Student: Vescan Catalin

**Group: 30235**

Table of Contents

[1. Requirements Analysis 3](#_Toc512562296)

[1.1 Assignment Specification 3](#_Toc512562297)

[1.2 Functional Requirements 3](#_Toc512562298)

[1.3 Non-functional Requirements 3](#_Toc512562299)

[2. Use-Case Model 4](#_Toc512562300)

[3. System Architectural Design 6](#_Toc512562301)

[3.1 Architectural Pattern Description 6](#_Toc512562302)

[3.2 Diagrams 7](#_Toc512562303)

[4. UML Sequence Diagrams 9](#_Toc512562304)

[5. Class Design 10](#_Toc512562305)

[5.1 Design Patterns Description 10](#_Toc512562306)

[5.2 UML Class Diagram 11](#_Toc512562307)

[6. Data Model 12](#_Toc512562308)

# 1. Requirements Analysis

## Assignment Specification

Use Java/C# API to design and implement an application for the employees of a pharmacy (chemists). The application should have two types of users (a regular user represented by the chemist 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 medication by name, ingredients, manufacturer.
* Sell medication.

The administrator can perform the following operations:

* + CRUD on medication (medication information: by name, ingredients, manufacturer, 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 medication out of stock.

## Functional Requirements

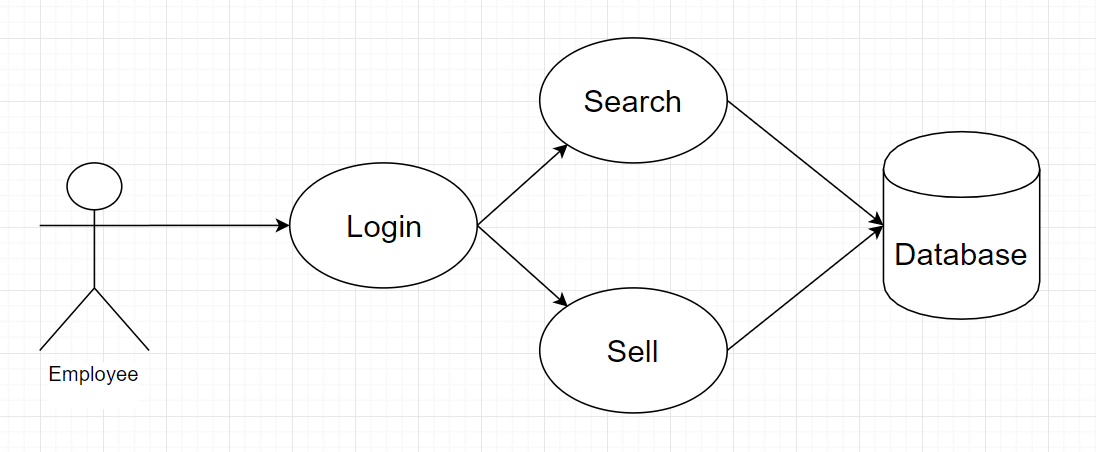
CRUD operations on medications and employees performed by administrator, and search and sell medication operations performed by employees.

## Non-functional Requirements

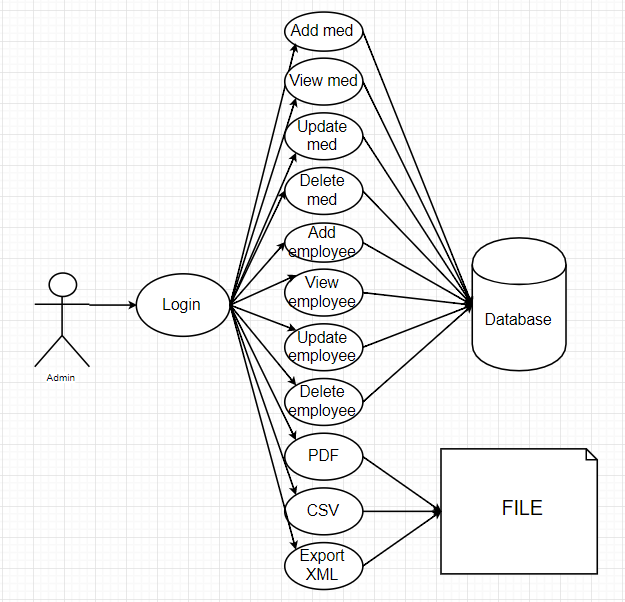
This application run only on desktop, on all operating system, return an answer in less than 1 second, with a back-up system who save the data at every 30 minutes, and this applications has 2 security application to protect the data.

# 2. Use-Case Model

Usecase employee



Usecase admin



An example to understand this usecase will be presented in next few rows:

In the login frame, the employee needs and username and a password for autentification, and he must introduce a valid username (only alphabetical character), otherwise he will see a warning message to prevent him, and a strongly password.

After login, the employee must add an medication name, ingredient, and manufacturer and he has two option: to search all medications with these characteristics from database, or to sell the medication with these characteristics.

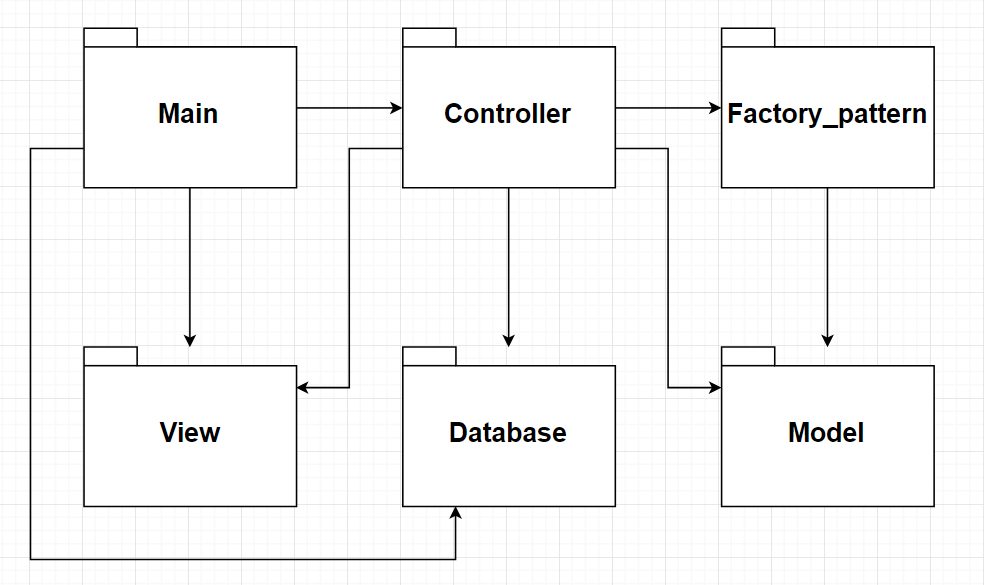
# 3. System Architectural Design

## 3.1 Architectural Pattern Description

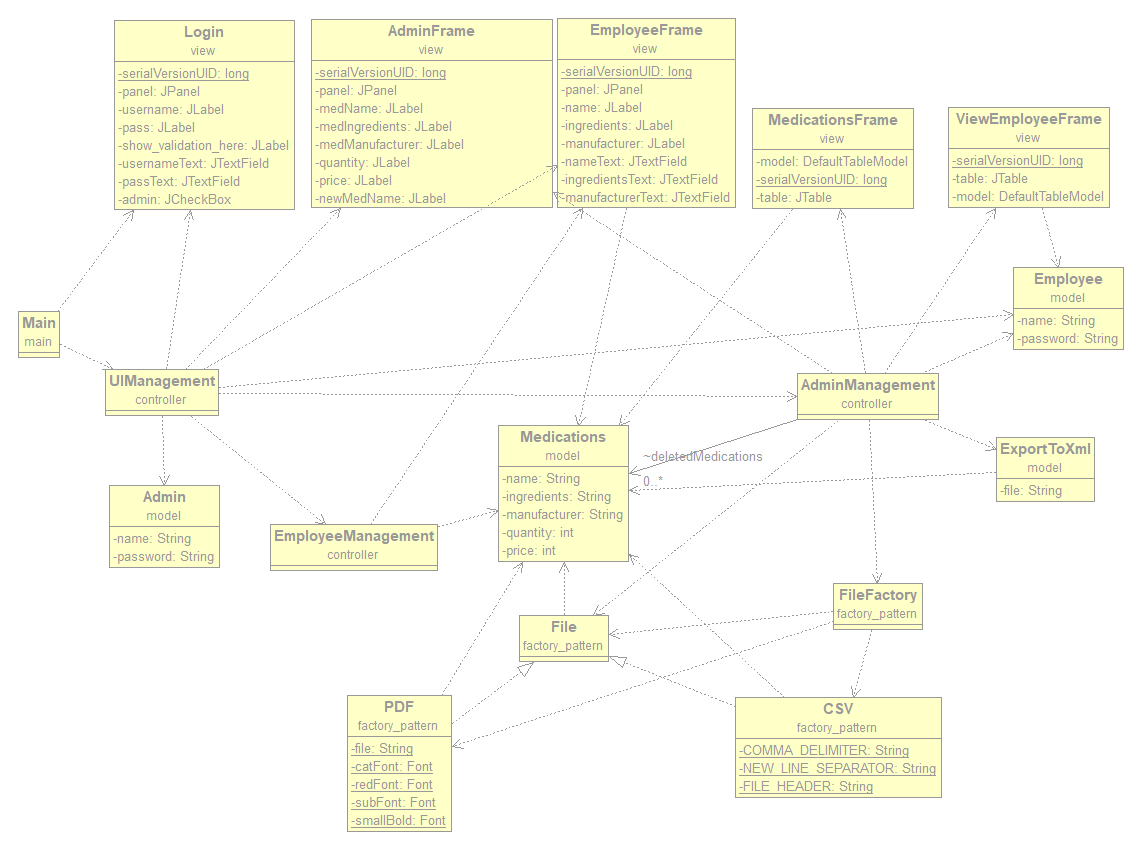
In this application I used model-view-controller (MVC) as architectural pattern which is divided in 3 parts, the model part where we have the classes that are mapped on database, the view part where we have the classes that represents the graphic user interface (GUI), and the last one, the controller part where we have the classes which make the connection between view, model and database.

## 3.2 Diagrams

Package diagram

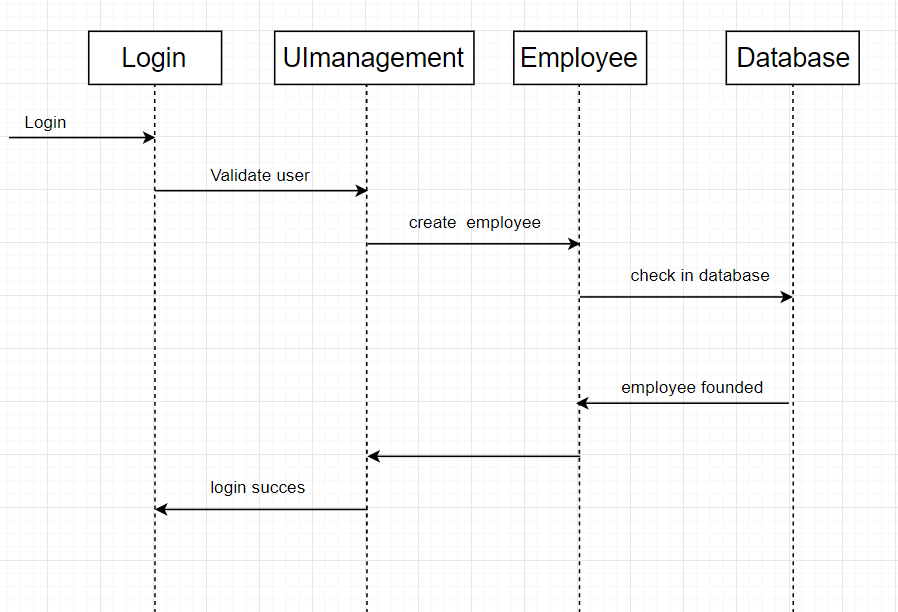


Class Diagram



# 4. **UML Sequence Diagrams**

Login sequence diagram

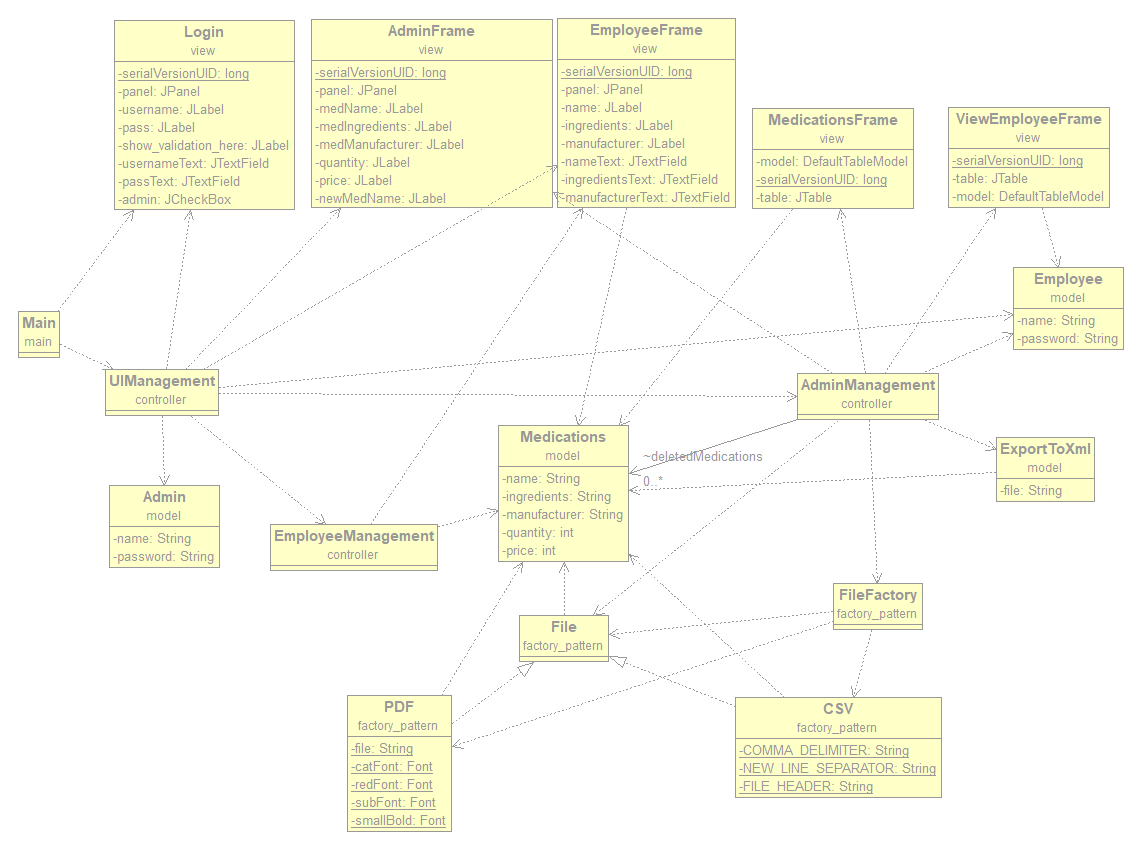


# 5. Class Design

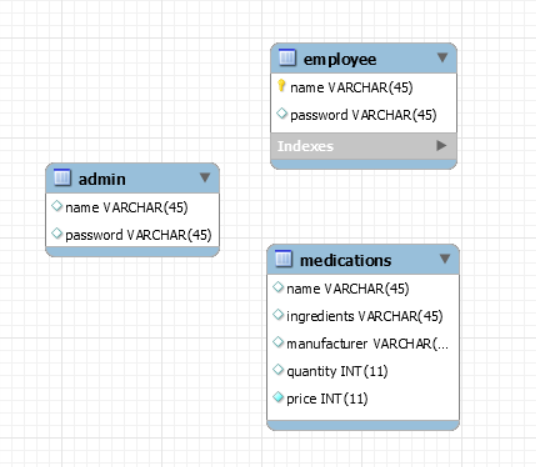
## 5.1 Design Patterns Description

I used the Factory Method Pattern as design pattern for generating the .pdf and .csv reports. This pattern use an interface that is implemented by all classes which may be a specifics file generators, in our case PDF and CSV, and a class which determine what kind of file are you choosen to generate.

## 5.2 UML Class Diagram



# 6. Data Model



7. **System Testing**

I used JunitTest to test the aplication, and i have no errors.

8. **Bibliography**

<https://ro.wikipedia.org/wiki/Model-view-controller>

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

<https://www.tutorialspoint.com/hibernate/index.htm>

<http://www.vogella.com/tutorials/JavaPDF/article.html>

<https://examples.javacodegeeks.com/core-java/writeread-csv-files-in-java-example/>