Student:Iancu Daiana-Sorana

**Group:30431**

Assignment 1

Documentation

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

The project represents a simulation of a ticket selling application for the Nationat Theater of Cluj-Napoca. The application is supposed to be used by the administrator to manage the shows, cashiers and reservations and by the cashiers to sell tickets.

# Functional Requirements

There are two users:

* administrator, which will be able to perform:

CRUD on cashiers’ information

CRUD on shows’ information

Export to .csv file the already sold tickets.

* Cashier, who will be able to:

Sell tickets to a show

Find out if the number of tickets per show were exceeded

See the tickets sold for a show

Cancel a reservation

Edit a seat

# Non-functional Requirements

The information is to be kept in a database.

The users must log in before using the application.

The passwords of the users are encrypted with a one-way encryption algorithm.

There are different layers for: accessing the data, logic and presentation.

2. Use-Case Model

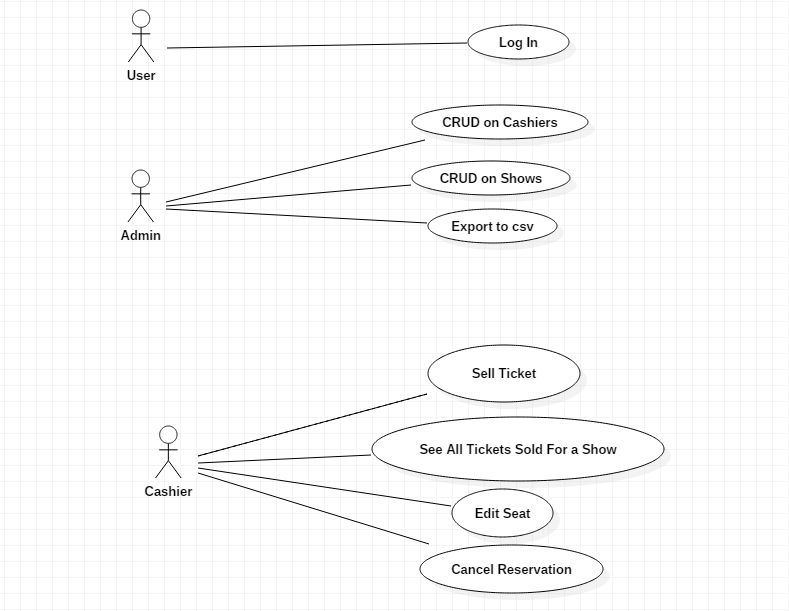
Use case: Sell ticket

Level: user-goal level

Primary actor: cashier

Main success scenario: a ticket is sold

Extensions: no ticket is sold

**

3. System Architectural Design

**3.1 Architectural Pattern Description**

The architectural pattern used is Layered Architecture.

This pattern can be used to create projects that can be decomposed into tasks. Each layer is at a different level of abstraction, and each of them provides services to the layer above, never to the one below.

The commonly used layers are:

Presentation Layer – this usually holds the GUIs

Business Layer – this one holds the services and the model

Data Access Layer – this one contains the classes which access and retrieve data necessary to the application.

This architectural pattern is mostly used in general desktop applications or e-commerce webapplications.

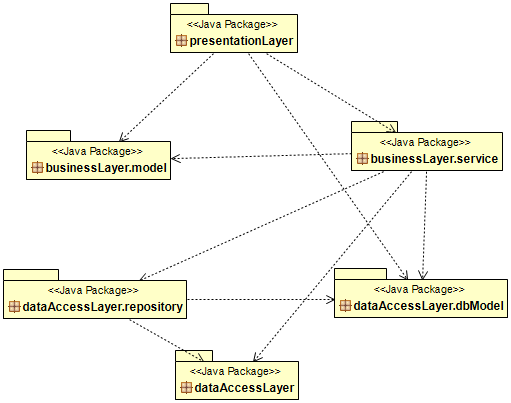
**3.2 Diagrams**

The program is, as explained above, divided into three main packages which represent layers: presentationLayer, businessLayer and dataAccessLayer.

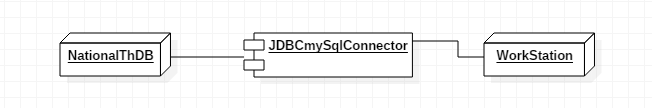
Inside the businessLayer and dataAccesLayer packages, there are more packages which separate the model at the given layer from the services/methods.

As it can be seen from the image, one layer depends only on the layer below, never on the layer above.

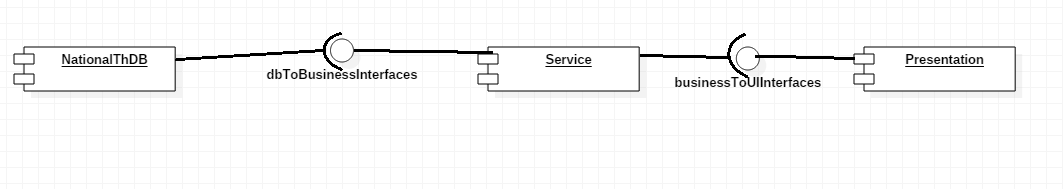
*Package Diagram*

**

*Deployment Diagram*



*Component Diagram*

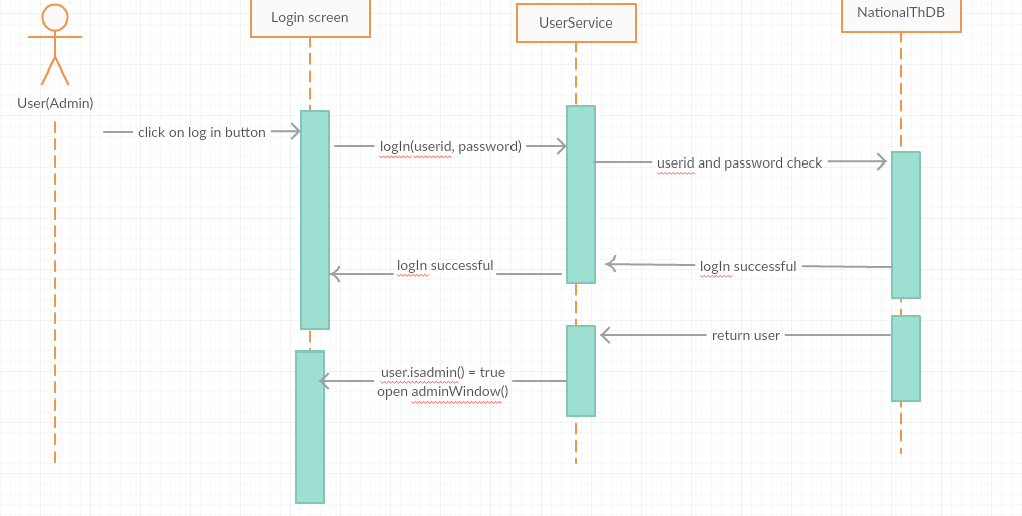


4. UML Sequence Diagrams

The sequence diagram presented in this section is for the case: Admin Log In.

The admin provides the password and user and presses log In, triggering the logIn function which will search for the user and password in the database. If they are found, they are returned together with the respective user.

The user.isAdmin function = true opens the AdminWindow().

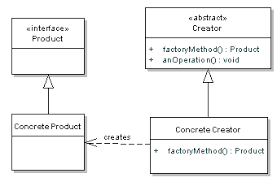
**

5. Class Design

**5.1 Design Patterns Description**

One of the design patterns that had to be used in this project is Factory Method.

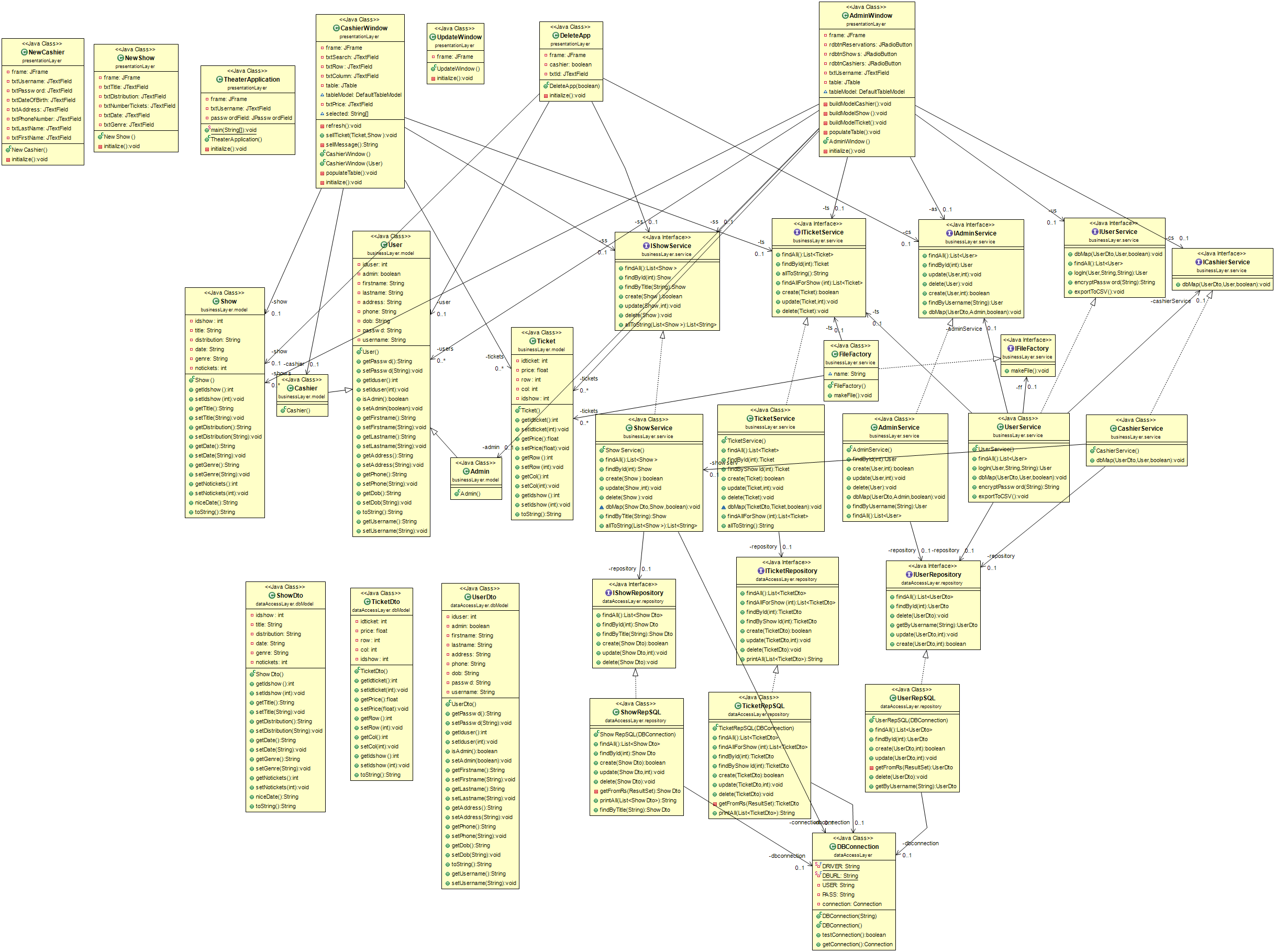
This method is a creational DP used to deal with the problem of creating objects without having to specify the exact class of the object that will be created.

**

**5.2 UML Class Diagram**

In the Class Diagram we can observe that the layers are well-defined:data access layer is at the base, connected through interfaces to the upper layer. The business layer accesses the data access layer through the interfaces provided by it. At the top, the presentation layer can be observed, accessing the business layer through the interfaces that it provides.

It is obvious by just looking at the direction of the dependencies, that the above layer uses the one beneath, never the other way around.

**

6. Data Model

The data model is represented by User, Ticket and Show. The classes UserDto, ShowDto and TicketDto are used to retrieve data from the database.

7. System Testing

The testing was done using a testing class called Test for the data access layer.

For the most important operations of the application, ticket selling and encryption of thepassword, the testing was done using Junit Test.

The test classes are called EncryptionTest and SellTicketTest. The first one checks if the encrypted string is equal to the result

8. Bibliography