Student: Luminita Marghescu
Group: 30234

Table of Contents

1. Requirements Analysis

1.1 Assignment Specification

The C# application implements an order manager of a furniture manufacturer. It has two types of users (a regular user represented by the order manager and an administrator user) which have to provide a username and a password in order to use the application.

1.2 Functional Requirements

The regular user can perform the following operations:

- Add/update/view order information (customer, shipping address, identification number, delivery date, status.).
- Create/update/delete/view product information (title, description, color, size, price, stock etc).
- Add products to order and update order value and stock accordingly.

The administrator user can perform the following operations:

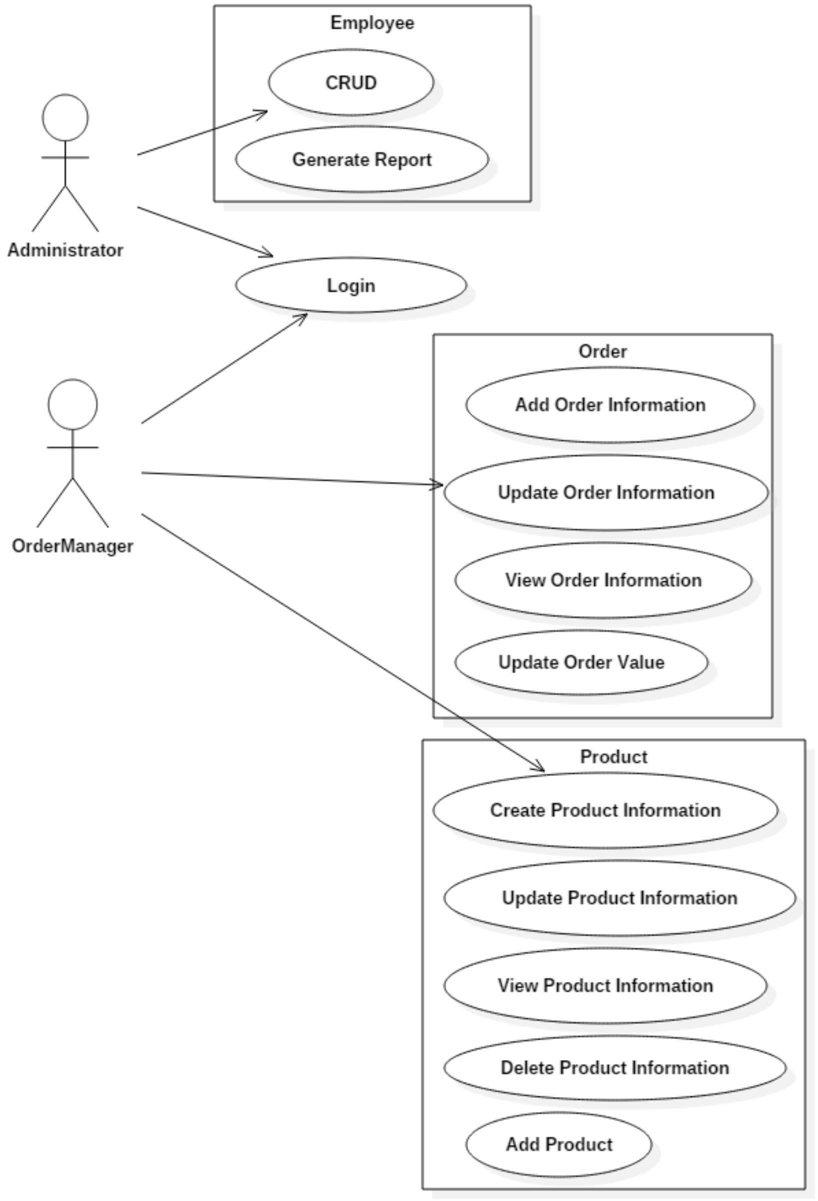
- CRUD on employees' information (Create, Read, Update and Delete operations).
- Generate reports for a particular period containing the activities performed by an employee.

1.3 Non-functional Requirements

- Accessibility
- Availability
- Data back-up
- Efficiency
- Price
- Privacy
- Portability
- Response Time
- Safety
- Security
- Testability

2. Use-Case Model

Use case: <Add Product>
Level: <sub-function>
Primary actor: <Order Manager>
Main success scenario: <Login as Order Manager, Add Product>
Extensions: <Scenario of failure: Logic as Administrator>



3. System Architectural Design

3.1 Architectural Pattern Description

[Describe briefly the used architectural patterns.]

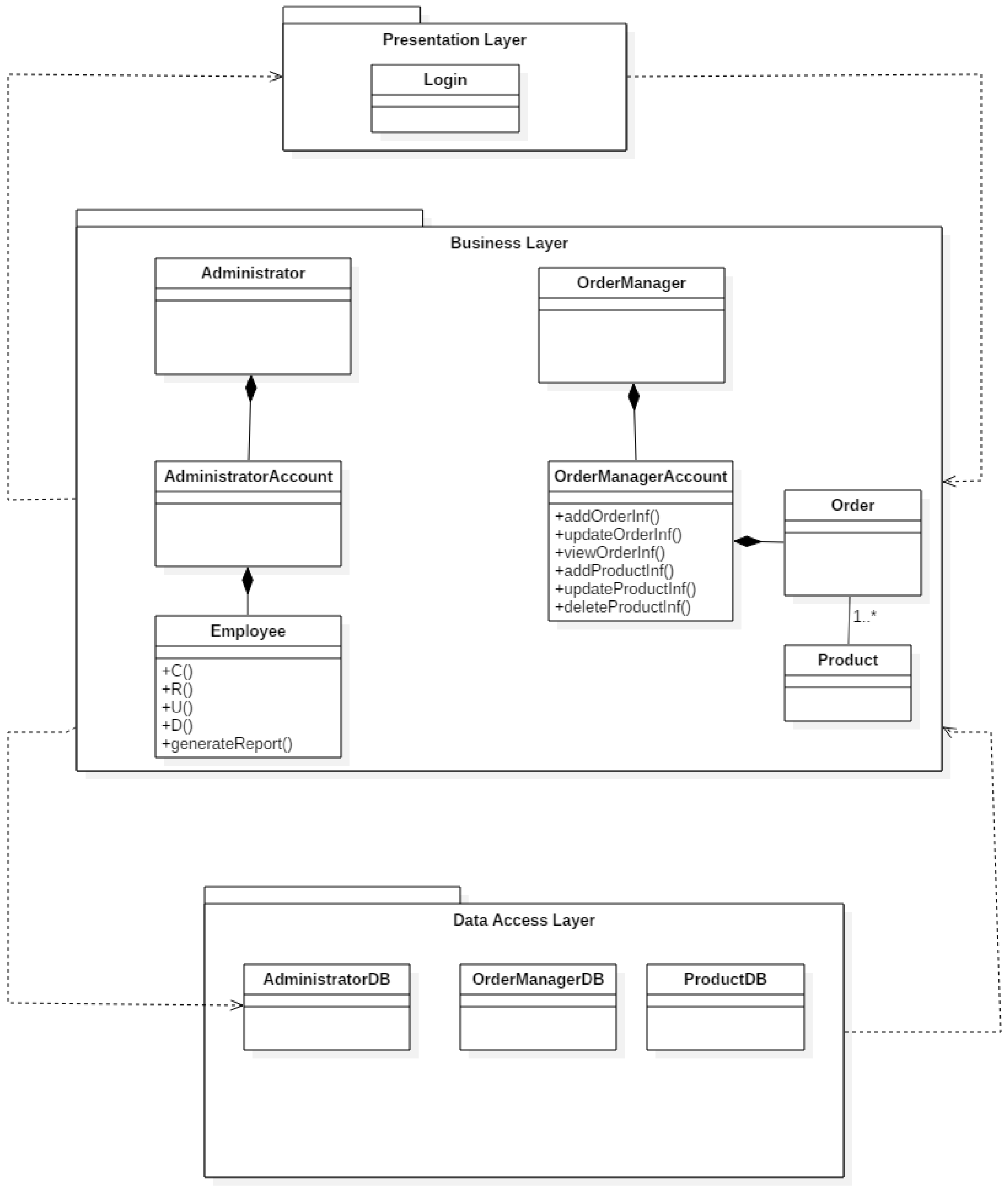
3.2 Diagrams

[Create the system’s conceptual architecture; use architectural patterns and describe how they are applied. Create package, component and deployment diagrams]

4. UML Sequence Diagrams

[Create a sequence diagram for a relevant scenario.]

5. Class Design



5.1 Design Patterns Description

[Describe briefly the used design patterns.]

5.2 UML Class Diagram

[Create the UML Class Diagram and highlight and motivate how the design patterns are used.]

6. Data Model

[Present the data models used in the system’s implementation.]

7. System Testing

[Present the used testing strategies (unit testing, integration testing, validation testing) and testing methods (data-flow, partitioning, boundary analysis, etc.).]

8. Bibliography