

# Web Front-End MPP Capstone Project

## Functional Specification

Author	Gerry O'Brien
Last Updated	12/17/2018 11:16 AM
Document Status	Final

### Table of Contents

1	Executive summary	2
2	Dependencies	2
3	Feature list	5
4	WorldWideImporters Functionality	8
4.1	Feature overview	8
4.2	Process flows / use cases	8
	Customer Visits Site	9
	Customer Displays Products	10
	Customer Adds Product to shopping cart	12
	Customer Checks Out	13
	Customer Wishes to Contact World Wide Importers	14
	Customer Wants More Information About world wide importers	16

5	Wireframes	18
6	Success Criteria	22
6.1	Goals	22
6.2	Non goals	22

## 1 EXECUTIVE SUMMARY

The capstone project for the Web Front-End MPP program is project that is intended to allow the student to apply their knowledge and skills gained from the courses that make up the Web Front-End MPP curriculum. It is a project that will showcase the student's abilities and provide them with an addition to their portfolio for presentation to prospective employers.

This functional specification will outline the aspects of the project that students must meet in order to be successful in achieving a passing grade on this capstone project. The project will require implementation of the knowledge and skills gained throughout the curriculum.

### *Scenario:*

WorldWideImporters is starting a small grocery segment to offer online shopping for their fruit and produce groceries. The website to support this segment will be constructed using the latest web technologies and frameworks in order to make the front-end portion of the site responsive and adaptive to a wide array of platforms and mobile devices.

The site will consist of a common layout with navigation, header, and footer sections to provide a common look and feel while offering a shopping experience that is easy to use and intuitive.

## 2 DEPENDENCIES

The site will have a dependency on a Microsoft Azure data source and API. Details of that service are as follows:

To access the data that describes the items (like name, price, image, ...etc) in the online store, there is a service that you can call using XMLHttpRequest or Fetch API to retrieve the data.

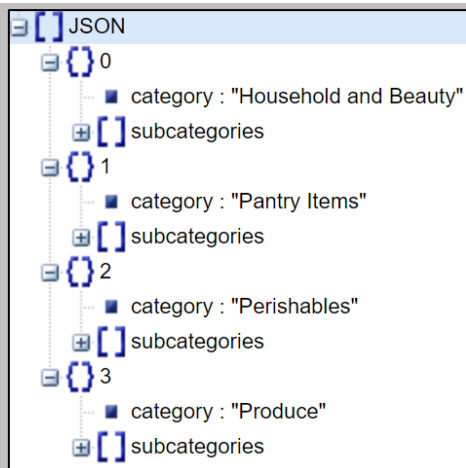
You can use either [Fetch API](#) or a tool like [Axios](#), [jQuery.ajax](#) or Angular's [\\$http](#) service to request data. The url for the service is: <https://webmppcapstone.blob.core.windows.net/data/itemsdata.json>

The returned data is in JSON format with the following structure:

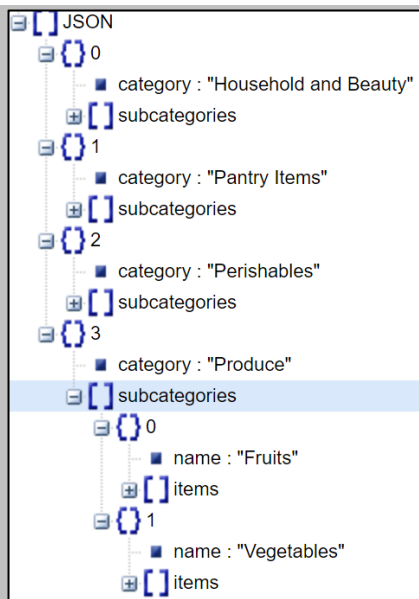
- 1- An array that contains 4 items representing the four main categories of products in the store. (Figure 1)
- 2- Each item in this array has two properties:
  - a. The "category" property is a string property that has the name of the category
  - b. The "subcategories" is an array that contains the subcategories in this main category. Each subcategory item in this array has two properties: "name" property which is a string property for the name of the subcategory and "items" property which is an array of the items in that subcategory. (Figure 2)

For example in figure 2 you will see we have two items in the subcategories array of the Produce category, those are the "Fruits" and the "Vegetables" subcategories, each of them has an array of "items".

- 3- The items array in each subcategory is where the data of the actual product is. Figure 3 shows the structure of the "items" array. Here are the properties of each product:
  - a. "name" a string property this is the name of the product
  - b. "description" a string property contains a small description about the item like how many pieces in the box or the weight in pounds in case of a produce item.
  - c. "price" a number property that has the price of the item in dollars
  - d. "imagelink" a string property that contains an online href link where you can find the image of the product to use to show the a picture of the product.
  - e. "rating" a string property that has the rating of the product in a range from 1 to 5.
  - f. "stock" a string property that has the amount available in stock of the product.
  - g. "category" a string property with the category of the product.
  - h. "subcategory" a string property with the subcategory of the product.



**Figure 1 Main Categories in JSON Data**



**Figure 2 Subcategories in JSON Data**

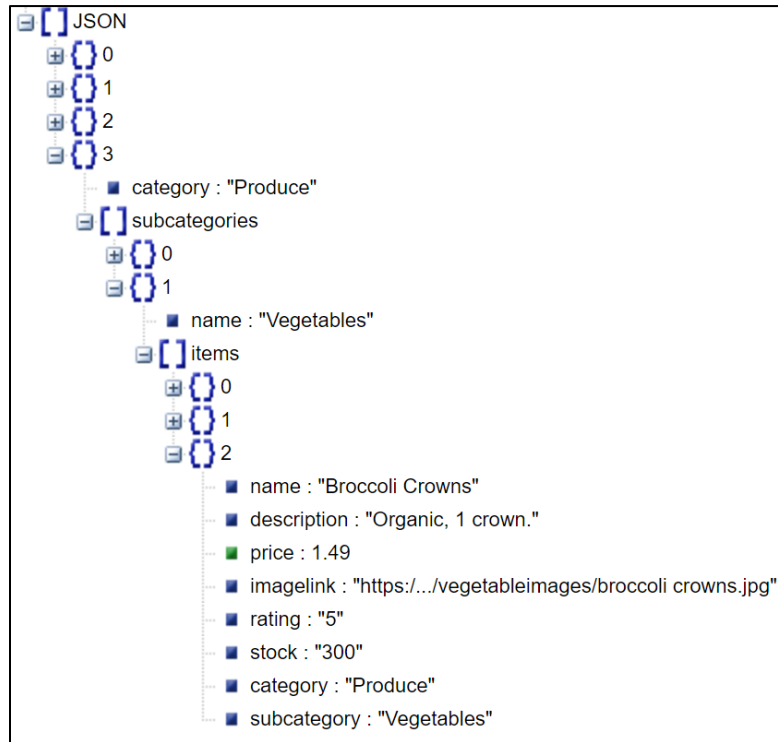


Figure 3 items in JSON Data

### 3 FEATURE LIST

- 1) Home Page
  - a. Home page will contain a carousel that cycles through featured products
  - b. Home page will contain a button that links to the Shopping Page
- 2) Shopping Page
  - a. Shopping Page will contain a collapsible category menu
  - b. Shopping page will display items based on the selected category and filters
  - c. Shopping page will contain a settings bar that allows the user to do the following:
    - i. Sort items by price
    - ii. Show only stocked items
    - iii. Filter items by price
- 3) Product Page
  - a. Product page will contain an image of the product
  - b. Product page will include the following product details:
    - i. Product Name
    - ii. Price
    - iii. Rating
    - iv. Stock status
    - v. Description
  - c. Product page will contain a button to add the product to the cart
  - d. Product page will contain an input field to specify the quantity of items to add to the cart
  - e. Product page will contain a button to go back to the Shopping page

#### 4) Cart Page

- a. Cart page will contain a list of products in the cart. Each line item will include the following:
  - i. Product Thumbnail
  - ii. Product Name
  - iii. Unit Price
  - iv. Quantity (with option to increase and decrease)
  - v. Line Item Total
  - vi. Delete Button
- b. Cart page will contain a cart summary section that lists the subtotal cost, tax cost, and shipping fee.
- c. Cart page will have a checkout button.

#### 5) Header Requirements

- a. Each page will have a header that includes:
  - i. Grocery Cloud title
  - ii. Link to Home Page
  - iii. Link to Shopping Page
  - iv. Link to Cart

#### 6) Footer Requirements

- a. Each page will have a footer that includes:
  - i. Copyright/Logo
  - ii. Email Address
  - iii. Contact Us (web form)
  - iv. About us

•

#### 7) CSS Requirements Listed Here

- a. CSS “must” be used to create the page design, layout, and styling.
- b. CSS “must” exist in a separate file and not be completely inline. Inline CSS should only be used where it is absolutely necessary.
- c. The use of id and classes should be properly utilized to ensure an understanding of the differences between their uses.

#### 8) Git Source Requirements

- a. As you will be required to pass the content off to a grader for review, you will need to make use of a GitHub repository. The Orientation Course for this curriculum offered some Git tutorials and links to Git and GitHub sources.
- b. Feel free to use a local Git repository for your own source and version control if you deem it necessary. This is an option and not required but it provides you with experience using Git for source control, which is one the most popular source control mechanisms in the industry.
- c. Note that by using a GitHub repository, your project files will be public and searchable. This has advantages and disadvantages.
  - i. Advantage is that graders and potential employers will be able to view your code and project files

- ii. Disadvantage is that other students could locate your source and copy it.
- iii. GitHub repositories are public by nature and can only be made private if you elect to pay for a private repository or you have access to one of the educational packages where you can make private repositories.

## 4 WORLDWIDEIMPORTERS FUNCTIONALITY

### 4.1 FEATURE OVERVIEW

As customers visit WorldWideImporters' online grocery ordering site, they will be presented with a home page that is visually appealing. The purpose of the home page is to present the customer with navigation to the entire site which includes access to product pages, and a check out process that displays the items they added to their shopping cart. The home page nav area must include links to Home, Products, Shopping Cart, and About Us pages.

The home page needs to also have a footer section with links to the following items:

- About Us
- Contact Us

The middle area of the home page must include a rotating banner style ad that displays current sales or specials and it also must contain content describing WorldWideImporters and the offerings available on this site.

Each category page (vegetable, fruits, etc.) must include a visually appealing presentation of the products in that category. Rendering must be performed with React or Angular components. Initial product display on the page should show an image and price per qty. Each item must also include an Add to Cart button and a numeric control that permits easy entry of quantity to add. Items must be correctly added to the shopping cart with total price based on quantity multiplied by price.

Shopping cart must display items that were added, quantity added, price per unit, and total cost per item. For shopping cart total, assume a 10% sales tax and standard shipping rate of \$10.00. Shopping cart must also provide an option to remove and/or change the quantity with a recalculate button to update totals accordingly.

About Us page is a static page that describes who WorldWideImporters is. It must still follow the same look and feel as the rest of the site.

Contact Us page must be a form that accepts a user's name, email address, and a text field for entering information that they wish to communicate to WWI.

All pages must be styled with CSS and follow the wireframe provided. They must also maintain a consistent look and feel throughout the site.

### 4.2 PROCESS FLOWS / USE CASES

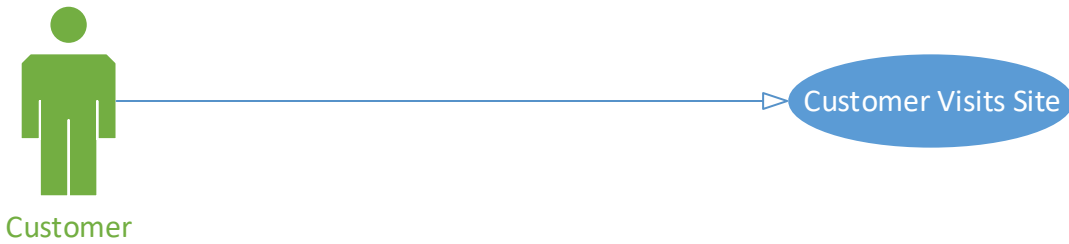
The following use cases will be used for feature implementation:

ID	Use Case Name
1	Customer Visits Site
2	Customer Displays Products
4	Customer Adds Item to Shopping Cart
5	Customer Checks Out
6	Customer Wishes to Contact WorldWideImporters
7	Customer Wants More Information on WorldWideImporters

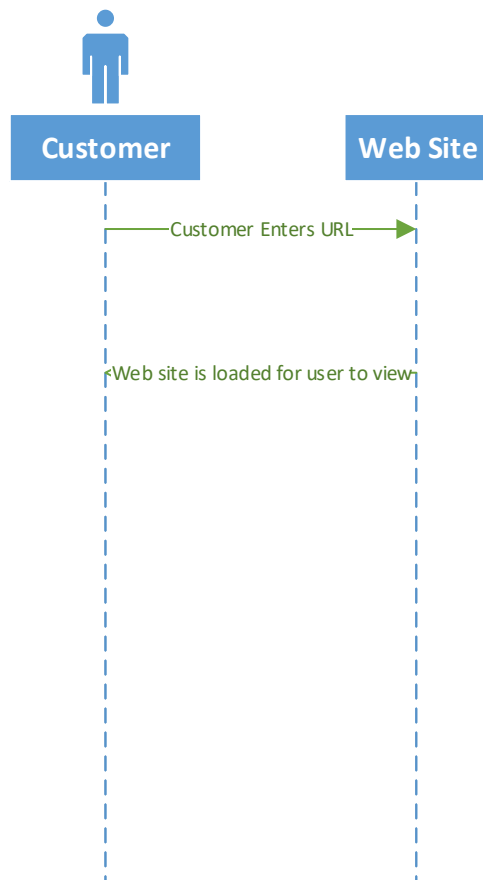


## CUSTOMER VISITS SITE

---



**Figure 2 Site Visit Use Case**



**Figure 3 Customer Site Visit Process Flow**

<b>Primary Actor</b>	Customer
<b>Secondary Actor</b>	
<b>Description</b>	Customer Visits Home Page of site
<b>Precondition</b>	Customer knows URL of site
<b>Trigger</b>	Home page is loaded
<b>Basic Flow</b>	Customer enters URL for home page into browser and the home page loads on the user's computer
<b>Alternate Flow</b>	None

## CUSTOMER DISPLAYS PRODUCTS

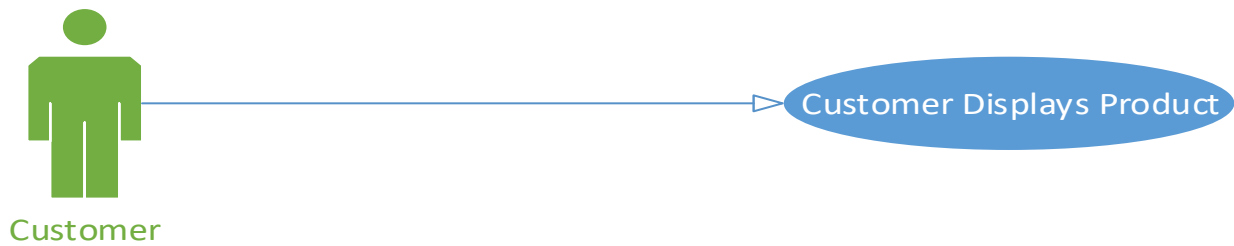
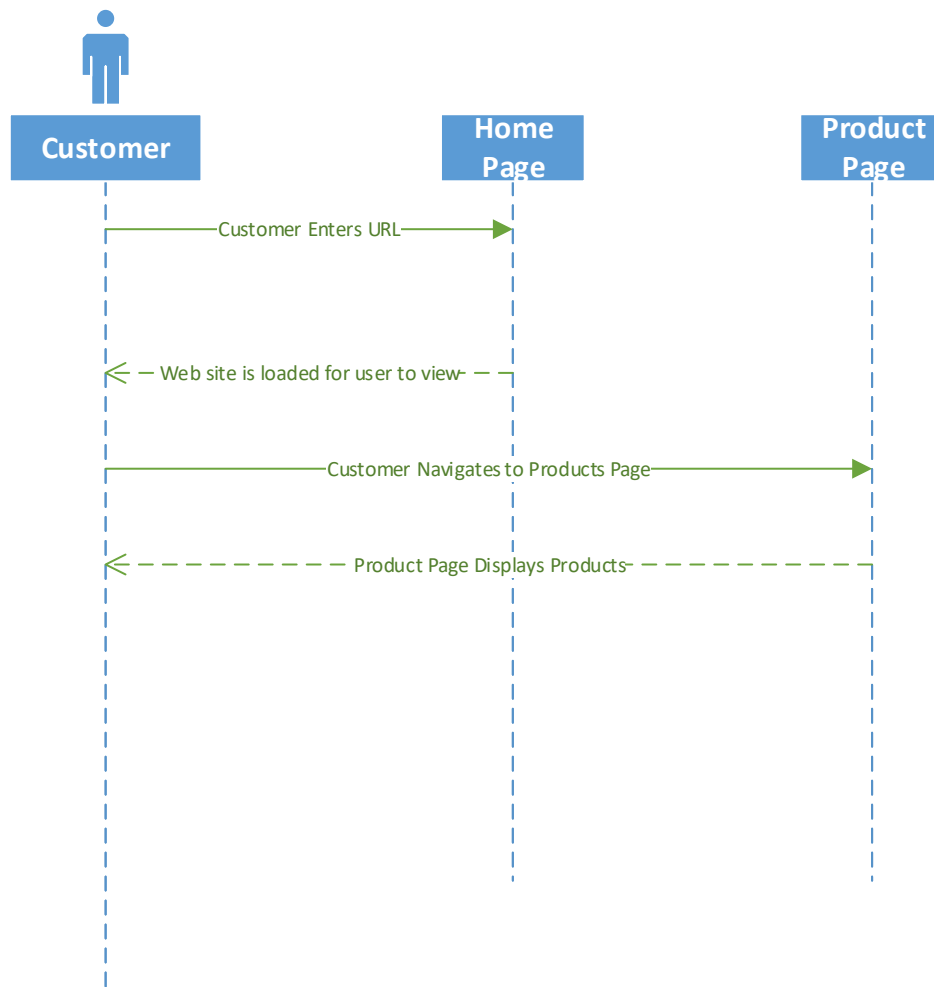


Figure 4 Display Product Use Case



**Figure 5 Display Product Process Flow**

<b>Primary Actor</b>	Customer
<b>Secondary Actor</b>	
<b>Description</b>	Customer navigates to products page
<b>Precondition</b>	Link to products is available in nav section
<b>Trigger</b>	Products page is loaded
<b>Basic Flow</b>	Customer is on the web site and wants to look at the available products so they click the product link in the nav section
<b>Alternate Flow</b>	None

## CUSTOMER ADDS PRODUCT TO SHOPPING CART

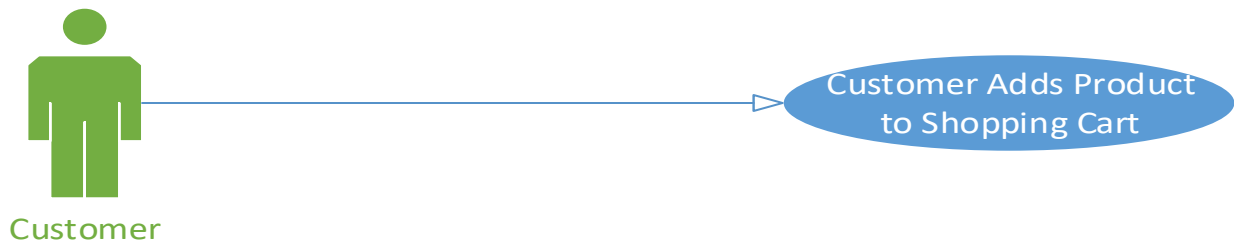


Figure 6 Add Product to Shopping Cart Use Case

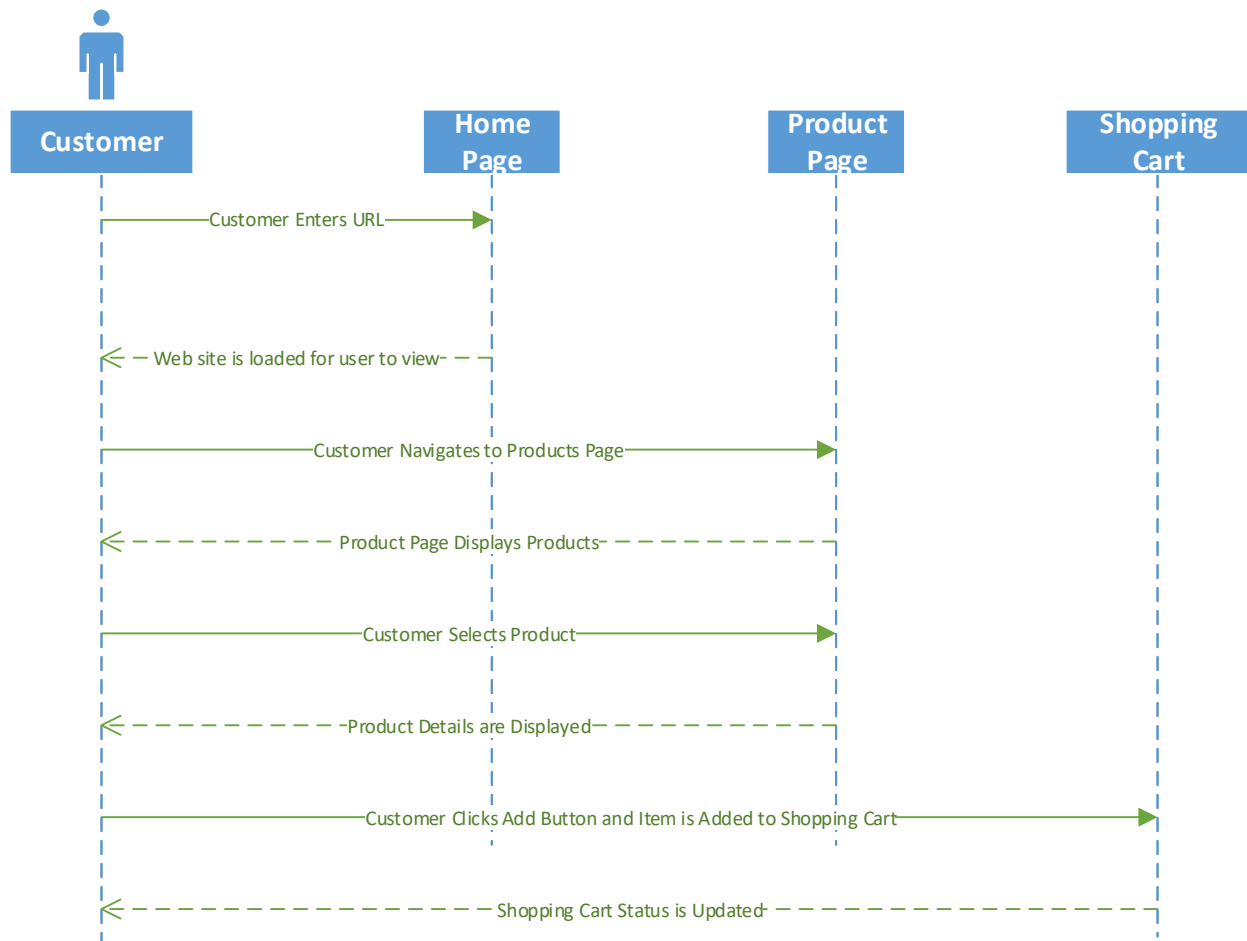


Figure 7 Add Product to Shopping Cart Process Flow

<b>Primary Actor</b>	Customer
<b>Secondary Actor</b>	
<b>Description</b>	Customer adds product(s) to shopping cart
<b>Precondition</b>	Customer has found a product they want to purchase
<b>Trigger</b>	Add button is clicked
<b>Basic Flow</b>	Customer is on the products page, locates the product(s) they want, the click the Add button, and the product is added to the shopping cart.
<b>Alternate Flow</b>	None

## CUSTOMER CHECKS OUT

---



Figure 8 Customer Checks Out Use Case

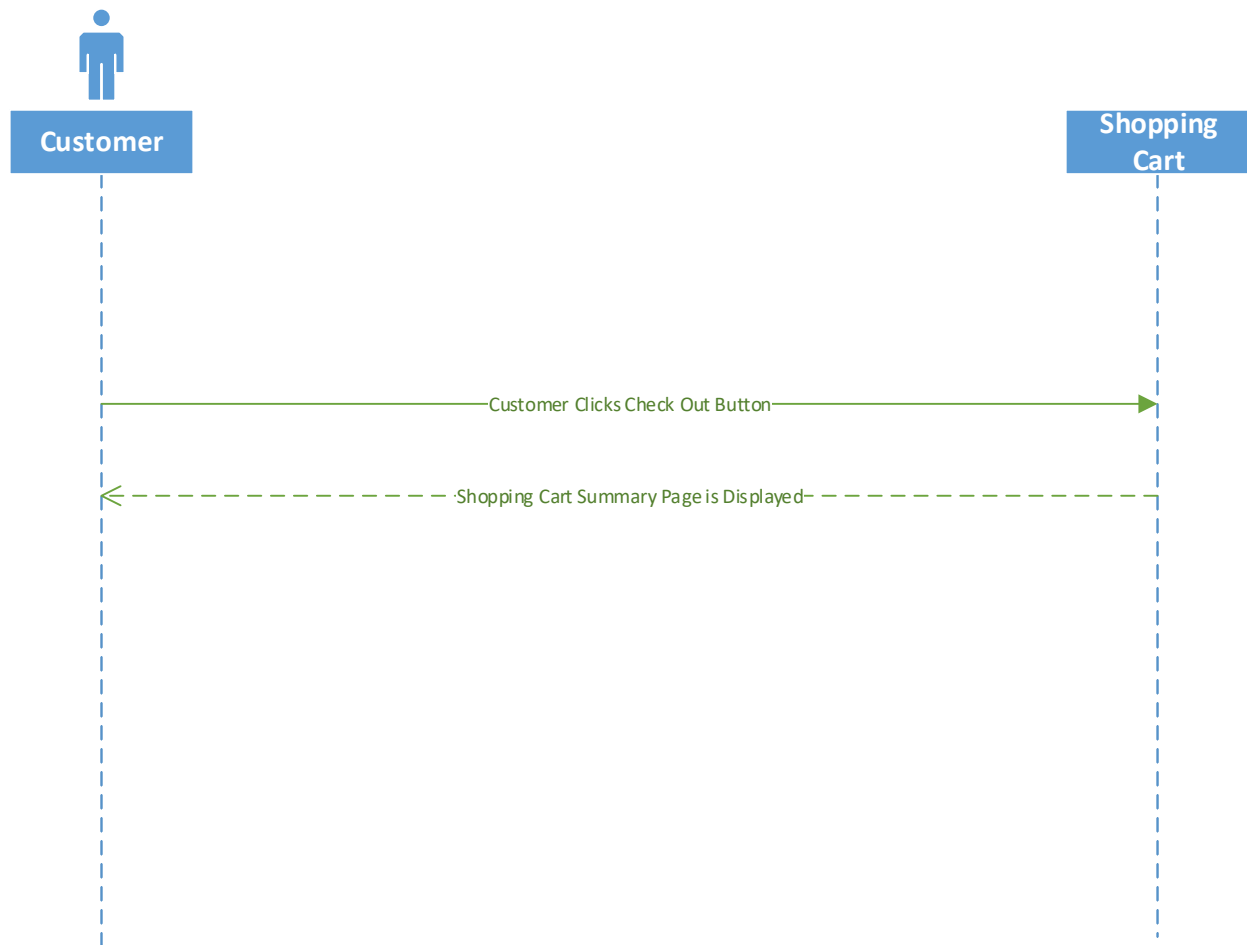
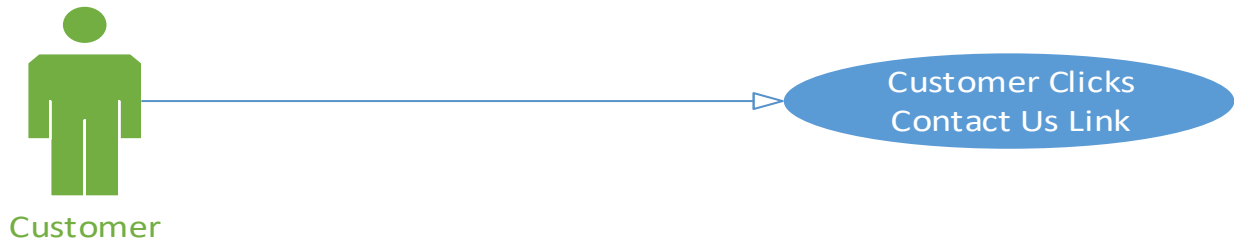


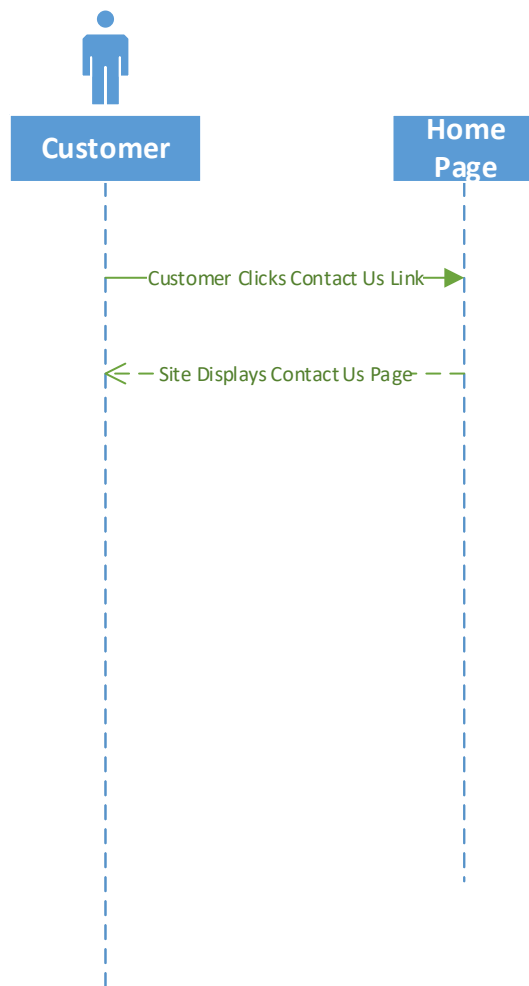
Figure 9 Customer Checks Out Process Flow

Primary Actor	Customer
Secondary Actor	
Description	Customer is finished shopping and wishes to check out and pay
Precondition	Shopping Cart has Items
Trigger	Customer clicks Check Out button
Basic Flow	Customer has added products to the shopping cart and now wishes to check out and end their session. The click the check out button
Alternate Flow	None

## CUSTOMER WISHES TO CONTACT WORLD WIDE IMPORTERS



**Figure 10 Customer Clicks Contact Us Use Case**



**Figure 11 Contact Us Process Flow**

<b>Primary Actor</b>	Customer
<b>Secondary Actor</b>	
<b>Description</b>	Customer clicks the Contact Us link
<b>Precondition</b>	Link for Contact Us is in footer of all pages
<b>Trigger</b>	Contact Us page is loaded
<b>Basic Flow</b>	Customer is on the web site and wants to contact someone at World Wide Importers so they click on the Contact Us link found in the footer of any page
<b>Alternate Flow</b>	None

### CUSTOMER WANTS MORE INFORMATION ABOUT WORLD WIDE IMPORTERS

---

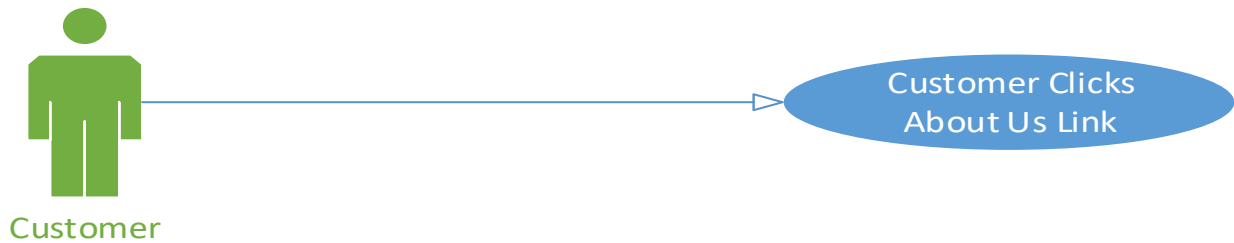
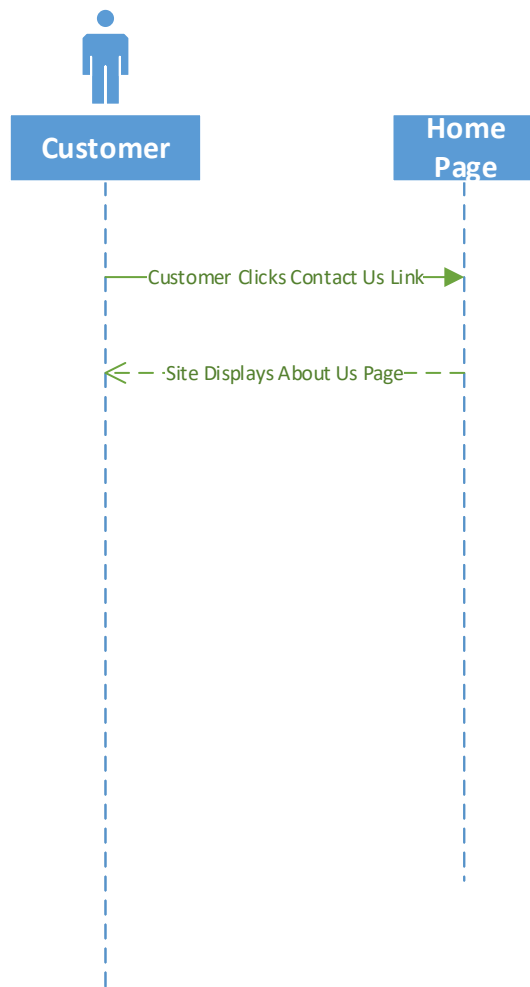


Figure 12 Customer Clicks About Us Use Case





**Figure 13 About Us Process Flow**

<b>Primary Actor</b>	Customer
<b>Secondary Actor</b>	
<b>Description</b>	Customer clicks About Us link to find out more about WWI
<b>Precondition</b>	About Us link is present in footer
<b>Trigger</b>	Customer clicks About Us link
<b>Basic Flow</b>	Customer is on the web site and wants to know more about World Wide Importers and they click the About Us link in the footer of any page
<b>Alternate Flow</b>	None

## 5 WIREFRAMES

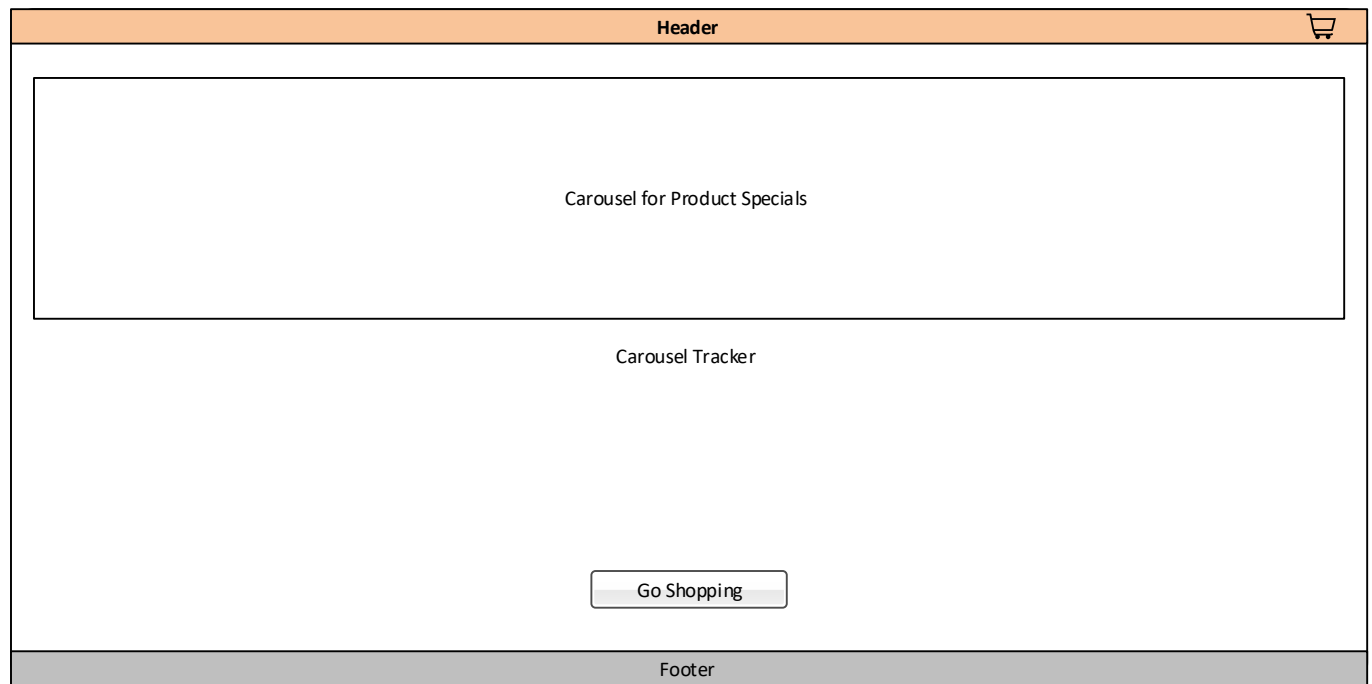


Figure 14 Home Page Wireframe

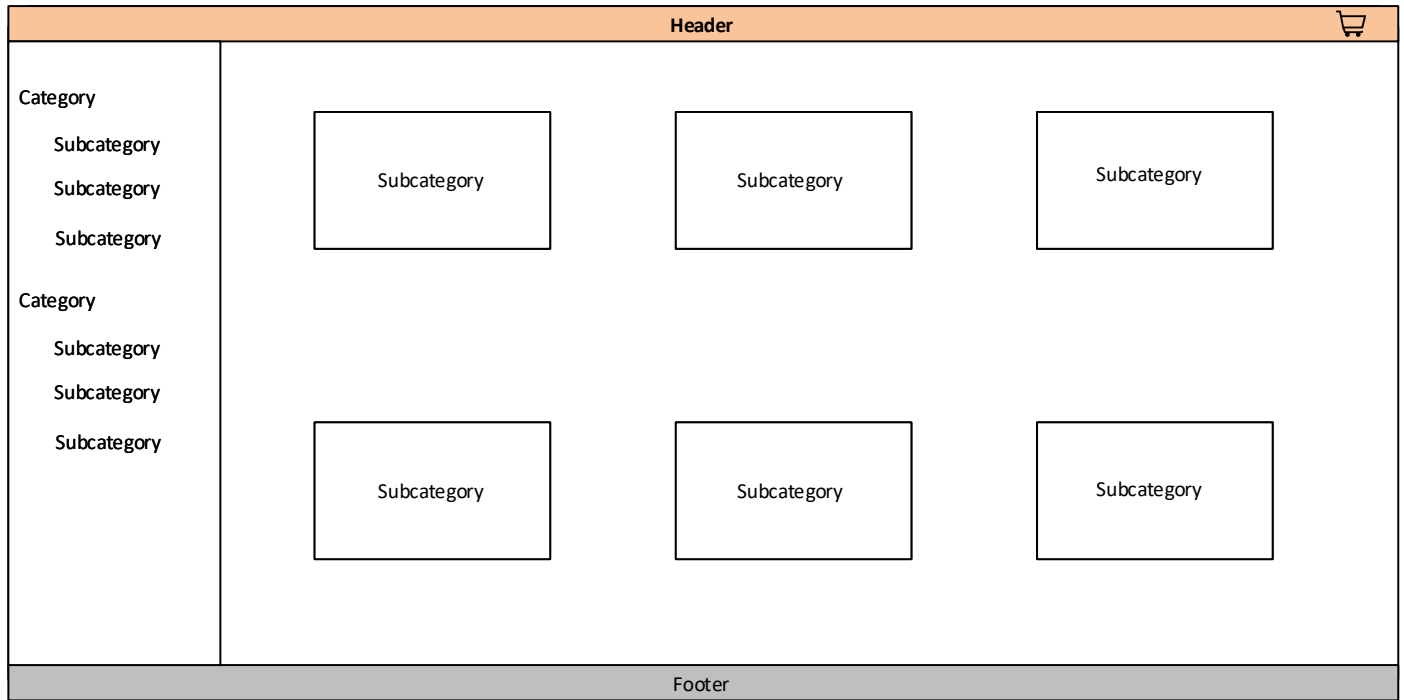


Figure 15 Category and Subcategory Wireframe

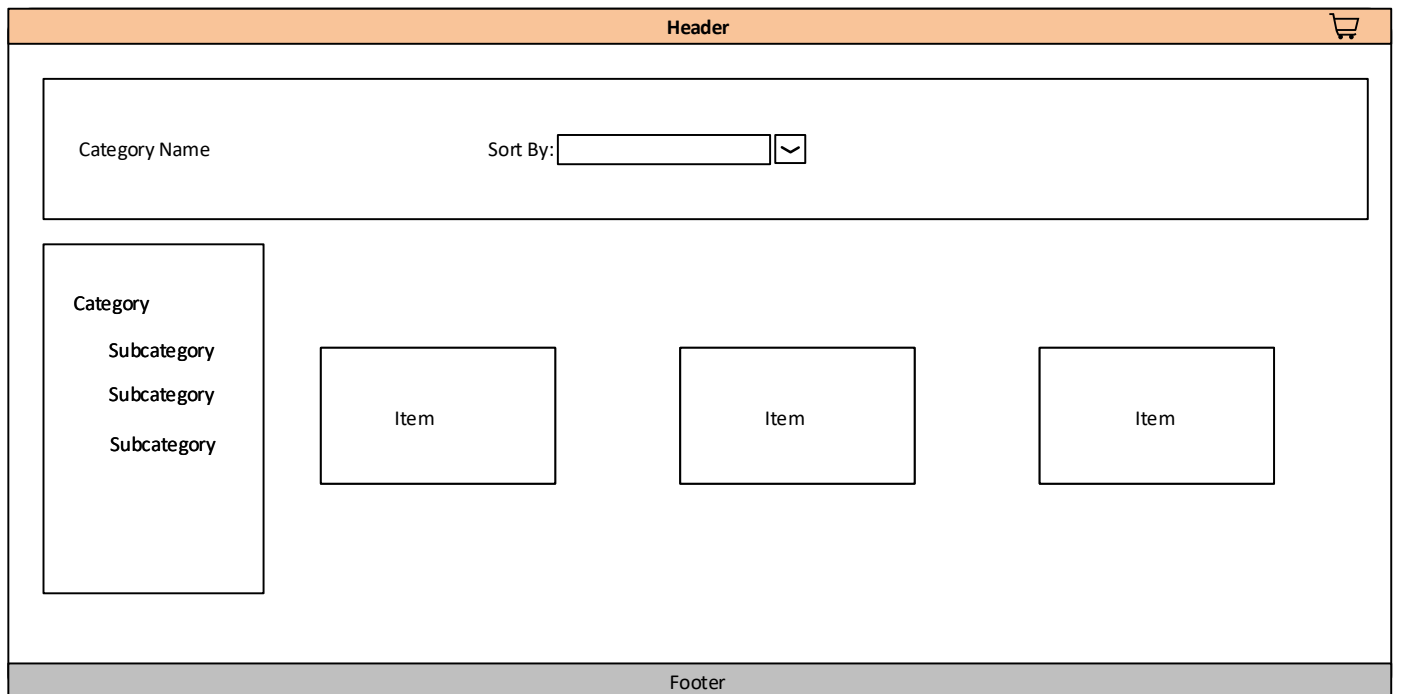


Figure 16 Product Category Wireframe

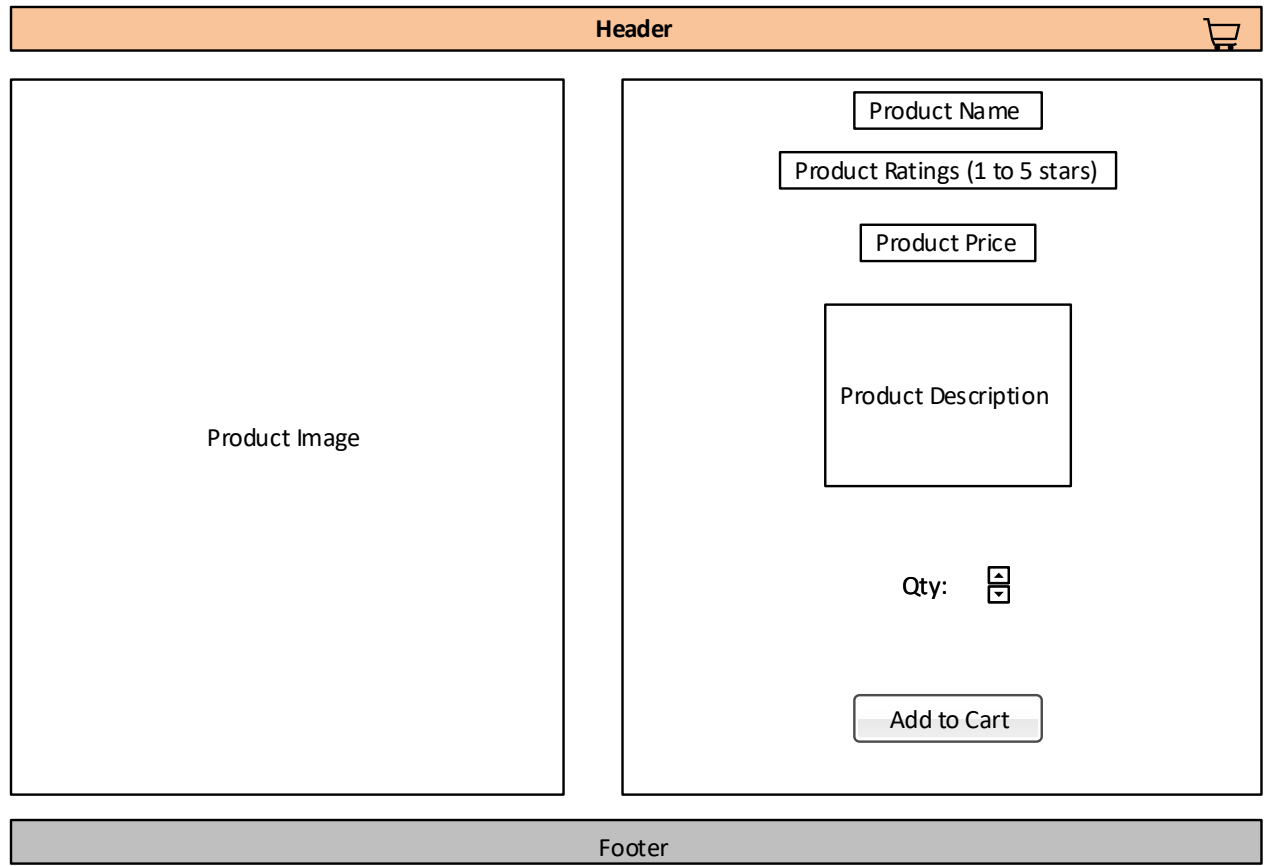


Figure 17 Product Detail Wireframe

Header

Item	Details	Price	Qty	Total
Item	Details	Price	Qty	Total
Item	Details	Price	Qty	Total

Subtotal

Cart Summary

Subtotal

Shipping

Tax

Total

Add to Cart

Footer

Figure 18 Shopping Cart Detail Wireframe

## 6 SUCCESS CRITERIA

### 6.1 GOALS

- 1) Create an online shopping site for WorldWideImporters grocery segment
- 2) Implement the site using the web technologies and frameworks covered in the Web Front-End curriculum
- 3) Follow this specification for specific requirements
- 4) Site must be designed using the wireframe that is provided as part of the requirements documentation
- 5) Fetch API, XMLHttpRequest or third-party library is used to pull data from the Microsoft Azure web service APIs.
- 6) Integrated use of React or Angular for rendering data source into page displays
- 7) Effective use of HTML 5 elements for semantic document structure
- 8) Effective use of Cascading Style Sheets (CSS) for site styling that supports multiple platforms and form factors, such as mobile devices
- 9) Effective use of JavaScript for interactivity, data binding, state management, and navigation as indicated in specific elements of this document pertaining to site design.
- 10) See the grading rubric on the course page for details on how the project will be graded. Note that the rubric is the list of elements that will be used to grade the project and you are required to identify each rubric element in your code as a code comment.

### 6.2 NON GOALS

- 1) Ecommerce is not required for this site. Simply having the ability to add items to cart is sufficient
- 2) Creation of web services or Azure accounts is not required
- 3) Hosting of site on Azure or any third party hosting services