1. Introduction

1.1 Purpose of this document

This document provides all of the requirements for an e-commerce grocery store application. It is intended for the customers and developers of this application.

1.2 Scope of this document

This document intends to cover the requirements for an e-commerce grocery store application. The expected users of this system will be consumers who purchase grocery store items.

1.3 Overview

The system is described as an on-line grocery store application. The primary users will be consumers who will select grocery store items for purchase. These consumers will have the ability to query the system concerning the availability of specific products they wish to select for purchase. In addition to providing availability of products, the system will also require consumers to provide personal information such as name, address, telephone, and e-mail address. Consumers will also be prompted an order fulfillment page which will include their order details and confirmation of order received. The system will also provide the consumer PayPal payment authorization for now and will extend to credit card payment in the future.

In the future, the system will allow consumers to recall a prior order and, depending on the availability of each item, duplicate that order to be their newest order request. Those items which are presently unavailable will be excluded from their newest order request. This "express order" function will benefit consumers by not having to entirely re-key their selections for those instances where they wish to buy similarly to a past order.

2. General Description

2.1 Product Functions

The application should:

- Display page will allow for the display of the website and provide links to the various web pages of the site.
- Has a search function which will retrieve available grocery items relative to a consumer's requests for products having similar characteristics.
- Will have a shopping cart where the items selected by the customer will be added.
- Have a checkout option which allows the consumer to review their order and pay.
- Have a payment function that will allow the consumers to pay by means of PayPal to begin with and will eventually extend to credit card option.
- Have order fulfillment – confirm customers that their order have been fulfilled

- update and maintain inventory levels to ensure consumers are able to purchase those items requested based on availability

2.1.1 Additional functions to be added in the future:

- Log in page
- permit consumers to retrieve a past order to facilitate their newest order request
- permit consumers to alter their selections prior to completing their purchase

2.2 User Characteristics

The user is expected to have experience using personal computers and the internet and should have available a credit card, check or cash.

2.3 General Constraints

- The system must be implemented by May 2013.
- No additional constraints are known at this time.

3. Functional Requirements

3.1 The system will maintain and update item inventory levels.

Description

The system will maintain item inventory levels based upon a consumer's order. The system will also update item inventory levels upon satisfying a consumer's order.

Criticality

High.

Technical issues

Access to a data repository is essential to accomplish this.

Cost and schedule

This is a primary requirement and should therefore constitute as much as 20% of the code development time and cost.

Risks

This requirement is crucial to the entire project's success.

Dependencies with other requirements

This is dependant on requirements 3.3 and 3.6.

3.2 Search function - The system will display available items.

Description

The system will display available items for the consumer to choose from based on their search.

Criticality

High.

Technical issues

Access to a data repository is essential to accomplish this.

Cost and schedule

This is a primary requirement and should therefore constitute as much as 10% of the code development time and cost.

Risks
This requirement is crucial to the entire project's success.
Dependencies with other requirements
This is dependant on requirement 3.1.

3.3 Shopping Cart - The system will accumulate orders.
Description
The system will make use of the shopping cart to accumulate orders of items selected by a consumer.
Criticality
High.
Technical issues
None
Cost and schedule
This is a primary requirement and should therefore constitute as much as 20% of the code development time and cost.
Risks
This requirement is crucial to the entire project's success.
Dependencies with other requirements
This is dependant on requirement 3.2.

3.4 Pay- The system will ask for consumer personal and financial (PayPal) data.
Description
The system will ask for consumer personal data such as name, address, and their PayPal information (optional) to ensure the delivery of products.
Criticality
High.
Technical issues
Access to a data repository is essential to accomplish this.
Cost and schedule
This is a primary requirement and should therefore constitute as much as 20% of the code development time and cost.
Risks
This requirement is crucial to the entire project's success.
Dependencies with other requirements
This is dependant on requirements 3.3 and 3.6.

3.5 Checkout - The system will display order summaries.
Description
The system will display order summaries based on what the consumer has selected.
Criticality
High.
Technical issues
None.
Cost and schedule
This is primary requirement and should therefore constitute as much as 10% of the code development time and cost.

Risks
This requirement is crucial to the entire project's success.
Dependencies with other requirements
This is dependant on requirement 3.3.

3.6 The system will provide an order fulfillment to consumer.
Description
The system will provide an order fulfillment page to the consumer detailing their order details such as items they have ordered, method of payment etc.
Criticality
High.
Technical issues
Access to an internet payment gateway and database repository is essential to accomplish this.
Cost and schedule
This is a primary requirement and should therefore constitute as much as 20% of the code development time and cost.
Risks
This requirement is crucial to the entire project's success.
Dependencies with other requirements
This is dependant on requirement 3.3.

3.7 The system will verify creditability.
Description
The system will verify creditability to determine whether or not to authorize a consumer's order.
Criticality
None.
Technical issues
Access to an internet payment gateway is essential to accomplish this.
Cost and schedule
This is a secondary requirement and will be added to the system in the future.
Risks
None.
Dependencies with other requirements
This is dependant on requirement 3.3.

3.8 The system will store order data.
Description
The system will store order data relative to a consumer's order.
Criticality
None.
Technical issues
Access to a data repository is essential to accomplish this.
Cost and schedule
This is a secondary requirement and will be added to the system in the future.
Risks
None.

Dependencies with other requirements
This is dependant on requirements 3.3 and 3.6.

3.9 The system will retrieve order data.

Description
The system will retrieve order data relative to a consumer's prior purchase
Criticality
None.
Technical issues
Access to a data repository is essential to accomplish this.
Cost and schedule
This is a secondary requirement and will be added to the system in the future.
Risks
None.
Dependencies with other requirements
This is dependant on requirement 3.6.

3.10 The system will have a login page.

Description
The system will retrieve order and consumer information upon logging in.
Criticality
None.
Technical issues
Access to a data repository is essential to accomplish this.
Cost and schedule
This is a secondary requirement and will be added to the system in the future.
Risks
None.
Dependencies with other requirements
This is dependent on requirement 3.6.

4. Interface Requirements

4.1 User interfaces

A Graphical User Interface (GUI) will be present throughout the application. The appearance of the interface will be dependent upon the state of the application the user is presently in.

4.2 Hardware interfaces

- Web Server capable of handling processing load
- Internet Connection capable of handling TBD traffic

4.3 Software interfaces

The application will require an "internet payment gateway" which will provide internet connectivity between the buyer, seller and their representative financial institutions.

Microsoft Visual Web Developer 2008 Express Edition, SQL Server 2008 Express Edition, and SQL Server Management Studio Express.

4.4 Communications interfaces

The application will require an “internet payment gateway” interface.

5. Performance Requirements

The application must perform at like speed, co-operate in like memory and require normalized disk utilization as that of similar e-commerce retail applications.

6. Other Non-functional Attributes

6.1 Security

The application must ensure privacy regarding PayPal information and other personal information entered by a user.

6.2 Reliability

The application must provide complete functionality whenever in use. Application failures are not to impair the data or the application.

6.3 Maintainability

The application is to be maintained by the proprietor once it has been delivered.

6.4 Portability

The application must account for all known portability issues.

6.5 Extensibility

The application must be modifiable to support future enhancements.

6.6 Reusability

The application components must be reusable for present and future versions of this application.

6.7 Resource Utilization

The application should be limited to those resources normally allocated to standard commercial PC applications.

7. Adopted Coding Standard

For HTML and CSS: W3C Markup validation service. This validator checks the markup validity of Web documents in HTML, XHTML, SMIL, MathML, etc.

For ASP: Validation server control is used to validate the data of an input control. If the data does not pass validation, it will display an error message to the user.

For SQL: Mimer SQL Validators for validating the sql code

This coding standard requires or recommends certain practices for developing out website using the languages C#, ASP.NET, HTML, CSS, SQL and any others we may choose to implement as the project progresses from one phase to another. The objective of this coding standard is to have a positive effect on

- Avoidance of errors/bugs due to naming conventions etc.
- Maintainability, by promoting a unified style
- Performance, by improving unified practices

As discussed in class, "A good coding standards document will contain guidance for naming of files, modules, functions, variables, and constants. In addition to describing the layout of the code at the small level, the document can also act as an official and easily reference able guide to the overall competent structure. When adding a feature to a component, the coding standards document should give a fairly clear idea of what file and module for feature belongs in, what its functions should be called, and how the calling system should look.

At the small scale, coding standards documents should contain stylistic guidelines for blocks and lines of code. This typically contains rules about the use of indentation, whitespace, brace positioning, comment style, and so on."

Since our initiative is to develop a small scale e-commerce website, this particular coding standard document will provide guidelines to naming conventions, indentation, comment style etc.

Case Sensitivity

The languages we have chosen to incorporate in our project are all case sensitive. Keywords, methods, and variables must be entered with the proper case. As any other language, this language has preference for lower case words. Keywords such as if, for, while, etc. are always written in lower case letters.

Indentation

Indentation is one of the hottest topics for developers. Depending on the developer, indentations type, size, etc. are changed. To create a unified format, these are the rules we choose:

- We always indent with tabs
- The tabs are ½ inch wide
- Code falls against the left margin

Whitespace

Whitespace will be included after conditional operators, between declaration, variables, after every comma, braces, argument lists, binary operators, and so on.

Statements

- Compound statements are statements that contain lists of statements enclosed in braces

Example: {statements}.

- The enclosed statements should be indented one more level than the compound statement.
- The opening brace should be at the beginning of the line following the line that begins the compound statement and be indented to the beginning of the compound statement. The closing brace should begin a line and be indented to the beginning of the compound statement.
- Braces are used around all statements, even single statements, when they are part of a control structure, such as an if-else or for statement. This makes it easier to add statements without accidentally introducing bugs due to forgetting to add braces.

Blank lines

Blank lines improve readability by setting off sections of code that are logically related. One blank line should always be used in the following circumstances:

- Between the local variables in a method and its first statement
- Between logical sections inside a method to improve readability
- After the closing brace of a code block that is not followed by another closing brace

Wrapping Lines

When an expression will not fit on a single line, break it up according to these general principles:

- Break after a comma
- Break after an operator
- Prefer higher-level breaks to lower-level breaks
- Align the new line with the beginning of the expression at the same level on the previous line

Comments

Comments provide a way for the reader to understand what the function of each function is. Therefore, comments will be included throughout our entire to provide easy navigation and help us to identify the purpose of a functions accurately one or two months from the day the code was written. Comments will be included before the function they are pertaining to.

- HTML & CSS: single line or multi-line comments will be delimited in the following format
  <!--- --->
- Other languages the comments are delimited by "//" for single line comments and /*…..*/ for block comments.

Naming

Classes are named using a noun or noun phrase. The first of each word will be capitalized. Do not use type prefix, such as C for class, etc. When appropriate, use a compound word to name a derived class. The second part of the derived class's name

should be the name of the base class. For example, the base class name is Exception. The derived class name is AppException. The file name should reflect the class(es) it contains.

Use descriptive parameter, variable, and constant names. Use names that describe the meaning of the parameter rather than its type. For example, if we are trying to create a constant for the value of Pi so that it could be used universally. Label this constant pi. So on and so forth.

Line endings

C# and ASP.NET will use a semicolon as a line termination character. Every line of code must end with this semicolon, except when you are defining a block structure such as a method, a conditional statement, or a looping construct. By omitting this semicolon, you can easily split a line of code over multiple physical lines.

8. Website Design

8.1 Number of Web Pages

This website will contain all three main categories of pages: Navigation Pages, Content Pages, and Functional Pages.

Navigational Pages – the purpose of navigational pages is to direct people to the content they are looking for; examples include home page, landing pages, and galleries.

- Home Page – provides a dashboard-like view into the rest of the pages, home pages direct visitors to key areas of the website.

Content Pages – These pages are the substance of your site and why people are ultimately there; examples include articles and product pages.

- Product Page – will include information such as product pictures, description, details, related products etc.

Functional Pages – These pages allow people to perform a task, such as conduct a search or check email; examples include search pages, submission forms, and applications.

- Search Page – will allow for the consumer to search for the product they are interested in.
- Submission form – allows people to submit information like the checkout form which will ask for consumer personal and financial data.

This website will contain: Home Page, Product List Page (Grocery, Health & Beauty, General, and Natural-Organic), Shopping Cart Page, Search page, Checkout Page, and Error Page.

Home Page contains the main interface between the consumer and the web. It will introduce the consumers to the website and the products on the website.

Product List Page will list the variety of products categorized by categories (Grocery, Health & Beauty, General, & Natural-Organic) available to the consumers to buy over the internet. These products will be categorized to ease the navigation for the consumer.

Shopping Cart Page - provides a list of the products that the consumer has added to the shopping cart. It will also provide the option for the consumer to remove or add items. This page will also provide a link to the checkout page.

Search page – the search option will be included on the home page and the search page will provide the search results of the user specific search.

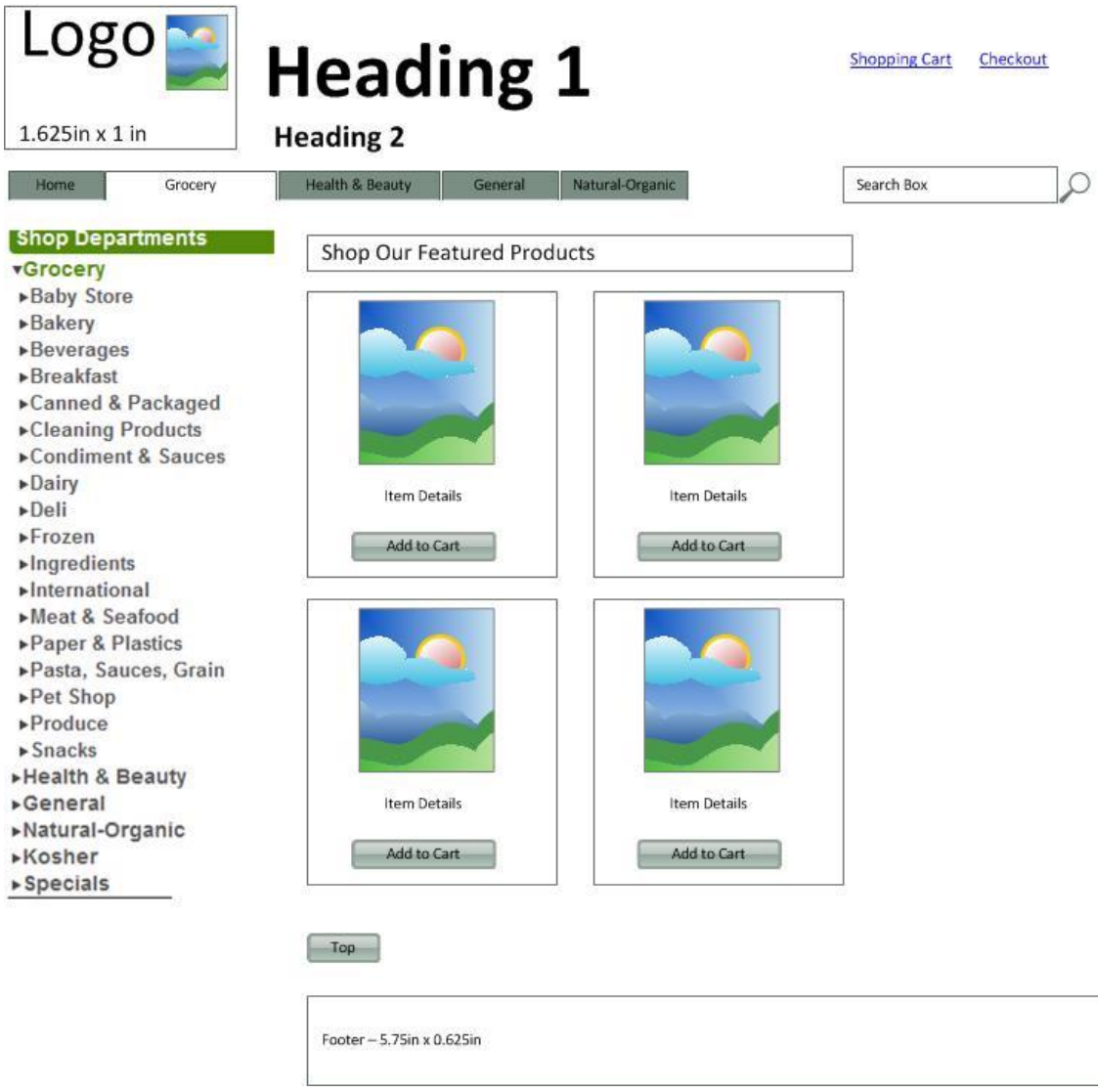
Checkout page – will be displayed via the shopping cart page after the user has pushed the checkout page. This page will contain a link to pay for the groceries via a third party payment processor in this case PayPal. Upon payment processing, the user will be leaded on to a checkout confirmation page which will let the user know if their order has been processed accurately or if there were any errors.

Error page is the page that will be displayed in the browser in case of an error occur in handling something on the website.

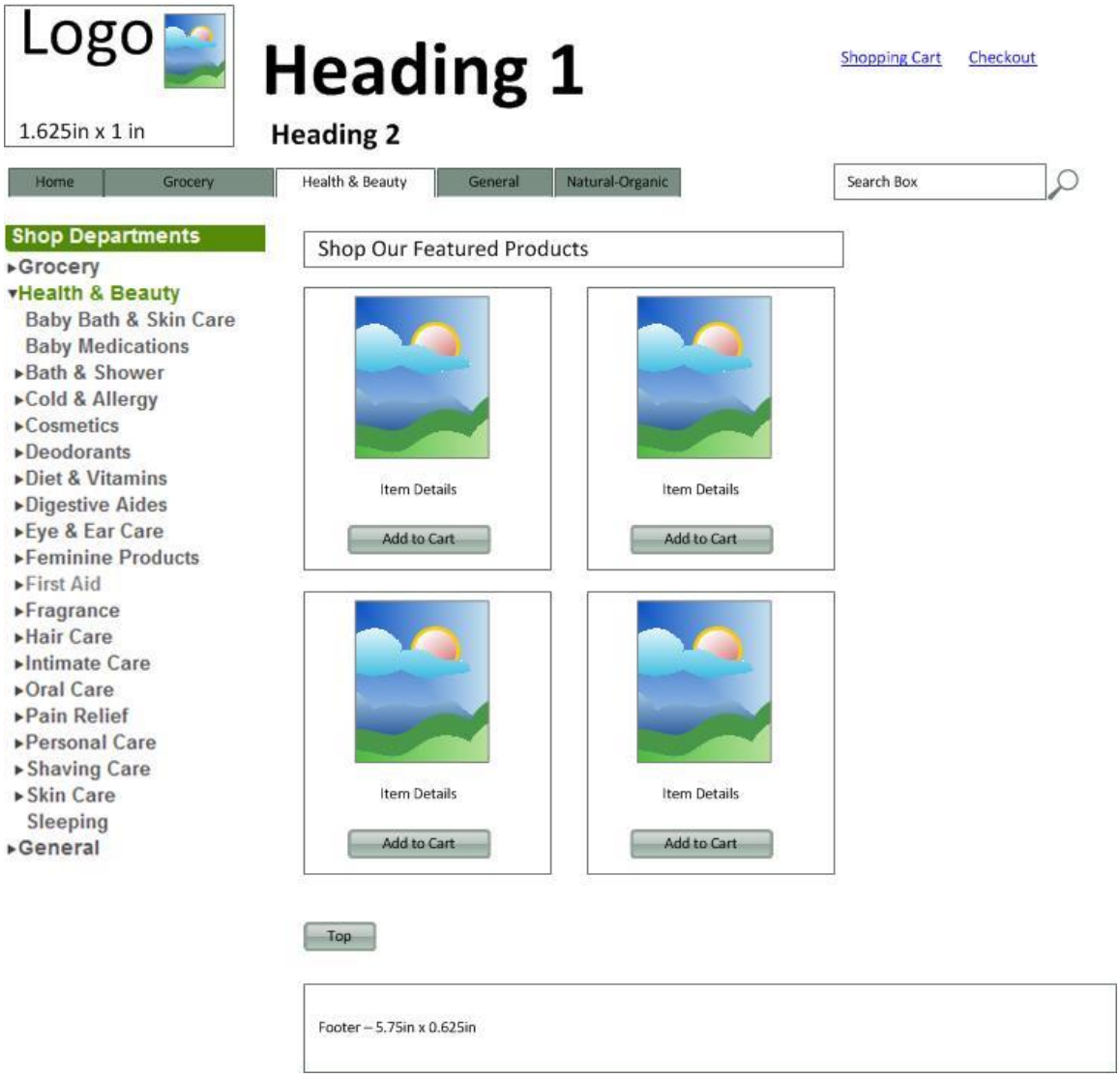
8.2 Wireframes

Home Screen

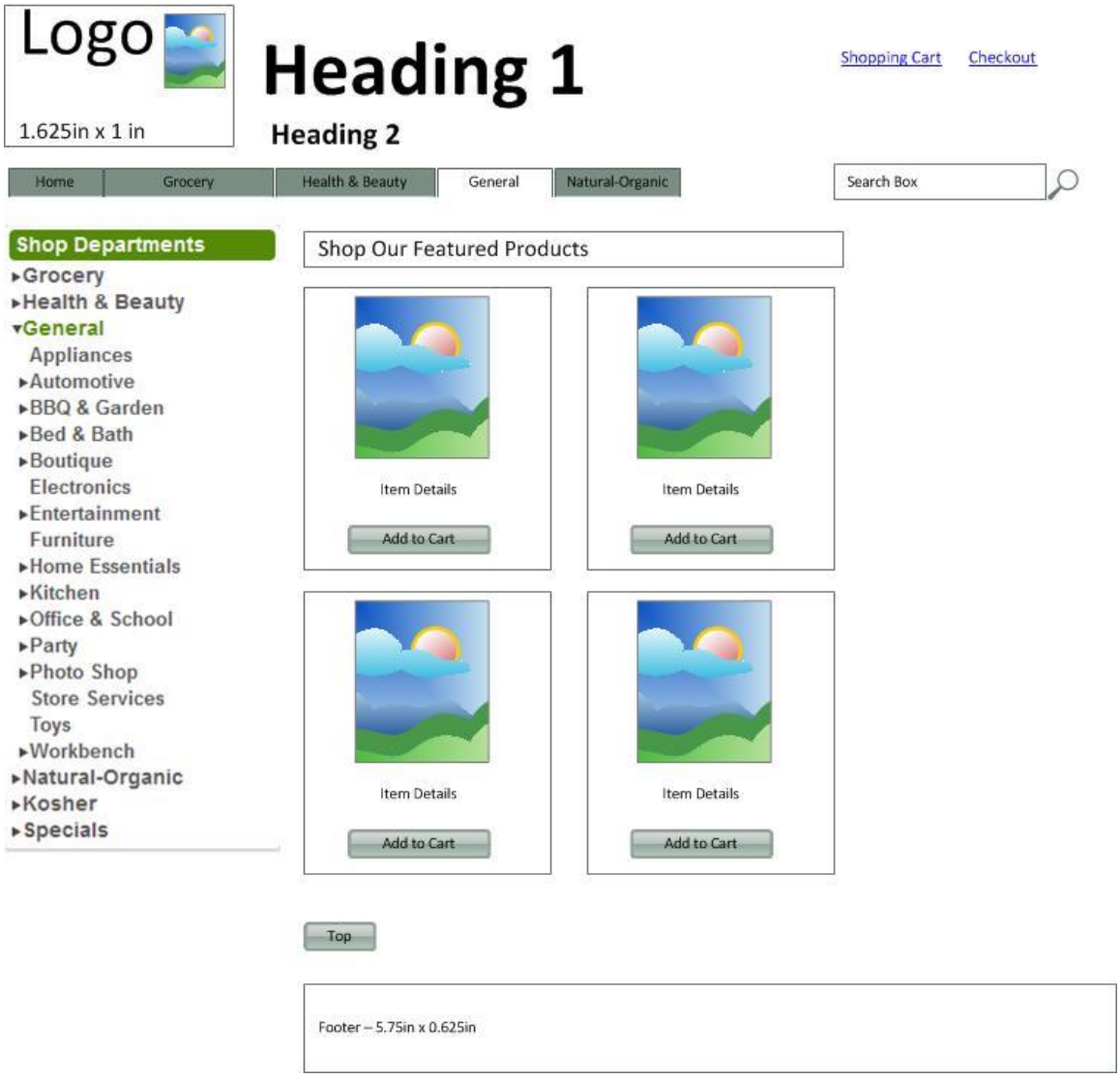
Grocery Page


Logo
1.625in x 1 in
Heading 1
Heading 2
Shopping Cart
Checkout
Home
Grocery
Health & Beauty
General
Natural-Organic
Search Box
Shop Departments
Grocery
Baby Store
Bakery
Beverages
Breakfast
Canned & Packaged
Cleaning Products
Condiment & Sauces
Dairy
Deli
Frozen
Ingredients
International
Meat & Seafood
Paper & Plastics
Pasta, Sauces, Grain
Pet Shop
Produce
Snacks
Health & Beauty
General
Natural-Organic
Kosher
Specials
Shop Our Featured Products
Item Details
Add to Cart
Item Details
Add to Cart
Item Details
Add to Cart
Item Details
Add to Cart
Top
Footer – 5.75in x 0.625in

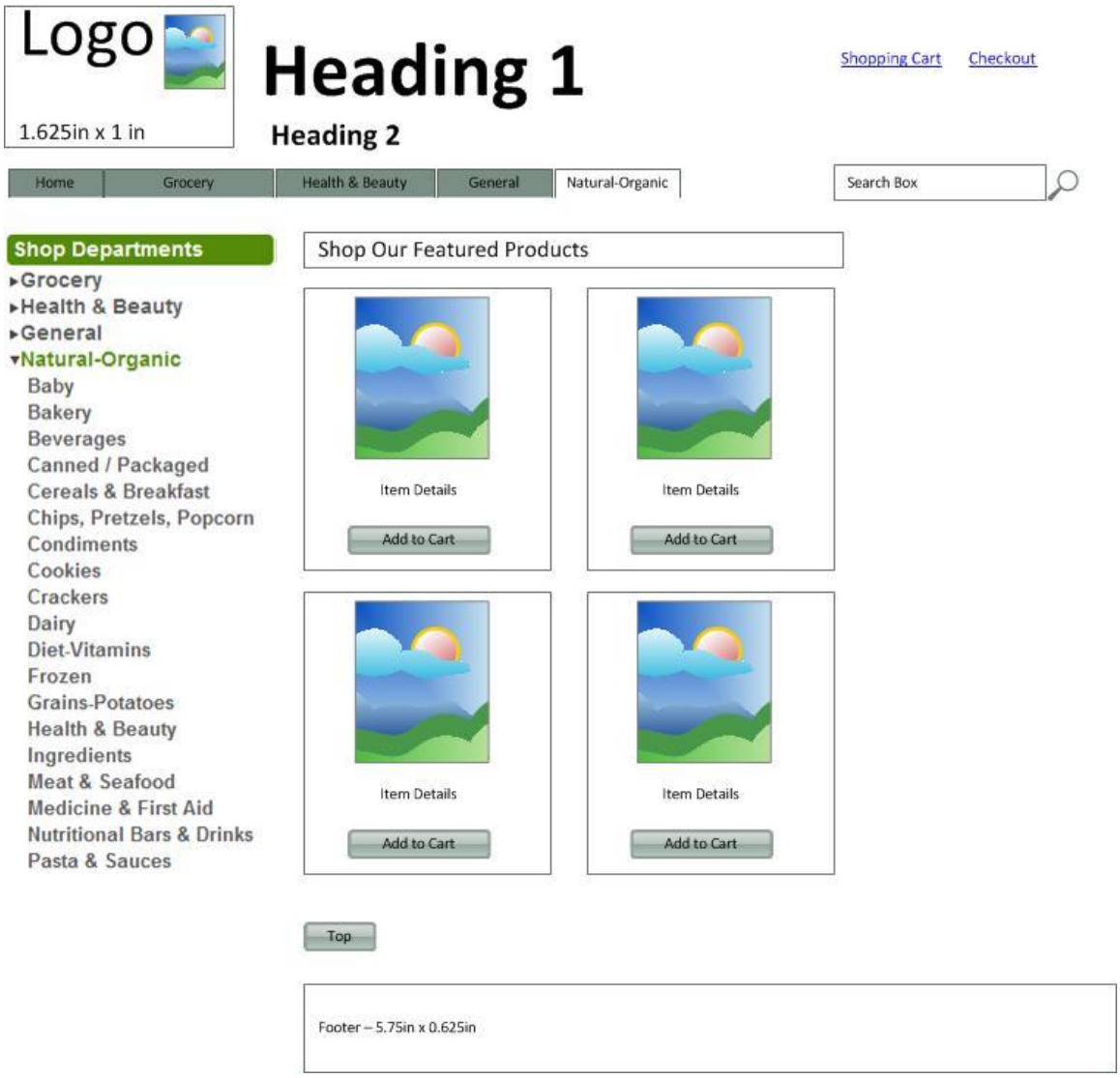
Health & Beauty



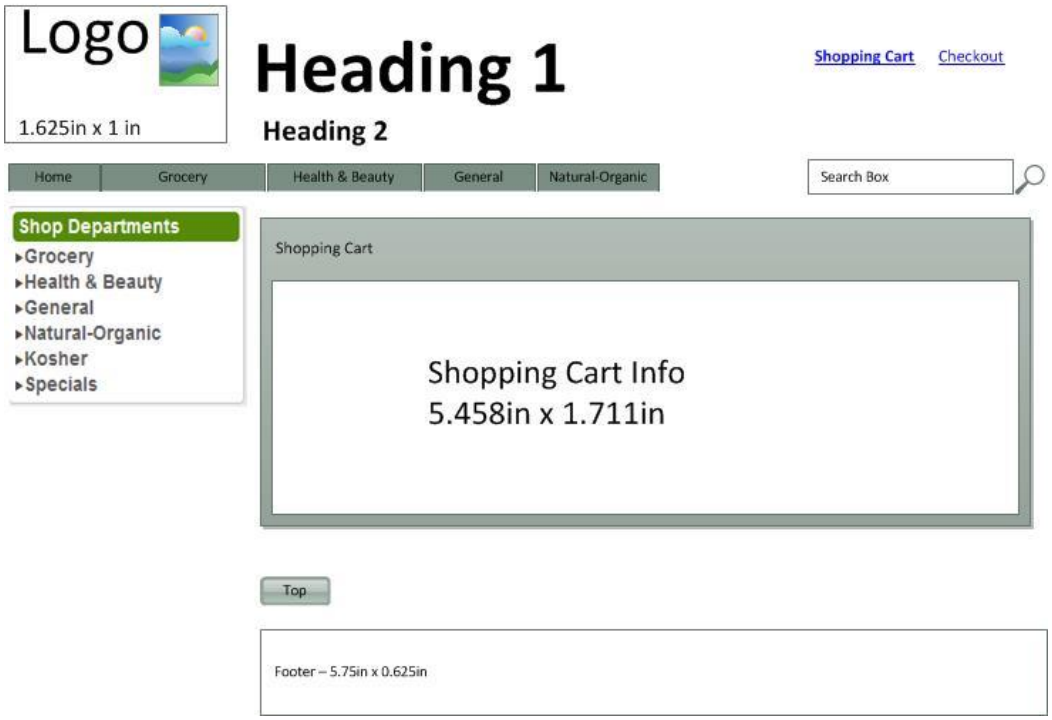
General


Logo
1.625in x 1 in
Heading 1
Heading 2
Shopping Cart
Checkout
Home
Grocery
Health & Beauty
General
Natural-Organic
Search Box
Shop Departments
Grocery
Health & Beauty
General
Appliances
Automotive
BBQ & Garden
Bed & Bath
Boutique
Electronics
Entertainment
Furniture
Home Essentials
Kitchen
Office & School
Party
Photo Shop
Store Services
Toys
Workbench
Natural-Organic
Kosher
Specials
Shop Our Featured Products
Item Details
Add to Cart
Item Details
Add to Cart
Item Details
Add to Cart
Item Details
Add to Cart
Top
Footer – 5.75in x 0.625in

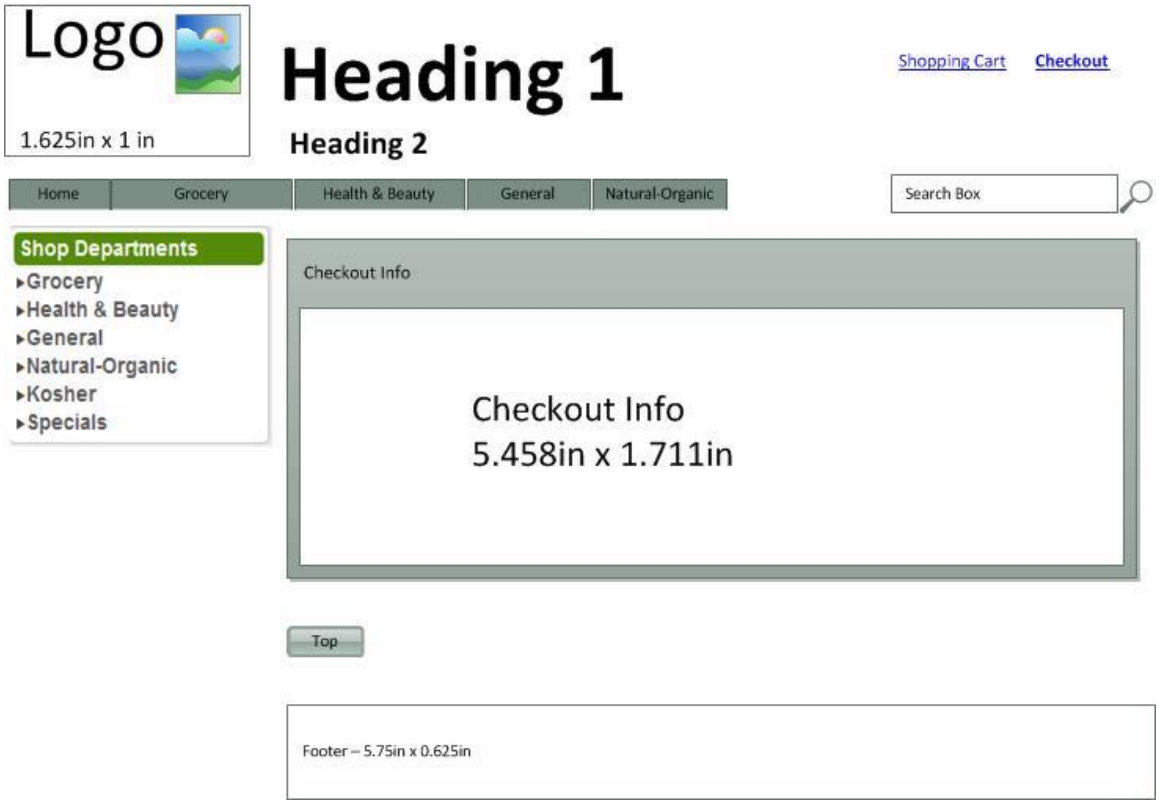
Natural Organic



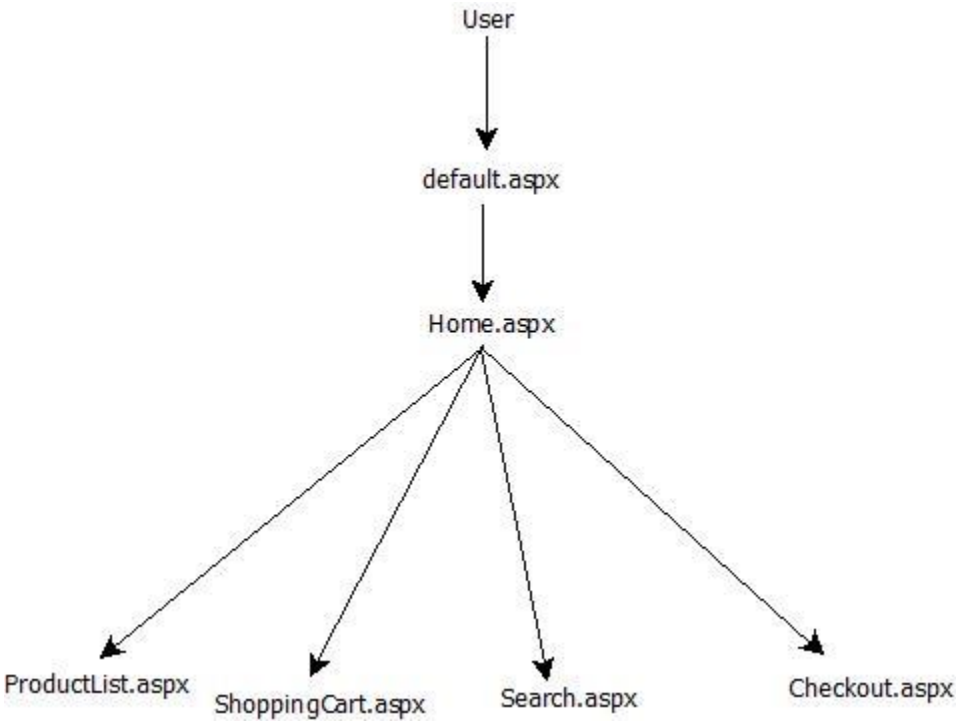
Shopping Cart



Checkout



8.3 Site Flow



9. User Story

The e-commerce grocery store application is a web based application which will be written to conform to version 6.0 or higher of Internet Explorer. Output will be displayed in HTML format in the default browser.

Pre-Condition: The pre-condition for each use case will require the user to be accessing the e-commerce grocery store web page.

9.1 Use case: View Grocery Items for Selection

9.1.1 Diagram

9.1.2 Brief Description

The use case View Grocery Items for Selection is initiated by the consumer to display items available for purchase.

9.1.3 Initial Step-by-Step Description

Before this use case can be initiated, the consumer has accessed the e-commerce grocery store web page.
1. The consumer specifies category of grocery item (all, dairy, health & beauty, frozen, etc).
2. The consumer specifies brand name or keyword.
3. The consumer presses submit.
9. The system displays results based on specifications in 1 and 2 above.

9.1.4 Detailed Description

Use Case Name	View Grocery Items for Selection
Priority	Essential
Trigger	Specifying search criteria and pressing submit.
Precondition	User has accessed the e-commerce grocery store application web page.
Basic Path	The consumer accesses the e-commerce grocery store application web page. The system provides consumer with categories for search. The consumer specifies search criteria and presses submit button. The system displays results based on specifications. The instance of this use case terminates.
Alternative Paths	None.
Postcondition	Available grocery item information is displayed.
Exception Paths	If an error occurs during retrieval the application provides a specific error message relative to the cause. The consumer has the ability to try again.
See Also	
Possible Enhancement	

9.2 Use case: Specify Grocery Item Order Quantity

9.2.1 Diagram

9.2.2 Brief Description

The use case Specify Grocery Item Order Quantity is initiated by the consumer to specify the quantity of grocery item(s) for an order.

9.2.3 Initial Step-by-Step Description

Before this use case can be initiated, the consumer has viewed grocery items for selection.

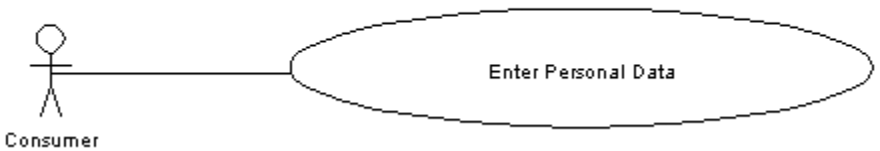
1. The consumer specifies quantity for selected grocery item(s).
2. The consumer presses order.
3. The system accumulates the consumer's order.

9.2.4 Detailed Description

Use Case Name	Specify Grocery Item Order Quantity
Priority	Essential
Trigger	Selecting submit.
Precondition	Grocery items appear in view.
Basic Path	The consumer specifies quantity for select items and presses order button. The system accumulates the consumer's order displaying results based on specifications. The instance of this use case terminates.
Alternative Paths	None.
Postcondition	Selected grocery item(s) information is displayed.
Exception Paths	If an error occurs while specifying order quantity the application provides a specific error message relative to the cause. The consumer has the ability to try again.
See Also	
Possible Enhancement	Allow consumer to view nutritional information for select products.

9.3 Use case: Enter Personal & Financial Data

9.3.1 Diagram



9.3.2 Brief Description

The use case Enter Personal Data is initiated by the consumer to enter personal data.

9.3.3 Initial Step-by-Step Description

Before this use case can be initiated, the consumer has specified order quantities for specific grocery items.

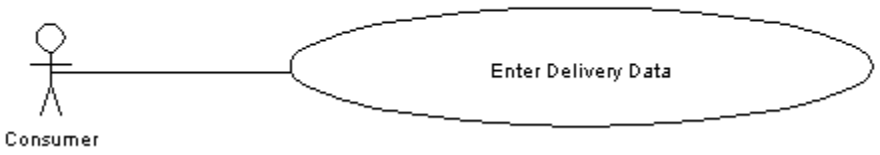
1. The consumer enters personal data.
2. The consumer presses continue.
3. The system stores consumer personal data.

9.3.4 Detailed Description

Use Case Name	Enter Personal & Financial Data
Priority	Essential
Trigger	Selecting checkout.
Precondition	The system has accumulated orders for a consumer.
Basic Path	The consumer enters personal data The system stores the consumer's personal data based on specifications. The instance of this use case terminates.
Alternative Paths	None.
Postcondition	Personal & Financial data is stored.
Exception Paths	If an error occurs while specifying personal data the application provides a specific error message relative to the cause. The consumer has the ability to try again.
See Also	
Possible Enhancement	

9.4 Use case: Enter Delivery Data

9.4.1 Diagram



9.4.2 Brief Description

The use case Enter Delivery Data is initiated by the consumer to enter delivery data.

9.4.3 Initial Step-by-Step Description

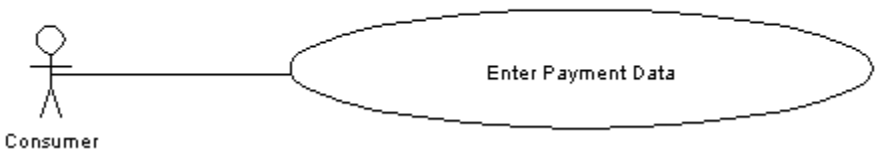
Before this use case can be initiated, the consumer has entered personal data.
1. The consumer enters delivery data.
2. The consumer presses continue.
3. The system stores consumer delivery data.

9.4.4 Detailed Description

Use Case Name	Enter Delivery Data
Priority	Essential
Trigger	Selecting continue.
Precondition	The system has accepted personal data.
Basic Path	The consumer enters delivery data The system stores the consumer's delivery data based on specifications. The instance of this use case terminates.
Alternative Paths	None.
Postcondition	Delivery data is stored.
Exception Paths	If an error occurs while specifying delivery data the application provides a specific error message relative to the cause. The consumer has the ability to try again.
See Also	
Possible Enhancement	

9.5 Use case: Enter Payment Data

9.5.1 Diagram



9.5.2 Brief Description

The use case Enter Payment Data is initiated by the consumer to enter payment data.

9.5.3 Initial Step-by-Step Description

Before this use case can be initiated, the consumer has entered delivery data.
1. The consumer enters payment data.
2. The consumer presses process.

3. The system accesses a payment gateway system then accepts or declines payment data.
4. If the system accepts payment data, the system stores order data.
5. If the system indicates payment data was declined, the system permits consumer to re-enter payment data or cancel.

9.5.4 Detailed Description

Use Case Name	Enter Payment Data
Priority	Essential
Trigger	Selecting continue.
Precondition	The system has accepted delivery data.
Basic Path	The consumer enters payment data If accepted the system stores order data else the system permits consumer to re-enter or cancel. The instance of this use case terminates.
Alternative Paths	None.
Postcondition	Payment data is stored.
Exception Paths	If an error occurs while specifying payment data the application provides a specific error message relative to the cause. The consumer has the ability to try again.
See Also	
Possible Enhancement	On acceptance, send e-mail to consumer detailing their order.

10. Sequence Diagrams

11. Class Diagrams

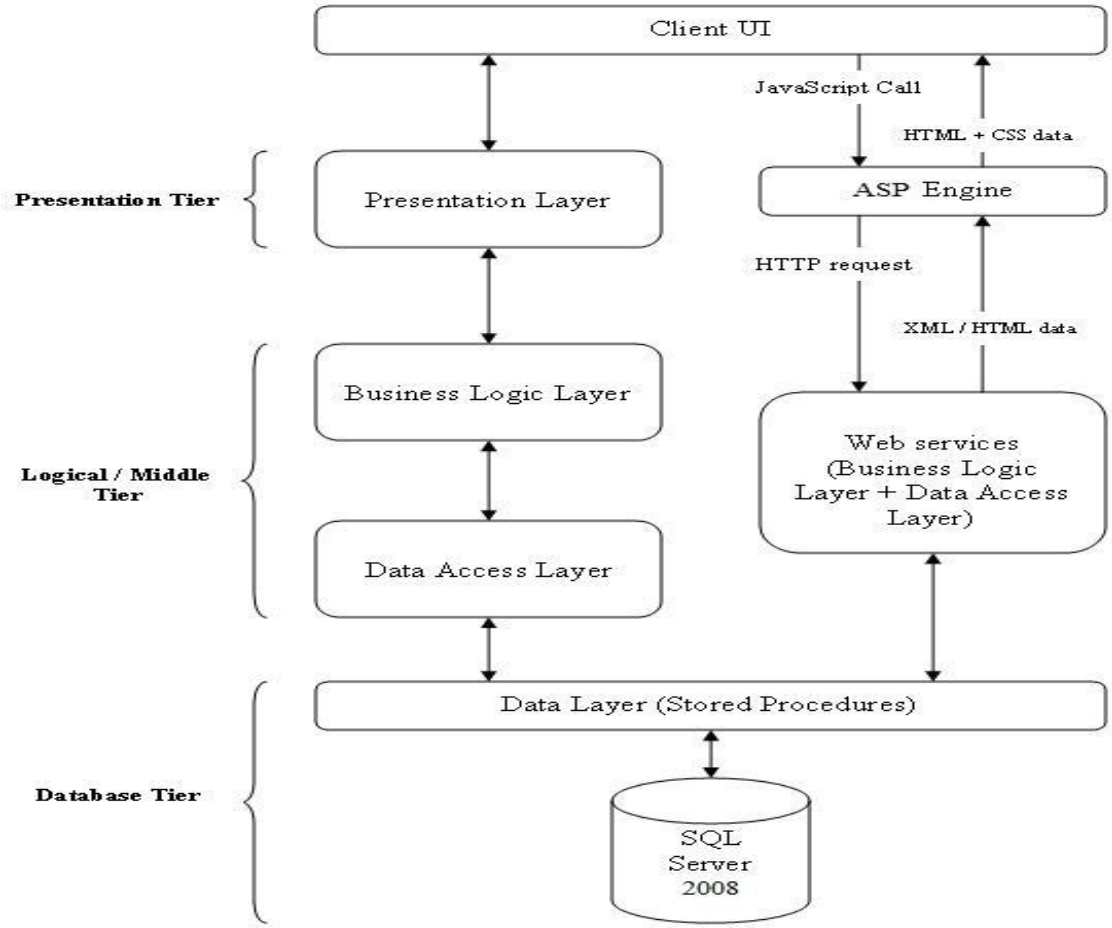
12. Database Design

12.1 Data Design

We will use database at the back end to display all the data on the front end. We will use Microsoft SQL Server 2008 to store and retrieve data using ASP.NET.

All informational data will be stored in the database and at the time of loading the webpage, it will be retrieved

12.2 Database Architecture



In the architecture shown above, Client UI is the client browser that is present in the client's machine. The Presentation layer consists of standard ASP.NET web forms, and documents, etc. This layer works with the results/output of the business logic layer and transforms the results into something usable and readable by the end user. Data Access Layer provides access to the database by executing a set of SQL statements or stored procedures. Database Tier consists of database and data layer which consists of stored procedures to manipulate and retrieve data from the database i.e. SQL Server

12.3 Database Numbers

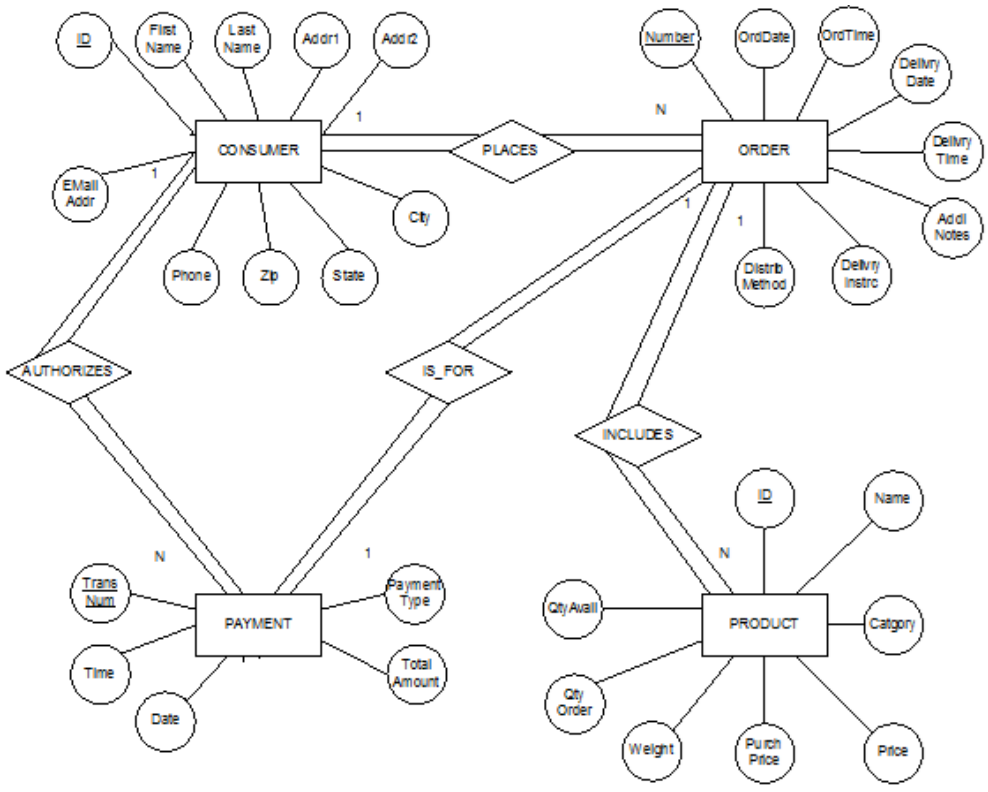
Size of the Database: 4MB
Number of Users: 5
Contains 4 tables: Consumer, Order, Payment, Product

Table Name	# of rows	# of columns	Column Names	Index Space	Data Space
Consumer	0	10	ID, FirstName, LastName, Addr1, Addr2, City, State, Zip, Phone, Email	0.008MB	0.008MB

Order	0	8	Number, OrdDate, OrdTime, DelivryDate, DelivryTime, AddlNotes, DelivryInstrc, DistrbMethod	0.008MB	0.008MB
Payment	0	5	TransNum, Payment Type, Time, Date, Total Amount	0.008MB	0.008MB
Product	62	8	ID, Name, Category, Price, PurchPrice, Weight, QtyOrder, QtyAvail	0.016MB	0.023MB
Totals	**62**	**31**		**0.040MB**	**0.047MB**

12.4 Entity Relationship Diagram

The following diagram represents the high level entities of the e-commerce grocery store application. This Entity Relationship Diagram (ERD) will most likely change as data entities are further explored throughout the design and perhaps into the initial development phase of this project.



13. Acknowledgement

I would like to thank my professor Dr. Yuke Wang for guiding me through the design of this project.