<Assignment Name>

Analysis and Design Document

Student:

**Group:**

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

1. **Objective**

The objective of this assignment is to allow students to become familiar with the Model View Controller architectural pattern and the Factory Method design pattern.

1. **Application Description**

Use Java/C# API to design and implement an application for the employees of a book store. The application should have two types of users (a regular user represented by the book store employee and an administrator user) which have to provide a username and a password in order to use the application.

The regular user can perform the following operations:

* Search books by genre, title, author.
* Sell books.

The administrator can perform the following operations:

* CRUD on books (book information: title, author, genre, quantity, and price).
* CRUD on regular users’ information.
* Generate two types of reports files, one in pdf format and one in csv format, with the books out of stock.

1. **Application Constraints**

* The information about users, books and selling will be stored in multiple XML files. Use the Model View Controller in designing the application. Use the Factory Method design pattern for generating the reports.
* All the inputs of the application will be validated against invalid data before submitting the data and saving it.

# Functional Requirements

As inputs we have all kinds of information about the users or books. These inputs are used in order to obtain our outputs, like: show the user/book information in a processed way. These information is saved in xml files and their procession takes place in the business logic part of the system (which is contained in the model).

# Non-functional Requirements

The criteria used to judge the operations made by this system are the following:

- Separate implementation from interface

- User friendly interface

- MVC application (separate implementation of view from model. The two are bound through the controller)

2. Use-Case Model

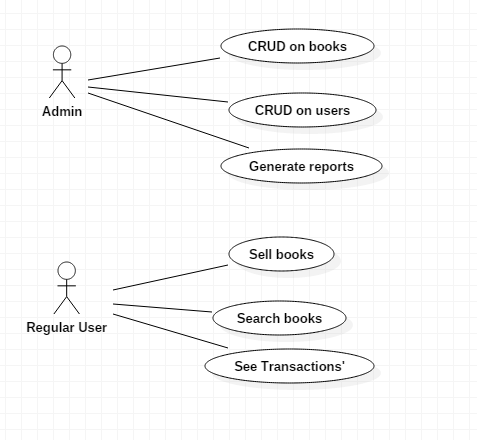
*Use case:* user log in process

*Level:* user-goal level

*Primary actor:* user who wants to sign in

*Main success scenario:* user successfully logs into account

*Extensions:* user enters wrong username or password, wrong input error



3. System Architectural Design

**3.1 Architectural Pattern Description**

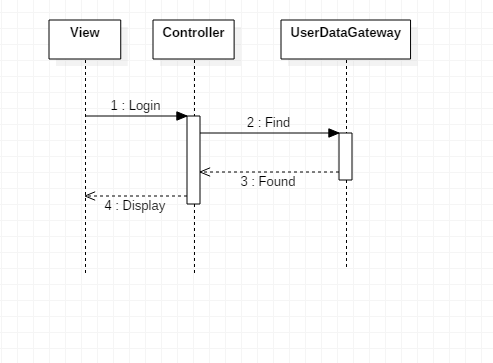
Model–View–Controller (MVC) is a software architectural pattern for implementing user interfaces on computers. It divides a given application into three interconnected parts in order to separate internal representations of information from the ways that information is presented to and accepted from the user. The MVC design pattern decouples these major components allowing for efficient code reuse and parallel development.

**3.2 Diagrams**



4. UML Sequence Diagrams

User log in:



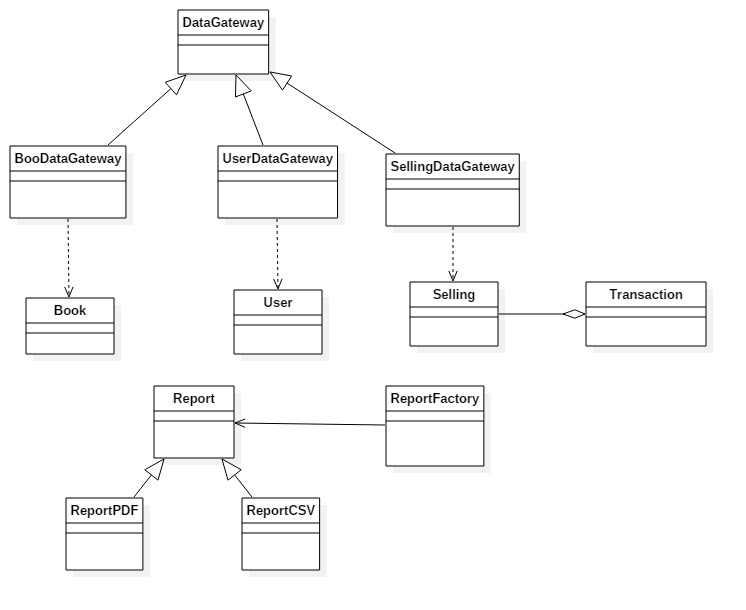
5. Class Design

**5.1 Design Patterns Description**

Factory pattern is one of the most used design patterns. This type of design pattern comes under creational pattern as this pattern provides one of the best ways to create an object.

In Factory pattern, we create object without exposing the creation logic to the client and refer to newly created object using a common interface.

**5.2 UML Class Diagram**

**

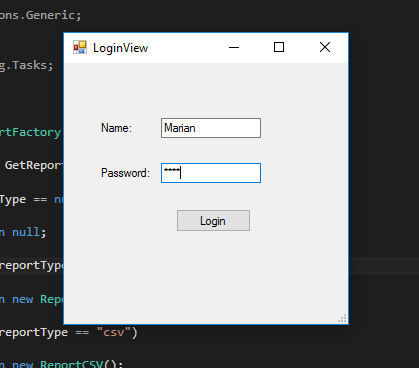
6. Data Model

The data model used in the application contains the classes above (both the objects that we works with and the data access ones). The model also contains the implementation of the Factory Design Pattern which is used to generate reports. The controller uses this model.

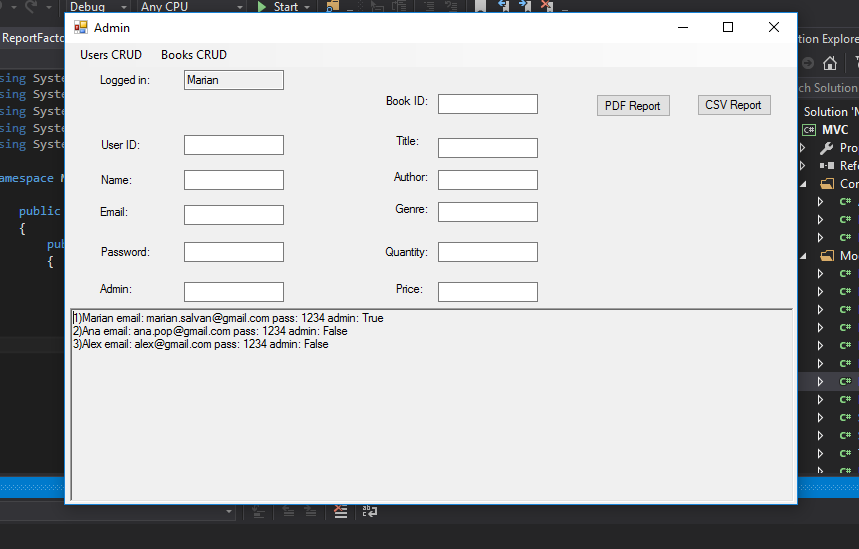
7. System Testing

The data input is tested when introduced in order to be valid.

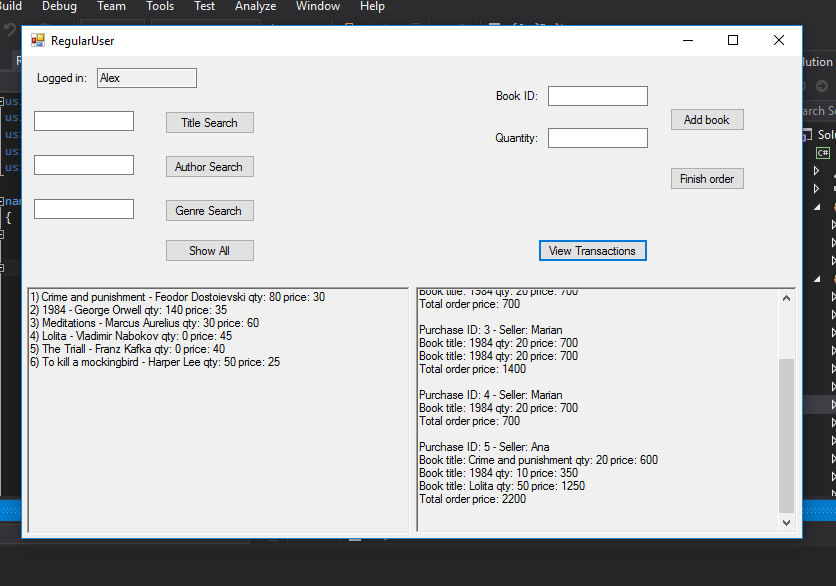
Login view:



Admin View:



Regular user view:



8. Bibliography

<https://msdn.microsoft.com/en-us/library/54xbah2z(v=vs.110).aspx>

<https://msdn.microsoft.com/en-us/library/e80y5yhx(v=vs.110).aspx>

<http://msdn.microsoft.com/en-us/library/system.xml.xmlreader.aspx>

<http://msdn.microsoft.com/en-us/library/system.xml.xmlwriter.aspx>

<http://msdn.microsoft.com/en-us/library/ms764730(VS.85).aspx>