

**Development of a User-Friendly Interface and Database to
Increase Waste Packaging Recycling**

by

Yichuan Xu

Thesis submitted in partial fulfillment of the requirements for the
Degree of Bachelor of Computer Science with Honours

Acadia University

Spring Convocation 2014

© Yichuan Xu, 2014

This thesis by Yichuan Xu

is accepted in its present form by the

Jodrey School of Computer Science

as satisfying the thesis requirements for the degree of

Bachelor of Computer Science with Honours

Approved by the Thesis Co-supervisors

Dr. Darcy Benoit

Date

Dr. Paul Arnold

Date

Approved by the Acting Director

Dr. Andre Trudel

Date

Approved by the Honours Committee

Dr. Matthew Lukeman

Date

I, Yichuan Xu, grant permission to the University Librarian at Acadia University to reproduce, loan or distribute copies of my thesis in microform, paper or electronic formats on a non-profit basis. I, however, retain the copyright of my thesis.

Yichuan Xu

Date

Table of Contents

Table of Contents	i
List of Figures	v
Abstract	vii
1 Introduction	1
1.1 Organization	5
1.2 Related Work	6
2 Software Specifications	7
2.1 Introduction	7
2.1.1 Purpose	7
2.1.2 Scope	7
2.2 Overall Description	8
2.2.1 Product Perspective	8
2.2.2 Product Function	8
2.2.3 User Aptitude	9
2.2.4 Constraints	9
2.3 Detailed Requirements	9
2.3.1 External Interfaces	9

2.3.2	Functional Requirements	9
2.3.3	Performance Requirements	10
2.3.4	Logical Database Requirements	10
2.3.5	Design Constraints	10
2.3.6	Use Case Diagram.....	11
3	Database Design and Data Collection.....	12
3.1	Overview	12
3.2	E-R model.....	13
3.2.1	Entity.....	13
3.2.2	Relationships.....	14
3.3	E-R Diagram Design	15
3.4	Database Tables.....	16
3.5	Data Collection.....	17
4	Website	17
4.1	Purpose	17
4.2	Tools.....	18
4.2.1	HTML	18
4.2.2	CSS	19
4.2.3	PHP	22
4.2.4	jQuery	26

4.3	Interface Design	29
4.4	User Login.....	36
4.5	Database Security	36
5	Android Application	37
5.1	Purpose	37
5.2	Functions	37
5.3	Tools.....	37
5.3.1	Java	37
5.3.2	Android SDK	38
5.4	Mobile Application Implementation	38
5.4.1	UI (User Interface) Design.....	38
5.4.2	Activities.....	42
5.4.3	Layouts.....	42
5.4.4	Barcode Scanner	43
5.4.5	Internet Connection.....	43
5.4.6	User Preference	43
6	Product Manual.....	44
6.1	Website.....	44
6.1.1	Information Retrieval.....	44
6.1.2	Registration.....	44

6.1.3 Adding new items	45
6.1.4 Modify or Remove Items	45
6.2 Mobile Application.....	46
6.2.1 Scan.....	46
6.2.2 Choose Region	46
6.2.3 Retrieve Product Information	47
7 Constraints	48
8 Conclusion and Future Work.....	49
9 References.....	50

List of Figures

Figure 1-1 Recycling Information Retrieval Process.....	4
Figure 1-2 Sample Data Format.....	6
Figure 2-1 Use Case Diagram For Website	11
Figure 2-2 Use Case Diagram For Mobile Application.....	11
Figure 3-1 An Example Of Entity-Relationship Model.....	13
Figure 3-2 An Example Of Entity In E-R Diagram.....	14
Figure 3-3 Many-To-Many Relationship.....	15
Figure 3-4 E-R Diagram	15
Figure 4-1 Hello World HTML Example	19
Figure 4-2 HTML With CSS	20
Figure 4-3 HTML Without CSS	21
Figure 4-4 jQuery Example- Default Number Of Constituents.....	27
Figure 4-5 jQuery Example- Updated Form.....	29
Figure 4-6 Home Page Design	30
Figure 4-7 Sign In Page Design	30
Figure 4-8 Product Page Design	31
Figure 4-9 Item Detail.....	32
Figure 4-10 3D Pie Chart.....	33
Figure 4-11 Tiles On Computers	35
Figure 4-12 Tiles On Mobile Phones.....	35
Figure 5-1 Home Page	39
Figure 5-2 Barcode Scanner Interface	39

Figure 5-3 Choose Region	40
Figure 5-4 Set Default Region	40
Figure 5-5 Item Information	41
Figure 5-6 Settings	41
Figure 6-1 Search Result.....	44
Figure 6-2 User Sign Up.....	45
Figure 6-3 Edit And Delete Button.....	45
Figure 6-4 Camera Button	46
Figure 6-5 Barcode Scanner.....	46
Figure 6-6 Select Region	47
Figure 6-7 Product Information	48

Abstract

Most of the recyclable materials in our waste stream originates from the grocery store, however, much of the non-recyclable waste from a typical household also originates from the same store. In order to maximize the recyclable portion of the waste stream from grocery store packaging waste, consumers must be knowledgeable in distinguishing between what is and what is not recyclable—no easy feat in many cases as products often contain combinations of recyclable and non-recyclable components. This project is intended to provide consumers with an electronic database and interface that tells the consumer what is and is not recyclable for a particular grocery store product. Through the use of each unique UPC code which is either entered manually or scanned, the software discriminates between the various components of packaged products from grocery stores in terms of their recyclability and their correct method of disposal. The database also clearly identifies companies that are producing products in more environmentally responsible packaging and therefore allows consumers to discriminate between products based on the recyclability of their waste.

1 Introduction

Waste disposal has been one of the most serious problems threatening the environment and has become one of the top concerns with municipalities that are tasked with its management. Maximizing waste use is very crucial in the process of sustainable development and in being eco-friendly. In Nova Scotia in general, consumers now have to choose between one of four waste streams—recyclables, paper products, organics and garbage. Unfortunately, when the packaging of products gets more and more complicated, through the inclusion of multiple materials, it becomes more difficult for average consumers to maximize their waste diversion practices. One of the main reasons for the confusion is that consumers often do not know if a certain part of the packaging is recyclable or not. So a complete platform that provides straight-forward information to customers about products and their recyclability will help consumers not only improve waste diversion, but give them the opportunity make informed purchasing decisions based on the relative recyclability of competing products. For this reason, a well-designed website with desktop/laptop access as well as an easy-to-use mobile application that provides easy access to recycling information is in order. To achieve this goal, the project was divided into several components: database design, data collection, website design and mobile application design; these components form the framework of the thesis.

The following sequence of diagrams shows what customers might experience when they are shopping in a store or a supermarket and how using the app will help them to decide which item would be the best to choose.



When customers are purchasing grocery items, it is always not easy to pick the best product especially the items being friendly to our environment.



Our program enables customers to choose the most eco-friendly product and where to dispose of the waste.





Customers enter the UPC code or the product name and all the information about that product will be displayed, which tells customers consumers the recyclability of the packaging as well as which stream the waste is to be placed in.



Users will be able to see the detailed information of the product, as well as the number of constituents and their recyclability. Thus users are able to judge which product is more eco-friendly with the straight-forward calculated percentage of recyclability.



What makes information retrieval easier is that customers can just use their phone to scan the barcode, and information will be displayed on the screen. Customers can make the decision with just several clicks.



Figure 1-1 Recycling Information Retrieval Process [1]

When customers are shopping in a grocery store they may want to find out the recyclability of packaging from a particular product, or find an eco-friendly product among tens of brands of the same product. If they have a smartphone, they can just open the application, scan the barcode of the product they are interested in and the application will return information that tells the customer its recyclability in a very straight-forward way. This helps customers choose the most recyclable product that is best for the environment. The mobile application simplifies all the steps and provides the core information that users may need, which is the recyclability of certain products and which waste stream it is disposed in. Because going to the website and entering the UPC (Universal Product Code) for information will be tedious for most users away from their desktop, using the scan function on a mobile phone will considerably shorten the time to find out about the recyclability of a product and the appropriate container. And since the recyclability of constituents varies among different regions within the Province of Nova Scotia, users are able to choose the region and set it as their default region so as to save time when using the app repeatedly.

In addition, when users are at home, they might want to learn about the recyclability of products they have just purchased. They can go to the website which provides a very user-friendly web layout for them to check information about the product in question using the UPC bar code as the unique product identifier. The website provides users with a platform where they are able to find information about products, read news and learn about recycling. The site is device-responsive so no matter what device users are using (desktop, tablet or mobile phone) they will all have a comfortable experience on this website. Not only can consumers use the information from the website, they can also be part of it. The registration function allows authorized users to sign up a new account and use the account to add or modify the product items stored. More detail about the website will be introduced in the later sections.

Behind the website and mobile application lies the database, which is used to store all the information related to the entire product, including product information, recyclability information, and user account details. The database design will be introduced in Section 3.

1.1 Organization

This thesis is organized sequentially. Section 2 describes the detailed requirements for this project based on IEEE protocols. Sections 3, 4 and 5 describe the implementation of the database, website and mobile application respectively. Section 6 introduces how to use this product while Section 8 concludes the thesis with product improvement recommendations.

As an important part of this project, the user-friendly website plays an important role in providing direct interaction with average consumers. To achieve this goal, PHP and jQuery were the main scripting languages as well as HTML and CSS. To provide consumers with immediate information at the point of purchase, a mobile application was developed to help consumers choose products based on the recyclability of their packaging. The Android platform will be the platform of initial choice since it is the most popular mobile operating system in the world at present.

1.2 Related Work

To collect data, information from 500 items were recorded after being purchased from a number of grocery stores. The packaging was saved and separated into the most basic constituents, then weighed and recorded in a formatted Excel spreadsheet along with the product information. Finally, the data were entered into the database for future query-use purposes. Figure 1-2 is an example of the sample data format.

Waste packaging constituents	Mass (g)	% of total product	Class	Recyclable?	Accepted Recyclable Waste (g)	Recyclable Waste (g)	Container
Laminated paper bag	35	10%	Plastic laminated paper	No			
Wire tie	4	1%	Paper coated wire	No			
Cellophane sealant bag	6	2%	Cellophane	No			
Polystyrene tray	15	4%	#6 plastic	Yes	15	15	blue bag (containers)
Totals	60	17%			15	15	
User Input: UPC code	345627001						
County	Kings						
Output: Product image							
Product	Christie cookies 350 g						
Parent Company	Mondelez International						
Class	Cookies						
Weight product (g)	350						
Weight packaging (g)	60						
Packaging/product %	17%						
Recyclable Waste/Total Waste %	25%						

Figure 1-2 Sample Data Format

2 Software Specifications

The following software requirements specifications are based on the standards established by the IEEE [2].

2.1 Introduction

2.1.1 Purpose

To build a website with internet access through desktops, laptops and mobile phones that contains at least 500 items of product packaging and show which constituents are and are not recyclable, recording results on a mass basis as well as on a percentage basis.

2.1.2 Scope

The website was implemented with PHP and MySQL as the database. The website was designed for those users who want to know about the composition of certain products and need help to decide which product is more eco-friendly. In addition, this site can have a positive influence on those companies producing packages. All the information was displayed in a visual way for easy interpretation.

2.2 Overall Description

2.2.1 Product Perspective

2.2.1.1 System Interface

The site interacts with the browser on a basic level for saving as well as loading cache and cookies. Since it is mobile-responsive, it is public and visible for users to get information at any time.

2.2.1.2 User Interface

The user interface plays a very important part in this site in order to provide the user with a direct and easily understandable appearance that leaves a very good impression. The user is able to select products and compare them among several other products. The proportion of each component within a product is also presented in a pie chart that is easy to read and understand.

2.2.1.3 Hardware Interface

Hardware interface is not applicable in this project.

2.2.1.4 Software interface

Desktop browser and mobile browser are needed to access content from the server.

2.2.1.5 Operations

Operations is not applicable in this project.

2.2.2 Product Function

This product consists of three parts that enable users to look for information about grocery items recyclability anytime and anywhere if they have access to the internet. The

information on the grocery items including product name, product weight, product packaging details, and product recyclability, are easily found with the input of the UPC code on the website, or the simple click of the scan button on mobile devices. Users will find it easy to locate all the information they need, such as to compare the recyclability among several items or to know where to throw the constituent waste items.

2.2.3 User Aptitude

Users should have basic knowledge about web-browsing on computers or mobile devices.

2.2.4 Constraints

A current constraint of this product is that users are not able to add or modify item details on mobile devices. In addition, since the barcode scanner is not integrated into the mobile application, a barcode scanner is needed for the mobile application to function; the user interface design could be refined to follow the trends of the web design.

2.3 Detailed Requirements

2.3.1 External Interfaces

Internet connection is needed to access the database.

2.3.2 Functional Requirements

There are several functional requirements that are needed for this product to work as intended and include the following features:

- The user will be able to know which product is the most eco-friendly among one group of products.

- The user will be able to see the ranking of products based on how eco-friendly they are.
- Selected users will be able to register/sign in to this site and post information.

2.3.3 Performance Requirements

Time searching in the database should be less than 3 seconds.

2.3.4 Logical Database Requirements

The logical database requirements make sure the database is not ambiguous and easy to implement and contains:

- A products table that holds the basic product information including UPC code, product name, weight/volume, company name, and parent company name.
- A constituents table that holds the information of constituents of product packaging including constituent name and constituent type.
- A relationship between products and constituents that describes which waste constituents are within a particular product.
- A relationship between the constituents table and a table of Nova Scotia regions that describes if one constituent is recyclable or not in a particular region of interest.

2.3.5 Design Constraints

At this point, the mobile application is working exclusively on the Android platform; in the future it can be expanded to other platforms.

2.3.6 Use Case Diagram

The Use Case Diagram describes the roles in the system and what activities they accomplish. Figures 2-1 and 2-2 illustrate roles and activities in the website and mobile applications respectively.

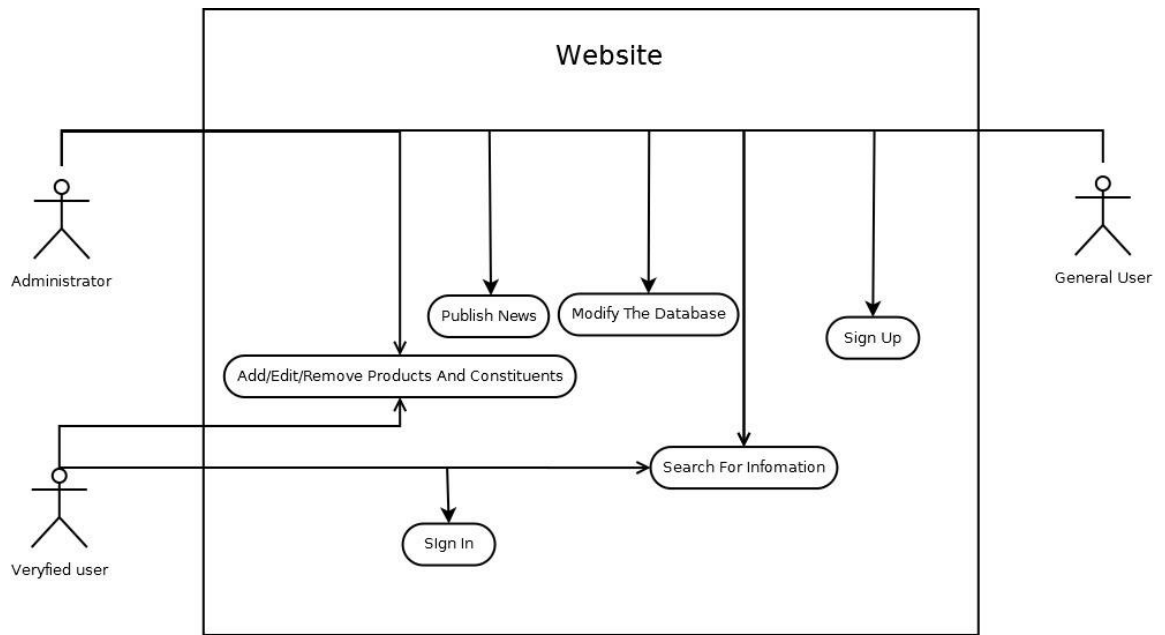


Figure 2-1 Use Case Diagram For Website

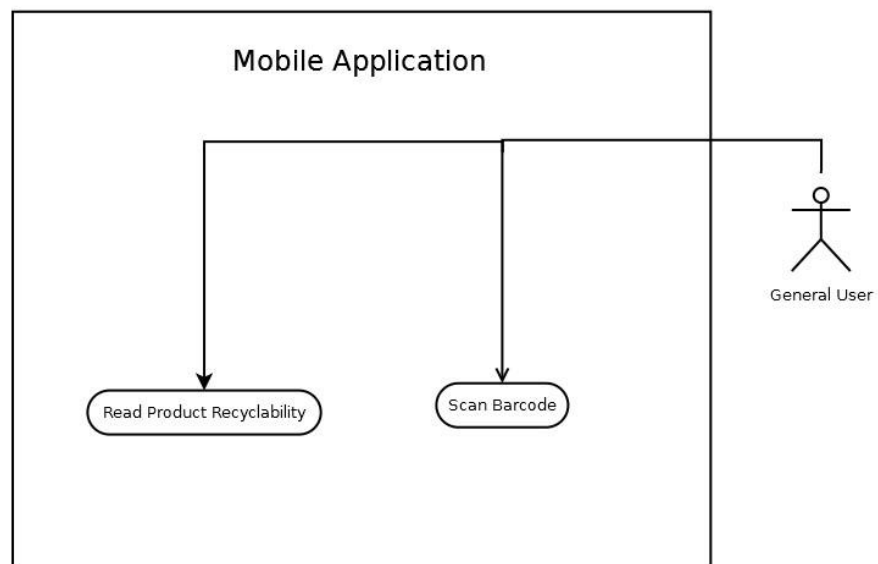


Figure 2-2 Use Case Diagram For Mobile Application

Arrows in the diagram above point to the activities a role in the system can do. For example, the user in this mobile application is able to scan the barcode of grocery items and read the item recyclability from the screen.

3 Database Design and Data Collection

3.1 Overview

The database will be using MySQL as the relational database management system (RDBMS), which is open-source and free. In order to collect the data, grocery store items were purchased every week until 500 items were registered, forming an initial database for this project. Once purchased, the entire product was weighed as well as the packaging and was divided into recyclable and non-recyclable waste and weighed separately.

This specific database consists of several core parts: products, constituents, regions and the relationship among these tables.

Every product has its own unique UPC code, each waste constituent has its name, and each region in the Province of Nova Scotia is represented to manage products that have multiple constituents, with some constituents differing in terms of their recyclability in different regions. It is therefore important to avoid data duplication or incomplete data when data is added or removed from the database.

3.2 E-R model

E-R model is short for Entity-Relationship model. As described in the specification, it is a data model for describing a database in an abstract way [3]. It provides for clarity in the rational representation of the database model in a more straightforward way and makes it easier to convert to a network or hierarchical data model. It is the first step to build the database before implementation.

Entities, relationships, and attributes are depicted by different figures using squares, diamonds and ovals to help most people understand how it works. A simple example is shown in Figure 3-1.

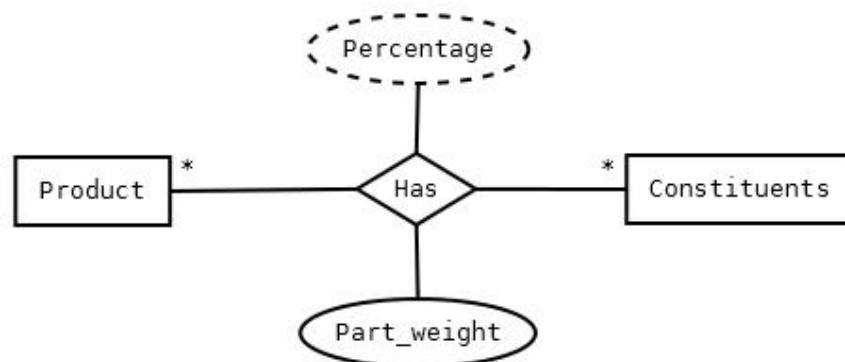


Figure 3-1 An Example Of Entity-Relationship Model

3.2.1 Entity

An entity is an abstract concept to describe an object. It can be any real item, like a person, a product, a company or even an event. In this project, there are several entities: product, constituents and regions. However, adjectives are required to specifically describe the object, thus the attribute acts as an adjective in the E-R model. For example,

a person can have short hair, long hair, or no hair, so if the person is an entity, it will have hairstyle as its attribute because different people can be distinguished by their different hairstyles. In the E-R diagram, it may look like Figure 3-2.

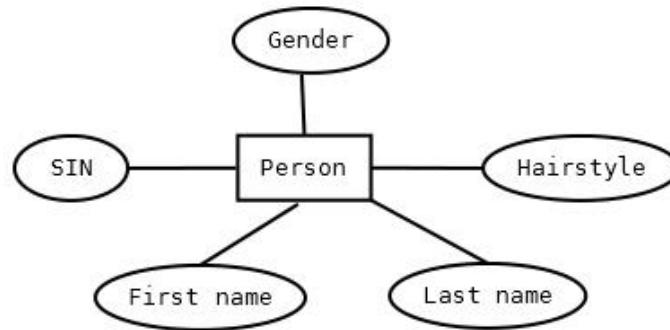


Figure 3-2 An Example Of Entity In E-R Diagram

3.2.1.1 Primary Key

Primary key is a column or set of columns that identifies a particular row. Almost any entity should have a primary key (except for weak entities). For example, at Acadia University, students' information is stored in the database, including the student name, student ID, and course list. But with all this information, a key is required to find a certain student without ambiguity, since students may have the same name, course list or professor name; the unique student ID can be the unique key to find the student. Thus, the student ID is chosen as the primary key when data are stored in the database.

3.2.2 Relationships

A relationship is how two entities are connected functionally to each other. They are like bridges. For instance, there are two tables stored in a university database (one is a list of students, another is a list of courses), one student may register for one course or several courses in the course list. On the other hand, one course can be selected by one student or many students. So the verb 'register' plays a role of relationship in this

example. Obviously, from the example, and pictured in Figure 3-3, there is a multiplicity in an association, where the asterisk (*) represents many and '1' represents one. In this case there is not a 1-to-1 relationship so there is no '1' in the diagram.

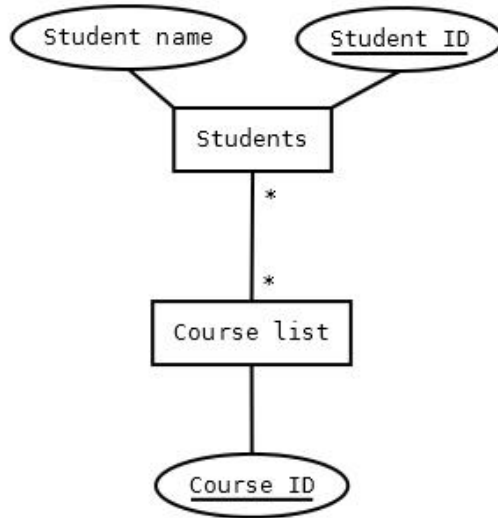


Figure 3-3 Many-To-Many Relationship

3.3 E-R Diagram Design

The final E-R diagram (Figure 3-4) is used to connect all the relevant entities associated with the product.

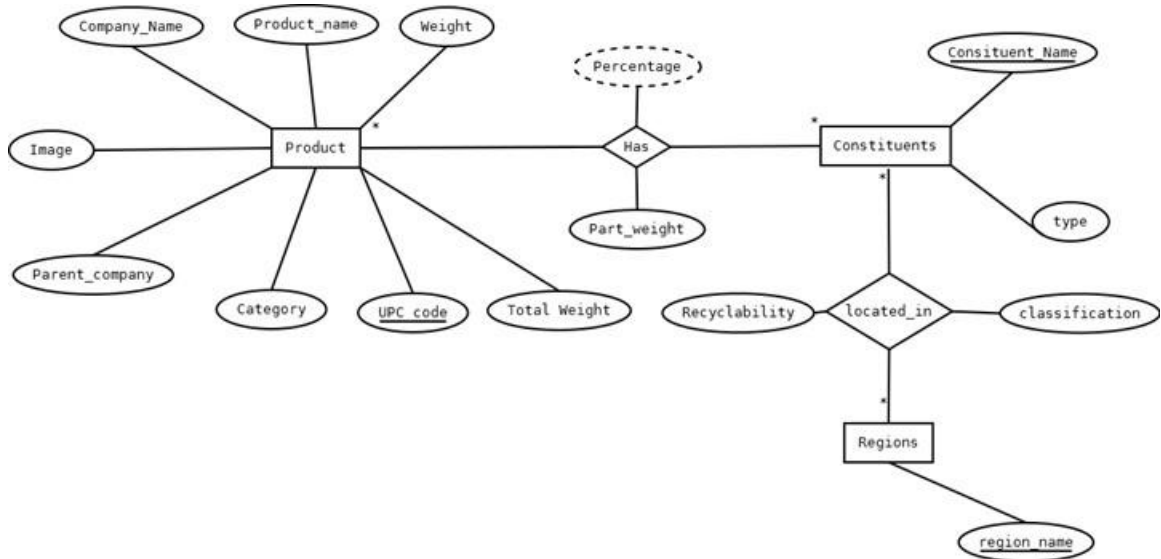


Figure 3-4 E-R Diagram

In this diagram there are two major entities: Product and Constituents. The Product entity, as the name implies, is to store all product information, which has eight attributes. Within these attributes, the UPC code being the primary key is used to distinguish different products since it is unique for all products. Each product item has several constituents with the name and type, so there is a relationship between the Product entity and Constituents entity. One product could have several constituents, and the weight of each constituent is part of the total weight of a product, so a derived value called ‘percentage’ appears in this relationship since it is calculated by the part weight and total weight. The Constituents table stores all the constituent names and their type, for example, a plastic bag. On the right side of the diagram there is a relationship between Constituents and Regions, for the reason that the recyclability of a certain constituent might be different in different regions; for example, the cap of a bottle of juice may be recyclable in the Valley region, but not be recyclable in Cape Breton.

3.4 Database Tables

The database consists of seven tables as follows:

1. Constituents
2. Products
3. Product-Constituents
4. Regions
5. Regions-Recyclability
6. Item-Class (to store all item classifications)
7. User-Information (to store user information)

Before creating the tables in the database, there are several steps. The following steps are based on the Products table as an example.

1. Make a Primary Key. The UPC is different among all the products and will not be repeated. Thus, it can be the unique identifier for all products.
2. Complete the table with other keys. Add the attributes from the E-R diagram to this table.
3. Figure out the type for all the keys. For instance, the UPC is a type of varchar of length of 20, and cannot be a null value.

3.5 Data Collection

All the data came from the product items purchased from stores. Before using them, the total weight was recorded. Once the product was consumed, the packaging was kept and separated into several constituents, weighed and recorded according to recyclability. All the data was entered into the database once the product was approved.

4 Website

4.1 Purpose

The website is to provide the user with a friendly interface that allows them to find information on a certain product like company name, weight, images and most importantly, if a product is recyclable and the recyclability of each constituent. For administrators, they should be able to add, change and remove items in the database.

4.2 Tools

In the development of this site, HTML, CSS and JQuery were chosen as the main languages or libraries for the front-end design and PHP to interact with the database. A brief description of these languages follows.

4.2.1 HTML

HTML (HyperText Markup Language) is the main markup language for creating web pages and other information that can be displayed in a web browser. It lays the structure of the website by tags, so the content of the website can be organized by CSS or jQuery.

HTML basically consists of various types of tags in angle brackets, for example, `<html>`, in a web page content. Most of the tags are paired, like `<p>` and `</p>`, however, some tags are unpaired, for instance ``. In paired tags, text, other tags and other types of text such as content are put between them, for instance, `<h1>This is a Header</h1>` [4].

A simple example that displays “Hello World” would be coded as follows:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Hello World example</title>
  </head>
  <body>
    <p>Hello World</p>
  </body>
</html>
```

The subsequent product would look as follows:

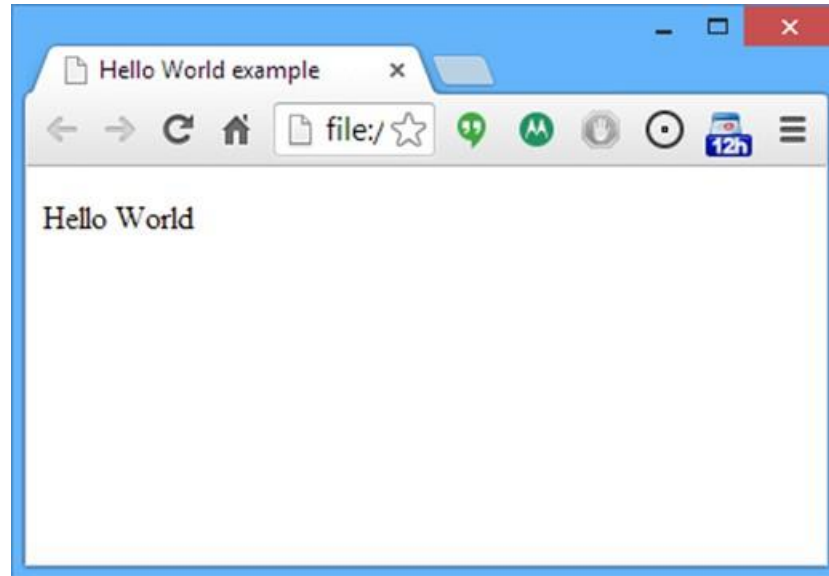


Figure 4-1 Hello World HTML Example

Content is then put into the tags. In HTML5, a lot of new elements in tags have been introduced including `<canvas>`, `<article>`, `<header>`, and `<nav>` as well as some new form controls such as calendar, date, time, and email.

4.2.2 CSS

If HTML is compared to the body of a man, then the CSS will be the clothes. Cascading Style Sheets (CSS) is the language to change the look of HTML tags in a way the web designer wants, instead of plain text. It can not only change the color, font, margin and other attributes, but also is able to trigger animation like slide, hide or show. CSS is also able to make web pages adapt to different screen sizes or resolutions. As a result, CSS makes it possible to show different content layouts and appearances to users on desktops, laptops, smartphones or tablets. CSS is independent of HTML and can be used with any XML-based markup language. Since CSS is independent from HTML, it is

possible for web designers to share the style sheets among different web pages and web designs, and is also easier to maintain [5].

In general, CSS ‘decorates’ HTML so that the web page will not look disorganized to users. The CSS code can be embedded at the front of the HTML file in the <head> tag, in every tag that needs CSS, or saved in the separate CSS file. The following example displays the difference with and without CSS.

The HTML with CSS:



Figure 4-2 HTML With CSS

HTML_with_css_example.html:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      body {
        background-color:#d0e4fe;
      }

      h1 {
        color:orange;
        text-align:center;
      }

      p {
        font-family:"Times New Roman";
        font-size:20px;
      }
    </style>
  </head>
  <body>
    <h1>CSS example!</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```



```
    </style>
  </head>

  <body>
    <h1>CSS example</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

The HTML without CSS:



Figure 4-3 HTML Without CSS

HTML_without_css_example.html:

```
<!DOCTYPE html>
<html>
  <body>
    <h1>No CSS example!</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

CSS is also able to implement the responsive design.

With the fast growth in mobile technology, smartphones and tablets are playing a more and more important role in people’s daily lives. “The global mobile data traffic has grown 81% in 2013, which reached 1.5 exabytes per month by the end of 2013, up from

820 petabytes per month at the end of 2012” [6]. Thus a web layout for mobile devices has become very necessary and fortunately CSS3 can be used to help achieve this goal.

To accomplish this, the following command:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

is added into the <head> tag to set the browser to display the page to the width of the device with an initial scale of 1.

The following command is then added:

```
@media only screen and (min-width: 150px) and (max-width: 750px)
{
}
```

at the bottom of the style.css file. The CSS code could be placed inside the block of what is shown to users using mobile devices. The reasoning is that most smartphones have the width of screen less than 720 pixels, so this could help to tell if users are browsing on mobile devices.

4.2.3 PHP

If HTML is the structure of a body, CSS the clothes, PHP would be the blood flowing within the body and nerves interacting with every part of the body to make them cooperate based on the information from the brain. As a server-side scripting language, it is prevailing as a general-purpose programming language.

PHP code is interpreted through a PHP processor module on the server end, which translates PHP into the web page that can display on the screen. PHP can be embedded

into an HTML document and used in a standalone applications thanks to its command-line interface capability [7].

In this project, PHP is used to interact with the database and manipulate operations, like checking the valid forms before submission, connecting to the database, or checking the user password.

PHP code can be saved in a .php file or embedded in an HTML file. The following example of an ‘Hello world’ program is written in PHP.

If saved in a .php file, it is simply:

```
<?php
    echo "Hello world";
?>
```

When embedded in the HTML file it looks as follows:

```
<!DOCTYPE html>
<html>
  <head>
    <title>PHP example</title>
  </head>
  <body>
    <?php
      echo "<p>Hello world</p>";
    ?>
  </body>
</html>
```

As mentioned before, PHP is used as a tool to interact with the database and runs at the server-end. The server interprets the PHP code and generates the related code that can be displayed by browsers. In addition, it can be integrated into an HTML file instead of using external files.

In order to manipulate the database, the PHP extensions called mysql or mysqli (after PHP version 5) are used. In this project, mysqli was chosen because of the following benefits in this product [8]:

1. Object-oriented interface
2. Enhanced debugging capabilities
3. Embedded server support

To connect to the database, a file called mysqli_connect.php is created. It just works to connect to the database by using the mysqli_connect function with four parameters (Host, Username, Password, Database name). In other .php files, using require_once('mysqli_connect.php') simplifies the code and makes it easy to access the database.

The following steps are followed to retrieve the data from the database:

1. Establish the connection by using require_once('mysqli_connect.php').
2. Save the result of mysqli_query(mysqli \$link , string \$query) to a variable called 'result';
3. mysqli_fetch_array(\$result, MYSQLI_ASSOC) will parse the result to arrays for data output.

This piece of code is an example of reading from the database:

```
<?php //To build up the connection between website and database

//Define the information required to log into the database
DEFINE('Username','user');
DEFINE('Password','password');
DEFINE('Host','localhost');
DEFINE('Database','database_name');
```

```
//set up the connection
$dbc = @mysqli_connect(Host,Username>Password,rp) OR die('Unable to
connect to the database '. mysqli_connect_error());

?>
```

The '@' sign in front of 'mysqli_connect' is the error control operator, the compiler will ignore the possible errors that might be generated. It is saved to a .php file so that every time a database connection is built, the file use is imported.

After the connection is built, the database is accessed and data is retrieved.

Initially, some SQL code is written for MySQL to do the query within the database, for example:

```
SELECT *
FROM constituents, products, prod_const, regions_recyclability
WHERE products.upccode = prod_const.upccode
AND prod_const.cname = constituents.cname
AND constituents.cname = regions_recyclability.cname
ORDER BY products.upccode
LIMIT 0 , 30;
```

This statement will fetch all the data from TABLE constituents, TABLE products, TABLE prod_const and TABLE regions_recyclability in the order of the UPC code.

The next step runs the query and deals with the data retrieved from the database:

```
$r = mysqli_query($dbc, $q); // Run the query.
$num = mysqli_num_rows($r); // get the number of rows

if($num > 0) {
    /* Output the title */
    /* some code */

    /* Output the result */
    while ($row = mysqli_fetch_array($r, MYSQLI_ASSOC)){
        /* some code */
    }
} else {
    echo '<p class="no_data">No data in the database</p>';
}
}
```

The data on the site is now available for viewing, but needs to be ‘decorated’ to make it straight-forward and easy to read. Therefore, some CSS code is added to change its appearance. Most of the CSS code is stored in the .css file.

4.2.4 jQuery

Since HTML provides the structure, CSS the clothes and PHP the blood and nerves in a body, jQuery is the part of the brain which has lots of knowledge that can make the body move in a way it wants or change the gesture to make the body look different. As a JavaScript library, it makes writing JavaScript easier and do the work in less code as its slogan says---“Write less, do more”.

jQuery can ‘select’ the elements in HTML and easily add or change their attributes. The following example consists of HTML code and jQuery. When users click the “Hello” on the webpage, a dialogue box saying “hello world” will show up.

jQuery_example_html.html:

```
<p class="example">Hello</p>

<!DOCTYPE html>
<html>
  <head>
    <title>jQuery Example</title>
  </head>
  <body>
    <p class="example">Hello</p>
  </body>
</html>
```

jQuery_example_jQuery.js:

```
$( document ).ready( function() {  
    $(". example").click( function() {  
        alert("hello world");  
    });  
});
```

jQuery can also make it easier in terms of adding animations by using slide(), fade(), and animate() functions, which makes web design simpler and saves time.

In this product, jQuery is used to make the website more dynamic and more interactive with users. There are several implementations with jQuery including news tiles and dynamic forms.

Below, a simple example of the dynamic form is introduced to illustrate. By default, the number of constituents is zero, which means no constituent. In HTML file, the number of constituents is labeled an ID called 'cnumber'. In the .js file, this ID has to be retrieved, so the '\$' sign is used to retrieve this ID and to do some changes on it.

A screenshot of a web form with a light green background. It contains four rows of input fields: 1. 'Number of constituents:' followed by a dropdown menu showing '0'. 2. 'Region:' followed by a dropdown menu showing 'Cape Breton'. 3. 'Upload an image:' followed by a 'Choose File' button and the text 'No file chosen'. 4. 'Author:' followed by a text input field containing the placeholder text 'Author name'.

Figure 4-4 jQuery Example- Default Number Of Constituents

The code of this function is as follows:

```
function displayinfo(){  
    // Read the ID from HTML  
    var constituent = $('#cnumber');  
    //Save its value to a variable  
    var cnum = constituent.val();
```

```

//HTML code for new rows of form
var clist_rest = "
<label>Constituents Name: </label><input type='text' name='cform[cname][]" size='15' maxlength='20' /> <br /> \
<label>Constituent weight:</label> <input type='text' name='cform[pweight][]" size='15' maxlength='20' />g <br /> \
<label>Type:</label> <input type='text' name='cform[Type][]" size='15' maxlength='20' /></p> \
<label>classification:</label> \
<select name='cform[classification][]"> \
  <option value='Green Cart'>Green Cart</option> \
  <option value='Blue Bag #1: Paper'>Blue Bag #1: Paper</option> \
  <option value='Blue Bag #2: Recyclables'>Blue Bag #2: Recyclables</option> \
  <option value='Clear Garbage Bag'>Clear Garbage Bag</option> \
</select><br /> \
</div>";

//Remove previously generated forms
$("div[id^='clist']").remove();
//Add the forms based on the user input by a loop.
for (var i = 0; i < cnum; i++)
{
  $('#cnumber').after (
    //HTML code for constituents info
    "<br /><div id='clist" + i + "'>" + clist_rest);
};
};
//Change the number of forms.
$('#cnumber').change(displaycinfo);

```

Every time a user changes the number of constituents, the form will be updated so if users choose 2, then the new rows will appear:

Number of constituents:	<input type="text" value="2"/>
Constituents Name:	<input type="text"/>
Constituent weight:	<input type="text"/> g
Type:	<input type="text"/>
classification:	<input type="text" value="Green Cart"/>
Constituents Name:	<input type="text"/>
Constituent weight:	<input type="text"/> g
Type:	<input type="text"/>
classification:	<input type="text" value="Green Cart"/>
Region:	<input type="text" value="Cape Breton"/>
Upload an image:	<input type="button" value="Choose File"/> No file chosen
Author:	<input type="text" value="Author name"/>

Figure 4-5 jQuery Example- Updated Form

4.3 Interface Design

Since the purpose of this website is to provide users with a very friendly interface and at the same time embody the theme of this project – recycling– green was chosen as the background color. Furthermore, a combination of vivid colors will deliver a feeling of comfort and ease that provides a visual effect. Figure 4-6 is the home page design that provides the combination of vivid colors. It would be consistent with the theme of the product—being eco-friendly and close to nature.



Figure 4-6 Home Page Design

This interface is the initial design. It looks straightforward and refreshing. At the center is a scrollable slider, which provides users with the latest information and a very comfortable visual experience at the same time. The tiles under the slider are dynamic; they will only display the title if there is no action, however, when users hover the mouse above a certain tile, the content in that section will show up and provide detailed information to users.



Figure 4-7 Sign In Page Design

Above is the sign up window; whenever users click the sign up button, this window will pop up and float on the screen.

Figure 4-8 is the product page that list all the items. By using Ajax, users are able to filter the result and search without going to another page.



Figure 4-8 Product Page Design

The whole idea of the design is image-driven, so the product uses a very large and eye-catching image to help propagate the idea of recycling the packaging of grocery items purchased. After users enter a keyword or click the item on the list, detailed information of a certain product will be given, which looks like Figure 4-9.



Figure 4-9 Item Detail

On this page, all the detailed information of a certain product is given. The list on the top-right will be filled with the product name, UPC code, company name, weight and rate of recyclability, which is followed by a list of constituents of this product. For example: a box of Oreo cookies can have a paper box package on the outside, a plastic package inside, and a plastic tray to hold the cookies. So this list will show all of these constituents and their individual weight, recyclability, type and region (recyclability of constituents may vary in different regions).

Having realized that most users are more concerned about the recyclability than the product information, information in this list is demonstrated by a 3D pie-chart for delivering a straight-forward and easy-to-understand form to help users know what the product is made of. The recyclability was made more prominent through the use of the 3D Pie-Chart (Figure 4-10).

constituent name	constituent weight	Percentage	type	region
plastic bag	12	0.12	#6 plastic	Cape Breton
plastic tray	13	0.13	#1 plastic	Cape Breton

Percentage of Constituents

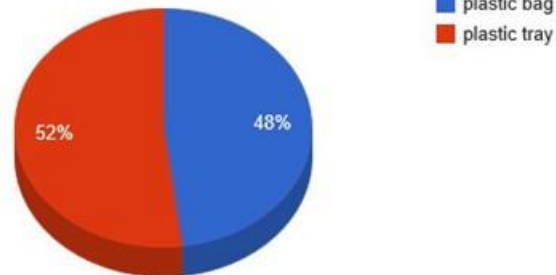


Figure 4-10 3D Pie Chart

Fortunately Google has provided such a tool to help web developers, so it is just added to the <head> tag in the .html file where the pie-chart is presented and is described below:

```

<html>
<head>
  <script type="text/javascript" src="https://www.google.com/jsapi"></script>
  <script type="text/javascript">
    google.load("visualization", "1", {packages:["corechart"]});
    google.setOnLoadCallback(drawChart);
    function drawChart() {
      var data = google.visualization.arrayToDataTable([
        ['Task', 'Hours per Day'],
        ['Work', 11],
        ['Eat', 2],
        ['Commute', 2],
        ['Watch TV', 2],
        ['Sleep', 7]
      ]);

      var options = {
        title: 'My Daily Activities',
        is3D: true,
      };

      var chart = new google.visualization.PieChart(document.getElementById('piechart_3d'));
      chart.draw(data, options);
    }
  </script>
</head>

```

```
<body>
  <div id="piechart_3d" style="width: 900px; height: 500px;"></div>
</body>
</html>
```

(Source: <https://developers.google.com/chart/interactive/docs/gallery/piechart#3D>)

With the index page, products page and item detail page completed, detail on the file accessibility follows.

On mobile devices, due to the limitations of a small screen, the web layout for computers would cause pain for users to browse on their smartphones. A different design for mobile devices is desired.

As mentioned in the CSS section, this is done by changing the design of the HTML element in the following bracket.

```
@media screen and (min-width: 150px) and (max-width: 859px)
{
}
```

For example, when users are browsing a computer, the tiles on the home page have three tiles in a row (Figure 4-11), which is too much for mobile phones. Beside, to show the content in a certain tile on computers, users need to use a mouse to hover on the tile while it is not possible for a touch screen. So the content has to show without a hover action. After the modification, the new layout for mobile phones are more acceptable for mobile phone users (Figure 4-12).

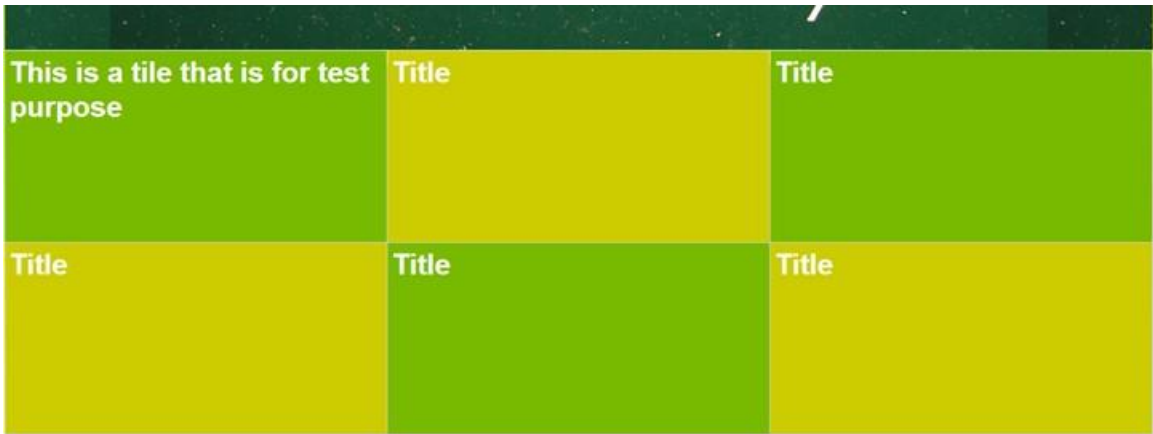


Figure 4-11 Tiles On Computers

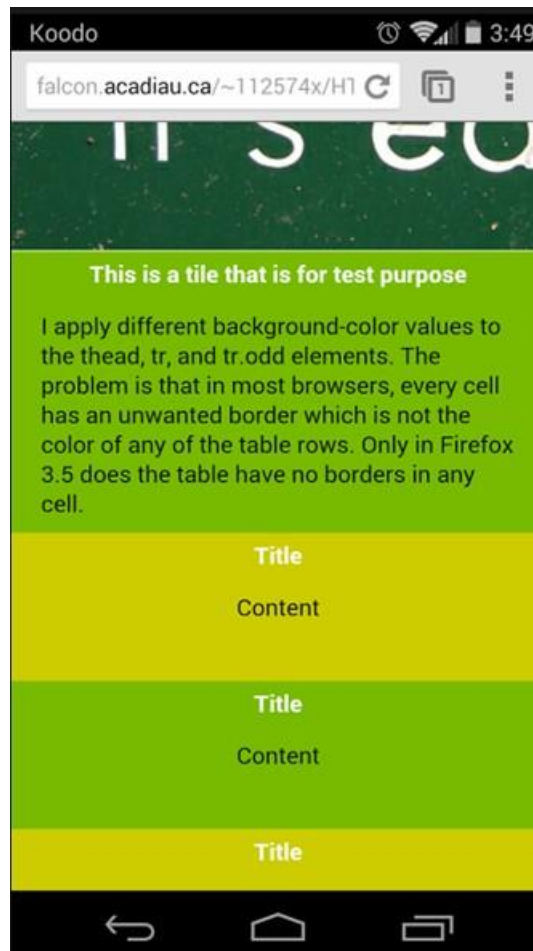


Figure 4-12 Tiles On Mobile Phones

4.4 User Login

Selected users have to register before they can add new products to the database, modify products, or delete items from the database. All passwords are encrypted in SHA-1 to prevent plain text passwords.

4.5 Database Security

Security is very important to keep the website stable and safe for users and data, so measures have been taken in case of any attack. When users are entering product information, it is saved to the database using SQL. If measures were not taken to check the data, the risk of database corruption through ‘SQL injection’ would be high. ‘SQL injection’ happens when malicious code is inserted into some fields that are to be passed to the SQL server for parsing and execution. So any procedure that is related to SQL statements should be checked for the possibility of SQL injection because the SQL server will execute all syntactically valid queries that it receives [9], so basically people can access the database through the form data if it isn’t taken into account.

The ‘mysql:real_escape_string’ can be used in this case to escape special characters in a string for use in an SQL statement, preventing malicious code inserted into the string. This could help to prevent the database being attacked by malicious software or users [10].

5 Android Application

5.1 Purpose

The market share of Android OS has grown dramatically in the past years in the mobile computing area. An application of the product running on Android will be much easier and more acceptable to consumers when compared with using a browser and typing the URL to get the same information.

5.2 Functions

The core function of this application is to provide consumers with a straightforward and easy experience to access the database. Users only need to open the app and scan the barcode of a certain product and then get the product information and the recyclability rate with one click.

Users are able to choose which region they are living in and to save it as a preference for future use. The application can be expanded based on future needs.

5.3 Tools

5.3.1 Java

Android applications are written in Java, which is one of the most popular programming languages in the world. It is intended to make applications ‘write once, run anywhere’, meaning that the code can run on different platforms. As Java is known to most computer science students, the introduction is omitted here.

5.3.2 Android SDK

For now, the Android SDK (Software Development Kit) is the most popular development tool for Android developers, though Android Studio is getting accepted by more developers. In order to run the test, an Android smartphone running 4.1.2 as the test device was used because this version was one of the most popular versions among different Android versions at the time of development.

5.4 Mobile Application Implementation

5.4.1 UI (User Interface) Design

Since this application is designed for users that have no knowledge of complicated software, simplicity matters. The design purpose is to make the interface as simple as possible so that consumers of any age are able to use the resource without any help. The home page has only one button to launch the barcode scanner (Figure 5-1).

The color is consistent with the website design so that a friendlier environment is provided to users to provide a similar experience to that of the website. After the user clicks the camera button, it will launch the barcode scanner and all the user needs to do is scan the barcode on the product packages (Figure 5-2).

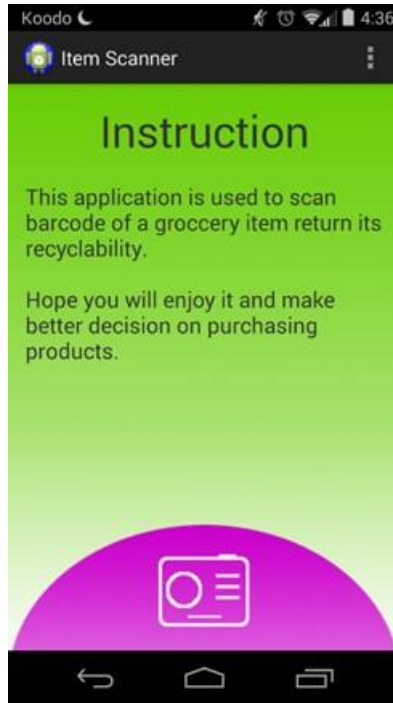


Figure 5-1 Home Page



Figure 5-2 Barcode Scanner Interface

As soon as the user scans the barcode, they will be guided to a new interface to choose the region because the recyclability varies through the regions (Figure 5-3).

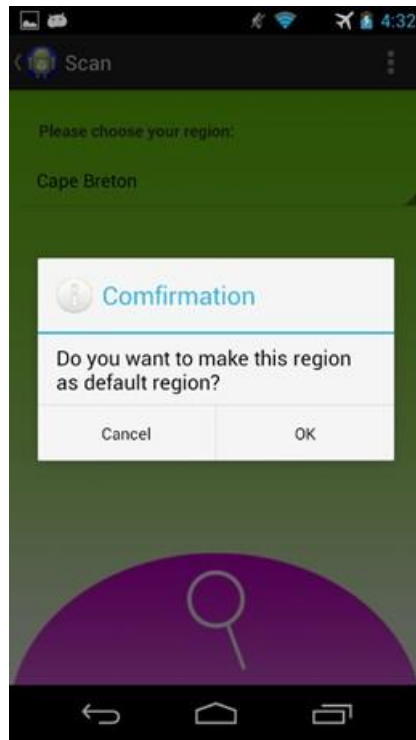


Figure 5-3 Choose Region

When the user chooses the region and clicks the search button, they can choose to save the region as their default (Figure 5-4).

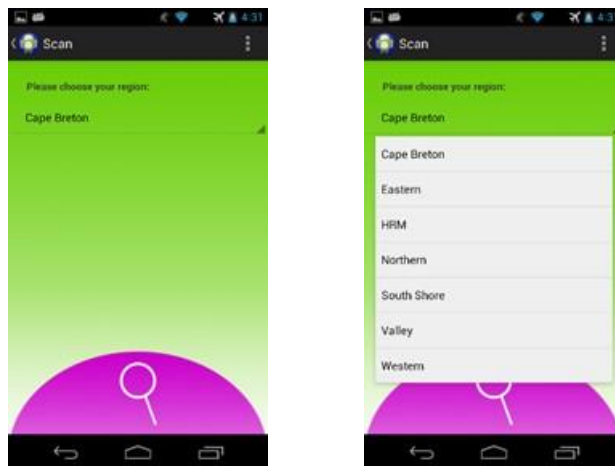


Figure 5-4 Set Default Region

Then the information will be shown properly for users as follows (Figure 5-5).



Figure 5-5 Item Information

On the details page, users will get straightforward information about a product and can easily read the recyclability of it as well as where to dispose of all the packages; they are also able to change the default region in the settings (Figure 5-6).

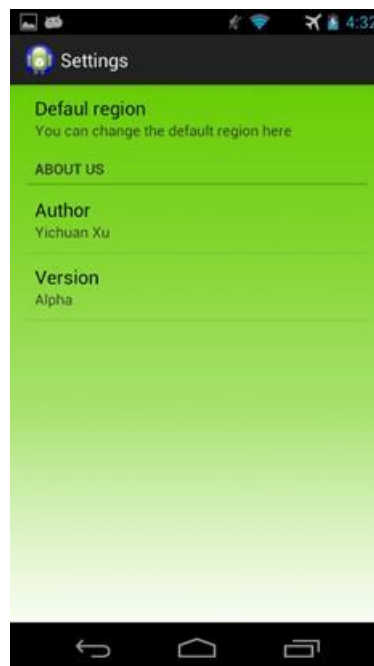


Figure 5-6 Settings

5.4.2 Activities

Activities in Android applications are like handlers to handle the widgets on the screen. It responds to the actions you make to the screen, like clicking a button, scrolling a list, or zooming in and zooming out a picture with two fingers. Activities are written in Java and all activities derive from 'Activity' Class. The library has lots of functions that are over-ridden to perform some operations by adding a '@' sign before those functions.

```
@Override
protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_main);

}
```

5.4.3 Layouts

If the activities are compared to the scripts in web development, layouts will be the combinations of HTML and CSS. It decides how the interface will look on an Android device. They are stored as xml files under the res/layout folder as shown:

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity" >

<TextView
    android:id="@+id/introduction_title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/introduction_title"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:gravity="center"
    android:textSize="40sp"
    android:typeface="sans" />

</RelativeLayout>
```

The text that is shown is stored in the strings.xml file under the res/values folder. It has one major advantage, which is to easily translate the whole application into another language. For example, when English is stored in strings.xml and simplified Chinese is desired for the app, a folder can easily be created called values-zh with the Chinese version of strings.xml placed in the folder.

5.4.4 Barcode Scanner

In this app, when the user clicks the scan button, it will launch the barcode scanner by zxing [11]. So the installation of the barcode scanner from the Google Play Store is necessary before the application is able to be fully functional.

5.4.5 Internet Connection

To check the internet connection the function called “isNetworkAvailable” was used that returns true if the network is connected. The code is as follows:

```
//Check if network is available
public boolean isNetworkAvailable(Context ctx)
{
    ConnectivityManager cm =
    (ConnectivityManager) ctx.getSystemService(Context.CONNECTIVITY
SERVICE);
    NetworkInfo netInfo = cm.getActiveNetworkInfo();
    if (netInfo != null && netInfo.isConnectedOrConnecting() &&
        cm.getActiveNetworkInfo().isAvailable() &&
        cm.getActiveNetworkInfo().isConnected())
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

5.4.6 User Preference

Users are able to store their preference of the default region. The setting is stored in the SharedPreferences interface. The SharedPreferences.Editor interface is used to

modify the preference in a SharedPreferences Object and uses lots of ‘get’ methods to recover the preference.

6 Product Manual

This section is to provide a quick instruction to show how to use this product and make operations.

6.1 Website

6.1.1 Information Retrieval

Click the ‘Product’ button on the nav-bar; users will enter the Product page. If users need to search for a product, they simply choose the class and region, then enter either the UPC code or product name and click ‘Search’. The result below will be shown:



UPC code	Product Name	Weight/Volume(g/L)
6393625006	Gravy mix	227.0

Figure 6-1 Search Result

6.1.2 Registration

For a select number of users, privileges will be granted to modify and update the database. To do so, click sign up on the nav-bar, enter the username, password and user information. Then a new user account will be created. If a username or email address has already existed, users will be prompted to enter a new one. Figure 6-2 is the sign-up page.

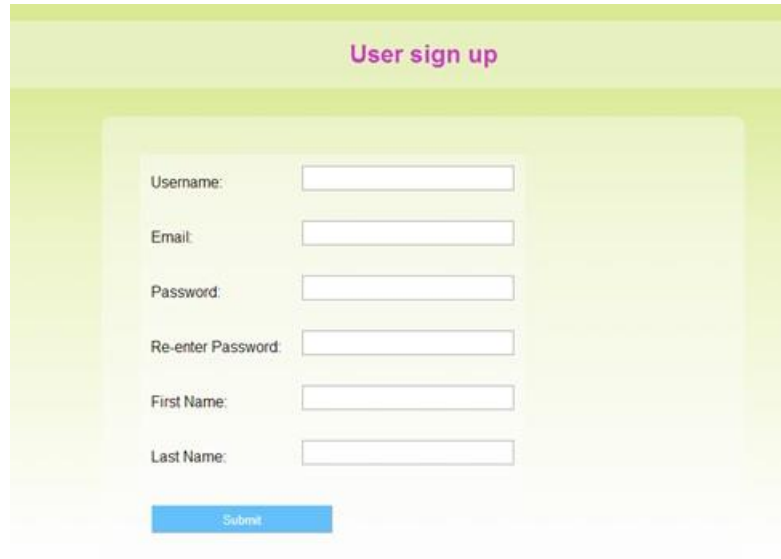
A screenshot of a 'User sign up' form. The form is titled 'User sign up' in purple text at the top. It contains seven input fields: 'Username:', 'Email:', 'Password:', 'Re-enter Password:', 'First Name:', and 'Last Name:'. Each field is followed by a white rectangular input box. At the bottom of the form is a blue 'Submit' button.

Figure 6-2 User Sign Up

6.1.3 Adding new items

In order to add new items, users need to register as a member. After users finish the User Sign up, or click Sign in and enter correct information, they click the Product page and a new button will appear--- ‘ [Add new product](#) ’. Then users are able to add new product information and when finished simply click the ‘Submit’ button.

6.1.4 Modify or Remove Items

On the product detail page, simply click ‘Edit’ or ‘Delete’ to modify or remove an item (Figure 6-3).



Figure 6-3 Edit And Delete Button

6.2 Mobile Application

6.2.1 Scan

After users open the application, click the camera button to get into the scan mode (Figure 6-4).

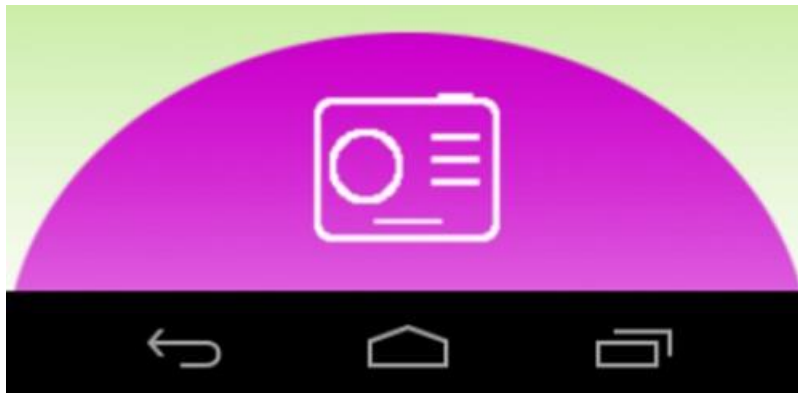


Figure 6-4 Camera Button

Then users will have to focus on the barcode and the barcode scanner will automatically read the value and save it (Figure 6-5).

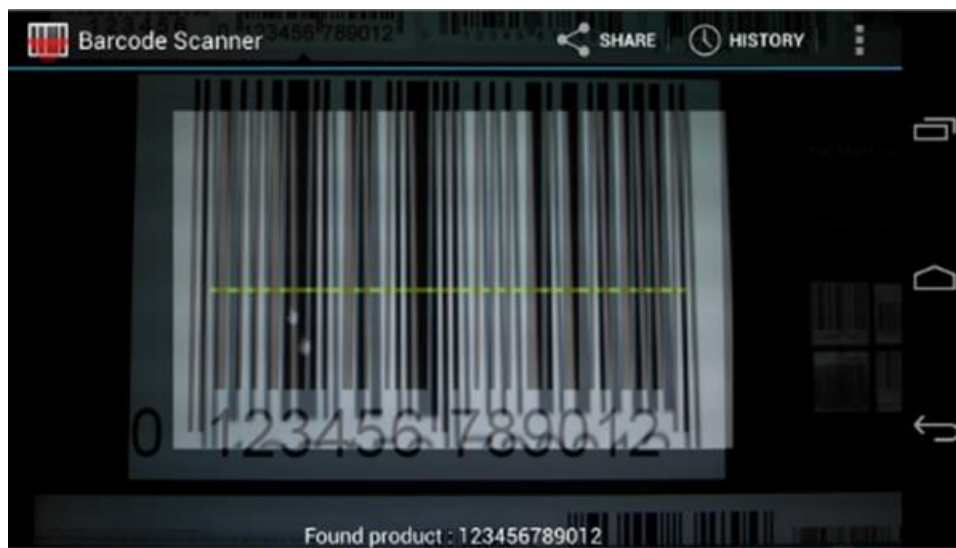


Figure 6-5 Barcode Scanner

6.2.2 Choose Region

After the barcode scanner reads the UPC code, users will be prompted to choose the region of interest. A default region could be made after users select one, and next time

they will not be prompted to choose again. If users need to change the default region, click the button on the top right side of the screen and click settings to initiate a change (Figure 6-6).

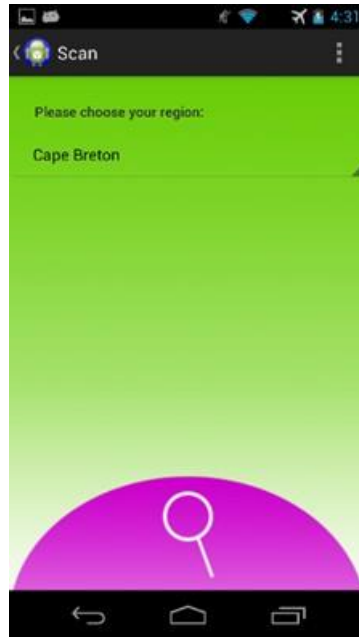


Figure 6-6 Select Region

6.2.3 Retrieve Product Information

In this phase, an internet connection is needed. The system will connect to the remote database and retrieve the data including product information and recycling information. The information will show details about the product and the destination of the packaging (Figure 6-7).



Figure 6-7 Product Information

7 Constraints

Though the product is fully functional, there are still improvements to be done in terms of use for commercial purposes including the following:

1. This mobile application only has the function to scan the barcode and load item data from the database.
2. The user interface on both the web and mobile device can be refined to make the product more user-friendly and attractive so as to provide a comfortable user experience.

8 Conclusion and Future Work

This thesis introduced a fully functional product that is able to help people better recycle their packaging waste, provide municipalities with an additional educational tool, and reduce the amount of garbage landfilled.

Although this product is fully functioning, there are many ways it could be expanded. This includes a better user interface design for both the website and the mobile application, and improved functionalities.

The user may want the function to compare different products at the same time on the website in order to have a better understanding of the difference among several products. Regarding the mobile application, functions can be expanded to meet more needs from end-users such as adding product information from mobile phones and compare product recyclability. Besides, when a product is scanned to get the information, the system could show a similar product with higher recyclability, which will help customers to decide which product is more eco-friendly and worth buying.

9 References

- [1] czq5494, "<http://www.nipic.com/>," 16 June 2010. [Online]. Available: <http://www.nipic.com/>. [Accessed 10 March 2014].
- [2] I. C. Society, IEEE Std 830-1998 IEEE Recommended Practice for Software Requirement Specifications, IEEE, 1998.
- [3] P. P. Chen, "The Entity-Relationship Model: Toward a Unified View of Data," 1976.
- [4] Wikipedia, "HTML," 2014. [Online]. Available: <https://en.wikipedia.org/wiki/HTML>.
- [5] W3C, "HTML & CSS," 2014. [Online]. Available: <http://www.w3.org/standards/webdesign/htmlcss#whatcss>.
- [6] Cisco, "Global Mobile Data Traffic Forecast Update, 2013–2018," 2014. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html.
- [7] PHP-Wikipedia, "PHP-Wikipedia," [Online]. Available: <http://en.wikipedia.org/wiki/Php>.
- [8] PHP.net, "PHP Mysql document," 2013. [Online]. Available: <http://us2.php.net/manual/en/mysqli.overview.php>.

- [9] Microsoft, "SQL Injection," 4 August 2013. [Online]. Available:
<http://technet.microsoft.com/en-us/library/ms161953%28v=SQL.105%29.aspx>.
- [10] PHP.net, "mysqli::real_escape_string," 2014. [Online]. Available:
http://ca2.php.net/mysqli_real_escape_string.
- [11] Zxing, "zxing," 2014. [Online]. Available: <https://github.com/zxing/zxing>.