Design and Implementation of an Online Shop Information System

1. General Information

Project Title: Online Shop Information System
Author: Belek Alisherov (Index: 55879)
Course: Software Development Methodologies
Semester: 5
Supervisor: Marcin Kacprowicz
Submission Date: 05.02.26

---

2. Project Description

2.1 Project Objective

The objective of this project is to design and implement an Online Shop information system that enables users to browse products, manage a shopping cart, place orders, and simulate online payments. The system is developed using the Scrum software development methodology and demonstrates the application of software engineering principles, UML modeling, architectural patterns, design patterns, and REST-based services.

2.2 Project Scope

The system consists of the following components:

- User Management Module

- Product Management Module

- Shopping Cart Module

- Order Management Module

- Payment Module

- Admin Management Module

---

3. System Requirements

3.1 Functional Requirements

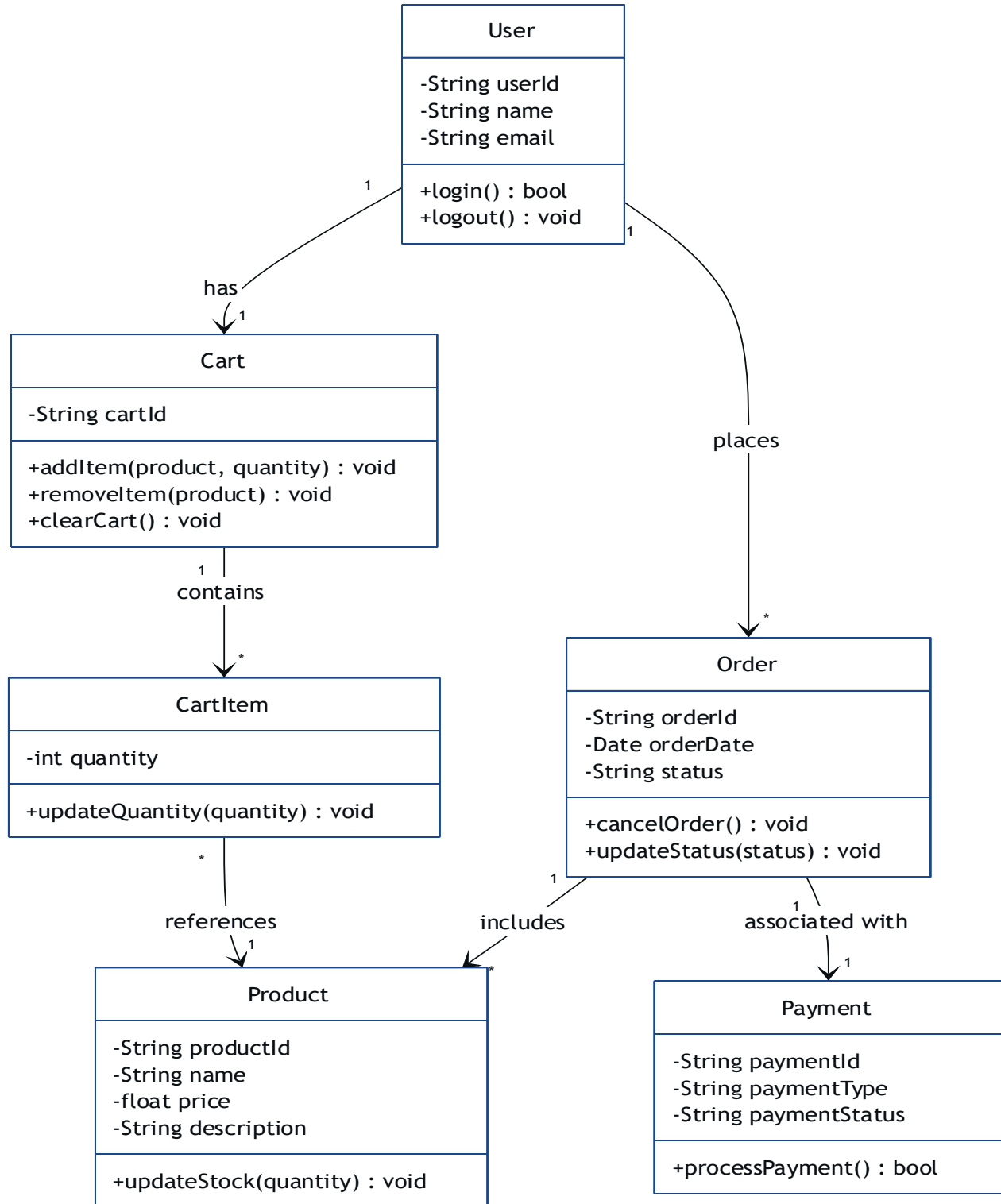| ID	Function Name	Description	Priority
F1	User Registration	Allows users to create an account	High
F2	User Login	Allows registered users to log in	High
F3	View Products	Users can browse available products	High
F4	Search Products	Users can search products by name/category	Medium
F5	Shopping Cart	Users can add/remove products from cart	High
F6	Place Order	Users can place orders	High
F7	View Orders	Users can view order history	Medium
F8	Admin Product Management	Admin can add, update, delete products	High

3.2 Non-Functional Requirements

ID	Requirement	Description	Category
N1	Performance	Response time ≤ 2 seconds	Efficiency
N2	Security	Encrypted passwords and secure access	Security
N3	Availability	System uptime ≥ 99%	Reliability
N4	Scalability	System supports increasing users	Scalability
N5	Usability	Easy-to-use interface	Usability

4. UML Diagrams

4.1 Use Case Diagram

The Use Case Diagram shows interactions between users/admins and the Online Shop system, clearly separating customer and administrator responsibilities.

4.2 Class Diagram

User

-String userId
-String name
-String email

+login() : bool
+logout() : void

has 1

Cart

-String cartId

+addItem(product, quantity) : void
+removeItem(product) : void
+clearCart() : void

1

contains

*

CartItem

-int quantity

+updateQuantity(quantity) : void

*

references

1

places

*

Order

-String orderId
-Date orderDate
-String status

+cancelOrder() : void
+updateStatus(status) : void

1

includes

*

1

associated with

1

Product

-String productId
-String name
-float price
-String description

+updateStock(quantity) : void

Payment

-String paymentId
-String paymentType
-String paymentStatus

+processPayment() : bool

5. Methodology Implementation

Selected Methodology: Scrum

The Scrum methodology was applied throughout the development process. The project was divided into multiple sprints:

- **Sprint 1:** Requirements analysis and system design

- **Sprint 2:** User and Product modules implementation

- **Sprint 3:** Cart, Order, and Payment modules

- **Sprint 4:** Testing, documentation, and final review

Each sprint included planning, development, testing, and review phases.

---

6. Design and Architectural Patterns

Design Patterns Used

- **Singleton Pattern:** Used for database connection management

- **Factory Pattern:** Used to create different payment methods

Architectural Pattern

- **MVC (Model-View-Controller):** Separates business logic, presentation, and control logic

---

7. Testing

Test ID	Description	Result
T1	User login unit test	Passed
T2	Product retrieval test	Passed
T3	Cart operations test	Passed
T4	Order creation test	Passed

Testing included unit testing and integration testing.

---

8. System Architecture

The system follows a layered MVC architecture and is divided into RESTful web services:

- User Service

- Product Service

- Order Service

- Payment Service

Each service communicates using HTTP and JSON.

---

9. Implementation Details

The system was implemented using modern web technologies:

- Backend: Java / Python / Node.js

- Frontend: HTML, CSS, JavaScript

- Database: MySQL

- API Style: REST

---

10. Conclusions and Future Work

The Online Shop system successfully meets the project objectives and demonstrates the use of Scrum methodology, UML modeling, and design patterns. Future improvements may include real payment gateway integration, mobile application support, and advanced security features.

---

11. References

- Software Engineering documentation

- REST API design guidelines

- Course materials