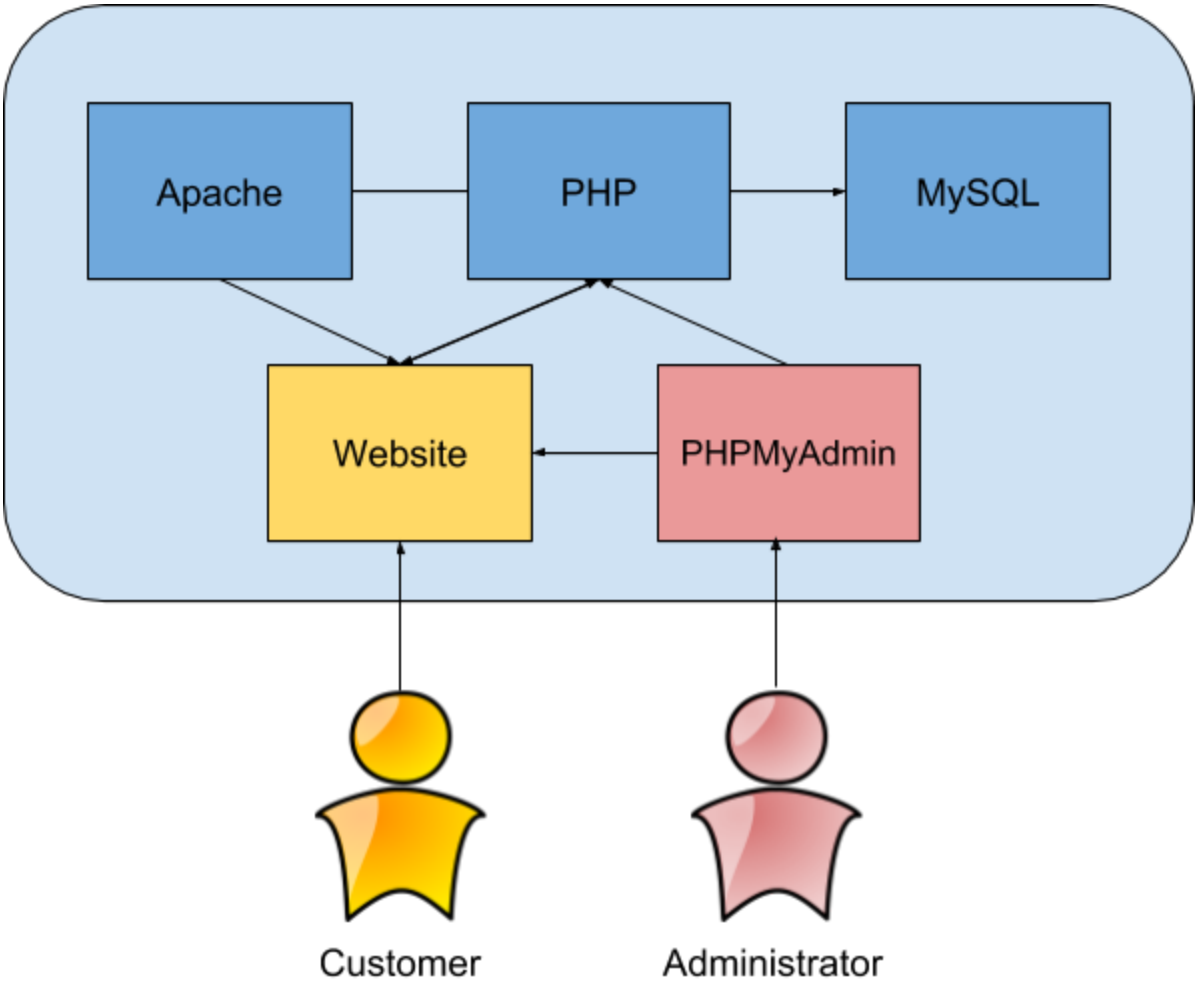
Team 3 - Electronic Retail Store
Part 2

Alan Lam
Johnny Lui
Danny Mou
Minh David Ngo

Table of Contents

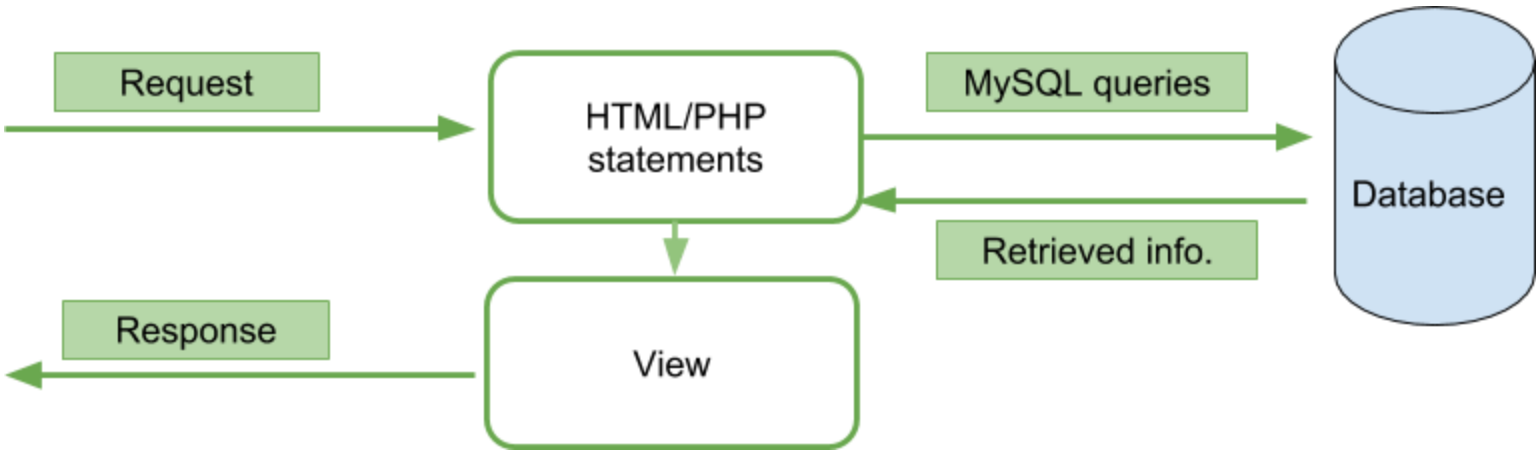
Architecture/High Level Design

For our project, we will be using MAMP, a local server environment, to host our domain and database servers such as Apache, MySQL, and PHP. MAMP only supports Windows and OS X. Apache and PHP will work as the front-end of the website, providing server sided code to clients. MySQL will be the database for all customer interactions such as inventory, transactions, user accounts, and etc. MAMP also includes PHPMyAdmin which helps handle MySQL queries and statements for administrators.



Flow

The basic flow for customers is a request/response sequence (see Diagram 1). First, the customer requests the page information. This send multiple queries from PHP code to the MySQL database to fetch products, information, images, and etc.



During purchases, there would be an extra step for validation of information. This will require extra queries from the user inputs. The view will depend on the page such as the search page will return multiple items or the account page will have forms to change passwords.

(For a more detailed architecture flow see Low-level Design p.5)

Hardware/Software/Tools

-A Laptop or Desktop with Windows (7 or higher) or Mac OS X installed.

Requirements:

-At least 2 GB RAM

-At least 2 GB HDD space

-MAMP 3.2.2

Includes:

-Apache 2.2.31

-MySQL 5.5.49

-PHP 5, PHP 7

Requirements:

-.NET Framework 4.0

-1 GB RAM

-Bootstrap for Web Development

-Google Maps API

-MySQL Workbench (optional)

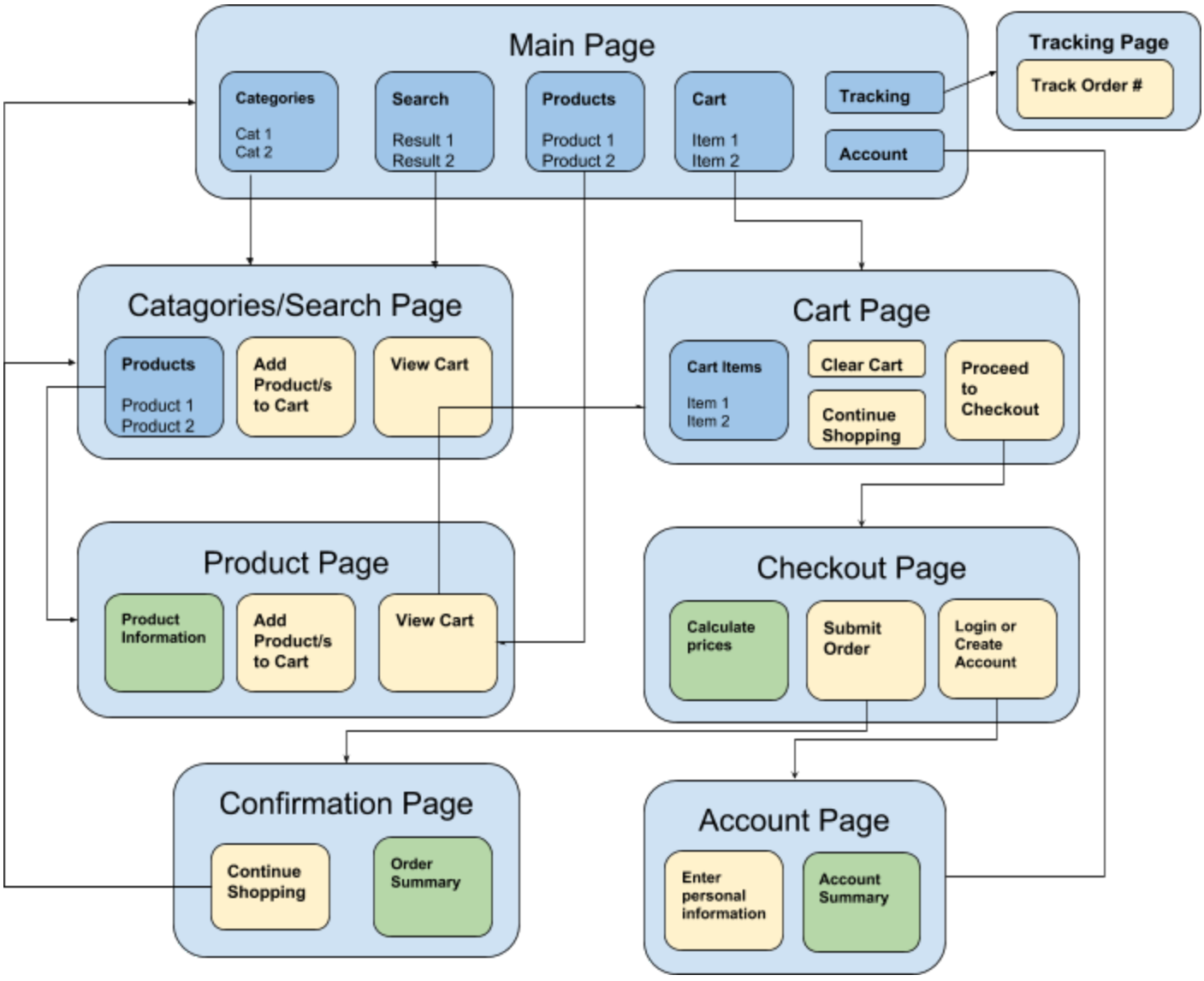
-Stable Internet Connection

-Updated Browser (Chrome, Firefox, IE, Opera, etc.)

-Mobile Device (iPhone, Android, Windows Phone)

-Various Text Editors

Low-level Design



Using the flow module (p.3), we can generate webpages from executing PHP code that retrieves database information using queries. The arrows of this diagram shows the tentative redirections of web pages, such as the cart page leads to the checkout page, to indicate the flow of a usual customer scenario.

PHP

PHP provides the server-sided database information to a web document without exposing the code to the clients (HTML). To do so, we need to include the following codes to connect to our MySQL database and to manipulate data.

Connect to DB:

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";

// Create connection
$conn = new mysqli($servername, $username, $password);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Queries (Insert, Select, Update, Delete):

```
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$sql = "UPDATE MyGuests SET lastname='Doe' WHERE id=2";
$sql = "DELETE FROM MyGuests WHERE id=3";
$conn->query($sql) //Evaluates to TRUE or FALSE
```

Create persistent session:

```
<?php
session_start();
?>
$_SESSION[“”]; // Variables set on previous pages.
```

HTML/CSS/JS

The HTML returns the view of the store. Some Javascript/CSS is needed to make the site more user-friendly such as navigation bars, images, categories, and etc. Below are some of the functions and pages.

Pages Overview

-Main
- The first page the customer sees. Contains featured items and products with links to most other pages.

-Categories/Search
- Returns a list of products either by category or search. Able to add items to cart or look at a specific product for more details.

-Product
- Contains the product information, stock, price, images, and an add to cart button.

-Login/Register
- Shows text fields for users to enter personal information to login or to create a new account.

-Account Settings
- Shows a list of account functions that can be changed such as changing a password or getting a purchase history.

-Tracking
- Tracks a specific transaction with the Google Maps API and returns the details of the shipping.

-Shopping Cart
- Keeps track of products added to the cart and allows the customer to purchase all at once.

-Purchase
- User enters personal information to make an order for the items requested in the cart.

-Order Review/Order Submitted
- A final review of the invoice and submission of the order for confirmation.

Some Functions:

Registering:

Show register form
If submit button pressed
    Check if all required information filled out
        If not, show error and show what still needs to be filled out
    Check if username/email has already been used
        If used, show error
    Else
        Generate hashes for password
        Insert account information to database
        Redirect back to main page

Adding to Cart:
If logged in
    Add to cart
    Run insert query to table
    Show message that adding was successful
Else
    Redirect to login page

Sample Google Maps Routing Function:

```
function calcRoute() {
  // First, clear out any existing markerArray
  // from previous calculations.
  for (i = 0; i < markerArray.length; i++) {
    markerArray[i].setMap(null);
  }
  // Retrieve the start and end locations and create
  // a DirectionsRequest using WALKING directions.
  var start = document.getElementById('start').value;
  var end = document.getElementById('end').value;
  var request = {
      origin: start,
      destination: end,
      travelMode: 'WALKING'
  };
```

```
  // Route the directions and pass the response to a
  // function to create markers for each step.
  directionsService.route(request, function(response, status) {
    if (status == "OK") {
      var warnings = document.getElementById("warnings_panel");
      warnings.innerHTML = "" + response.routes[0].warnings + "";
      directionsDisplay.setDirections(response);
      showSteps(response);
    }
  });
}
```

Purchase:

Show debit/credit card form
If Submit button is pressed
    Check if user entered valid card information
    If valid
        Check if user wanted to save credit card information onto their account
            If true
                Add credit card onto their account in database
        Store user order information to database
        Remove purchased items from inventory (quantity)
        Direct user to order summary page
    Else
        Prompt user to recheck their information and try again
If Back button is pressed
    Return to shopping cart page

Search:

Show search text field
If user begins to type in text field
    Show autocomplete suggestion list as they type (i.e. co shows computer/copier)
    Suggestion list will taken from database
    If user pressed enter or search button (magnifying glass)
        If similar items are found
            Redirect user to a page that lists related items with the top items
            Being the closest to user query
        Else

Prompt user that no items can be found or query can be invalid

Tracking Package

User types tracking # in box
User presses search button
Search database for tracking #
If tracking # incorrect or does not exist
    Display incorrect tracking # message
Else
    Display FROM address, package status, delivery time?, truck driver location?, destination, google map

Sample Example Image Slider

```
var slideIndex = 1;
showDivs(slideIndex);

function plusDivs(n) {
        showDivs(slideIndex += n);
}

function showDivs(n) {
   var i;
   var x = document.getElementsByClassName("mySlides");
   if (n > x.length) {slideIndex = 1}
   if (n < 1) {slideIndex = x.length} ;
   for (i = 0; i < x.length; i++) {
      x[i].style.display = "none";
        }
  x[slideIndex-1].style.display = "block";
}
```

*Sources copied from www.w3Schools.com

MySQL

Table Overview

Address - Contains customer addresses
Account - Account information
Cart - Current items in cart
Inventory - Items available in inventory
Order - Placed Orders/Transactions
Order_Items - Items placed in the order
Warehouse - Warehouse addresses

Schemas

Address

- id - Primary Key int(11) NOT NULL
- accountId - Foreign Key int(11) NOT NULL : Account ID that the address is associated to
- name - varchar(255) NOT NULL
- address - varchar(255) NOT NULL : address of customer
- city - varchar(20) NOT NULL
- state - varchar(20) NOT NULL
- zip - int(5) NOT NULL

Account

- id - Primary Key int(11) NOT NULL
- email - varchar(255) NOT NULL
- username - varchar(20) NOT NULL
- password - varchar(255) NOT NULL
- salt - varchar(255) NOT NULL
- first_name - varchar(255) NOT NULL
- last_name - varchar(255) NOT NULL
- creation_date - timestamp NOT NULL

Cart

accountId - Primary/Foreign Key 1 int(11) NOT NULL
itemId - Primary/Foreign Key 2 int(11) NOT NULL
quantity - int(3) NOT NULL
date_added - timestamp NOT NULL

Inventory

id - Primary Key int(11) NOT NULL
name - varchar(255) NOT NULL
img_src - varchar(255) NOT NULL : link to image
price - decimal(10, 2) NOT NULL
quantity_remaining - int(11) NOT NULL

Order

id - Primary Key int(11) NOT NULL
accountId - int(11) NOT NULL
addressId - int(11) NOT NULL : ID of address to deliver to
warehouseId - int(11) NOT NULL : ID of which warehouses to deliver from
total - decimal(10,2) NOT NULL : Order total
date - timestamp NOT NULL : Date placed

Order_Items

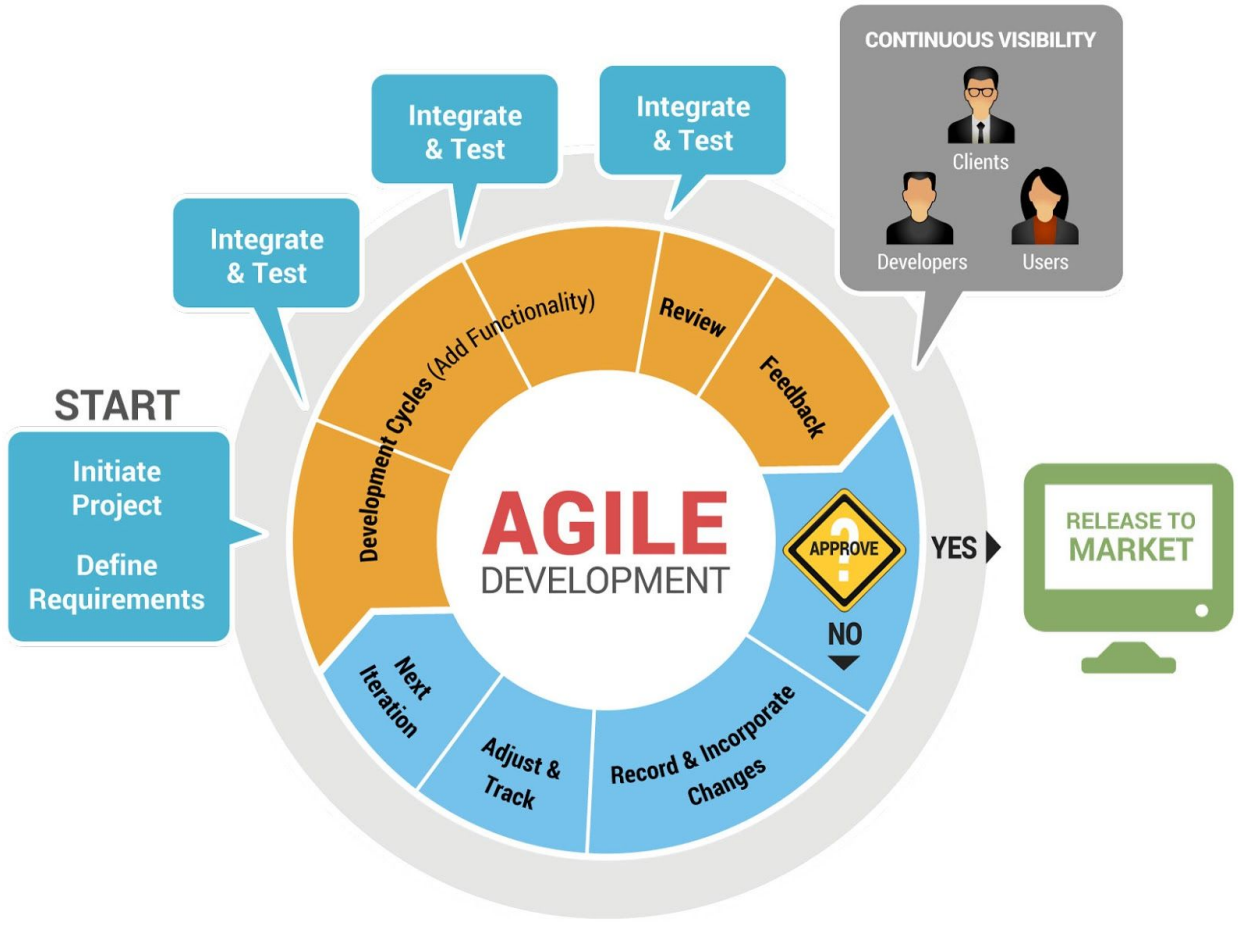
accountId - Primary/Foreign Key 1 int(11) NOT NULL
itemId - Primary/Foreign Key 2 int(11) NOT NULL
quantity - int(3) NOT NULL
price - decimal(10,2) NOT NULL : price paid

Warehouse

Id - Primary Key int(11) NOT NULL
address - varchar(255) NOT NULL
city - varchar(20) NOT NULL
state - varchar(20) NOT NULL
zip - int(5) NOT NULL

High-level Test Plan

We will have weekly sprints to tackle items on the product backlog and have deliverables prior to the next week's sprint. Daily scrum meetings will be held to get a better understanding of where each member is in their individual contributions. Different tests will be done during each week to check the quality of work, fix bugs, and judge whether or not we are on schedule. The test plan is expanded in the categories below.



Strategy

The plan is to have each member of the team complete one or more items on the product backlog each sprint so that they can be tested before moving on with the project. By testing pieces of the project each week, we will be able to have a better understanding of the project as a group and how much time we have to spend on other items in the backlog. We will be using the following types of testing throughout the project:

- Unit Testing
    - Each member makes sure their individual contributions are bug free
- Integration Testing
    - Everyone's code will be pieced together to test the program as a whole and any problems that come up will be solved during this phase
- Functional Testing
    - This is where we test the functionality of the product that we will have developed
- System Testing
    - Usability, reliability, and various performance tests will be done here to ensure that the product meets requirements
- Alpha Testing
    - This form of testing will be done by the team and other individuals to ensure that everything runs fine
- Beta Testing
    - This form of testing will come after fixing bugs that are found from alpha testing, if any. The bugs will be fixed and the product will be prepared for black-box testing.
- Black-box Testing
    - Different groups will evaluate our product and we will address any issues that are found.

Goals/Expectations

Our overall goal is to create a user-friendly full functional website. The website will allow users to browse through our amazing inventory and purchase the items that they desire. We fully expect to follow the Agile development process to improve our communication and workflow. Here are our goals and expectations.

Store/Website:

- Perfectly Placed Stores
    - We will be setting up 10 different stores for maximum profit.
    - We have decided on opening 2 stores per county.
- Fully Functional Website
    - Customers will be able to purchase electronic devices.
    - The website will include an account registration function along with login.
    - It will also include a virtual shopping cart, and a simulated transaction.
    - The inventory, customer, transaction information stored in a database.
    - Ease of access and navigability.
    - Be able to track delivery truck carrying the customer's package via Google Maps

Team Goals:

We expect to learn all of the necessary tools to complete this project.
As a team, we plan to create a detailed and clear product backlog.
15 minute scrum meetings and team communication will occur daily.
We plan to finish the project by Sprint 7 or 8.
Each member is expected to carry out their load of work.
Problems will occur and should be resolved as a team.

Metric/Tools/Methodology

We will test the website using various platforms such as Chrome, Mozilla Firefox, Internet Explorer, Safari, and other web browsers.

For site usability and navigability, we will poll our peers on different platforms and operating systems to ensure that the website is easy to use on all devices (Windows, Mac OS X, Linux, iOS, Android, Windows Phone).