# **System Design Document**

***UMBC Bazaar***

**Client**

Ericka Marable

**Team 3: UMBC Obsidian**

Gene Burchette

Jerson Guansing

Karl Justice

Darren Stevens

Victor Wu

***UMBC Bazaar***

**System Design Document**

#### **Table of Contents**

Page

## Introduction 3

1.1 Purpose of This Document 3

1.2 References 3

1. System Architecture 3

2.1 Architectural Design 3-4

2.2 Decomposition Description 4-6

1. Persistent Data Design 6

3.1 Database Descriptions 6-7

3.2 File Descriptions 8

1. Requirements Matrix 9
2. Appendix A – Agreement Between Customer and Contractor 10
3. Appendix B – Team Review Sign-off 11
4. Appendix C – Document Contributions 12
5. Appendix D - Changes Made 13-16

**1. Introduction**

**1.1 Purpose of This Document**

The purpose of this document is to describe the design of the UMBC Bazaar web application. Key topics covered in this document covered in this document include the high level system architecture, lower level class designs, and the persistent data design of UMBC Bazaar.

**1.2 References**

Throughout this document references will be made to :

1. The UMBC Bazaar System Requirements Documents
2. The UMBC Bazaar User Interface Design Document
3. UML Distilled, 2nd edition, Martin Fowler and Kendall Scott, 2000
4. Pressman, R.S. (2010). *Software engineering: A practitioner's approach* (7th ed.). New York: McGraw-Hill Higher Education.

**2. System Architecture**

**2.1 Architectural Design**



The UMBC Bazaar application will be built using PHP Frameworks. The basic components of the framework include Models, Views and Templates which follow a Model View Controller architecture (MVC).

The model layers consists of classes which refers to specific tables in the database. The model classes describe either user, product or order data, and will be described in more details later in this document.

Templates are the specific pages served to the user. Templates are made up of PHP, HTML, CSS and other static files. The templates used in this application will be described more fully in the User Interface Design Documentation.

The views are a layer abstraction between the model and template layers. These include access to the email system, PHP libraries such as curl, mysql connector, etc. A view, or the controller in MVC, is a function which takes input and returns output to the user through the template layer. Views in this application include the dynamically generated pages / results based on inputs and subset data. These include search, view details, browse and others which will be described more fully later in this document.

**2.2 Decomposition Description**

UMBC Bazaar is using PHP Framework that cannot adequately describe the structural decomposition diagram, and so a custom illustration is used.



Each model in UMBC Bazaar represents a specific database table. The model that is related to users are the data stored in the users table. More specific details on the functionality of each PHP file are in the functional requirements section of the Systems Requirement Specification documentation. The users database model encapsulates the UMBC ID, username, password and all other user specific data. Procedurally, this model is accessed by the login, register, confirm, profile and password reset PHP files.

The Products database model represents the products database table. This includes all published and inactive listings of products or services offered by all UMBC bazaar users. Each model entry encapsulates the product’s name, description, image and all other details pertaining to the product or service. Procedurally, this model is accessed by list seller’s product , add product, search, and product details PHP files.

The Orders database model represents that orders database table. This model encapsulates each order transaction with the UMBC Bazaar application. It encapsulates the users involved in the transaction, payment or trade options, product details, and all other details pertaining to the order. Procedurally, this model is accessed by the cart, process order, list order, view, and order details PHP files.



The UMBC Bazaar is using a PHP Framework, and so the view decomposition diagram cannot adequately describe it.

The register view takes an HTTP Request to retrieve user entered criteria, but it also makes use of the template response for user feedback. Furthermore, it requires access to additional input systems such as the UMBC Directory and output system such as the email system to send account confirmation emails.

The same applies to all other views that require HTTP Request (POST or GET), models through the database, template response by generating dynamic pages as feedback, and other third party systems such as email and payment services.

**3. Persistent Data Design**

**3.1 Database Descriptions**

An entity-relationship of our database schema for our UMBC Bazaar database can be found below.

All user, product and order information are each kept in a separate database named users, products and orders respectively. These databases were created automatically by the MariaDB framework. After creating the users, products, and order related classes, a built in tool created the databases to our specification. The schema for our databases is modeled below:



3.2 File Descriptions

FileList:

**CommonMethods.php**: This PHP file contains all the common functions used by several files

**DBConnection.php**: This PHP file encapsulates the database interface for the project

**README**: add src folder

**UMBCDirectory.php**: This PHP file uses curl to scrape information from the UMBC Directory given an ID

**bottom.html**: Bottom template

**captcha.php** : This PHP file produces a simple captcha image and saves the value in the session

**confirm.php**: This PHP file handles the account verification via the email after a user registers for an account

**icon.png**: For the main login PHP file

**index.php:** Initial Index

**login.js:** For the main login PHP file

**login.php:** The main login PHP file  
**logout.php:** Destroys the current session and creates a new one. Use redirected back to login

**middle.html:** Middle template

**register.php:** PHP file that handles the registration process

**resetpass.php:** This resets the password for users who forgot their login credential. Also, the login system will block any attempts after 15 failed logins based on their IP address. The file, resetpass.php will circumvent this by resetting the attempt counter back to zero after a new password has been sent to the user's email

**retrievers.jpg:**  This PHP produces JSON output for the XMLHttpRequest during the registration process

**search1.png:** For the PHP that produces JSON output for the XMLHttpRequest during the registration process

**students.php:** This PHP produces JSON output for the XMLHttpRequest during the registration process

**styles123.css:** CSS styles

**top.html:** Top template

**4. Requirements Matrix**

System Components:

1. Login.php
2. Register.php
3. Confirm.php
4. Profile.php
5. Resetpass.php
6. Sellproduct.php
7. Addproduct.php
8. Search.php
9. Cart.php
10. Order.php
11. Orderdetail.php

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | System  Components | A | B | C | D | E | F | G | H | I | J | K |
| Functional Requirements |  |  | | | | | | | | | | |
| FR-001 Register Account |  | X | X |  | X |  |  |  |  |  |  |
| FR-002 Login to Account | X |  |  |  |  |  |  |  |  |  |  |
| FR-003 Logout |  |  |  | X |  |  |  |  |  |  |  |
| FR-004 Update Profile |  |  |  | X | X |  |  |  |  |  |  |
| FR-005 Search for Product |  |  |  |  |  |  |  | X |  |  |  |
| FR-006 View Product |  |  |  |  |  | X |  | X |  |  |  |
| FR-007 Add Product |  |  |  |  |  |  | X |  |  |  |  |
| FR-008 Buy product |  |  |  |  |  |  |  |  | X | X | X |
| FR-009 Payment Purchase |  |  |  |  |  |  |  |  |  | X | X |
| FR-010 Trade Purchase |  |  |  |  |  |  | X |  |  | X | X |
| FR-011 View Orders |  |  |  |  |  |  |  |  |  | X | X |

**Appendix A – Agreement Between Customer and Contractor**

The client agrees to a *UMBC Bazaar* web application with registering, searching, buying, trading, and other user-based capabilities. Use cases are included in the functional requirements section above of the behavior between the system and user. Additional features will be provided in further development iterations.

When and if future changes to this document occur, a new drafted document shall be created. Both a hard and electronic copy of all versions shall be presented to the client for review. Upon approval, the draft will be finalized and signed off by both the client and contractor.

**Client**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:

Ericka Marable

**Team**

\_Gene Burchette\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/3/2016

Gene Burchette

\_\_\_\_\_\_Jerson Guansing\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/3/2016

Jerson Guansing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:

Karl Justice

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:

Darren Steven II

Victor Wu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/3/16

Victor Wu

**Appendix B – Team Review Sign-off**

This document has been collaboratively written by all members of the team. Additionally, all team members have reviewed this document and agree on both the content and the format. Any disagreement or concerns are addressed in team comments below.

\_Gene Burchette\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/3/2016

Gene Burchette

\_\_\_\_\_\_\_Jerson Guansing\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/3/2016

Jerson Guansing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:

Karl Justice

\_\_\_\_\_\_DSII\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/3/2016

Darren Stevens II

Victor Wu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 11/3/16

Victor Wu

**Appendix C – Document Contributions**

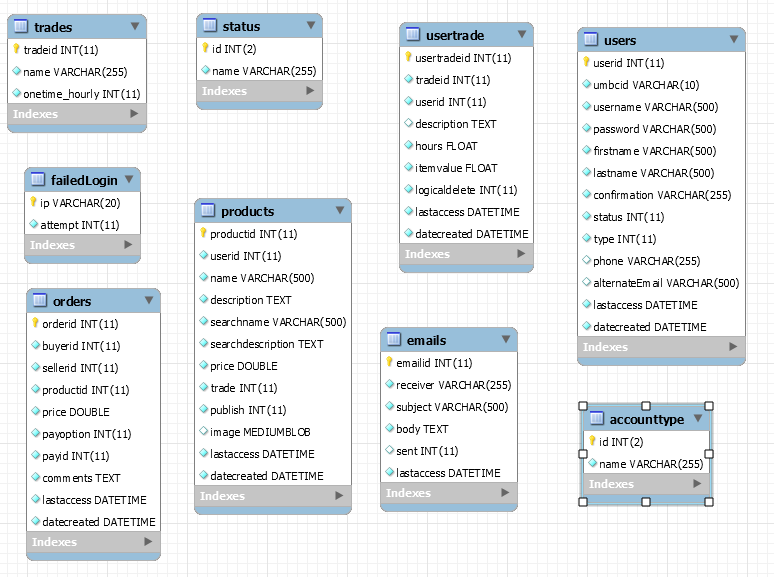
Jerson - Contributed on Section 2.1 Architectural Design, section 2.2 Decomposition Description, and Section 3.1 Database Description.

**Appendix D – Changes Made**

2.2 Decomposition Description

Cart was not implemented instead checkout.php was used.

3.1 Database Descriptions



Did not implement OrderDetails table since the shopping cart was scrapped. Instead productid column was added to the orders table.

3.2 File Descriptiions

|  |  |
| --- | --- |
| .htaccess | Contains rule for Error Documents |
| about.php | The “About Us” page |
| addproduct.php | Template to add or edit product listings |
| backup.php | Generates an SQL dump to backup the database |
| bottom.html | HTML template for the bottom/footer section of pages. |
| captcha.php | Generates a captcha image that needs to be matched. |
| checkout.php | In place of shopping cart. Select either cash on delivery or trade options. |
| CommonMethods.php | PHP file that contains all the commonly used functions. |
| confirm.php | PHP file that activates a newly registered account. |
| contactus.php | An interface to contact via email the administrators. |
| DBConnection.php | Class object and abstraction to connect to the MySQL database. |
| email.php | Interface that allow users to send another user an email. |
| error.php | Generic error page. |
| icon.png | Favicon for the website |
| index.php | The main page of the website |
| listtrade.php | Lists all the current “My Trade” options for a user. |
| login.js | JavaScript used for the login page. |
| login.php | The login page authenticates login credentials before using the website. |
| logout.php | Invalidates or destroys the current session. |
| middle.html | HTML template for the middle section of the pages. |
| No\_Image\_Available.png | Image to be displayed if the listing don’t have an image assigned. |
| orders.php | This PHP file lists the order history and details of the user. |
| productDetail.php | Displays more details about a specific product listing. |
| profile.php | Allows the use to edit their user information and change their password. |
| register.php | The PHP interface for users to register a new account. |
| resetpass.php | Allows users to reset forgotten password. Sends new password via email. |
| retrievers.jpg | UMBC Bazaar Logo |
| search.php | Processes keyword submission and displays relevant search results. |
| search1.png | Image for the search submit button. |
| sellproduct.php | Lists all the user’s product or services. |
| students.php | JSON application that checks the UMBC directory given an ID. |
| styles123.css | CSS Stylesheet for the webpages. |
| top.php | HTML template that displays the top navigation and UMBC Bazaar logo. |
| tradeoptions.php | PHP file that allows user to add new “My Trade Options” based on a predetermined list. |
| UMBCDirectory.php | PHP function that allow client-side code to authenticate if a UMBC ID is valid. |