Student:Roman Tudor

**Group:30234**

Table of Contents

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

1. Requirements Analysis

# Assignment Specification

Use Swing/C# API to design and implement an application for the order managers of a furniture manufacturer. The application should have 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.

# 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.
* Generate reports for a particular period containing the activities performed by an employee.

# Non-functional Requirements

To store data into a database.

To use the Layer architectural pattern

Login performed in a secured manner (the password cannot be stored in it’s original form, it needs to be crypted).

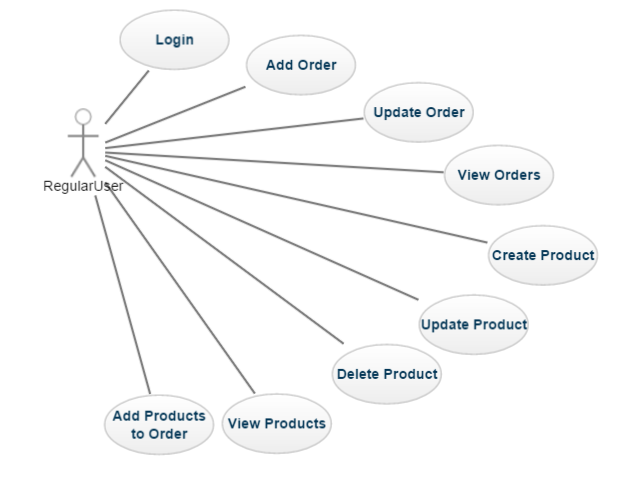
2. Use-Case Model

Use case: adding a product

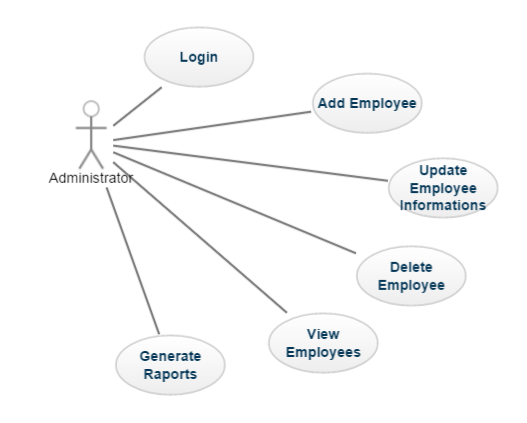
Level: user-goal level

Primary actor: regular user

Main success scenario: login (username and password required)-go to products tab – fill in the information about the product – click “Add Product” button

Extensions: failure : the product already exists in the database

[Use case diagram for a regular user]

[Use case diagram for an administrator user]

3. System Architectural Design

**3.1 Architectural Pattern Description**

Three-tier architecture is a client–server software architecture pattern in which the user interface (presentation), functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms.

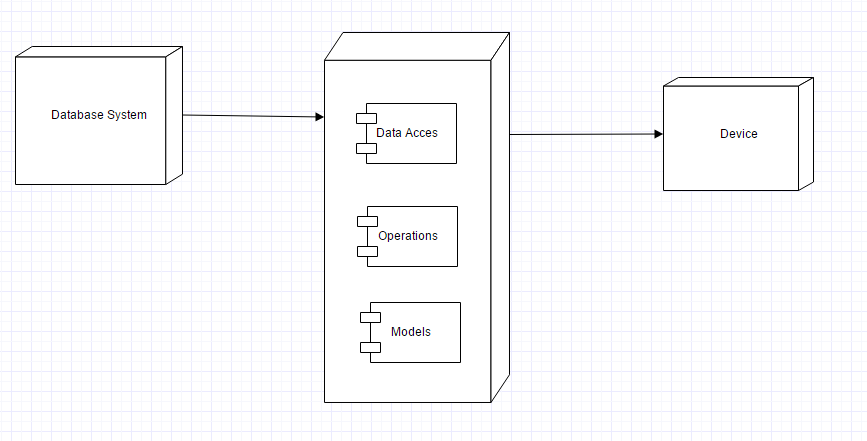
**3.2 Diagrams**

**3.2.1. Conceptual Diagram**



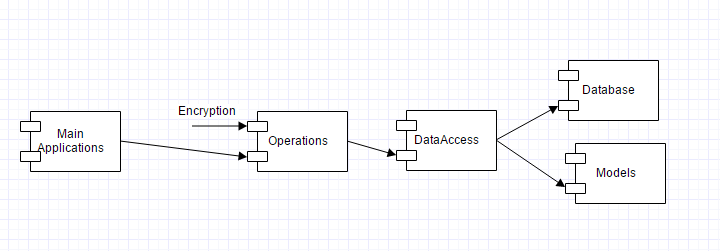
[3-Layer architecture conceptual diagram]

**3.2.2 Package Diagram**

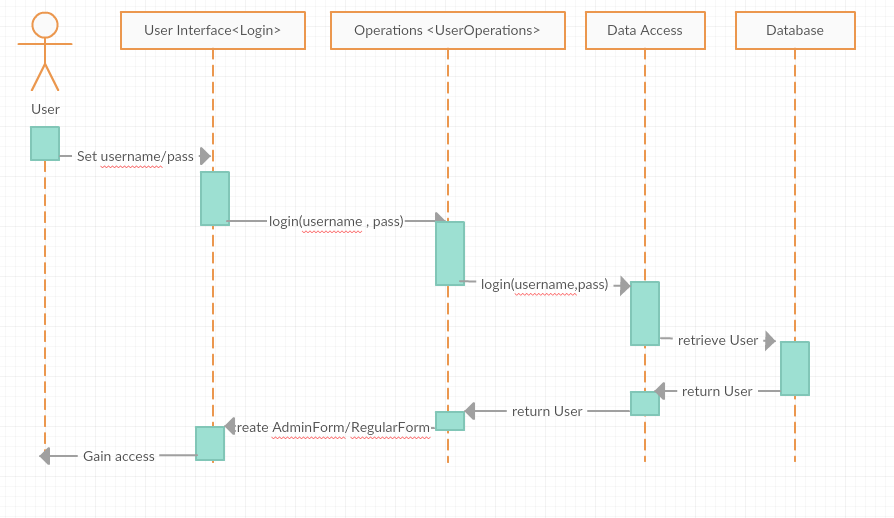
3.2.3 Deployment Diagram

[Deployment Diagram]

3.2.4 Component Diagram



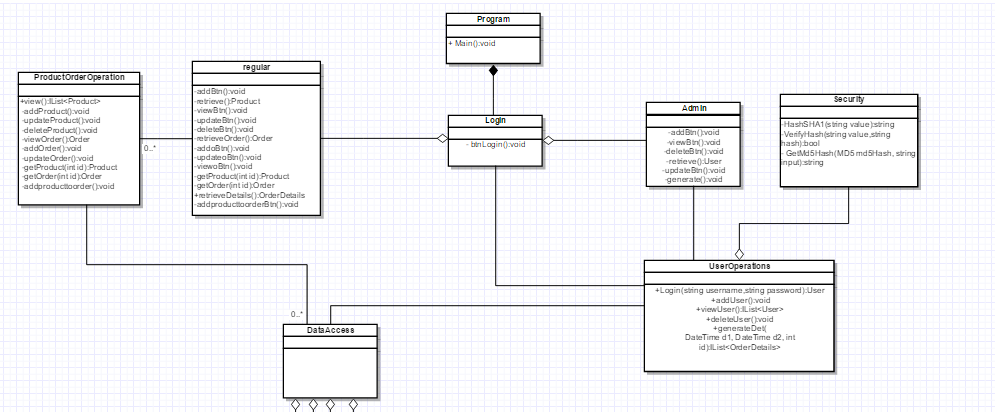
[Component Diagram]

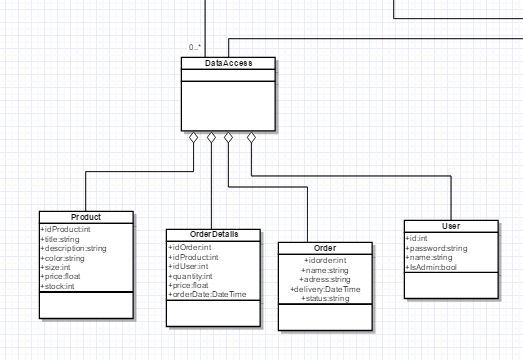
4. UML Sequence Diagrams

5. Class Design

**5.1 Design Patterns Description**

A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is a software design pattern and a well-established software architecture

**5.2 UML Class Diagram**

****

[UML Class Diagram]

6. Data Model

As data models I used in my implementation the following:

* Product
* Order
* User
* Order Details

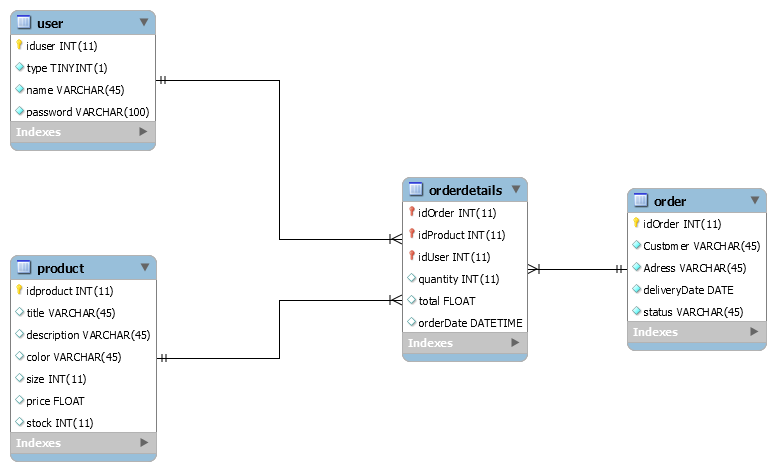
These represents the four classes that I used to represent real world objects.

The products have attributes like: title ,description, identification number, color, size, price and number in stock.

The orders have attributes like: identification number, customer information, address for delivery, delivery date and status.

The users have basic attributes like username, password, identification number and type.

For even more information about orders there are Order Details objects, which have attributes like order identification number, product identification number, total, user identification number and order date.

**

7. System Testing

Unit testing was the main method used for testing the system. Using assertion database operations and method based on data that were retrieved from the database were tested.

Unit testing is based on black box testing, basically the tester chooses an input for a specific method, knowing the expecting output, and with an assert operation tests if the output is satisfying.

8. Bibliography

<https://www.techopedia.com/definition/24649/three-tier-architecture>

<https://en.wikipedia.org/wiki/Multitier_architecture>

<http://exponential.io/blog/2015/03/05/3-layer-architecture-in-detail/>