Why Not Zoidberg?

System Design Document

Version 1.5
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 Design Document

Table of Contents

Document Versioning Control

Version Number	Date	Changes from Previous Version
1.0	10/17/12	Introduction, Purpose, Architectural Design, Persistent Data Design
1.1	10/26/2012	Fixed System Architecture,
1.2	10/30/12	Class Diagrams, Database Schemas, Requirements Matrix, Appendices
1.3	10/30/12	Persistent Data Design
1.4	11/7/12	Technology Diagram - we realized it would be better to switch to PHP/HTML for the system instead of Java.
1.5	11/29/12	Updated Class Diagrams, Technology Diagram, Persistent Data to match current system design.

1. Introduction

1.1 Purpose of this Document

This System Design document outlines the logical structure and architecture of the online shopping system for the developers' use and for our customer, Ms. Lisa Matthews. This document provides architecture diagrams that describe the logical flow of information and organization of the system so that the developers have a clear and concise backbone from which to construct a framework for the system. A list of classes, methods, database schemas and how each one fulfills each one of the functional requirements outlined in the System Requirements Specification, is also provided for clarity.

1.2 References

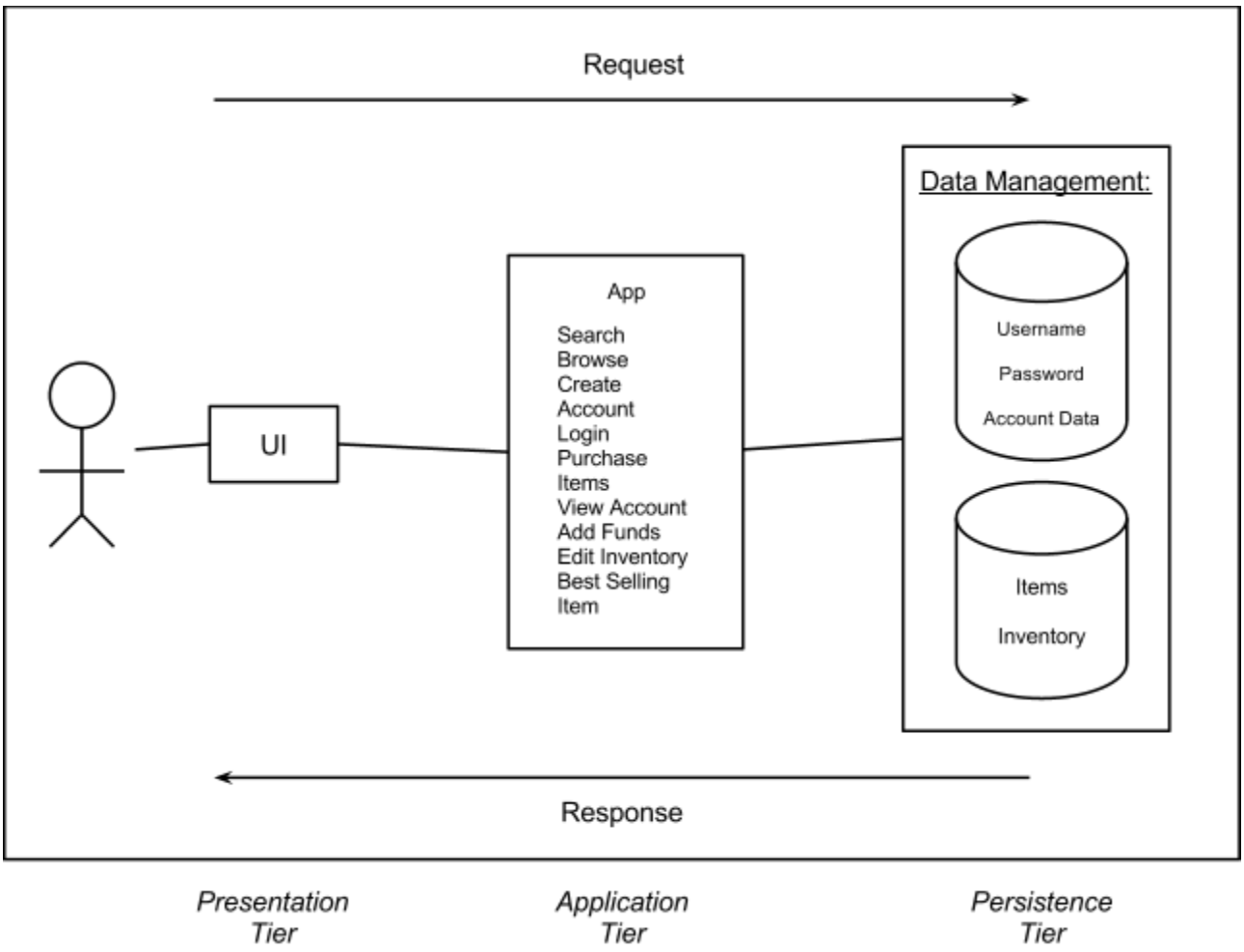
These are the references that helped us create the logical architecture and class diagrams:

- Roache, Eddie. Software Design - Requirements to Code. CMSC 345 Speaker Presentation (for architecture diagrams).

- www.uml-diagrams.org/class-diagrams.html

2. System Architecture

Diagram 2.1:



The UI is responsible for all output that the user has to view and interact with. It will possess corresponding views that allow the user to navigate the system and provide the appropriate services for the user. More specific information will follow in the UI Document.

The system will require the user to login in order to be assigned to one of two roles in the system: a customer and an administrator. A user who is not logged in will still be able to browse the website, but will not have access to the customer and administrator abilities. The administrator may login and access the admin menu, where he/she can edit the inventory and view the best selling items. The administrator account will be provided by the developers. Any other user who creates an account will automatically be assigned the role of customer, and they will have access to a shopping cart and a balance so that they may purchase those items in the shopping cart.

Specific Application Services:

Browse - All of the item information that is contained in the inventory database is returned to the user interface and displayed appropriately. The option to purchase the item that has been selected and displayed will be provided to the user (unless the user is an administrator).

Search - The search criteria will include: ISBN, Title, Author, Genre. The user will be able to narrow their search by selecting one of the attributes from the drop down box to search by. The default criteria will be title. The string that is input by the user will be matched with the appropriate attributes within the inventory database.

Login - The user will be prompt for a username and password which will both be searched for in the user database. If they are found, the user will be logged in with the appropriate account corresponding to the username and password they have input. If there is no account in the user database, the user will be prompted to create an account or return to the home page.

Create Account - The user will provide a username or password to create an account with. If the user inputs a username that is already present in the user database, they will be notified that the account username that they have chosen has already been selected and will be prompt to enter another or to return to the home page.

Purchase Items: The user will be given the option to purchase the item(s) that they wish to buy. If there is stock available in the inventory, then the purchase will be verified, the inventory number for that item will be updated, and the user's funds will be deducted. If the user funds are insufficient, the user will have to add adequate funds to complete the purchase, or edit their shopping cart until their balance is sufficient for their cart.

View Account: The user will be able to view and edit their balance. Their account information such as their name, address and funds will be available here. A link will be present that when accessed will take the user to a page where they can add funds to their balance.

Add Funds: The user will be given the option to add funds to their account after signing up/logging in. The amount added and their total balance will be verified and updated afterwards. Users will not be able to remove funds once they've been added.

Edit Inventory: The admin will be able to edit the inventory by searching the inventory database by ISBN number. A search field will be displayed for the admin to input the ISBN number for the book they wish to add/edit. The number will be checked for validity based on the character(s) input as well as the total number of characters. If the ISBN is found, the admin will be able to edit the book correlating to the ISBN integer entered. If it is not found, the admin will be taken to an insert book page where they will be able to add the information for the book they are appending to the inventory database.

View Bestselling Items: The administrator will be able to view the top selling items from the store. The administrator will be able to enter how many top selling items they want to view. The database will then be queried and will return the number of items specified that have the most copies sold, and will display them to the administrator.

Technology Diagram:

Diagram 2.2:

HTML will be used to create forms for user input, and make the interface for the user to interact with through a web browser. CSS will provide the style to the HTML, but will not affect the functionality at all. PHP will be used for the backend of the HTML, working with SQL to query the database and manipulate data to produce content for the user. PHP will also take care of holding local userdata for the system within a PHP session (e.g. username, shopping cart). Javascript will be used for mostly data validation as well as page redirection. SQL will be used within the PHP code to query the database and access data from the tables. phpMyAdmin will be used to create tables, fields, and store customer and inventory data.

2.2 Decomposition Description

Classes/Objects:

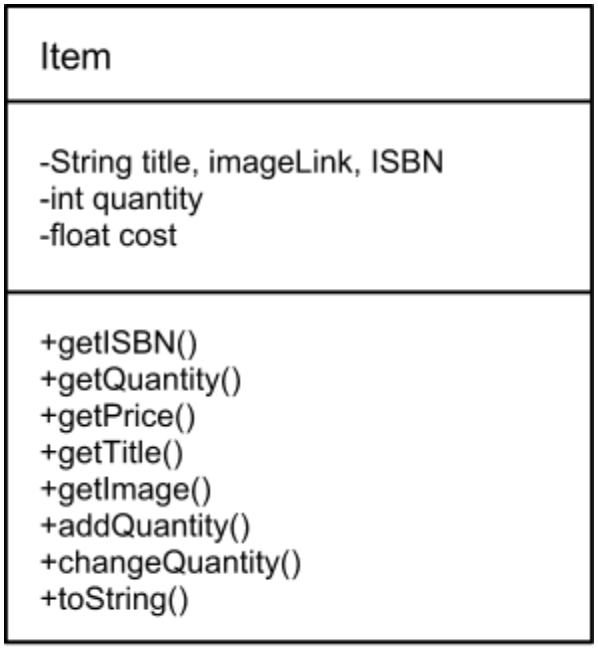


Figure 2.3a

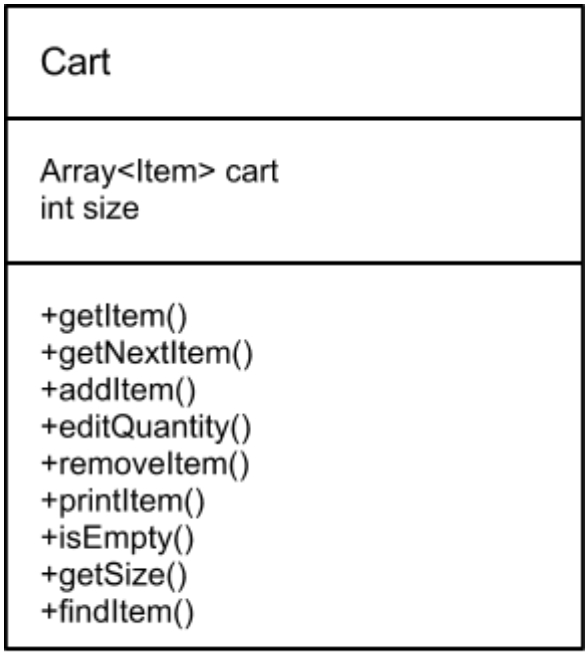


Figure 2.3b

The Item and Cart classes are to be used to hold data about the items that are added to a user's shopping cart. This will make things easier to modify by storing the data locally rather than having to query the database multiple times. The Item class (Figure 2.3a) will hold the essential information required to display the item to the user (while it's in the shopping cart). The Cart class (Figure 2.3b) will hold an array of Items, and will have multiple methods to make the shopping cart easy to use within other PHP files.

In order to display the shopping cart on the pages, we will implement a shoppingCart file which will display the contents of the shopping cart. This file will need to have access to the methods in the Cart class (Figure 2.4), which in turn must have access to the methods in the Item class.

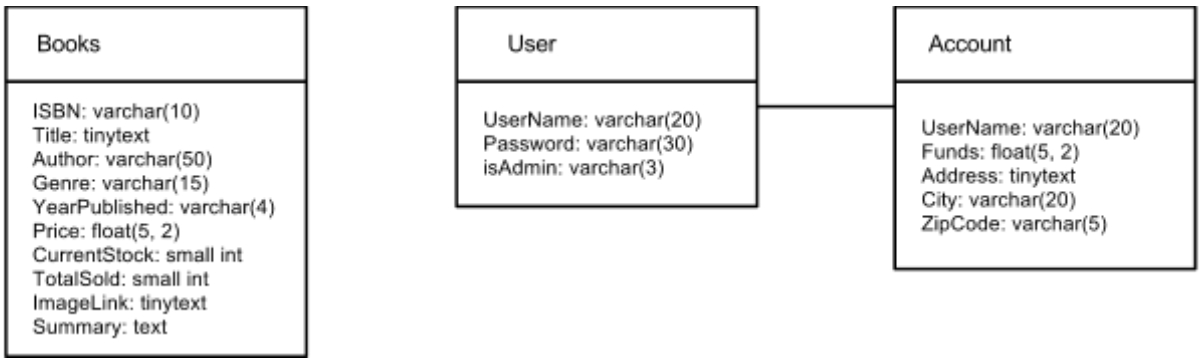
Figure 2.4

On the next page is a hierarchical diagram (Figure 2.5) of the overall system. Basically any page will be able to be accessed from the home page. For many of the components that require user input, there will be two or more pieces to the module. For example, the create account page will require a page to gather the user's input, and then a script for validation (to let the user know that they input something wrong), and then another script to register their account with the database and set up their session.

1. **Home Page**
    a. **Login**
        i. **HTML Form**
        ii. **Validation**
    b. **Create An Account**
        i. **HTML Form**
        ii. **Validation**
    c. **Logout (Customer/Admin Only)**
    d. **View Account (Customer Only)**
        i. **Add Funds**
            1. **HTML Form**
            2. **Validation**
            3. **Confirm Add Funds**
    e. **Edit Inventory (Admin Only)**
        i. **HTML Form**
        ii. **Validation**
        iii. **Update Item**
        iv. **Remove Item**
    f. **Search**
        i. **Browse**
            1. **HTML Form**
            2. **Validation**
            3. **Add to Cart (Customer Only)**
    g. **Popular Genres**
        i. **Browse**
            1. **HTML Form**
            2. **Validation**
            3. **Add to Cart (Customer Only)**
    h. **Shopping Cart Mini (Customer Only)**
        i. **Shopping Cart**
            1. **HTML Form**
            2. **Validation**
            3. **Update Cart**
            4. **Checkout**
    i. **About Us**

3. Persistent Data Design

3.1 Database Descriptions



These are the database schemas we will be using in this system.

3.2 File Descriptions

No files will be used in the implementation of this system.

4. **Requirements Matrix**

Functional Requirements	**Files which satisfy the FR**
1) Search	Search.php
2) Purchase	checkout.php
2a) Add to Shopping Cart	addToCart.php, shoppingCart.php
3) Browsing	browse.php
4) Create Account	createAccount.php
5) Add Funds	addFunds.php, confirmFunds.php
6) Account Login	login.php, getLogin.html
7) Edit Inventory	AdminSubmit.php, UpdateRemove.php

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 the current design is satisfactory and has been carried out and elaborated to the fullest extent. We guarantee that everyone has read this document and agrees to the content, or alternatively, 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 and the team will work with the customer 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 system design 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.

_____

Christopher Raborg, Team Facilitator

_____

David Guldan, Requirements Leader

_____

Kevin Yu, Design Leader

_____

Zachary Hisley, Implementation Leader

_____

Matt Sperbeck, Delivery Leader