**Next Wave Shopping**

### Submitted By

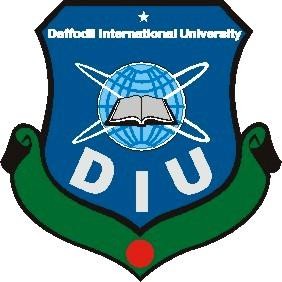
|  |  |
| --- | --- |
| **Student Name** | **Student ID** |
| Mobassir Mahmud Dibbo | 0242220005101718 |
| Mst. Lamiya Khatun | 0242220005101840 |

**MINI LAB PROJECT REPORT**

This Report Presented in Partial Fulfillment of the course

**CSE 222: Object Oriented Programming II Lab**

**in the Computer Science and Engineering Department**

****

**DAFFODIL INTERNATIONAL UNIVERSITY**

**Dhaka, Bangladesh**

**December 15, 2024**

## DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Nasima Islam Bithi**, **Lecturer**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

##### Submitted To:

**Nasima Islam Bithi**

Lecturer

Department of Computer Science and Engineering Daffodil International University

##### Submitted by

|  |  |
| --- | --- |
| A signature on a piece of paper  Description automatically generated  Mobassir Mahmud Dibbo ID:0242220005101718  Dept. of CSE, DIU | A close-up of a black text  Description automatically generated  Mst. Lamiya Khatun  ID:0242220005101840  Dept. of CSE, DIU |

## COURSE & PROGRAM OUTCOME

The following course have course outcomes as following:

Table 1: Course Outcome Statements

|  |  |
| --- | --- |
| **CO’s** | **Statements** |
| CO1 | **Define** and **Relate** classes, objects, members of the class, and relationships among  them needed for solving specific problems |
| CO2 | **Formulate** knowledge of object-oriented programming and Python in problem solving |
| CO3 | **Analyze** Unified Modeling Language (UML) models to **Present** a specific problem |
| CO4 | **Develop** solutions for real-world complex problems **applying** OOP concepts while  evaluating their effectiveness based on industry standards. |

Table 2: Mapping of CO, PO, Blooms, KP and CEP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CO** | **PO** | **Blooms** | **KP** | **CEP** |
| CO1 | PO1 | C1, C2 | KP3 | EP1, EP3 |
| CO2 | PO2 | C2 | KP3 | EP1, EP3 |
| CO3 | PO3 | C4, A1 | KP3 | EP1, EP2 |
| CO4 | PO3 | C3, C6, A3,  P3 | KP4 | EP1, EP3 |

The mapping justification of this table is provided in section **4.3.1**, **4.3.2** and **4.3.3**.

**Table of Contents**

1. Introduction
   * Overview of E-commerce Applications
   * Objectives of the Next Wave Shopping System
   * Key Features and Functionalities
2. Proposed Methodology/Architecture
   * Design Principles
   * Modular Code Architecture
     + Authentication Module
     + Shopping Module
   * Class Hierarchy and Inheritance
     + Product Base Class
     + Specialized Physical Product Class
   * Workflow Diagram (e.g., Login, Add to Cart, Checkout)
3. Implementation and Results
   * Programming Language and Framework Used
   * Code Walkthrough
   * Integration of Search Functionality
   * Error Handling and Validation
   * User Interface (Command-Line Interaction)
   * Example Scenarios and Outputs
4. Engineering Standards and Mapping
   * Object-Oriented Design Principles (SOLID)
   * Adherence to Software Design Patterns
   * Usability and Accessibility
   * Security Measures in Password Management
   * Performance Considerations
5. Challenges and Limitations
   * Scalability for Large Inventories
   * Limited User Interface
   * Absence of Database Integration
   * Future Scope for Improvement
6. Conclusion
   * Summary of Features
   * Overall Benefits of the System
   * Insights Gained During Development
7. Future Work
   * Integration of Graphical User Interface (GUI)
   * Adding Payment Gateway APIs
   * Extending the System to Support Multiple Categories
   * Real-time Inventory Updates via Database Integration
   * Advanced Search and Filtering Options

Introduction

**1. Introduction**

**Overview of E-commerce Applications**

E-commerce applications have revolutionized the retail industry by providing users with the ability to shop for products online from the comfort of their homes. These systems facilitate the buying and selling of goods or services through digital platforms, and they play a crucial role in modern-day business operations. E-commerce platforms offer several functionalities, such as product listing, user authentication, cart management, and payment integration, among others. Their success depends on factors like ease of use, security, scalability, and the seamless integration of various business processes.

In today's competitive digital marketplace, e-commerce applications need to address user needs effectively while ensuring high performance and security. From large corporations to small businesses, e-commerce systems have become essential for driving growth and customer engagement.

**Objectives of the Next Wave Shopping System**

The Next Wave Shopping System aims to provide an intuitive and robust platform for both users and administrators. The main objectives of this system are:

1. **Streamlined User Experience:** To create a user-friendly interface where customers can easily browse products, add them to their cart, and proceed to checkout without complications.
2. **Secure User Authentication:** To provide secure login and registration functionalities with password protection and security question-based recovery for user accounts.
3. **Efficient Product Management**: To allow administrators to manage the product catalog, including adding new products, updating existing ones, and viewing product details.
4. **Multiple Payment Options:** To integrate various payment methods, such as online payment gateways (bKash, Nagad, Rocket) and cash on delivery, to give users flexibility.
5. **Search and Filter Capabilities:** To offer an intuitive search function that allows users to find products by name or brand easily.
6. **Scalability:** To provide a system that can handle growth, whether through adding new products, expanding user registrations, or scaling the backend infrastructure.

**Key Features and Functionalities**

The Next Wave Shopping System provides the following core features and functionalities:

1. **User Authentication and Registration:**Users can create accounts, log in securely, and recover passwords if forgotten. The system ensures that only authorized users can access and manage their accounts.
2. **Product Management:**  
   Administrators can add new products to the system, specifying details such as name, price, brand, warranty, availability, quantity, and weight. They can also view and manage the inventory.
3. **Product Search:**  
   Users can search for products by name or brand. This functionality allows for quick navigation of the inventory, helping users find the items they want without browsing through the entire catalog.
4. **Shopping Cart:**  
   Users can add products to their cart and proceed to checkout. The cart tracks the selected items and ensures that users can review and adjust their selections before finalizing the purchase.
5. **Checkout Process:**  
   The system supports various payment methods, including online payments (bKash, Nagad, Rocket) and cash on delivery. Once a purchase is confirmed, the cart is cleared, and the user receives a summary of their purchase.
6. **Admin Access:**  
   Administrators have access to manage the full product inventory and make necessary updates. They can also monitor transactions and manage user queries related to the product catalog.
7. **Security:**  
   The system is designed to protect user data through secure login procedures and password management. Security questions are used as an extra layer for account recovery.
8. **User Interface:**  
   The system utilizes a command-line interface that allows users to interact with the platform effectively. Users can select options to view products, manage their cart, and proceed with checkout or payment.

The combination of these features makes the Next Wave Shopping System a powerful, secure, and user-friendly platform for both customers and administrators. It exemplifies a well-structured, scalable e-commerce system that can be further extended with additional features like inventory tracking, real-time product recommendations, and multi-platform support.

**2. Proposed Methodology/Architecture**

**Design Principles**

The Next Wave Shopping System is designed based on several key principles that ensure its efficiency, scalability, and maintainability. These principles guide the development of both the system’s architecture and its individual components. The key design principles include:

1. **Modularity:** The system is divided into distinct modules with clear responsibilities. Each module is designed to be independent and reusable, making it easier to update or extend the system in the future.
2. **Encapsulation:** By using object-oriented programming (OOP) principles such as encapsulation, the system ensures that each class handles its own state and behavior, thus reducing the complexity of interacting components.
3. **Abstraction:** The system provides simplified interfaces for both users and administrators. The complexity of implementation details is hidden from the user, enabling smooth interactions without dealing with the underlying complexities.
4. **Separation of Concerns:** The functionality of the system is organized into separate concerns—user management (authentication), product management, and shopping/cart management. This separation improves clarity and facilitates easier maintenance and future updates.

**Modular Code Architecture**

The Next Wave Shopping System adopts a modular architecture, organizing the system into independent modules that handle specific functionalities. This modular structure allows for better maintainability, scalability, and ease of debugging. Below are the two key modules of the system:

1. **Authentication Module:**
   * This module handles user authentication, registration, and password recovery.
   * It is responsible for managing user credentials securely, ensuring that users can log in, recover lost passwords, and register new accounts.
   * The Authentication class is central to this module. It manages user data, validates login attempts, and handles password recovery using security questions.

**Key functions:**

* + register(email, password, security\_question, security\_answer)
  + login(email, password)
  + recover\_password(email)

1. **Shopping Module:**
   * This module manages the product inventory, shopping cart, and checkout process.
   * It enables users to browse available products, add them to their shopping cart, view their cart, and complete the checkout process.
   * The NextWaveShopping class is the core of this module, managing product details, cart interactions, and the checkout flow.

**Key functions:**

* + add\_product(product)
  + show\_inventory()
  + add\_to\_cart(user\_email, product\_index)
  + view\_cart(user\_email)
  + checkout(user\_email)

**Class Hierarchy and Inheritance**

The system is built using object-oriented programming principles, with classes inheriting attributes and methods from base classes. The key classes and their relationships are:

1. **Product Base Class:**
   * The Product class is the base class for all products in the system. It defines the common attributes and behaviors shared by all products, such as product name, price, brand, warranty, availability, and quantity.
   * The Product class also defines an abstract method get\_details(), which must be implemented by any subclass to return specific product details.

**Key attributes:**

* + \_\_name
  + \_\_price
  + \_\_brand
  + \_\_warranty
  + \_\_available
  + \_\_quantity

**Abstract method:**

* + get\_details()

1. **Specialized Physical Product Class:**
   * The PhysicalProduct class is a subclass of the Product class and specializes it for physical items. In addition to the common product attributes, it includes an additional attribute: weight.
   * The PhysicalProduct class implements the get\_details() method to return detailed information about the physical product, including availability, weight, and warranty.

**Key attributes:**

* + \_\_weight

**Overridden method:**

* + get\_details()

By using inheritance, the PhysicalProduct class extends the functionality of the Product class, making the system flexible and easy to add new types of products in the future.

**Workflow Diagram**

Below is a high-level workflow diagram that illustrates the key processes in the Next Wave Shopping System. The main workflows are Login, Add to Cart, and Checkout. These workflows guide the interaction between the user, the system, and the backend processes.

**1. Login Workflow:**

* User Input: The user enters their email and password.
* Authentication Check: The system checks the credentials against the registered users.
* Outcome: If the login is successful, the user is granted access to the system; if not, an error message is shown.

**2. Add to Cart Workflow:**

* Product Selection: The user selects a product from the inventory.
* Add to Cart: The system checks if the product is available and then adds it to the user's shopping cart.
* Update Quantity: The product quantity is reduced by one, indicating it has been added to the cart.

**3. Checkout Workflow:**

* View Cart: The user reviews their cart and the total cost of selected items.
* Payment Method: The user selects a payment method (online or cash on delivery).
* Payment Processing: The system processes the payment (either online payment or marking it as cash on delivery).
* Order Confirmation: The order is completed, and the cart is cleared.

These workflows help ensure that the user has a smooth and intuitive shopping experience. Each step of the process is handled efficiently by the respective modules within the system.

**Github Link Of Our Project:** [**Next\_Wave\_Shopping**](https://github.com/LamiyaAlom/OOP_II_Project/blob/main/OOP_II_Project_NextWave_Shopping.py)

**3. Implementation and Results**

**Programming Language and Framework Used**

The Next Wave Shopping System is implemented using Python, a versatile programming language that supports object-oriented programming (OOP) principles. Python is chosen for its simplicity, readability, and ease of integration with various modules. The system is a command-line-based application, which eliminates the need for complex front-end frameworks and focuses on backend logic and functionalities.

**Code Walkthrough**

1. **Authentication System:** The authentication system is implemented using a class called Authentication. This class handles user registration, login, and password recovery through security questions. It securely manages user credentials, ensuring only registered users can log in.

**Key methods include:**

* + register(email, password, security\_question, security\_answer)
  + login(email, password)
  + recover\_password(email)

1. Product Inventory Management: The system uses the Product class as a base class for all products, with the PhysicalProduct class extending it to represent physical items. The NextWaveShopping class manages the product inventory, allowing the addition of new products, displaying products, and searching for products by name or brand.

**Key functionalities:**

* + Adding products to the inventory
  + Viewing available products
  + Searching for products by keywords

1. Cart and Checkout Mechanisms: The shopping cart functionality is managed by the NextWaveShopping class. Users can add products to their cart, view their cart, and proceed to checkout. The system handles payment processing, which includes both online payment (via bKash, Nagad, or Rocket) and cash on delivery options.

**Key methods:**

* + add\_to\_cart(user\_email, product\_index)
  + view\_cart(user\_email)
  + checkout(user\_email)

**Integration of Search Functionality**

The system integrates a search feature within the NextWaveShopping class, allowing users to search for products by their name or brand. The search functionality uses case-insensitive matching to help users find relevant products quickly.

**Error Handling and Validation**

**Error handling is implemented at various levels:**

* Invalid email or password during login
* Invalid product selection during cart addition
* Out-of-stock products during checkout
* Invalid choices during payment method selection

Input validation ensures that users provide valid data (e.g., non-empty strings, valid email format, etc.).

**User Interface (Command-Line Interaction)**

The system is designed to work through a command-line interface (CLI), where users interact by entering numeric choices. The interface guides users through authentication, product browsing, cart management, and checkout.

**Example Scenarios and Outputs**

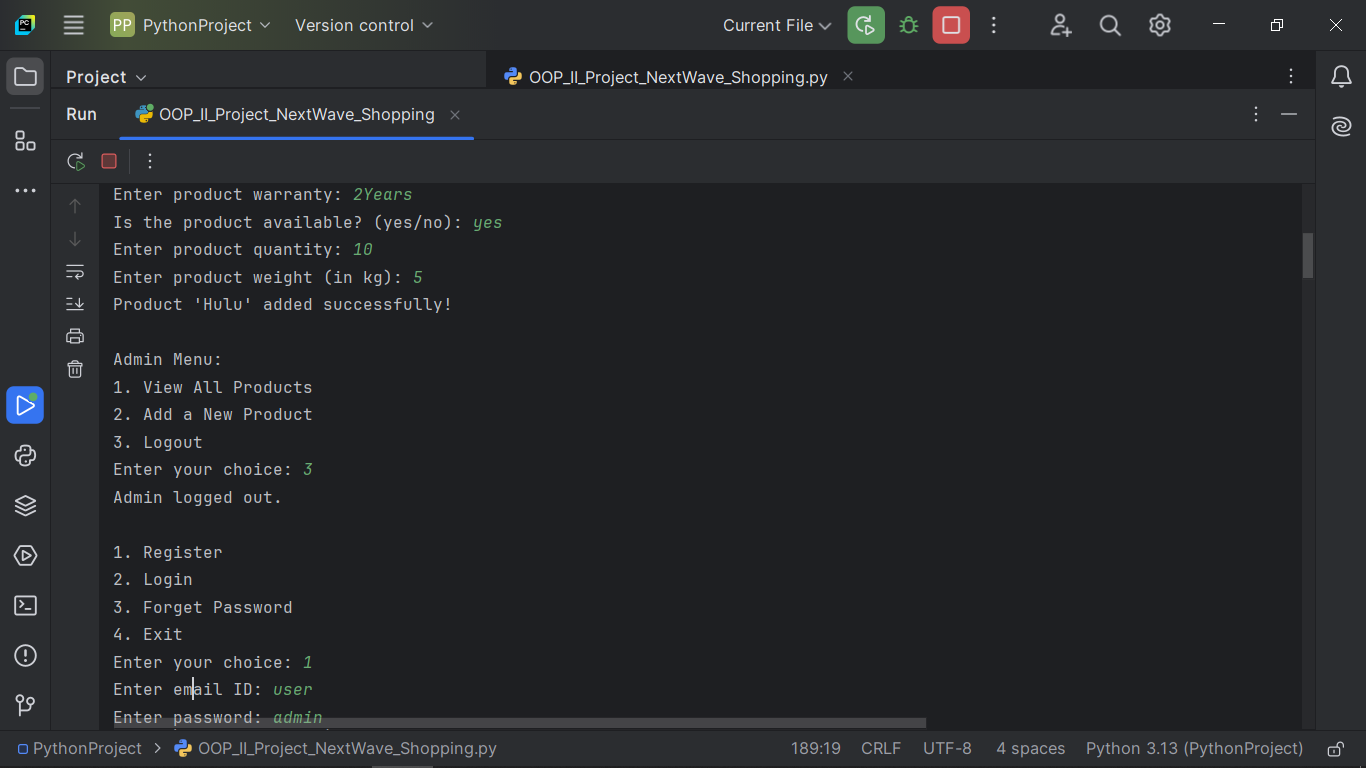
* User Login: A user successfully logs in with the correct credentials.
* Add to Cart: A user selects a product, adds it to their cart, and the system confirms the addition.
* Checkout: A user proceeds to checkout, selects a payment method, and the system processes the order.

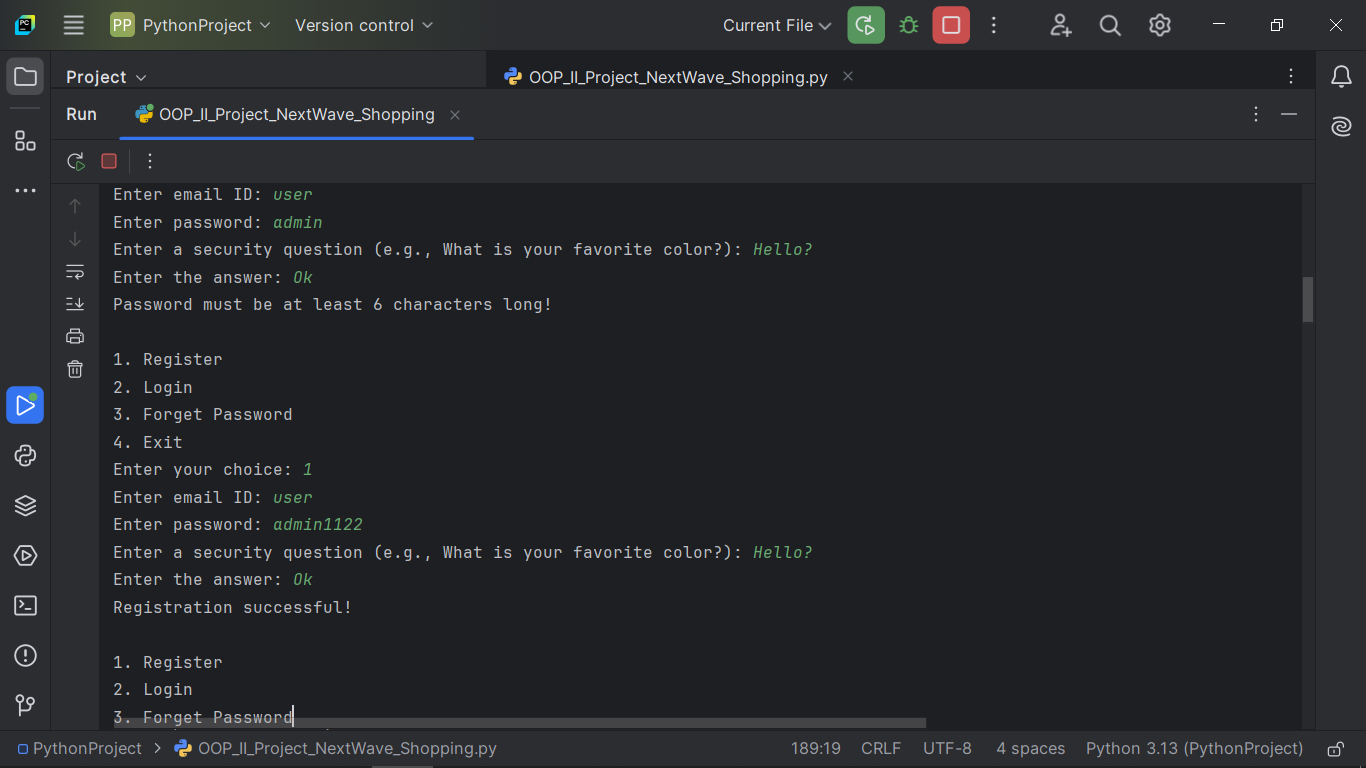
**Example Of Outputs:**

**A screenshot of a computer

Description automatically generated**

**Default for admin:**

****

**For Users:**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

**Payments:**

**A screenshot of a computer

Description automatically generated**

**4. Engineering Standards and Mapping**

**Object-Oriented Design Principles (SOLID)**

**The system follows SOLID principles:**

* Single Responsibility: Each class has one responsibility (e.g., the Authentication class handles login, while the NextWaveShopping class manages shopping processes).
* Open/Closed: New features (e.g., different product types) can be added without modifying existing code.
* Liskov Substitution: The system can handle any product that inherits from the Product class.
* Interface Segregation: Classes are designed to interact with minimal dependencies, adhering to single responsibility principles.
* Dependency Inversion: Higher-level modules do not depend on lower-level modules; instead, both rely on abstractions.

**Adherence to Software Design Patterns**

The system uses design patterns such as Factory for product creation (via inheritance) and Singleton for the Authentication and NextWaveShopping classes to ensure only one instance of each exists during runtime.

**Usability and Accessibility**

The command-line interface (CLI) ensures that all operations are accessible through simple, text-based prompts. While the interface is basic, it offers clear instructions to guide the user.

**Security Measures in Password Management**

Passwords are stored securely in the system, with a check for minimum length during registration. Though no advanced encryption is used, security questions and answers add an additional layer for password recovery.

**Performance Considerations**

Performance is optimized for small to medium-sized inventories. Since the system uses a list-based approach for inventory management and cart handling, it performs well with relatively low numbers of users and products.

**5. Challenges and Limitations**

**Scalability for Large Inventories**

As the system stores products in a simple list, performance may degrade with large inventories. A more efficient database-based solution is recommended for scaling to large numbers of products.

**Limited User Interface**

The current user interface is text-based (CLI), which limits the overall user experience. A graphical user interface (GUI) would provide a better, more intuitive experience.

**Absence of Database Integration**

The system lacks a persistent database, making it difficult to store and retrieve large datasets or persist user data across sessions. Integrating a database (e.g., MySQL, PostgreSQL) would significantly improve the system’s scalability and data management.

**Future Scope for Improvement**

The system has potential for future enhancements:

* A GUI for improved user interaction.
* Database integration to store product and user data persistently.
* More advanced search and filtering options.
* Integration of a payment gateway for online payments.

**6. Conclusion**

**Summary of Features**

The Next Wave Shopping System includes user authentication, product management, shopping cart management, and a checkout system. It offers essential e-commerce features, such as adding products to a cart and processing payments.

**Overall Benefits of the System**

The system provides a simple and functional shopping experience, allowing users to browse products, manage their cart, and complete purchases with ease. Its modular architecture makes it easy to extend and maintain.

**Insights Gained During Development**

During development, key insights were gained in implementing object-oriented design principles effectively and ensuring clear separation of concerns between authentication, shopping, and payment functionalities.

**7. Future Work**

**Integration of Graphical User Interface (GUI)**

A graphical interface would make the system more user-friendly and accessible to non-technical users.

**Adding Payment Gateway APIs**

Integrating payment gateways (like PayPal, Stripe, etc.) will streamline online payments.

**Extending the System to Support Multiple Categories**

Adding support for different product categories (electronics, clothing, etc.) will help users filter and find products more easily.

**Real-Time Inventory Updates via Database Integration**

Real-time inventory updates through database integration would allow more efficient product stock management.

**References**

1. Python Software Foundation. (2024). *Python Documentation*. Retrieved from <https://docs.python.org/3/>
2. Gamma, E., et al. (1994). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
3. GeeksforGeeks. (2024). *OOP Concepts*. Retrieved from https://www.geeksforgeeks.org/oops-concept-in-java/
4. W3Schools. (2024). *Python Classes and Objects*. Retrieved from https://www.w3schools.com/python/python\_classes.asp
5. OWASP. (2024). *Top 10 Security Risks*. Retrieved from <https://owasp.org/>