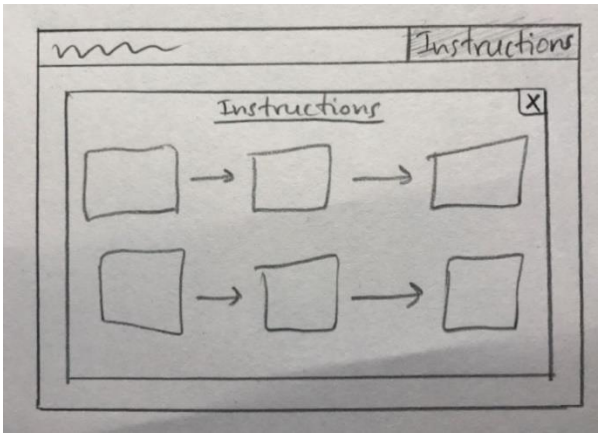

Figure 15 created by Eleftheria Sarsaroudi

2.3. Illia's Idea

2.3.1. User Interface

While the web-based systems is aiming to be flexible and integrable into any form of UI or enterprise bus, the basic UI will be provided for the users to improve their recycling experience. UI will allow user to search for the closest bin with applying some filters.

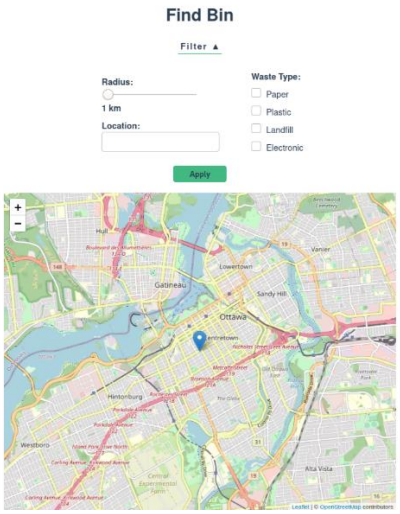

Figure 16 created by Illia Negovora

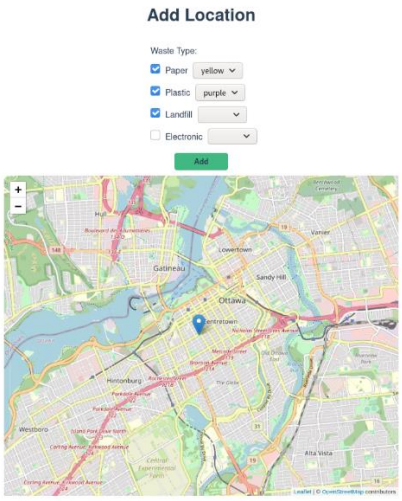

Figure 17 created by Illia Negovora

2.3.2. Identification of recyclable material and where it should go

Web based solution allows to search for a specific type of bin within the particular radius. In addition, when locations are filtered one can see the location of each bins that fulfill the search parameters and can access additional information, including: bin color, how full the bin is reported to be and even organization responsible for servicing the bin.

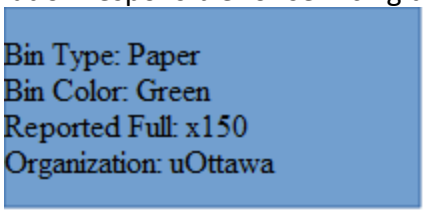

Figure 18 created by Illia Negovora

2.3.3. Scalability

Web service implemented in form of REST API using modern web technologies such as Node.js. Nature of RESTful API is such that it is easy to integrate into applications, websites and even enterprise grade infrastructure. Using Node.js allows for asynchronous processing of requests, which allows to scale application, and with good DevOps practices it can be scaled to the international level.

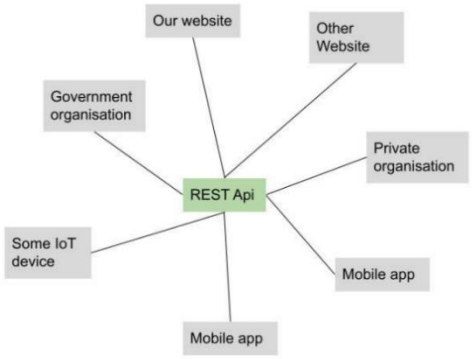

Figure 19 created by Illia Negovora

2.3.4. Documentation

Documentation must contain comprehensive description of API and relevant object and functions of application. Documentation will be posted and hosted on GitHub and will include API end points, query parameters, JSON post request description, etc.

```
# API Documentation
- To fetch bins within particular radius from certain location:
/public/bins?lat=<latitude>$long=<longitude>&rad=<search radius>

- To add new bin, make a post request with content type `application/json` with json following format"
```
"latitude": <latitude: float>
"longitude": <longitude: float>
"color": <bin color: string>
"type": <type of recyclable material: string>
```
```

Figure 20 created by Illia Negovora

2.4. Rui's Idea

2.4.1. User Interface

A user agreement will be shown when user first starts the application.

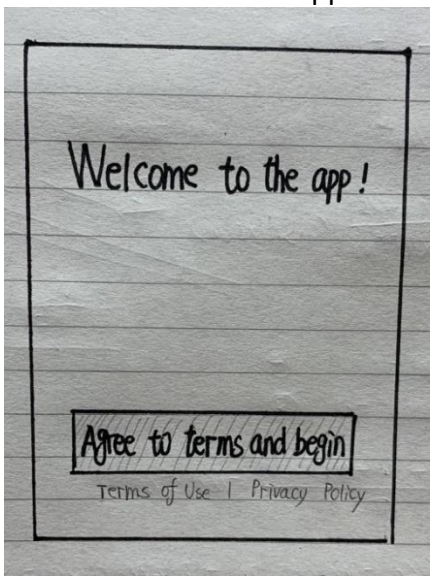

Figure 21 created by Rui Pang

Users need to give permission for photos, camera and the location before using.

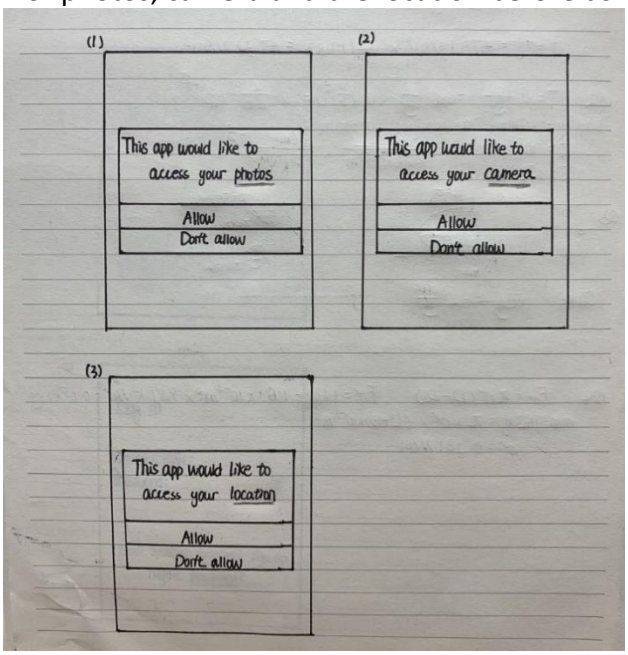

Figure 22 created by Rui Pang

2.4.2. Identification of recyclable material and where it should go

Just take pictures of rubbish and it can quickly recognize the type of garbage and which section or coloured-bin it belongs to. Based on your location, the application will provide the detailed addresses of bins that automatically listed in an ascending order in distance. Sometimes, if it cannot be recognized, we can manually search by the garbage type. “More Information" shows what type of trash corresponds to what coloured-trash bin, as well as the updated garbage classification policy. Users can get more information about the garbage classification with ease.

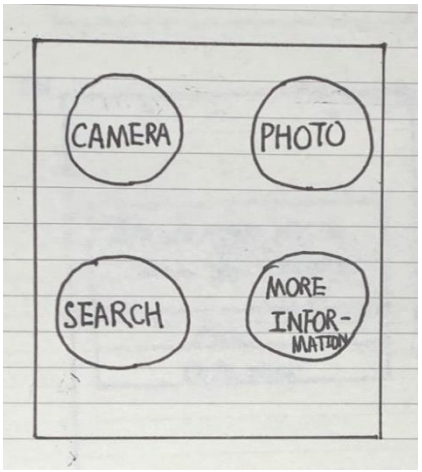

Figure 23 created by Rui Pang

2.4.3. Scalability

When a user enters the application, the app will ask for user’s log-in information to access the user's account or create a new account. This is a free software provided to users. If they do not want to register an account, they can click "skip".

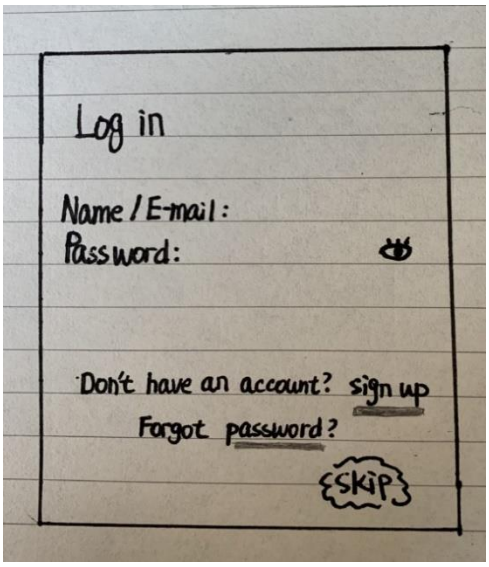

Figure 24 created by Rui Pang

2.4.4. Documentation

For example, user clicks the "photo" and selects one of the pictures to recognize. Then, the application automatically presents the name and the type of garbage, as well as the brief roadmap for user to find the near bins. Users can click "Back" and turn back to the homepage for searching next one.

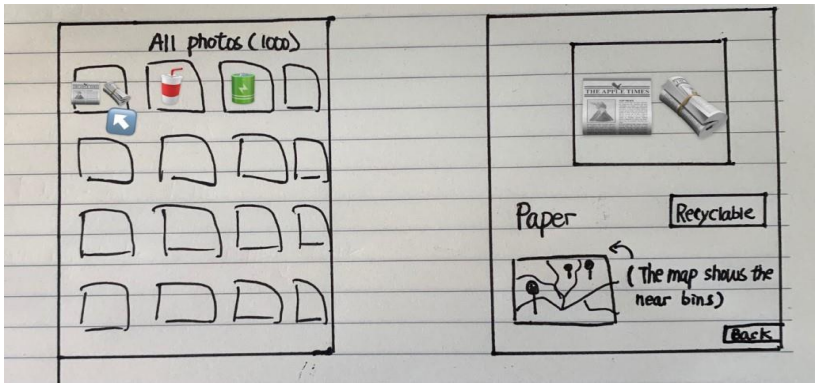

Figure 25 created by Rui Pang

2.5. Adrian's Idea

2.5.1. User Interface

The user interface would be a small set of LED lights above each bin, and a small indicator that tells you where to hold the object that you are trying to recycle (so the device can take a picture of it). These would all be in the front panel of a recycling bin, to point the user towards the correct bin for their specific item.

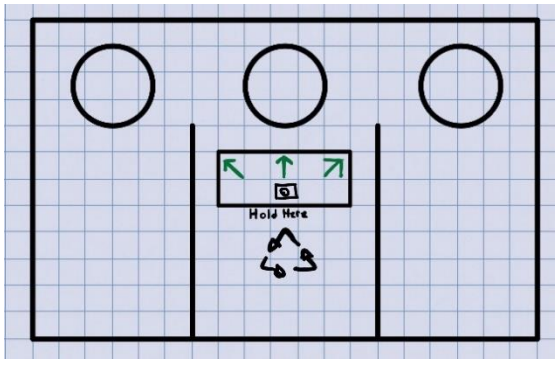

Figure 26 created by Adrian Perras

2.5.2. Identification of recyclable material and where it should go

This solution would use machine learning to recognise objects in the picture taken by the front facing camera, which could either be processed locally, or sent to a more powerful computer

for image processing. Using machine learning for object recognition means that the software can get better with more training to the algorithm, allowing us to recognize more objects over time.

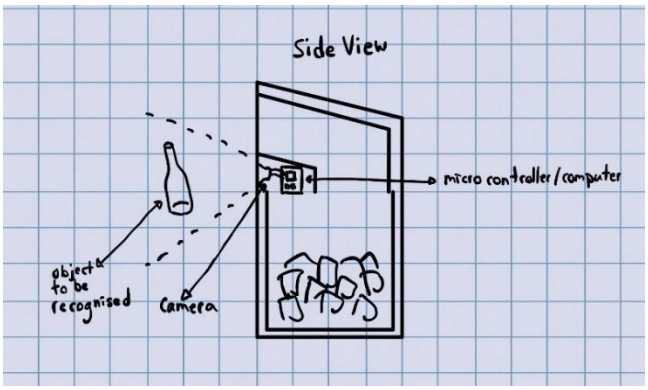

Figure 27 created by Adrian Perras

2.5.3. Scalability

Scalability of hardware is generally more challenging than an app, as there are upfront costs of manufacturing and a need to sell the product, relying on consumer or commercial interest. However, there is also greatly reduced “friction” for the user, as you do not need to open an app or use any service. This would likely be something that a municipality would purchase, meaning that everyone in that area would automatically be adopting it just by using these new recycling bins installed around the city/town/etc. There are also two options: on-board processing or server-side processing. By doing all the data processing locally, you do not need to have it connected to the internet, making it easier to have these spread out over a city. However, costs would perhaps increase as the computer would need to be more powerful and have more memory to store the machine learning models. Or by doing server-side processing, you can make cheaper computers do the work, and offload it to a separate server. This will decrease upfront costs but make it harder to scale, as the recycling bins with this will need to be connected to the internet.

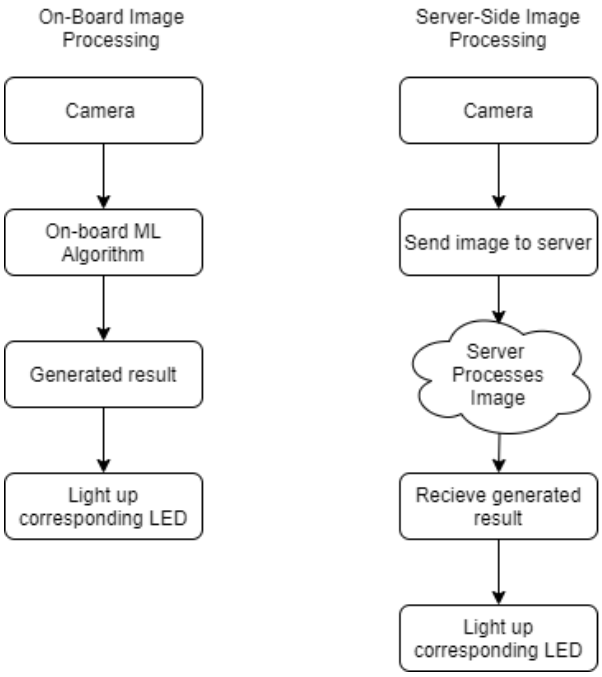

Figure 28 created by Adrian Perras

2.5.4. Documentation

This solution would not necessarily require documentation, as the purpose of the service is to remove the need of a consumer to have to think about the recycling process. For those who do want extra resources, there could be a QR code on the front of the Recycling bin that directs them to a website giving information about how the recycling bin system works, and further details on how it is implemented in your area.

3. Global Designs

After collecting and discussing all the concepts for all subsystems, we have developed three global solutions based on the team's input. To clearly explain each of the solutions, we will look at each subsystem that the solution is made of.

3.1. Solution 1: Public web API with UI

This solution will help users find and identify the proper recycling bin by providing precise location and detailed information about the bin. The information about the bin will include the bin's colour, the organization responsible for the bin and additional information provided by users in the case recycling bin is not easy to find. In this way, the user will always be able to identify the correct container, even if it lacks the necessary identification. The user will also have access to local rules and regulations related to recycling based on geographical location.

The web-based solution will come with a well-designed UI, including large buttons, labelled input fields and an intuitive interface. Responsive design will be a priority, with a strong emphasis on mobile devices. Also, UI will become a gateway for organizations and developers to register and start using the public API.

Scalability is a priority for the selected solutions. To ensure that our solution addresses a wide range of problems, strong emphasis will be made on reusability and ease of system integration into existent software infrastructure. Therefore, REST architecture was chosen for the development of the API portion of our system. This approach will ensure that any organization will rapidly deploy our service and use it to manage and monitor their recycling bins. It will also allow independent developers to integrate our system in applications, websites, digital devices, and even embedded devices. In addition, to attract more people, users who add and query the most bins will have a reward system, allowing to collect coins or similar virtual rewards.

Documentation is another essential element of the solution. Since public API is a critical element of the system, the interface should be well documented and readily available for developers. API documentation will be stored and hosted alongside the source code of the application on GitHub. Every query parameter and API endpoint will be listed as well as requirements for the JSON data for every post request to the service.

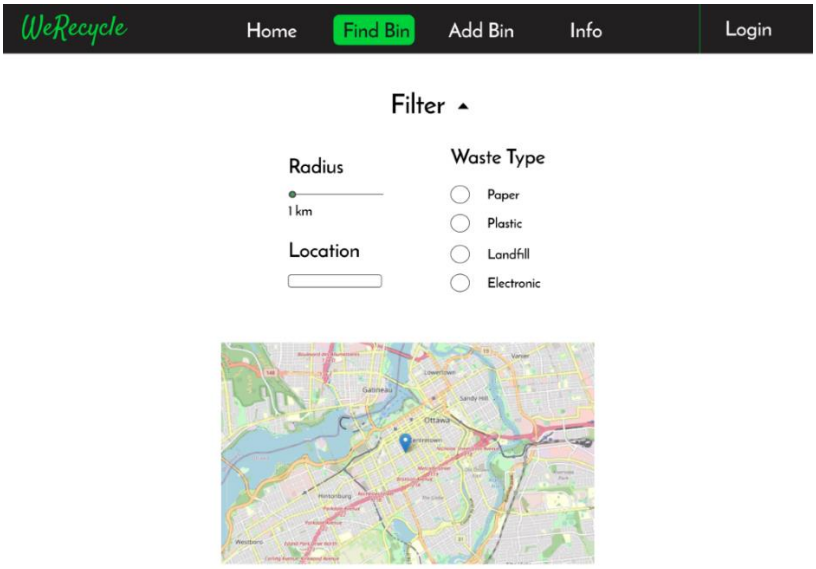

Figure 29 Solution 1 UI Example

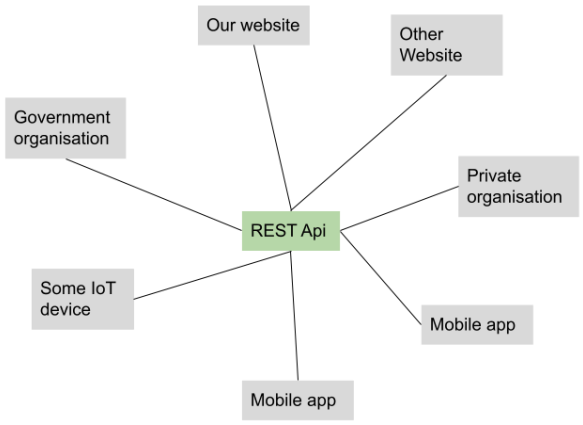

Figure 30 Solution 1 Logic Diagram

```
# API Documentation
- To fetch bins within particular radius from certain location:
/public/bins?lat=<latitude>$long=<longitude>&rad=<search radius>

- To add new bin, make a post request with content type `application/json` with json following format"
```
"latitude": <latitude: float>
"longitude": <longitude: float>
"color": <bin color: string>
"type": <type of recyclable material: string>
```

*Figure 31 Solution 1 Documentation Example*

## 3.2. Solution 2: Mobile Application

The mobile application will allow users to find an appropriate recycling bin by merely scanning the item. The application will then look for the closest recycling bin that is suitable for the item. The system will perform a search and return all the relevant information, including bin colour, location, etc. In such a way, user will be able to find appropriate recycling bin quickly and efficiently.

UI of the mobile application is the most critical element of this solution. Instead of building an application around features, we will develop our mobile application around simplicity. It will include a small number of large and visible buttons, with the scan button in the middle of the screen. Once the item is scanned, clear direction and additional information will appear on the screen. Further information will include bin colour and other directives.

The scalability of a mobile solution is not relying on reusability but on widespread usage. The application will be developed for the major mobile platforms, possibly using cross-platform technologies (React Native). The application will also be distributed under a permissive open-
```

source license such as MIT, so other developers can add features or create new applications based on our solutions. In such a way we will ensure the scalability of the application.
While the features of the application will be developed to be clear and intuitive, proper documentation can help with application support and modifications. Therefore, every object and element of application will be clearly documented. In addition, code will be commented appropriately, including all essential algorithms.

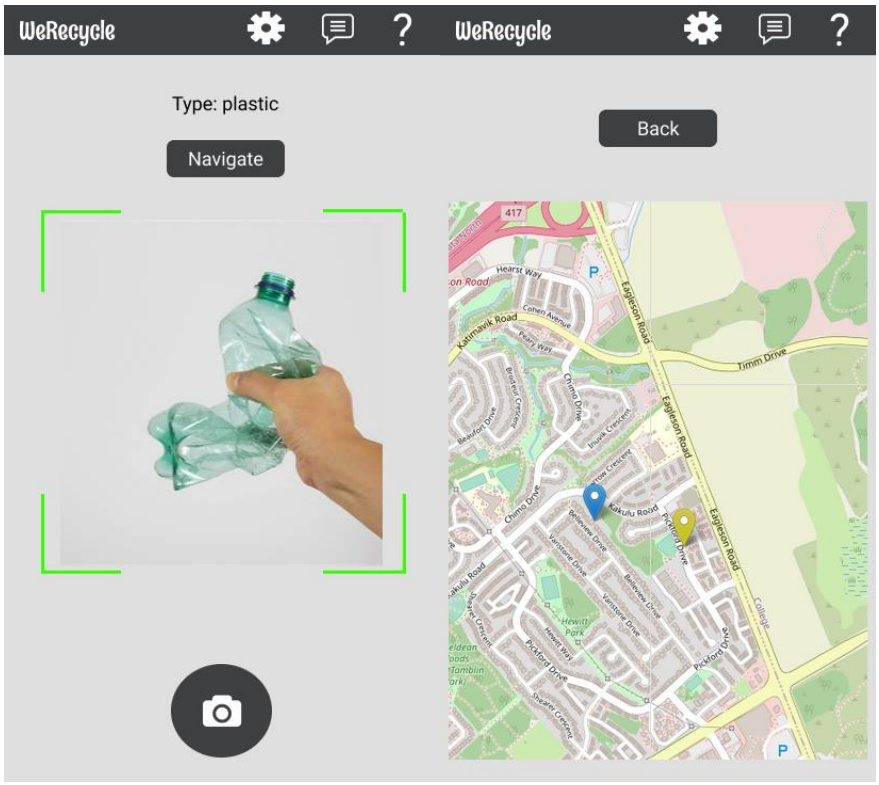

Figure 32 Solution 2 UI Example *Figure 33 Solution 2 UI Example*

3.3. Solution 3: Open-Source Library

This product is a software library that allows to scan and identify the type of recyclable product. With this library, an organization can choose what information will be displayed to the user and how it will be presented. In such a way, our system allows the organization to precisely adjust how it will help users find a correct recycling bin.
UI of this system is very flexible since the UI's implementation is delegated to the organization and independent developers. For example, this system can be easily integrated into digital advertisement boards and kiosks, IoT devices and small embedded systems. The UI of the system is entirely up to the organization.

Software library solution is highly scalable. Since it is an open-source library available for any organization or developer, many new products can be developed based on our solution. The library source code will be hosted and distributed using GitHub, making it a free project to develop since no hosting budget is required.

This solution requires the most comprehensive documentation. API, library objects and functions should contain documentation for organizations and developers. Besides, code should contain comprehensive and descriptive comments that will allow developers to use the library effectively.

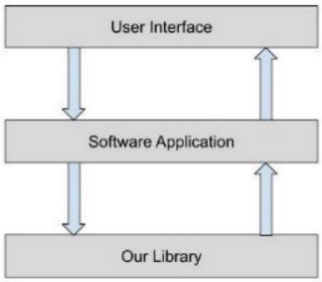

Figure 34 Solution 3 Logic Diagram

4. Benchmarking

In the previous deliverable (Deliverable C), our team researched applications regarding the collection stage of recycling. Technical benchmarking was done to identify which application was more suitable that will influence our prototype for this project. The result was that our prototype will follow the 3 apps Intellibins[1], BC Recyclepedia[2], and iRecycle[3], since they scored the most points as well as the same number of points. Though, all three are apps that the user will need to download on their electronic device. Therefore, our team came up with three similar but different fully functional designs, as mentioned previously, that will help citizens recycle properly: Public web API with UI, Mobile Application, and Open-Source Library.

The following table benchmarks the three global designs (fully functional solutions) to identify which one will be the prototype of our project. It is obvious that the best global design is the first one, public web API with UI. It is important to note that changes to the prototype may occur, therefore any changes will be mentioned in the following deliverables, if necessary.

Table 1: Global Design Prioritization and Ranking

Specifications	Importance (out of 5)	Global Design 1: Public web API with UI	Global Design 2: Mobile Application	Global Design 3: Open-Source Library
Assists with recycling	5	Yes	Yes	Yes
Cost	5	Not free	Not free	Free
Ease of use	5	Yes	Yes	Yes
Platform	4	Web service	Android and iOS	Cross-platform for single board computers
Scalability	4	Yes: Integration of API with any system and contest	Yes: Cross-platform (React Native) and Open-source license such as MIT	Yes: Development of new products based on our solution (GitHub)
View of recycling locations (i.e., map etc.)	3	Yes	Yes	No
UI/UX	5	Yes: Buttons and search forms, no use of camera	Yes: Buttons and use of camera for identification	Yes: Use of camera for identification
Tutorial/ Instructions	3	Yes	Yes	Yes

Table 2: Global Design Numerical Ranking

Specifications	Importance (out of 5)	Global Design 1: Public web API with UI	Global Design 2: Mobile Application	Global Design 3: Open-Source Library
Assists with recycling	5	3	3	3
Cost	5	2	2	3
Ease of use	5	3	3	3
Platform	4	3	3	3
Scalability	4	3	2	3
View of recycling locations (i.e., map etc.)	3	3	3	1
UI/UX	5	3	3	1
Tutorial/ Instructions	3	3	3	3
Total		**97**	93	86

5. Conclusion

The team has generated 5 ideas, which were condensed down into 3 global concepts: a public web API, mobile application, and an open-source library for embedded systems. Evaluating the 3 global concepts against the 4 main criteria defined by the team, weights were given to each category and a numerical rating for each was generated. Collecting these weighted rankings for each concept, we were able to identify that the web API best solved the needs defined in the initial problem statement, and so this is the chosen solution.

References

[1] "Intellibins helps you find recycling bins on city streets," *Inhabitat Green Design Innovation Architecture Green Building*. [Online]. Available: https://inhabitat.com/intellibins-helps-you-find-recycling-bins-on-city-streets/. [Accessed: 06-Feb-2021].

[2] Rcbc, "BC Recyclepedia Smart Phone App," *Recycling Council of British Columbia*, 14-Jan-2016. [Online]. Available: https://www.rcbc.ca/services/recyclepedia-app. [Accessed: 06-Feb-2021].

[3] "Download the iRecycle App," *Earth 911*, 29-Jul-2019. [Online]. Available: https://earth911.com/irecycle/. [Accessed: 05-Feb-2021].