Student:Tomoiaga Anamaria

**Group:30235**

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 4

3. System Architectural Design 4

4. UML Sequence Diagrams 4

5. Class Design 5

6. Data Model 5

7. System Testing 5

8. Bibliography 5

1. Requirements Analysis

# Assignment Specification

Utilizati Java/C# API pentru a implementa o aplicatie pentru angajatiii unei farmacii. Aplicatia trebuie sa aiba doua tipuri de useri (un user regular reprezentat de un chemist si un user administrator) care vor folosi un username si o parola pentru a putea folosi aplicatia.

Userul poate realiza urmatoarele operatii:

* Cauta medicamente dupa nume, ingredient si producator
* Vinde medicamente

Administratorul poate realiza urmatoarele operatii:

* CRUD pe informatiile medicamentelor
* CRUD pe informatiile userilor
* Genera doua tipuri de rapoarte, unul in format pdf si unul in format csv, cu medicamentele care nu mai sunt in stoc

# Functional Requirements

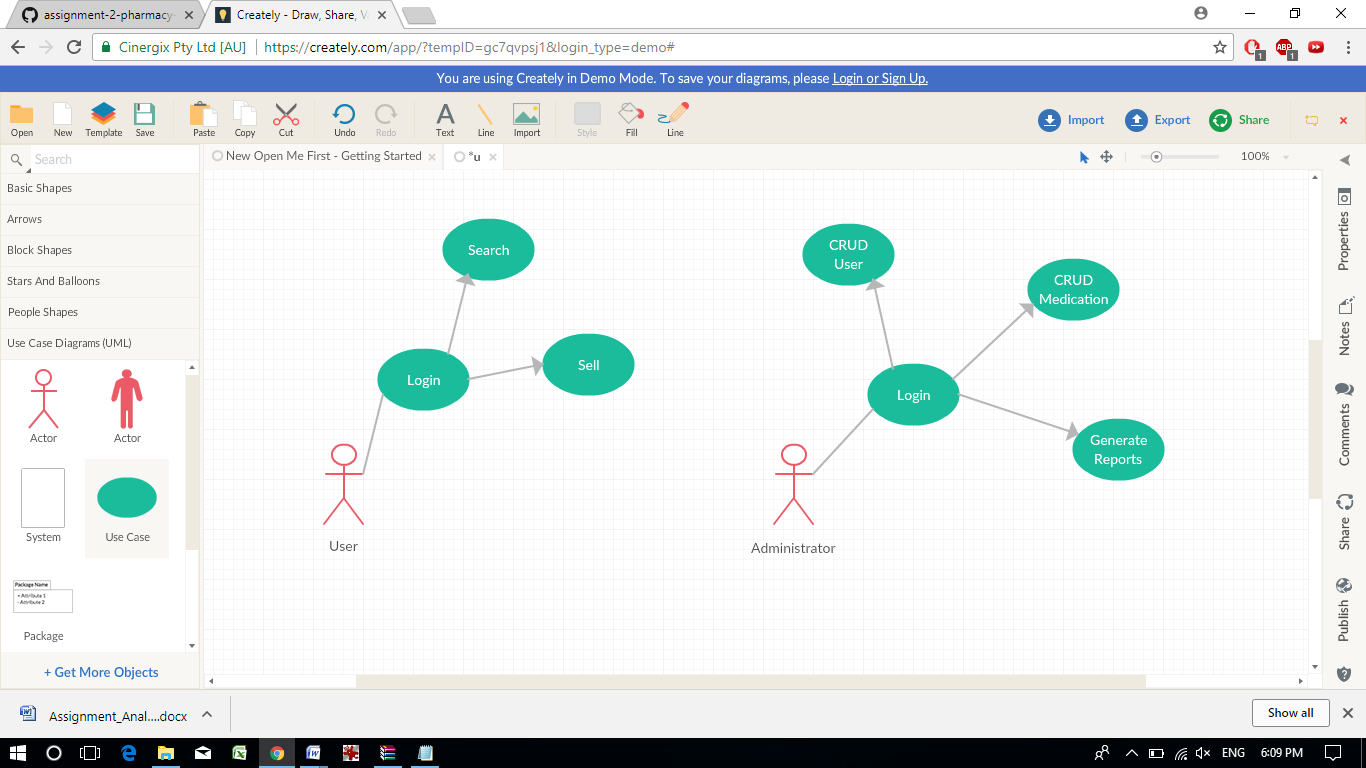
Administratorul nu se poate loga ca un user obisnuit.

# Non-functional Requirements

Securitate: sistemul este securizat prin cererea unui username si a unei parole pentru a putea utiliza aplicatia.

Performanta: sistemul trebuie sa raspunda repede la cererile userului.

2. Use-Case Model



Use case: <adaugare medicament>

Level: <user goal level>

Primary actor: <administrator>

Main success scenario: <Introducem toate informatiile despre medicament, care vor fi salvate in baza de date>

Extensions: <Daca nu introducem toate informatiile despre rezervare vom primi un mesaj de eroare sau daca introducem o cantitate negativa>

3. System Architectural Design

**3.1 Architectural Pattern Description**

Sistemul foloseste MVC (Model-View-Controller) ca model architectural, care imparte aplicatia in trei parti interconectate.

Model – este cel mai mic nivel si este responsabil pentru mentinerea datelor.

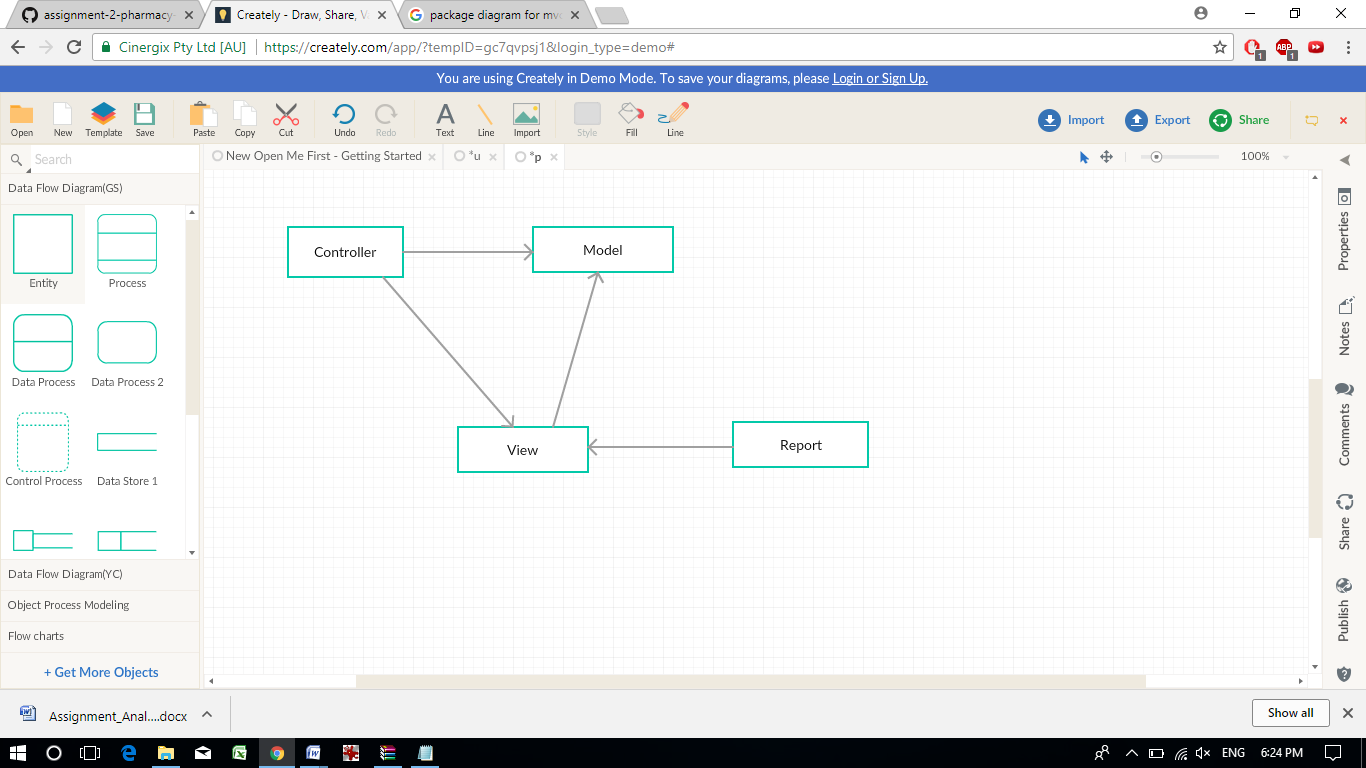
View – contine clasele care definesc interfata cu utilizatorul.

Controller – controleaza interactiunile dintre model si partea de vizualizare.

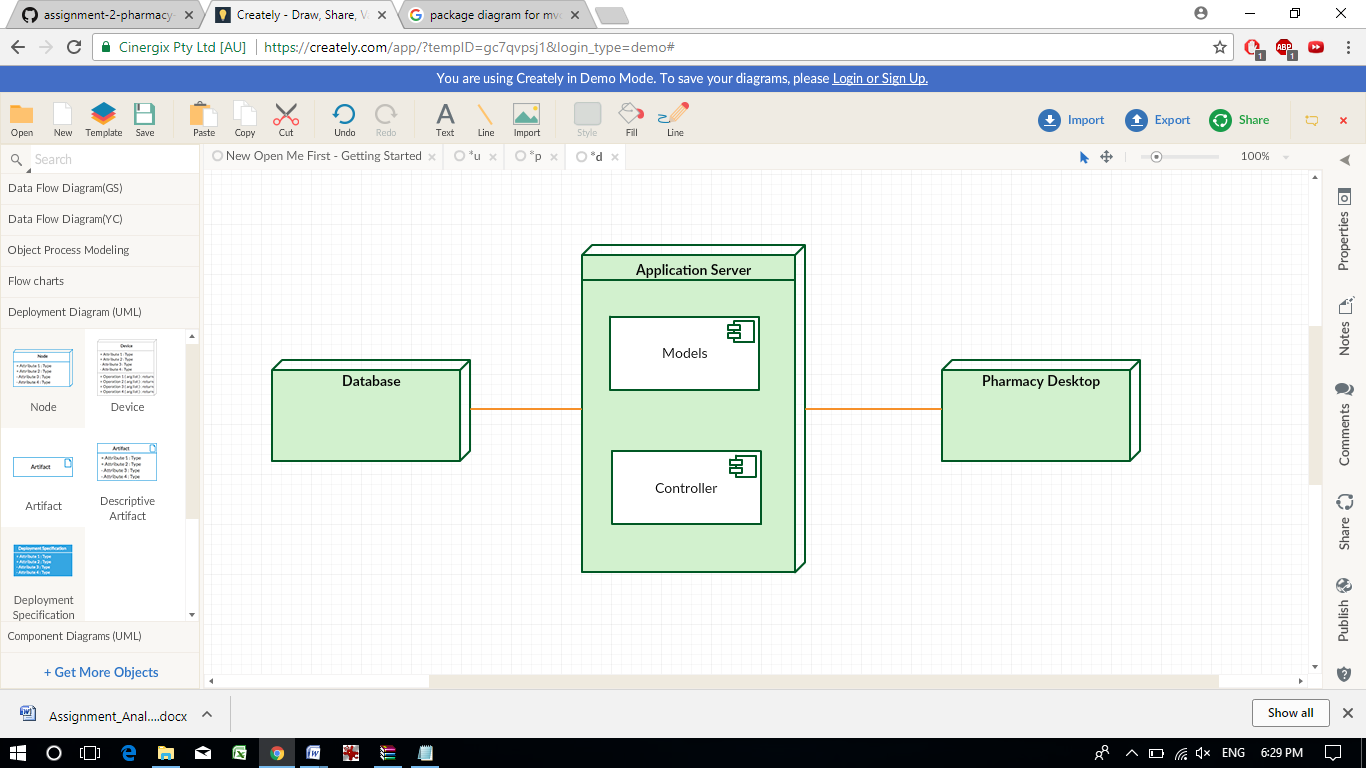
Am folosit Factory Pattern pentru a realiza generarea celor doua tipuri de rapoarte.

**3.2 Diagrams**

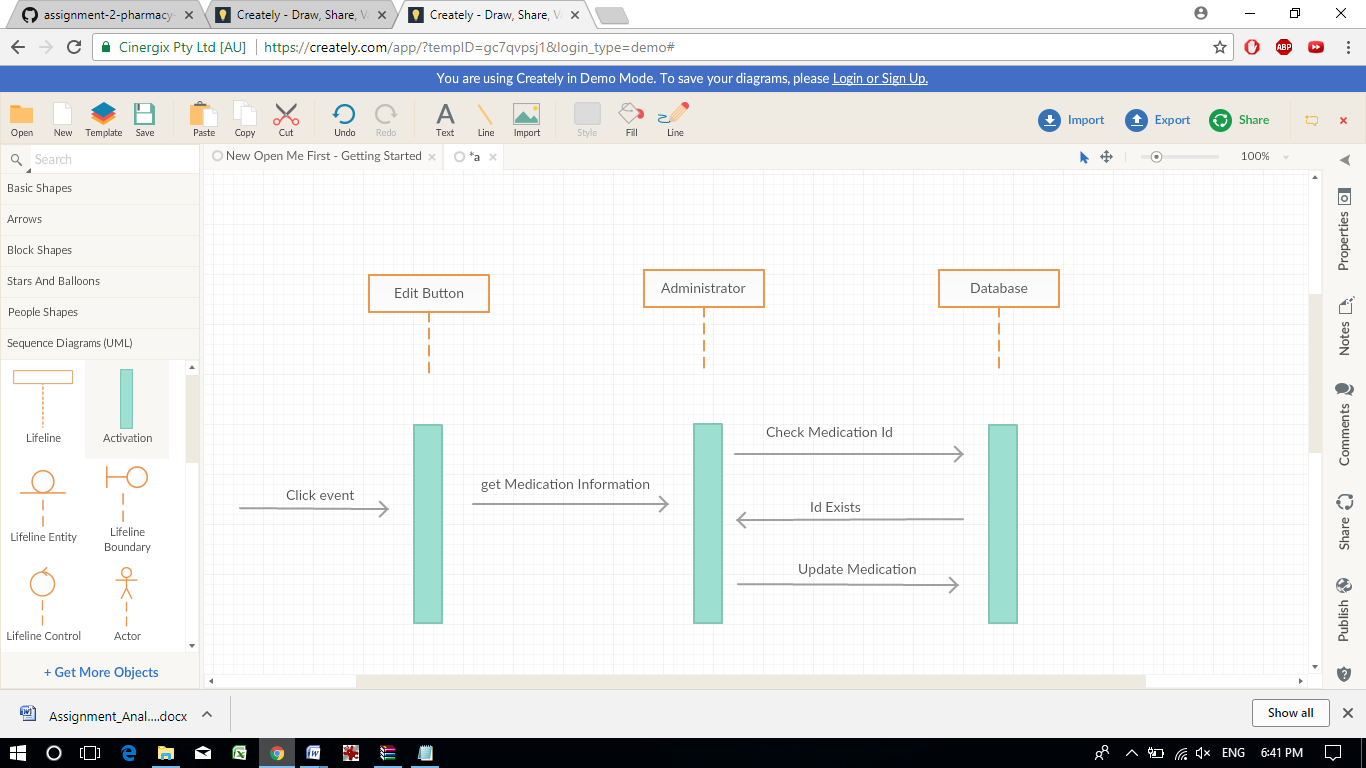
Package Diagram



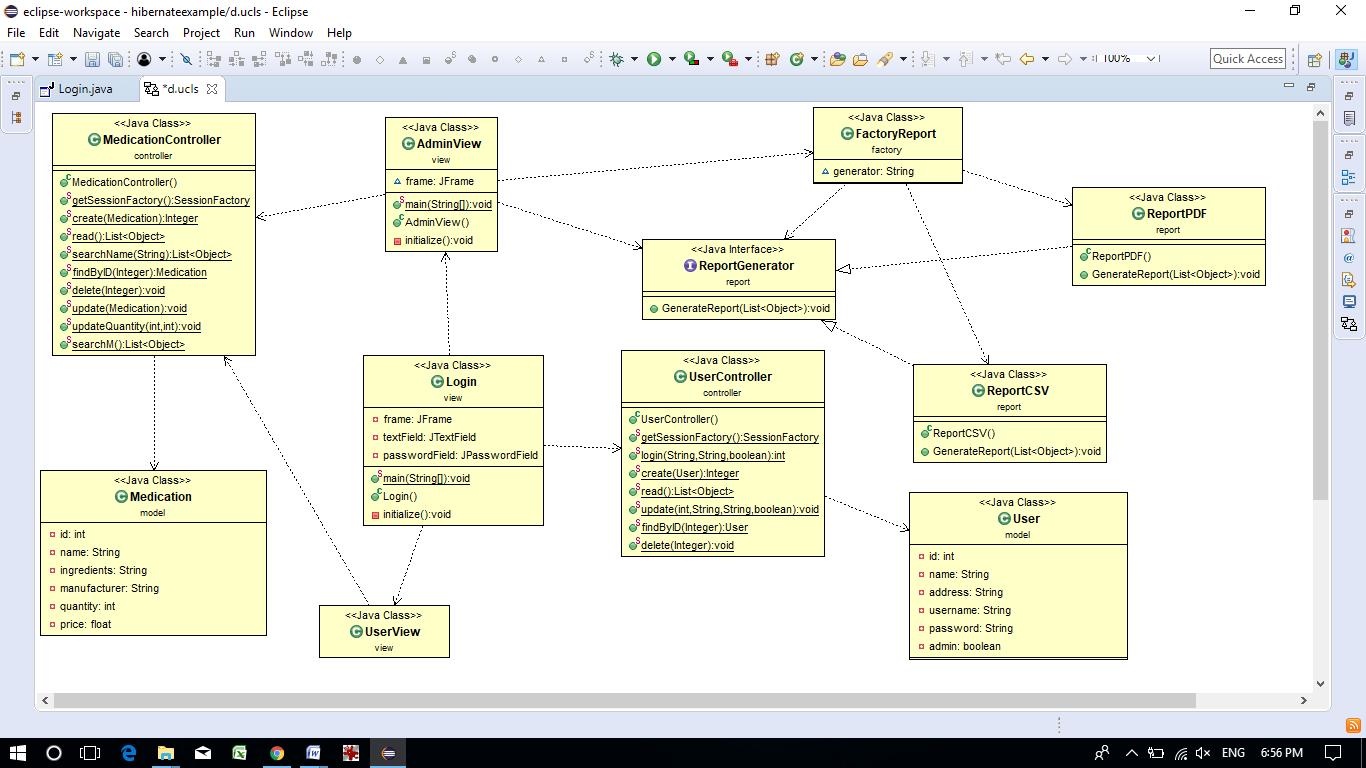
Deployment Diagram



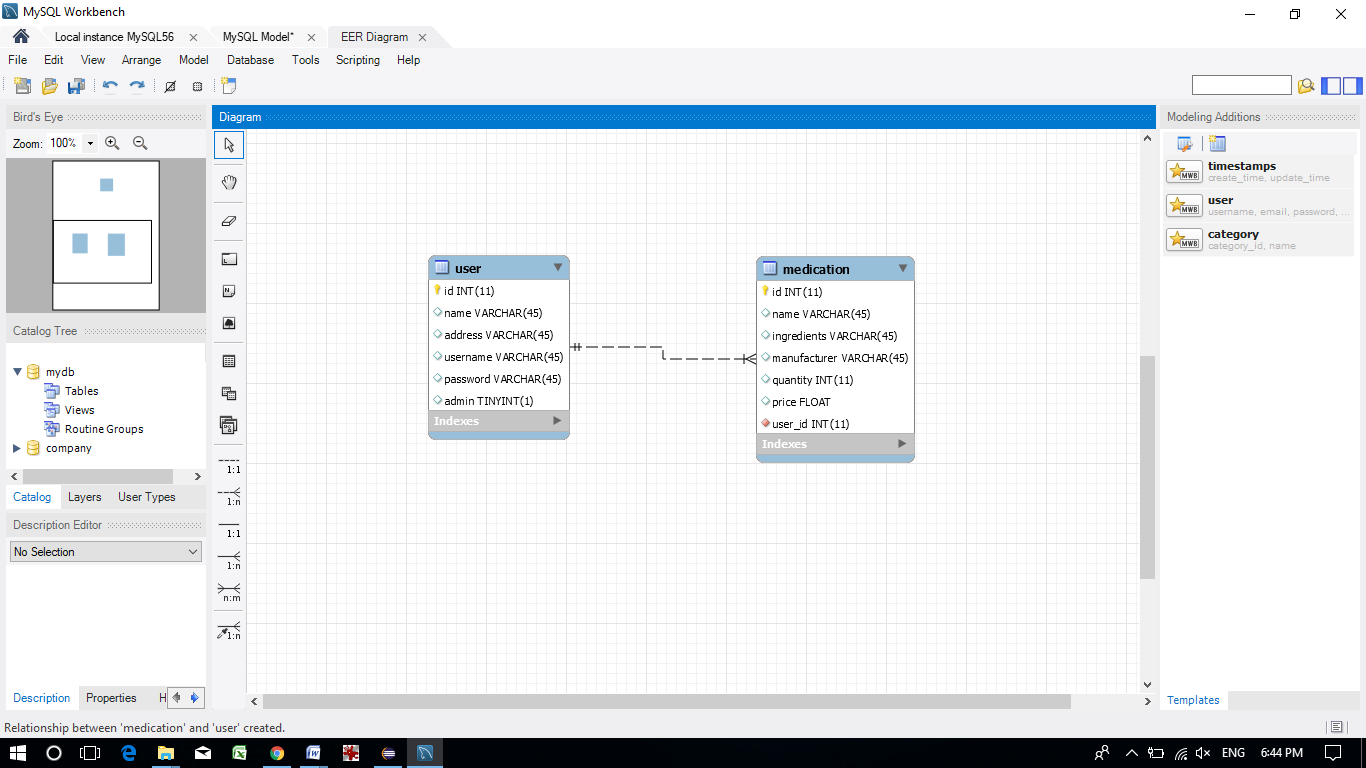
4. UML Sequence Diagrams



5. Class Design



6. Data Model

**

7. System Testing

Am verificat daca codul functioneaza prin realizarea de Unit Tests si prin validarea datelor de intrare.

8. Bibliography