

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Project Title:
E-Commerce Database System

Team Members: Aday Devora Delgado, Daniel Jesus Gamon, Joshua Michael Doucette, Natan Farhy, Robens Duverny

Product Owner: Robens Duverny

Mentor(s): Robens Duverny

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Abstract

This document presents a comprehensive overview of the DataDash E-commerce Database System, detailing its functionalities, design principles, and validation procedures. The system aims to address the limitations of current e-commerce platforms by providing a robust, scalable, and secure solution for managing product inventories, customer data, orders, and transactions.

Through the implementation of user stories derived from stakeholder requirements, the DataDash system offers enhanced features such as personalized user accounts, advanced product search capabilities, seamless shopping cart management, and detailed order tracking. The document outlines the project's agile development approach, including sprint planning and execution, which allowed for iterative improvements and timely delivery of features.

The system's architecture is built on modern design patterns and principles, ensuring modularity, maintainability, and extensibility. It employs a multi-tier architecture, separating concerns between the presentation layer, business logic, and data access layers. This design facilitates easier updates and scalability as the system grows.

Security and performance are key considerations in the DataDash system. The document details the measures implemented to ensure data protection, including secure user authentication, encrypted data storage, and protected transaction processing. Performance optimization techniques are also discussed, highlighting how the system handles high concurrent user loads and large data volumes efficiently.

This document serves as a comprehensive guide for developers, system administrators, and stakeholders, providing insights into the system's functionality, architecture, and development process. It

includes detailed UML diagrams, user interface designs, and validation procedures, offering a complete picture of the DataDash E-commerce Database System's capabilities and potential for future enhancements.

Table of Contents

Introduction.....	6
Current System.....	6
Purpose of New System.....	6
User Stories.....	7
Implemented User Stories.....	7
Pending User Stories.....	9
Project Plan.....	10
Hardware and Software Resources.....	10
Sprints Plan.....	11
Sprint 1.....	11
Sprint 2.....	11
Sprint 3.....	12
Sprint 4.....	13
Sprint 5.....	13
System Design.....	15
Architectural Patterns.....	15
System and Subsystem Decomposition.....	16
Deployment Diagram.....	17
Design Patterns.....	17
System Validation.....	18
Glossary.....	19
Appendix.....	22
Appendix A - UML Diagrams.....	22
System Architecture Diagram.....	22
Database Diagram.....	23
Appendix B - User Interface Design.....	24
Appendix C - Sprint Review Reports.....	29
Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents.....	35
References.....	37

Introduction

This document provides a comprehensive overview of the DataDash E-commerce Database System, a dynamic and user-friendly platform designed to empower online businesses. DataDash seeks to address the limitations of existing e-commerce solutions by offering a robust, scalable, and secure system for managing products, customers, orders, and transactions.

Driven by a user-centric approach, the development of DataDash has been guided by stakeholder requirements, translated into actionable user stories. The system delivers essential e-commerce features, including personalized user accounts, a comprehensive product catalog with advanced search capabilities, a streamlined shopping experience with shopping cart and wishlist management, and secure checkout and order processing.

Built on a multi-tier architecture and utilizing key design principles, DataDash prioritizes modularity, maintainability, and extensibility. The system is designed to accommodate growth and adapt to changing business needs, with a clear separation of concerns between the presentation layer, business logic, and data access layers.

Current System

The current system has limitations in managing product inventory, customer data, orders, and transactions. Issues include inefficiencies, data security vulnerabilities, and challenges in handling large data volumes and integrating various system components.

Purpose of New System

The new system aims to address the shortcomings of the current system by enhancing data security, improving efficiency in managing large data volumes, and integrating seamlessly with other components. The goal is to provide a robust and scalable solution for product inventory management, customer data handling, order processing, and transaction management.

USER STORIES

Implemented User Stories

1. As a customer, I want to be able to create an account and securely store my personal information.
2. As a customer, I want to be able to search for products by category, brand, or keyword.
3. As a customer, I want to be able to add items to the shopping cart, update quantities, and remove items.
4. As a customer, I want to receive email confirmation emails with detailed summaries of my purchases.
5. As a customer, I want to be able to view my order history and track the status of my current orders.
6. As a user, I want the homepage to be visually appealing, with clean layouts.
7. As a database administrator, I want to define the tables needed for storing product information, including attributes.
8. As a database administrator, I want to establish relationships between tables representing the connections between products, categories, brands, etc.
9. As a customer, I want to be able to view product details, including descriptions, images, and reviews, so that I can make informed purchasing decisions.
10. As a customer, I want to be able to leave reviews and ratings for products so that I can help others make informed purchasing decisions.
11. As a customer, I want to access a Frequently Asked Questions, Customer Service, and Terms & Conditions page.
12. As a customer, I want to receive email confirmation emails with detailed summaries of my purchases.

13. As a customer, I want to be able to cancel my order before it ships so that I can make changes if needed.
14. As a customer, I want to reset my password if I forget it so that I can regain access to my account.
15. As a customer, I want to see a confirmation page with a summary of my purchase when processed.
16. As a developer, I want a new user account to be recorded in the database when created.
17. As an administrator, I want all usernames and emails to be unique.
18. As a user, I want to see a success message when I update something relating to my account.
19. As a user, I want to add products to a wishlist so that I can save items for future purchase.
20. As a user, I want the search and category selection buttons and functionality to work.
21. As a user, I want to see items related to the one that I am currently viewing.
22. As an admin, I want the functionality to create, update, and delete products.
23. As an admin, I want an account so that I can login to make changes.
24. As an admin, I want users to only access certain webpages if signed in.
25. As an admin, I want to modify and create new banners.
26. As a user, I want to apply discount codes to take advantage of promotions.
27. As a user, I want to view all the reviews that I posted so that I can modify or delete them.
28. As an admin, I want to edit, delete, and create new brands for the catalog.
29. As a user, I want there to be a buy now button.
30. As an admin, I want URL rewriting to be done on the necessary pages.
31. As a user, I want to create a wishlist.
32. As an admin, I want to add discounts in the store.

- 33. As an admin, I want to retrieve my password if forgotten.
- 34. As a user, I want to be able to view all the reviews that I posted so that I can modify or delete them.
- 35. As an admin, I want to be able to edit, delete and create new brands for the catalog.
- 36. As an admin, I want to be able to view and update the product inventory.
- 37. As a user, I want there to be a buy now button.
- 39. As a user, I want to be able to create a wishlist.
- 41. As an admin, I want to be able to retrieve my password just in case If I were to have forgotten it.

Pending User Stories

- 38: As an admin, I want to have URL rewriting to be done on the necessary pages.
- 40: As an admin, I want to be able to add discounts in the store.

PROJECT PLAN

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

Hardware: Development and testing computers, servers for deployment.

Software: HTML, CSS, JavaScript, PHP, MySQL, Apache web server.

Sprints Plan

Sprint 1

- User Story 1: As a customer, I want to be able to create an account and securely store my personal information.
- User Story 2: As a customer, I want to be able to search for products by category, brand, or keyword.
- User Story 3: As a customer, I want to be able to add items to the shopping cart, update quantities, and remove items.
- User Story 4: As a customer, I want to receive email confirmation emails with detailed summaries of my purchases.
- User Story 5: As a customer, I want to be able to view my order history and track the status of my current orders.
- User Story 6: As a user, I want the homepage to be visually appealing, with clean layouts.
- User Story 7: As a database administrator, I want to define the tables needed for storing product information, including attributes.
- User Story 8: As a database administrator, I want to establish relationships between tables representing the connections between products, categories, brands, etc.

Sprint 2

- User Story 9: As a customer, I want to be able to view product details, including descriptions, images, and reviews, so that I can make informed purchasing decisions.
- User Story 10: As a customer, I want to be able to leave reviews and ratings for products so that I can help others make informed purchasing decisions.

- User Story 11: As a customer, I want to access a Frequently Asked Questions, Customer Service, and Terms & Conditions page.
- User Story 12: As a customer, I want to receive email confirmation emails with detailed summaries of my purchases.
- User Story 13: As a customer, I want to be able to cancel my order before it ships so that I can make changes if needed.
- User Story 14: As a customer, I want to reset my password if I forget it so that I can regain access to my account.
- User Story 15: As a customer, I want to see a confirmation page with a summary of my purchase when processed.
- User Story 16: As a developer, when a new user account is created, I want it to be recorded in the database.

Sprint 3

- User Story 25: As a user, I want to add products to a wishlist so that I can save items for future purchase.
- User Story 26: As a user, I want the search and category selection buttons and functionality to work.
- User Story 27: As a user, I want to see items related to the one that I am currently viewing, because it will allow me to view other possible options.
- User Story 28: As an admin, I would like to have functionality that allows me to create, update, and delete products.
- User Story 29: As an admin, I would like to have functionality that allows me to create, update, and delete products.

- User Story 30: As an admin, I would like to have an account so that I can login to make changes.

Sprint 4

- User Story 27: As a user, I want to see items related to the one that I am currently viewing, because it will allow me to view other possible options.
- User Story 31: As an admin, I want users to only be able to access certain webpages if they are signed in.
- User Story 32: As an admin, I want to have the functionality to be able to modify and create new banners.
- User Story 33: As a user, I want to apply discount codes so that I can take advantage of promotions.
- User Story 34: As a user, I want to be able to view all the reviews that I posted so that I can modify or delete them.

Sprint 5

- User Story 37: As a user, I want there to be a buy now button.
- User Story 35: As an admin, I want to be able to edit, delete, and create new brands for the catalog.
- User Story 23: As a user, I want to be able to see a success message when I update something relating to my account, not just resetting a password.
- User Story 38: As an admin, I want URL rewriting to be done on the necessary pages.
- User Story 39: As a user, I want to be able to create a wishlist.
- User Story 40: As an admin, I want to be able to add discounts in the store.

- User Story 41: As an admin, I want to be able to retrieve my password just in case I were to have forgotten it.

SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

Architectural Patterns

The DataDash e-commerce system utilizes a multi-tier architecture, also known as a layered architecture, to organize its functionality and ensure modularity, maintainability, and scalability. This architectural style separates the system into distinct layers, each with specific responsibilities:

1. Presentation Layer (Frontend):

Purpose: Responsible for handling user interactions, displaying information, and presenting the user interface.

Technologies:

- HTML
- CSS
- JavaScript.

Responsibilities:

- Gathering user input.
- Displaying product information, shopping cart contents, user account details, etc.
- Handling user navigation and interactions.
- Validating user input on the client-side.
- Making requests to the backend (business logic layer) to retrieve or update data.

2. Business Logic Layer (Backend):

Purpose: The core of the application, responsible for processing data, applying business rules, and managing application logic.

Technologies: PHP.

Responsibilities:

- Processing user requests from the presentation layer.
- Enforcing business rules (e.g., inventory checks during checkout, discount code validation).
- Interacting with the data access layer to retrieve or store data in the database.
- Orchestrating workflows such as order processing, payment handling, and email notifications.

3. Data Access Layer:

Purpose: Provides an abstraction layer for interacting with the database, isolating the business logic from the specific database technology.

Technologies: MySQL (database) and PHP Data Objects (PDO).

Responsibilities:

- Executing database queries (CRUD operations: Create, Read, Update, Delete).
- Retrieving and persisting data to the MySQL database.
- Ensuring data integrity and security.

System and Subsystem Decomposition

Packages:

- User Management: Handles user registration, login, authentication, password management, and user profiles.
- Product Catalog: Manages products, categories, brands, inventory, and product reviews.
- Shopping Cart and Wishlist: Implements the shopping cart functionality and wishlist management.
- Checkout and Order Processing: Handles the checkout process, order placement, and payment handling.
- Admin Panel: Provides functionalities for website administrators to manage products, inventory, categories, banners, and other store settings.

Subsystem Description:

- Each package represents a distinct subsystem within the DataDash application.
- Subsystems communicate with each other through well-defined interfaces (function calls, API requests).
- Clear subsystem boundaries improve modularity and maintainability.

Deployment Diagram

The DataDash application is deployed on a web server that can be either a shared hosting environment or a dedicated server. The web server runs the PHP code and serves the frontend content (HTML, CSS, JavaScript).

The database server, which can be separate or hosted on the same machine as the web server, houses the MySQL database.

Deployment diagram in Appendix A.

Design Patterns

While the exact design pattern in DataDash is still being refined, the project emphasizes important design principles and makes use of design patterns to improve the codebase's structure, flexibility, and maintainability. Here are some of the key patterns and concepts that are likely being used:

1. Separation of Concerns:

Description: A fundamental principle that advocates dividing the application's functionality into distinct modules or components. This makes the code easier to understand, modify, and test.

Example: The frontend (HTML, CSS, JavaScript) handles user interactions and display, while the backend (PHP, MySQL) deals with data processing, business logic, and database interactions. This separation makes each part of the system more manageable.

2. Encapsulation:

Description: Bundling data and the methods that operate on that data within a single unit (class or module). This hides internal implementation details and provides a controlled interface for interacting with the data.

Benefits:

- Data Protection: Prevents accidental data corruption.
- Code Organization: Promotes modularity and code reuse.

SYSTEM VALIDATION

Functional Testing:

- Unit Testing: Individual functions and methods within each component are tested to ensure correct behavior.
- Integration Testing: Components are tested together to verify they interact correctly.
- System Testing: The entire system is tested as a whole to ensure all functionalities work as expected.

Usability Testing:

- Real users test the website to identify any usability issues or areas for improvement in terms of navigation, clarity, and ease of use.

Performance Testing:

- Load Testing: Simulates a high number of concurrent users to test the system's performance under load.
- Stress Testing: Pushes the system to its limits to identify bottlenecks and breaking points.

GLOSSARY

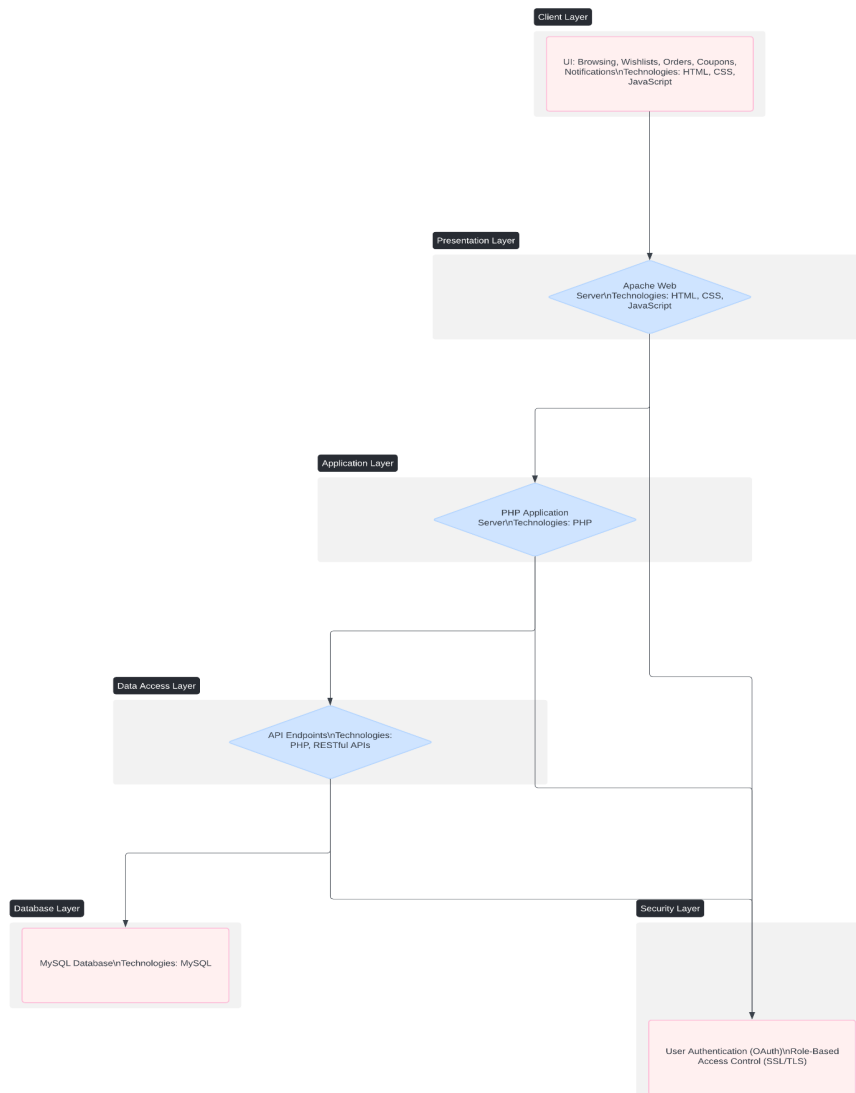
Term	Definition
API (Application Programming Interface)	A set of rules and specifications that software programs can follow to communicate with each other.
Agile Development	An iterative software development methodology that emphasizes flexibility, collaboration, and customer feedback.
Backend	The server-side of a website or application responsible for processing data, business logic, and database interactions.
Business Logic	The rules and procedures that govern how data is processed and manipulated within a system.
CSS (Cascading Style Sheets)	A language used to style the presentation of web pages (layout, colors, fonts).
Database	An organized collection of data, typically stored and accessed electronically.
Deployment Diagram	A UML diagram that shows the physical deployment of software components on hardware.
ERD (Entity-Relationship Diagram)	A visual representation of data entities and their relationships in a database.

Frontend	The client-side of a website or application, typically the user interface displayed in a web browser.
HTML (Hypertext Markup Language)	The standard markup language used to create the structure and content of web pages.
JavaScript	A scripting language used to create interactive and dynamic elements on web pages.
Load Testing	A type of performance testing that simulates multiple users accessing a system concurrently to measure its behavior under load.
MySQL	An open-source relational database management system.
PHP (Hypertext Preprocessor)	A server-side scripting language often used for web development.
SKU (Stock Keeping Unit)	A unique code used to identify a specific product variant (e.g., different size, color, or style of a shirt).
Sprint	A short, time-boxed period in agile development, typically 1-4 weeks long, during which a specific set of features are developed.

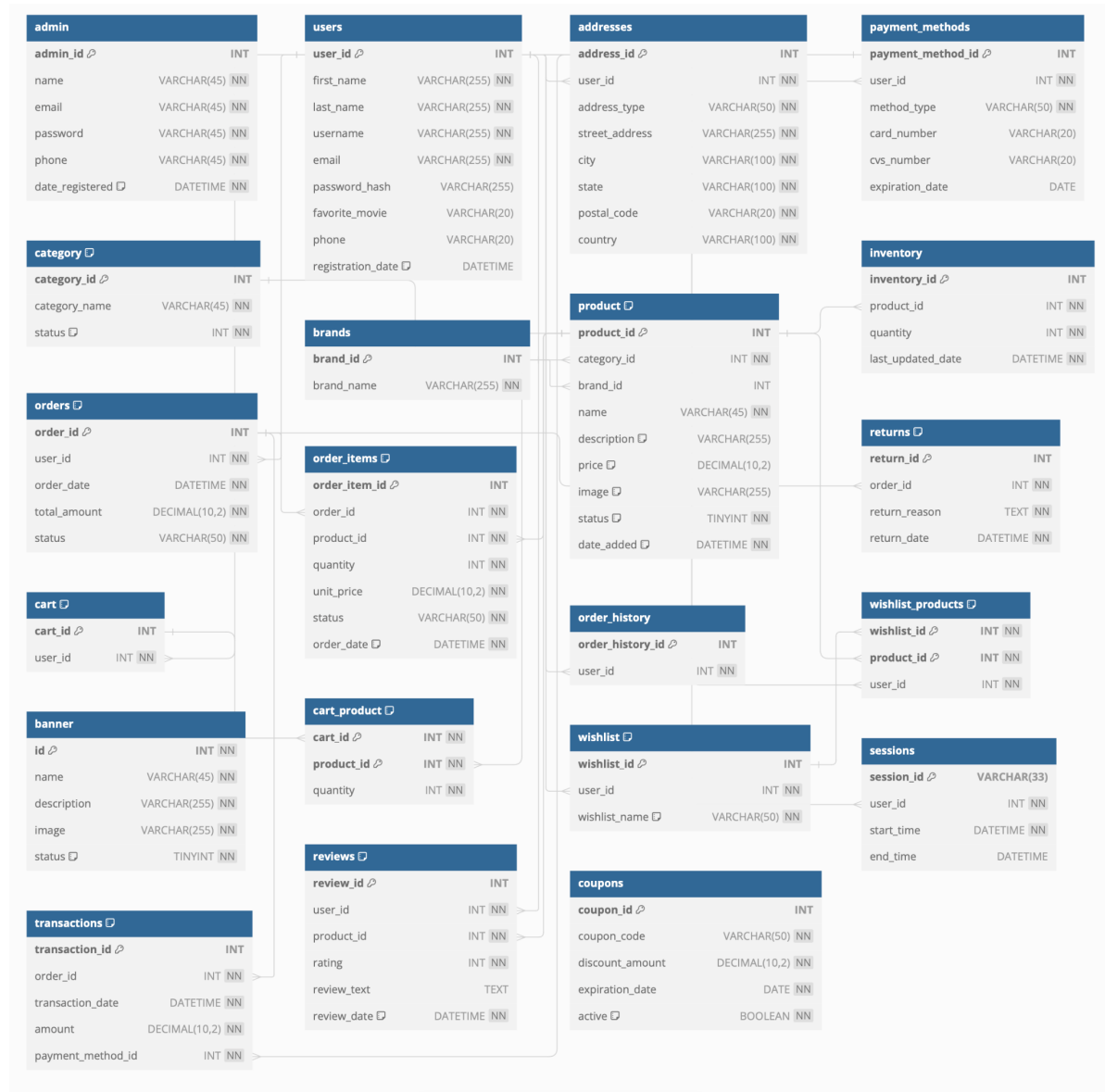
Stress Testing	A type of performance testing that pushes a system beyond its normal operating capacity to identify breaking points and bottlenecks.
Transaction	A single unit of work in a database, typically involving multiple operations that must all succeed or fail together.
UML (Unified Modeling Language)	A standard graphical language for visualizing, specifying, constructing, and documenting software systems.
User Story	A short, simple description of a feature from the perspective of a user, used in agile development.
Usability Testing	A process of evaluating a product or system's ease of use and user experience by observing real users interacting with it.
Wishlist	A feature that allows users to save items they are interested in purchasing for later.
MVC (Model-View-Controller)	A software design pattern that separates concerns: the model (data), the view (user interface), and the controller (business logic).
Singleton	A design pattern that ensures only one instance of a class exists, providing global access to that instance.

APPENDIX

Appendix A - UML Diagrams



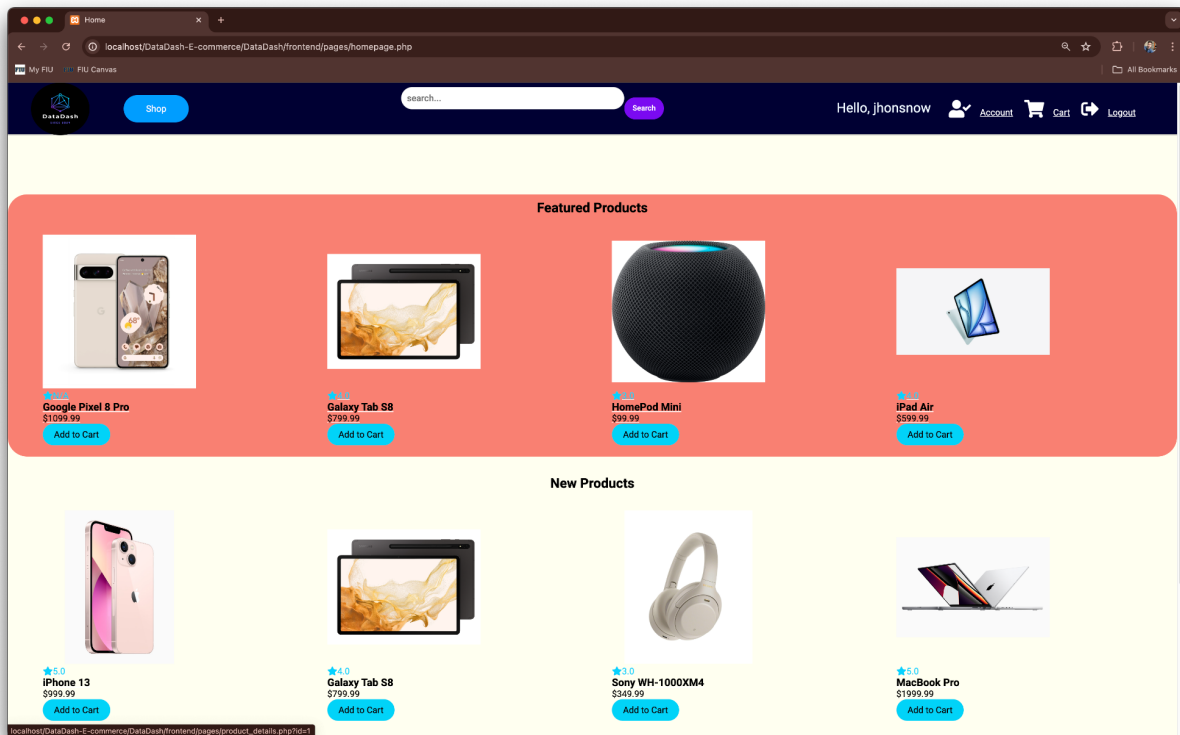
System Architecture Diagram



Database Diagram

Appendix B - User Interface Design

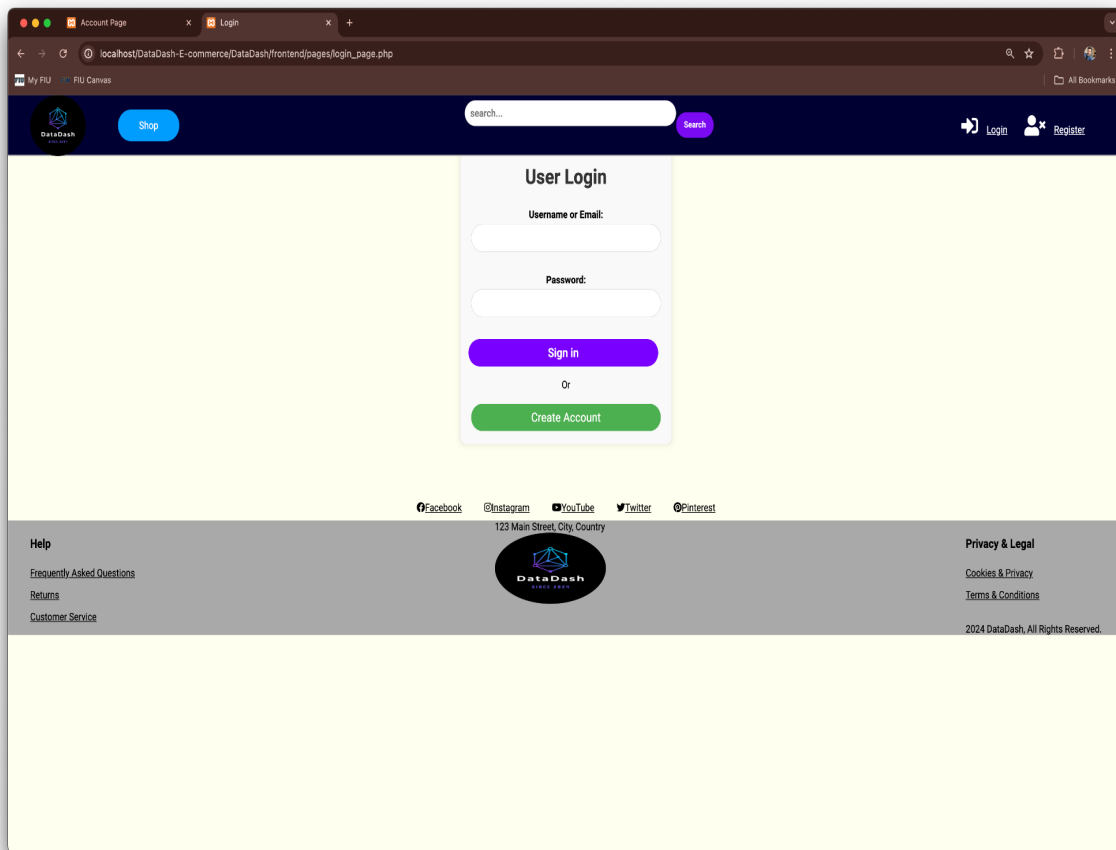
Homepage:



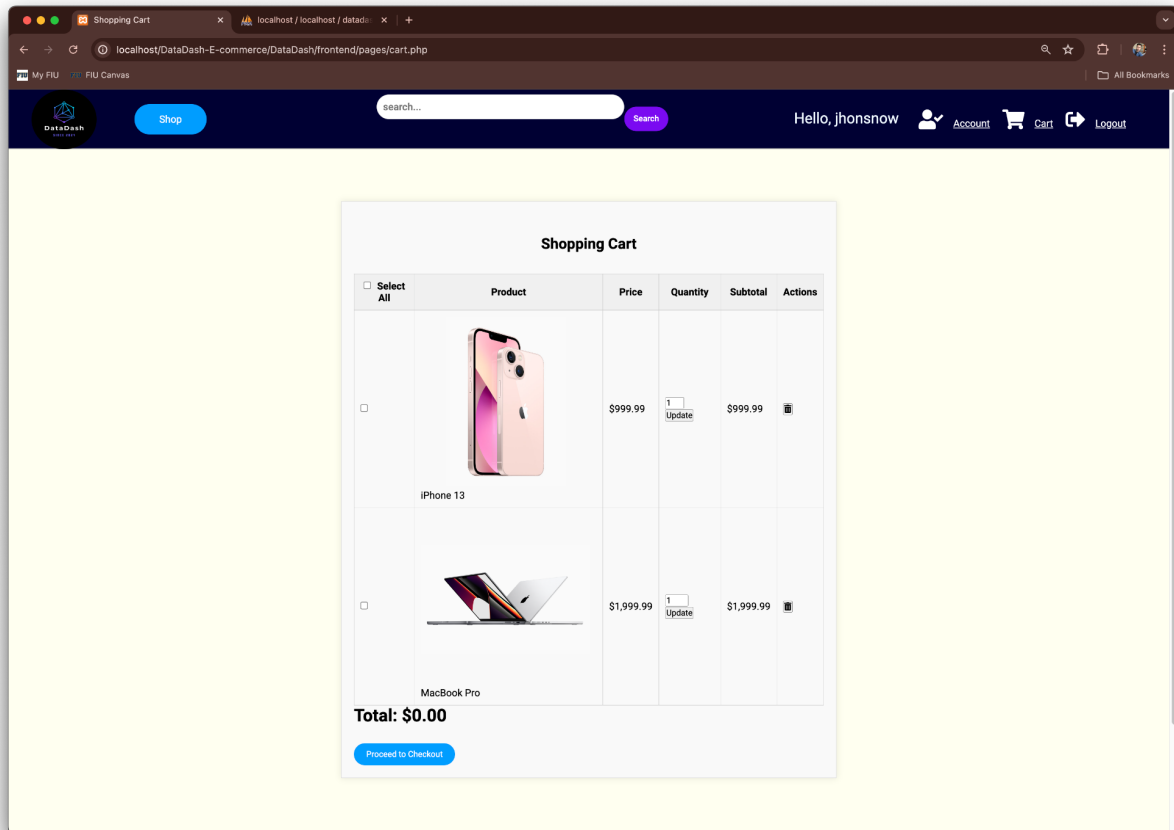
User Authentication:

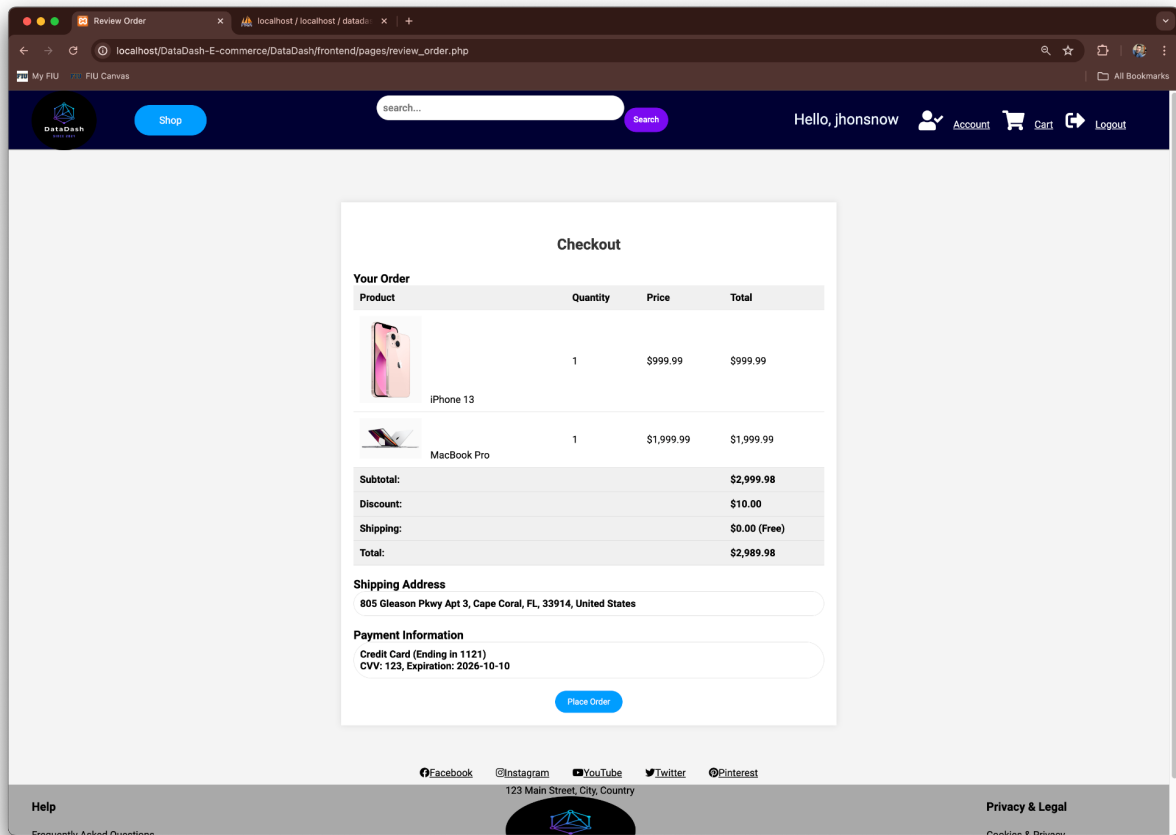
The screenshot displays a web browser window with the following elements:

- Browser Tabs:** 'Account Page' and 'Create Account'.
- Address Bar:** 'localhost/DataDash-E-commerce/DataDash/frontend/pages/create_account.php'.
- Page Header:** A dark blue navigation bar containing a 'DataDash' logo, a 'Shop' button, a search bar with the placeholder 'search...', and links for 'Login' and 'Register'.
- Main Content Area:** A light yellow background with a central white 'Create Account' form.
- Create Account Form:**
 - Fields: First Name, Last Name, Username, Email, Password, Confirm Password, Favorite Movie, and Phone (Optional).
 - Buttons: A purple 'Submit' button at the bottom of the form.
- Page Footer:** Social media icons for Facebook, Instagram, YouTube, Twitter, and Pinterest.



Shopping Cart and Order Checkout:





Appendix C - Sprint Review Reports

Sprint 1

Sprint Review Meeting Minutes

Attendees: Aday Devora Delgado, Daniel Jesus Gamon, Joshua Michael Doucette, Natan Farhy, Robens Duverny

Start time: 5/26/24, 8:35pm

End time: 5/26/24, 9:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story 1: As a customer, I want to be able to create an account and securely store my personal information.
- User Story 2: As a customer, I want to be able to search for products by category, brand, or keyword.
- User Story 3: As a customer, I want to be able to add items to the shopping cart update quantities, and remove items
- User Story 5: As a customer, I want to be able to view my order history and track the status of my current orders.
- User Story 6: As a user, I want the homepage to be visually appealing, with clean layouts.
- User Story 7: As a database administrator, I want to define the tables needed for storing product information, including attributes.
- User Story 8: As a database administrator, I want to establish relationships between tables representing the connections between products, categories, brands, etc.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- User Story 4

How this should be reflected on the user story definition in Mingle:

Title: Email confirmation

Description: As a customer, I want to receive email confirmation emails with detailed summaries of my purchases.

Acceptance Criteria:

Status: Moved from sprint 1, to be reassigned in sprint 2

Estimation: 6 Points

Priority: Medium

Sprint 2

Sprint Review Meeting Minutes

Attendees: Aday Devora Delgado, Daniel Jesus Gamon, Joshua Michael Doucette, Natan Farhy, Robens Duverny

Start time: 6/9/24, 8:35pm

End time: 6/9/24, 9:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story 9: As a customer, I want to be able to view product details, including descriptions, images, and reviews, so that I can make informed purchasing decisions.
- User Story 11: As a customer I want to be able to access a Frequently Asked Questions, Customer Service, and Terms & Conditions page.
- User Story 12: As a customer, I want to receive email confirmation emails with detailed summaries of my purchases.
- User Story 14: As a customer, I want to reset my password if I forget it so that I can regain access to my account.
- User Story 15: As a customer, I want to see a confirmation page with a summary of my purchase when processed.
- User Story 16: As a developer, When a new user account is created, I want it to be recorded in the database

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- None.

Sprint 3

Sprint Review Meeting Minutes

Attendees: Aday Devora Delgado, Daniel Jesus Gamon, Joshua Michael Doucette, Natan Farhy, Robens Duverny

Start time: 6/23/24, 8:35pm

End time: 6/23/24, 9:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- User Story 17: As an administrator, I want all usernames and emails to be unique.
- User Story 18: As a user, I want to be able to shop around.
- User Story 20: As an admin, I want to be able to add, update and delete items from the catalog.
- User Story 21: As an admin, I want to be able to add, update and delete categories.
- User Story 25: As a user, I want to add products to a wishlist so that I can save items for future purchase.
- User Story 26: As a user I want the search and category selection buttons and functionality to work.
- User Story 27: As a user, I want to see items related to the one that I am currently viewing, because it will allow me to view other possible options.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- None.

Sprint 4

Sprint Review Meeting Minutes

Attendees: Aday Devora Delgado, Daniel Jesus Gamon, Joshua Michael Doucette, Natan Farhy, Robens Duvernay

Start time: 7/7/24, 8:35pm

End time: 7/7/24, 9:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User story 36: As an admin, I want to be able to view and update the product inventory.
- User story 31: As an admin, I want users to only be able to access certain webpages if they are signed in.
- User story 33: As a user, I want to apply discount codes so that I can take advantage of promotions.
- User story 34: As a user, I want to be able to view all the reviews that I posted so that I can modify or delete them.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- User story 35: As an admin, I want to be able to edit, delete and create new brands for the catalog.

Sprint 5

Sprint Review Meeting Minutes 5

Attendees: Aday Devora Delgado, Daniel Jesus Gamon, Joshua Michael Doucette, Natan Farhy, Robens Duvernay

Start time: 7/21/24, 8:35pm

End time: 7/21/24, 9:00pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- User story 35: As an admin, I want to be able to edit, delete and create new brands for the catalog.
- User Story 37: As a user, I want there to be a buy now button.
- User Story 39: As a user, I want to be able to create a wishlist.
- User Story 41: As an admin, I want to be able to retrieve my password just in case If I were to have forgotten it.
- User Story 23: As a user, I want to be able to see a success message when I update something relating to my account, not just resetting a password.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- None.

Appendix D - User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

1. Development Environment Setup:

- Required Software:
- PHP 8.1 or later
- MySQL 8.0 or later
- Apache Web Server (or another compatible web server)
- Composer (dependency manager)
- Text Editor or IDE (Integrated Development Environment)

Steps:

1. Install PHP, MySQL, and Apache.
2. Configure Apache to serve the project directory.
3. Create a MySQL database for DataDash.
4. Import the schema.sql file into the database. (Include the actual file path)
5. Install project dependencies using Composer: `composer install`
6. Update the database connection settings in `backend/config/config.php`. (Include code snippet)

2. Server Deployment:

Steps:

1. Upload project files to the server.
2. Configure the web server (Apache or Nginx) to point to the public directory.
3. Create a MySQL database on the server.
4. Import the schema.sql file.
5. Update database connection settings in `backend/config/config.php` with server credentials.

3. Database Backups and Restores:

Backups:

Use `mysqldump` command (or a database backup tool) to regularly create backups of the database:

Example command: `mysqldump -u [username] -p[password] [database_name] > backup.sql`

Restores:

Use `mysql` command to restore a database from a backup file:

Example command: `mysql -u [username] -p[password] [database_name] < backup.sql`

Shortcomings:

- URL Rewriting (User Story 38): Not fully implemented due to time constraints.
- Admin Discount Management (User Story 40): The UI for managing discounts in the admin panel is not yet complete.

Wishlist (Future Enhancements):**Improved User Interface:**

- Responsive design for mobile devices.
- Implement more modern CSS frameworks for a visually appealing and consistent user experience.

-

Enhanced Search Functionality:

- Integrate auto-complete suggestions in the search bar.
- Implement advanced filtering options (e.g., by product attributes, price ranges).

Product Recommendations:

- Develop a recommendation system that suggests products based on user browsing history or past purchases.

-

Social Media Integration:

- Allow users to share products or their wishlist on social media platforms.

Email Marketing:

- Integrate with an email marketing service to send promotional emails and newsletters.

REFERENCES

1. Technologies and Tools:

- PHP Documentation: <https://www.php.net/manual/en/>
- MySQL Documentation: <https://dev.mysql.com/doc/>
- JavaScript Documentation (MDN): <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- HTML and CSS Documentation (W3Schools): <https://www.w3schools.com/>
- Composer Documentation: <https://getcomposer.org/doc/>

2. Design Patterns and Principles:

- Refactoring Guru (Design Patterns): <https://refactoring.guru/design-patterns>
- SOLID Principles: <https://en.wikipedia.org/wiki/SOLID>
- Other Design Pattern Resources: List any websites, books, or articles that were influential in your design decisions.

3. E-commerce Best Practices or Articles:

- Baymard Institute (Usability Studies): <https://baymard.com/>
- E-commerce Platform Documentation: If you researched specific e-commerce platforms for inspiration, include links to their documentation.
- Articles on E-commerce Development: If you found any articles or tutorials that were helpful, list them.

4. Github Repository: <https://github.com/dangamtechno/DataDash-E-commerce>