Why Not Zoidberg?

Software Requirements Specification

Version 1.4
11/29/2012

Prepared for
Ms. Lisa Matthews
math1@umbc.edu

Team Name: Planet Express
Team Facilitator - Chris Raborg
Requirements Leader - David Guldan
Design Leader - Kevin Yu
Implementation and Testing Leader - Zach Hisley
Delivery Leader - Matt Sperbeck

Why Not Zoidberg?
System Requirements Specification

Table of Contents

Document Versioning Control

Version Number	Date	Changes from Previous Version
1.0	10/3/2012	N/A
1.1	10/9/2012	Edits based on feedback
1.2	10/12/2012	Functional Requirements, Top Level Use Case
1.3	10/15/2012	Sub Use Case Diagrams
1.4	11/29/2012	Post-grade updates

1. **Introduction**

1.1 Purpose of This Document

This Software Requirements Specification provides a complete description of all the functions and specifications of the Online Shopping System. The expected audience of this document is Ms. Lisa Matthews because she is hiring us to make an online shopping website for her bookstore. The developers (Team Planet Express) are also an expected audience because we will be the ones designing the system.

1.2 References

These are the references that we used to guide us in creating the use cases and corresponding diagrams:

- Cockburn, A., Basic Use Case Template. 1998, Humans and Technology.

- Sommerville, Ian and Pete Sawyer, Requirements Engineering: A Good Practice Guide. 1997: Wiley

1.3 Purpose of the Product

The situation that triggered the need for Ms. Matthews' request was to streamline her order process so that many people could have access to purchase her books. Our group shall collaborate to create a shopping system that aligns with and fulfills Ms. Matthews needs and requirements for it. Ms. Matthews noted that she wanted the ability to view the top sold books within the system. Ms. Matthews left it to the discretion to the team, Planet Express, to come up with more optional functionality for the system.

1.4 Product Scope

This use case illustrates the functionalities of the system. The scope of the system includes being able to search for items, add items to a shopping cart, add funds/money to a personal balance, and purchase/checkout items. The two main stakeholders of the software include a customer and a systems administrator. A customer has access to all of the prior noted functionalities, while a systems administrator may only edit the inventory (add/remove items from the website).

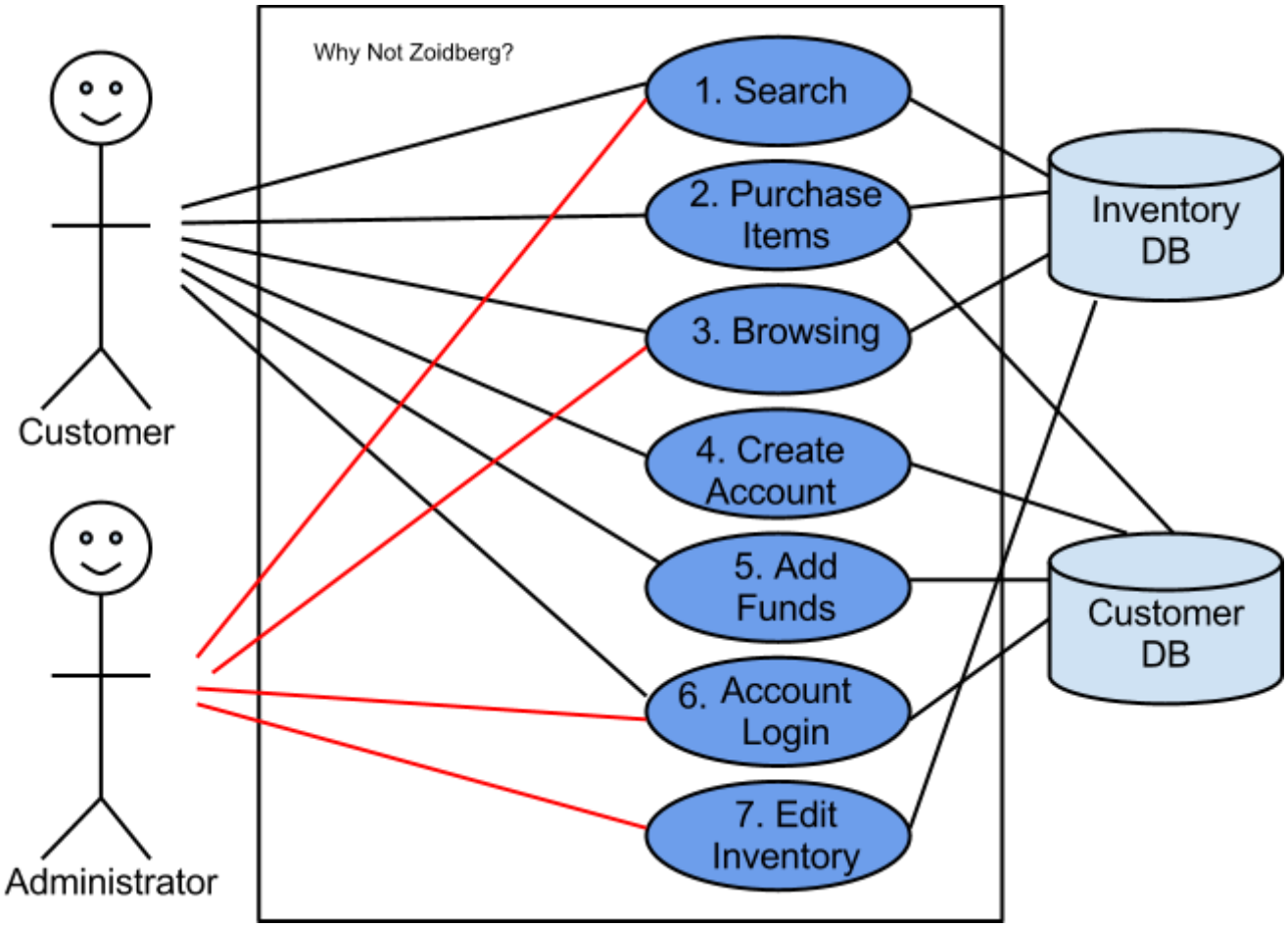


Figure 1.1: Top Level Use Case Diagram describes the overall outline of the system. There are two primary actors, the Customer and Administrator. Each primary actor has access to particular functionality in the system (indicated by lines between an actor and a functional requirements.

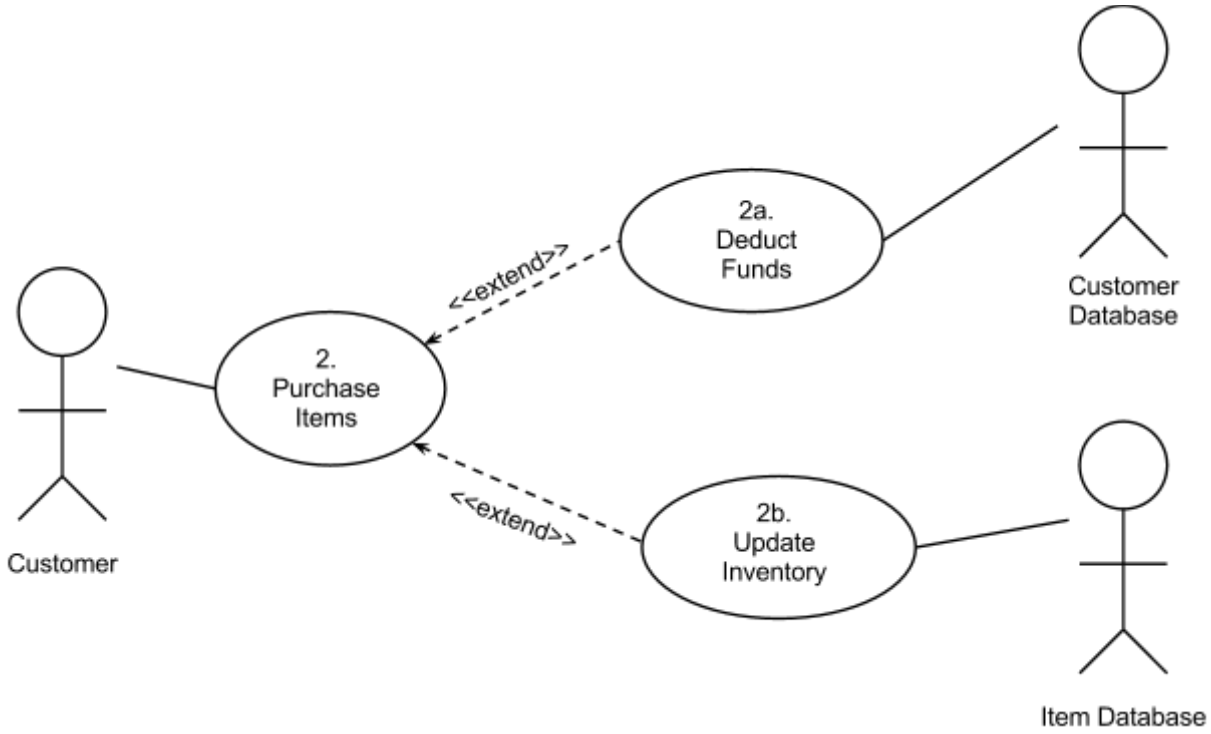
2. **Functional Requirements**

<table>
<tr><td>Number</td><td>1</td></tr>
<tr><td>Name</td><td>Search</td></tr>
<tr><td>Summary</td><td>The User can choose a selection criteria and provide a keyword to find an item which matches the keyword.</td></tr>
<tr><td>Priority</td><td>5</td></tr>
<tr><td>Preconditions</td><td>User chooses a selection criteria.
User enters a keyword into the system.</td></tr>
<tr><td>Postconditions</td><td>System displays a list of matching items</td></tr>
<tr><td>Primary Actor(s)</td><td>● Customers
● Administrator</td></tr>
<tr><td>Secondary Actor(s)</td><td>● None</td></tr>
<tr><td>Trigger</td><td>User enters the keyword to initiate the search.</td></tr>
<tr><td>Main Scenario</td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>User selects a selection criteria to narrow their search.</td></tr>
<tr><td>2.</td><td>User enters one or more keywords.</td></tr>
<tr><td>3.</td><td>User initiates the search.</td></tr>
<tr><td>4.</td><td>System displays search results.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>2a.</td><td>If the user does not type anything into the search box, the system shall display all of the books in the system.</td></tr>
</table>
</td></tr>
<tr><td>Open Issues</td><td>N/A</td></tr>
</table>

<table>
<tr><td>Number</td><td>2</td></tr>
<tr><td>Name</td><td>Purchase Item(s)</td></tr>
<tr><td>Summary</td><td>The system allows the user to buy an item that they want.</td></tr>
<tr><td>Priority</td><td>5</td></tr>
<tr><td>Preconditions</td><td>User selects one or more items to purchase.
User has a balance.</td></tr>
<tr><td>Postconditions</td><td>The total cost of the item(s) is deducted from the user’s balance.
The system shall update their inventory.</td></tr>
<tr><td>Primary Actor(s)</td><td>● Customers</td></tr>
<tr><td>Secondary Actor(s)</td><td>● None</td></tr>
<tr><td>Trigger</td><td>User has selected to checkout.</td></tr>
<tr><td>Main Scenario</td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>The User selects an item(s) to purchase.</td></tr>
<tr><td>2.</td><td>The System shall check the Inventory Database to see if there is at least 1 copy of the item available. Cancel transaction if there are no items available.</td></tr>
<tr><td>3.</td><td>The User selects to checkout.</td></tr>
<tr><td>4.</td><td>The System checks if the User has an account in the Customer Database.</td></tr>
<tr><td>5.</td><td>The System checks if the User has enough money in their balance to purchase the item(s).</td></tr>
<tr><td>6.</td><td>The System subtracts the total from the User’s balance.</td></tr>
<tr><td>7.</td><td>The System updates the Inventory Database.</td></tr>
<tr><td>8.</td><td>The System shall give the User a receipt of the transaction.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td>Extensions</td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>2a.</td><td>If the User requests more copies of an item than are currently available, the System shall notify the User that not enough copies are available.
They may purchase enough copies up to the maximum number available, or cancel the transaction.</td></tr>
<tr><td>3a.</td><td>If the User does not have an account, they are prompted to create an account, or they may cancel the transaction.</td></tr>
<tr><td>4a.</td><td>If the User does not have enough funds in their balance to purchase the item(s), they are prompted to add more funds to their balance, or they may cancel the transaction.</td></tr>
<tr><td>7a.</td><td>The Inventory Database updates the total number of copies purchased.</td></tr>
</table>
</td></tr>
<tr><td>Open Issues</td><td>N/A</td></tr>
</table>

Diagram 1.2:

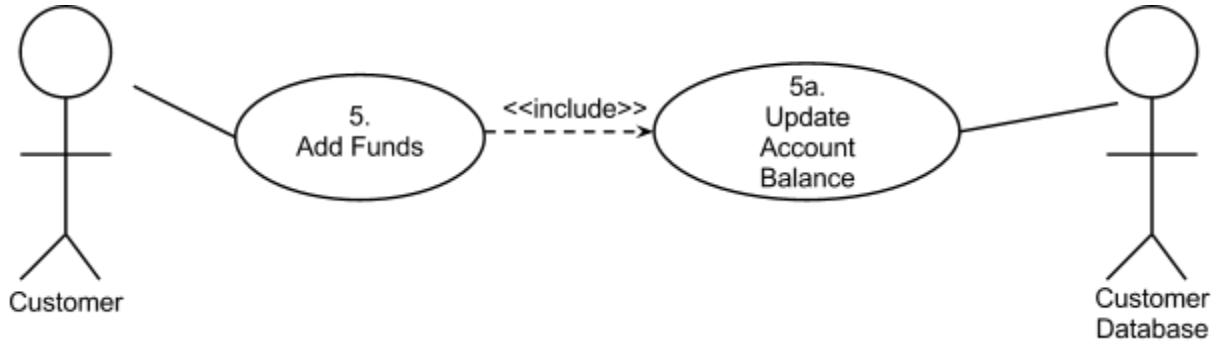


<table>
<tr><td>Number</td><td>3</td></tr>
<tr><td>Name</td><td>Browse</td></tr>
<tr><td>Summary</td><td>The User shall be able to look at all relevant information about a particular item.</td></tr>
<tr><td>Priority</td><td>3</td></tr>
<tr><td>Preconditions</td><td>The User has selected an item to browse.</td></tr>
<tr><td>Postconditions</td><td>The information is displayed for the User.</td></tr>
<tr><td>Primary Actor(s)</td><td>Customers
Administrators</td></tr>
<tr><td>Secondary Actor(s)</td><td>None</td></tr>
<tr><td>Trigger</td><td>The User clicks on a link for the item.</td></tr>
<tr><td>Main Scenario</td><td><table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>The User clicks on a link for the item.</td></tr>
<tr><td>2.</td><td>The System shall display all stored information about the item.</td></tr>
</table></td></tr>
<tr><td>Extensions</td><td>N/A</td></tr>
<tr><td>Open Issues</td><td>N/A</td></tr>
</table>

<table>
<tr><td>Number</td><td>4</td></tr>
<tr><td>Name</td><td>Create Account</td></tr>
<tr><td>Summary</td><td>Allows the User to create an account on the System, which allows them to make purchases.</td></tr>
<tr><td>Priority</td><td>4</td></tr>
<tr><td>Preconditions</td><td>The User does not have an account already.</td></tr>
<tr><td>Postconditions</td><td>The User is given an account.</td></tr>
<tr><td>Primary Actor(s)</td><td>• Customers</td></tr>
<tr><td>Secondary Actor(s)</td><td>• None</td></tr>
<tr><td>Trigger</td><td>The Customer enters their information and selects to create their account.</td></tr>
<tr><td>Main Scenario</td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>The User selects to create an account.</td></tr>
<tr><td>2.</td><td>The User inputs their email address and a password for their new account.</td></tr>
<tr><td>3.</td><td>The System checks to see if the User already has an account in the system.</td></tr>
<tr><td>4.</td><td>If the User does not already have an account, make one. Notify the User that his/her account has been created!</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>3a.</td><td>If the User has an account already, prompt them to login instead.</td></tr>
</table>
</td></tr>
<tr><td>Open Issues</td><td>N/A</td></tr>
</table>

<table>
<tr><td>Number</td><td>5</td></tr>
<tr><td>Name</td><td>Add Funds</td></tr>
<tr><td>Summary</td><td>Allows the User to add funds to his/her account.</td></tr>
<tr><td>Priority</td><td>5</td></tr>
<tr><td>Preconditions</td><td>The User has logged into their account.</td></tr>
<tr><td>Postconditions</td><td>The User’s balance has been updated.</td></tr>
<tr><td>Primary Actor(s)</td><td>• Customers</td></tr>
<tr><td>Secondary Actor(s)</td><td>• None</td></tr>
<tr><td>Trigger</td><td>The Customer enters the amount of funds to add, presses a button.</td></tr>
<tr><td>Main Scenario</td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>The User selects that they wish to add funds to their balance.</td></tr>
<tr><td>2.</td><td>The User enters the amount of funds to add and presses a button to confirm.</td></tr>
<tr><td>3.</td><td>The System shall update the User’s balance.</td></tr>
<tr><td>4.</td><td>The System shall notify the User that the balance has been updated and provide the new balance.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>2a.</td><td>If the User provides a negative amount, reprompt.</td></tr>
</table>
</td></tr>
<tr><td>Open Issues</td><td>N/A</td></tr>
</table>

Diagram 1.3:

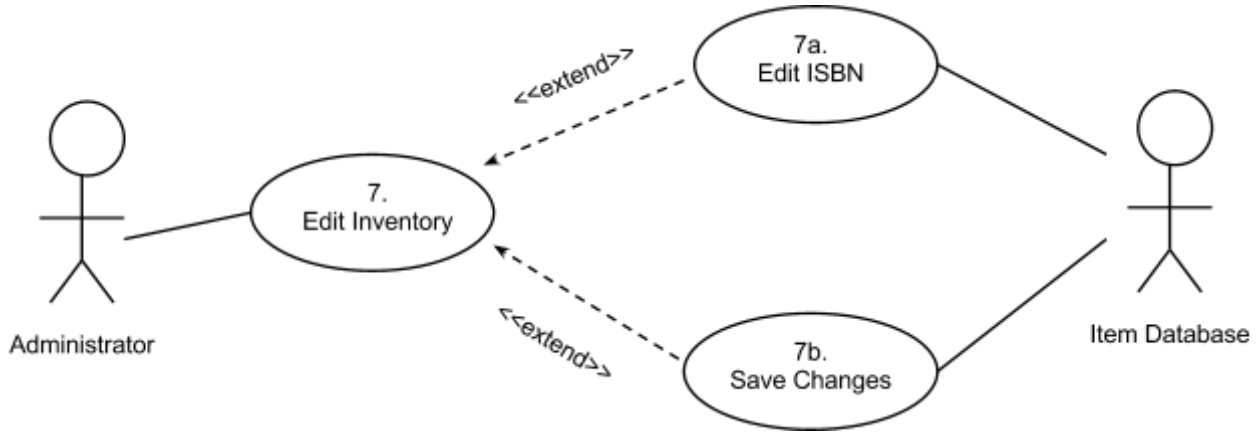

Customer
5.
Add Funds
<<include>>
5a.
Update
Account
Balance
Customer
Database

<table>
<tr><td>Number</td><td>6</td></tr>
<tr><td>Name</td><td>Account Login</td></tr>
<tr><td>Summary</td><td>Allows the User to login to their account.</td></tr>
<tr><td>Priority</td><td>5</td></tr>
<tr><td>Preconditions</td><td>The User has an account in the System.</td></tr>
<tr><td>Postconditions</td><td>The User is logged into their account.</td></tr>
<tr><td>Primary Actor(s)</td><td>Customers
Administrator</td></tr>
<tr><td>Secondary Actor(s)</td><td>None</td></tr>
<tr><td>Trigger</td><td>The Customer enters their information and presses a button.</td></tr>
<tr><td>Main Scenario</td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>The User selects to login to their account.</td></tr>
<tr><td>2.</td><td>The User inputs their email address and password.</td></tr>
<tr><td>3.</td><td>The System checks if the User’s email address is contained in the Customer Database. If it is not, prompt the user to re-type their email address.</td></tr>
<tr><td>4.</td><td>The System checks to see if the password provided by the User matches the password in the Customer Database.</td></tr>
<tr><td>5.</td><td>The User is logged on.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>3a.</td><td>If the email address does not match, the User is prompted to re-type their email address.</td></tr>
<tr><td>4a.</td><td>If the password does not match, the User is prompted to re-type their password.</td></tr>
</table>
</td></tr>
<tr><td>Open Issues</td><td>N/A</td></tr>
</table>

<table>
<tr><td>Number</td><td>7</td></tr>
<tr><td>Name</td><td>Edit Inventory</td></tr>
<tr><td>Summary</td><td>Allows the Administrator to edit the Inventory Database.</td></tr>
<tr><td>Priority</td><td>5</td></tr>
<tr><td>Preconditions</td><td>The Administrator has logged into the System.</td></tr>
<tr><td>Postconditions</td><td>The Inventory Database is modified.</td></tr>
<tr><td>Primary Actor(s)</td><td>● Administrator</td></tr>
<tr><td>Secondary Actor(s)</td><td>● Inventory Database</td></tr>
<tr><td>Trigger</td><td>The Administrator provides an ISBN to the System.</td></tr>
<tr><td>Main Scenario</td><td>
<table>
<tr><th>Step</th><th>Action</th></tr>
<tr><td>1.</td><td>The Administrator selects to edit the Inventory Database.</td></tr>
<tr><td>2.</td><td>The Administrator enters an ISBN.</td></tr>
<tr><td>3.</td><td>The System checks to see if the ISBN is already contained in the Inventory Database.</td></tr>
<tr><td>4.</td><td>The System shall notify the Administrator of the changes that have occurred.</td></tr>
</table>
</td></tr>
<tr><td>Extensions</td><td>
<table>
<tr><th>Step</th><th>Branching Action</th></tr>
<tr><td>3a.</td><td>If the item is already contained, prompt the Administrator to either edit the item information OR to delete the item from the database.</td></tr>
<tr><td>3a1.</td><td>If the Administrator wishes to edit the item, the System shall display all of the item information that may be changed (does not include the ISBN). The Administrator confirms the changes by pressing a button.</td></tr>
<tr><td>3a2.</td><td>If the Administrator wishes to delete the item, the System shall remove the item from the Inventory Database.</td></tr>
</table>
</td></tr>
</table>

	3b. | If the ISBN is NOT contained, then the administrator is prompted to enter all other information available for the item. The Administrator confirms the changes by pressing a button.
Open Issues	N/A

Diagram 1.7:



3. **Non-Functional Requirements**

3.1 Customer Constraints

The system shall be a web application located on one of the team member's "userpages" directory on the GL server. We're using HTML and CSS to create the backbone of the system, PHP to create the backbone of the system, JavaScript to supplement the PHP code, and SQL to run the database queries.

The database that we are using is contained on the GL server at UMBC. The database will contain tables to hold item information (ISBN, title, author, price, etc), usernames, and user account information.

Our customer also asked for specific functionality to track and display the top purchased items on the website.

3.2 External Interfaces

Our system does not contain any external interfaces.

3.3 Other

N/A

4. **Deliverables**

The figure below details the documents that will need to be delivered to the customer, Ms. Matthews, and the dates to be delivered. We will be presenting her with hard, paper copies of these documents so that she may inspect and review our document and sign off if she finds our work satisfactory.

The final copy of our source code will be saved on a CD to be presented to our customer on the date of our presentation. Final copies of all of our documents will also be saved on the CD in addition to the source code.

Item:	**Date to be delivered:**
System Requirement Specification	October 16, 2012
System Design Document	October 30, 2012
User Interface Design Document	October 30, 2012
Code Inspection Report	November 30, 2012
Testing Report	December 5, 2012
Administrator Manual	December 5, 2012
Source Code, Final CD	December 5, 2012

5. **Open Issues**

There are no open issues at this time.

Appendix A – Agreement Between Customer and Contractor

We, the undersigned, agree as developers that we have fully understood all requirements which have been requested by our customer to the best of our ability. Our customer acknowledges that these requirements have been satisfactory and have been carried out to their full extent. If anyone finds any issue with the document, it will be presented to the rest of the team and our customer, either in person or via email. It will then be discussed to come up with the best solution.

Team Signatures:

_____

Christopher Raborg, Team Facilitator

_____

David Guldan, Requirements Leader

_____

Kevin Yu, Design Leader

_____

Zachary Hisley, Implementation Leader

_____

Matt Sperbeck, Delivery Leader

Customer Signature:

_____

Ms. Lisa Matthews

Appendix B – Team Review Sign-off

We agree as a team that we have documented the requirements to the fullest possible extent and that we have all read the entire document and agree with the contents. If there is an issue within the team about the document, the team shall resolve the issue at the next team meeting.

Team Signatures:

_____

Christopher Raborg, Team Facilitator

_____

David Guldan, Requirements Leader

_____

Kevin Yu, Design Leader

_____

Zachary Hisley, Implementation Leader

_____

Matt Sperbeck, Delivery Leader