Analysis and Design Document

Student:Tomoiaga Anamaria

**Group:30235**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <10/04/2018> | <1.0> |  | <Tomoiaga Anamaria> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

# Acest proiect prezintă o aplicație web de forma unui magazin online de cosmetice.

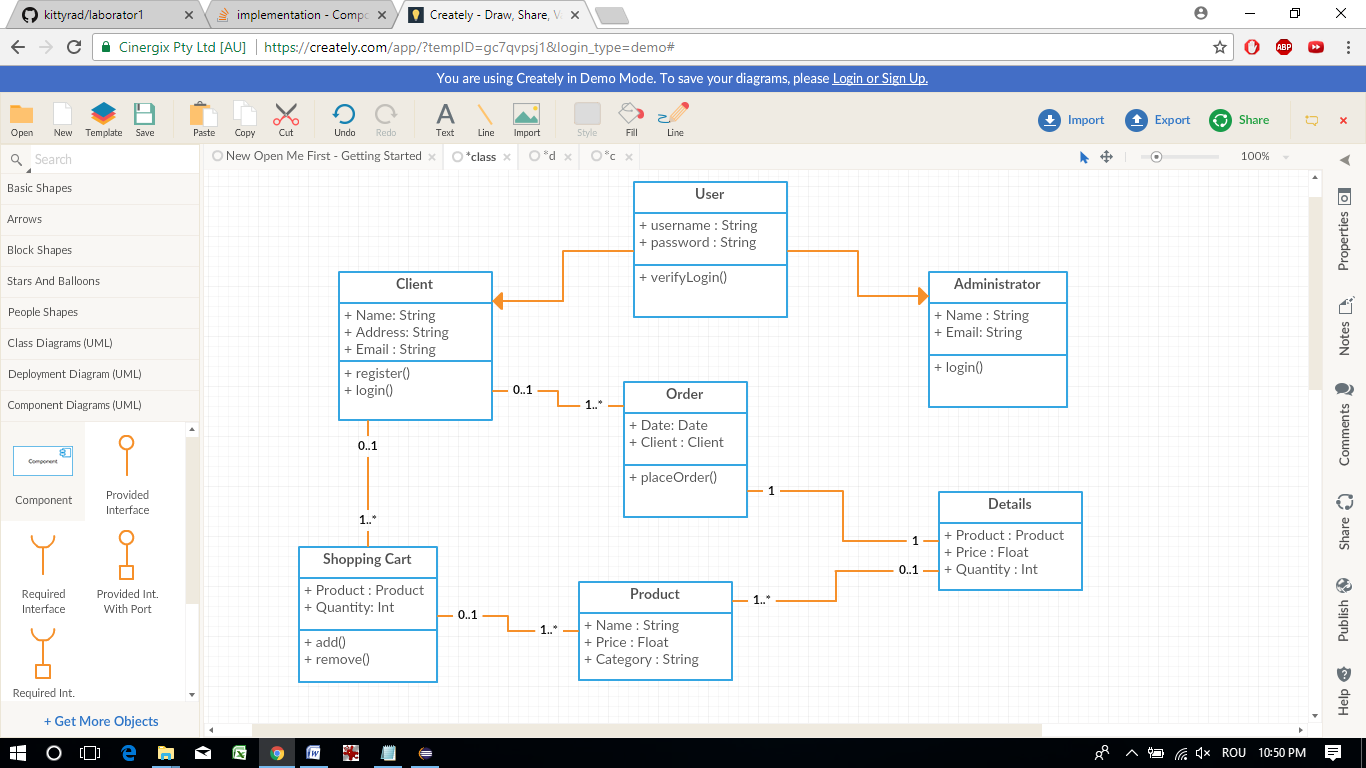
La accesarea aplicatiei, vizitatorul are posibilitatea de a vizualiza o pagină de prezentare, care conține un meniu cu toate funcționalitățile, și anume: vizualizarea produselor, vizualizarea produselor pe categorii, căutarea unui anumit produs si crearea unui cont de utilizator. Avem două tipuri de utilizatori, client și administrator.

Administratorul va fi cel care va avea grijă de întreţinerea sistemului. Acesta va putea adăuga noi produse, șterge produse, schimba prețul produselor și va putea face update la stoc. Însă, pentru a avea acces la toate aceste facilităţi, administratorul va trebui să se autentifice cu un username și o parolă.

User-ul, adică utilizatorul obişnuit, va putea să se înregistreze, să caute un anumit produs, să vizualizeze produsele pe categorii, să afle detalii despre produse, să adauge sau să șteargă produse din coșul de cumparaturi și să realizeze o comandă.

# Elaboration – Iteration 1.1

# Domain Model



# Architectural Design

Proiectul va utiliza 3-tier architecture pattern, care împarte aplicația în diferite layere. Fiecare layer are un scop anume și apelează funcții din layerele de dupa el.

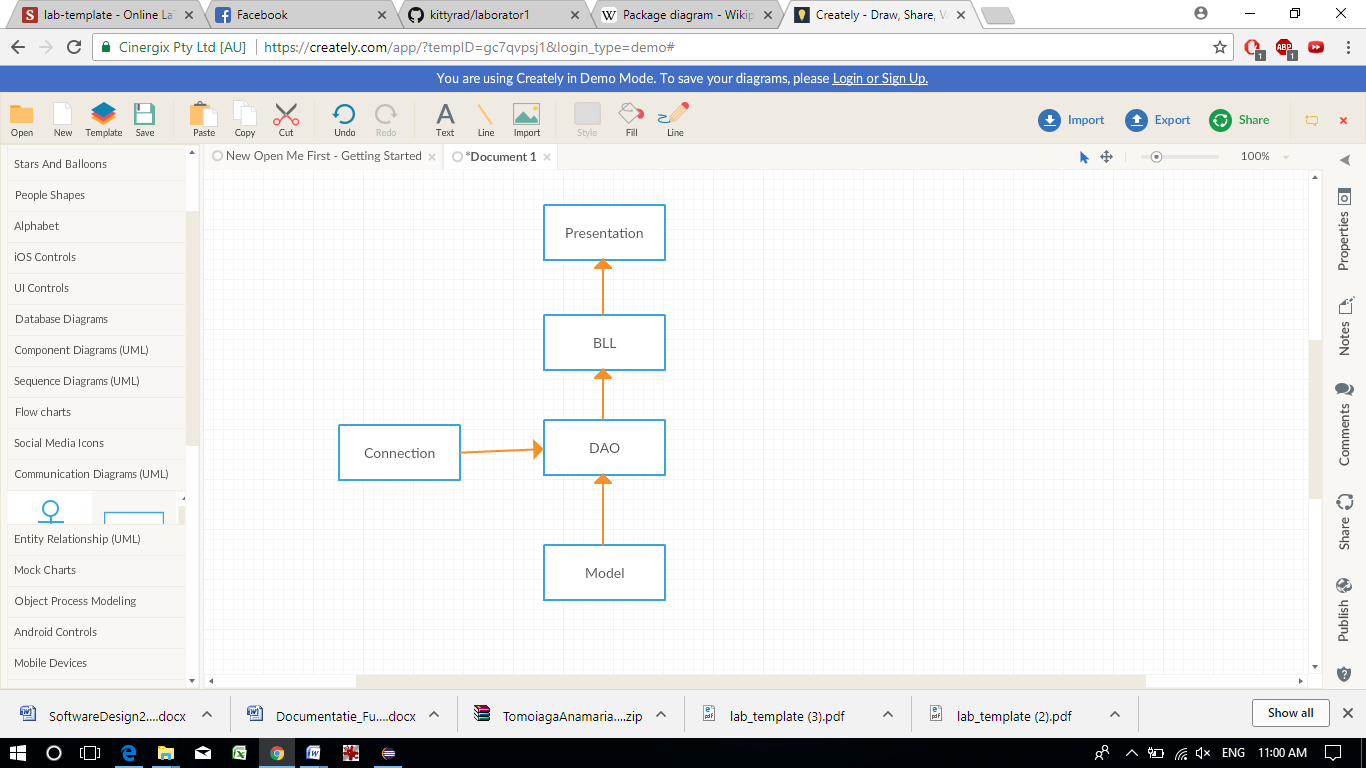
Pimul layer (Data Access Layer) cuprinde clasele care conțin interogări și conectarea cu baza de date.

Al doilea layer (Business Logic) conține clasele care incapsuleaza logica aplicației.

Al treilea layer (Presentation Layer) conține clasele care definesc interfața cu utilizatorul.

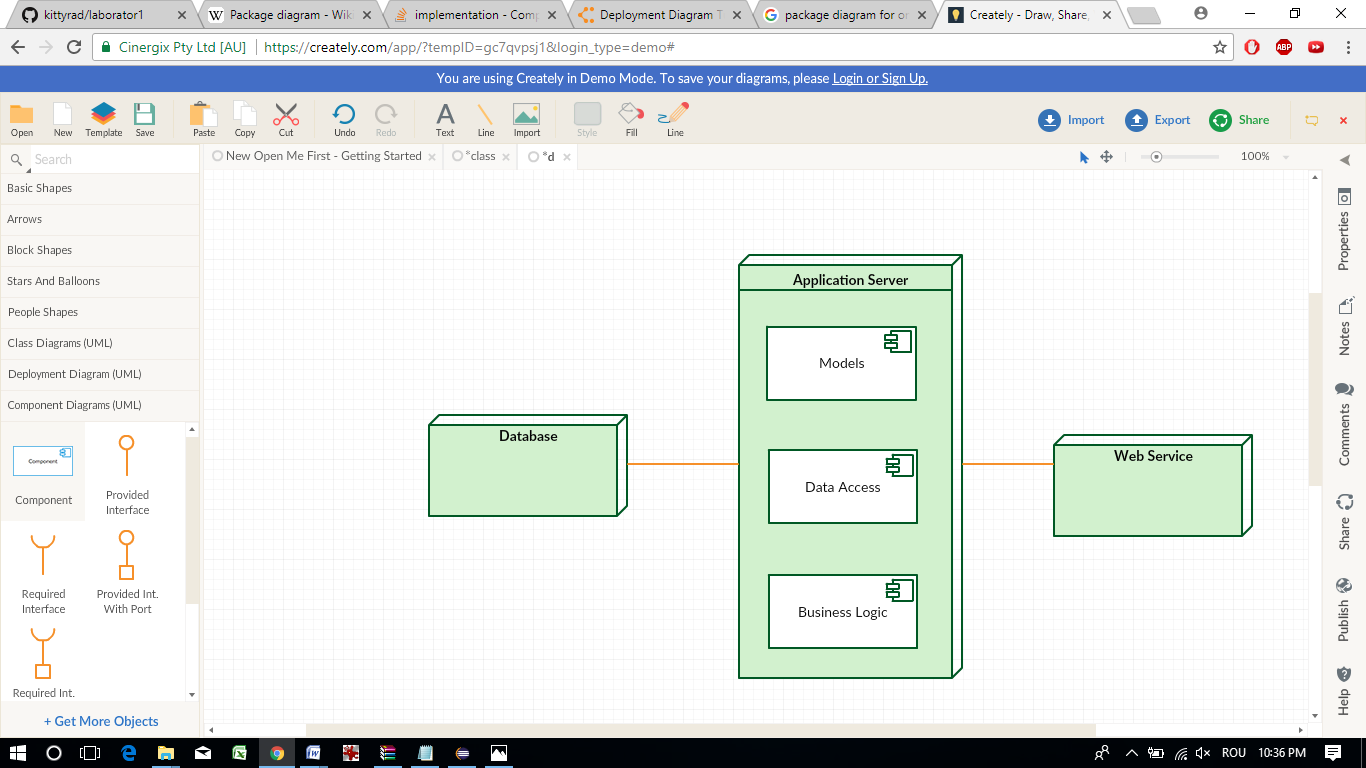
## Conceptual Architecture

## Package Design

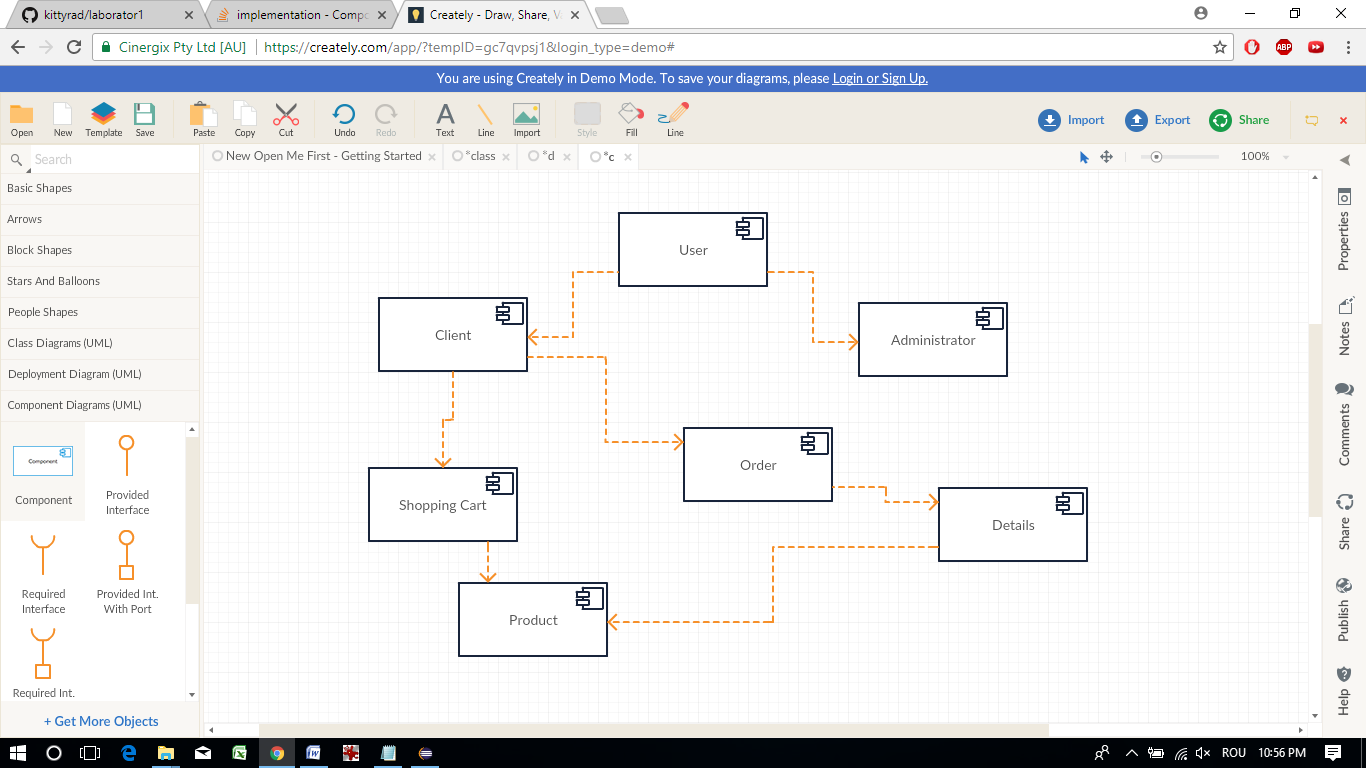


## Component and Deployment Diagrams

# Deployment



**Component**

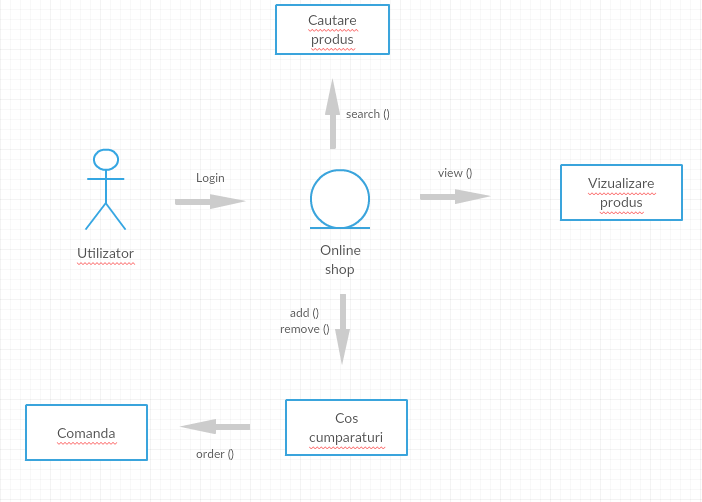
****

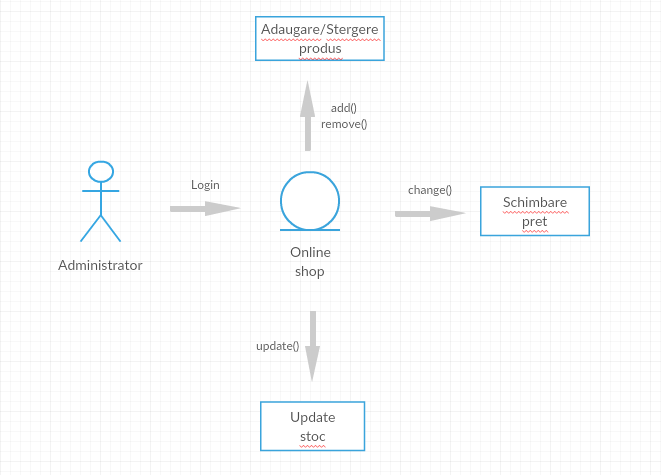
# Elaboration – Iteration 1.2

# Design Model

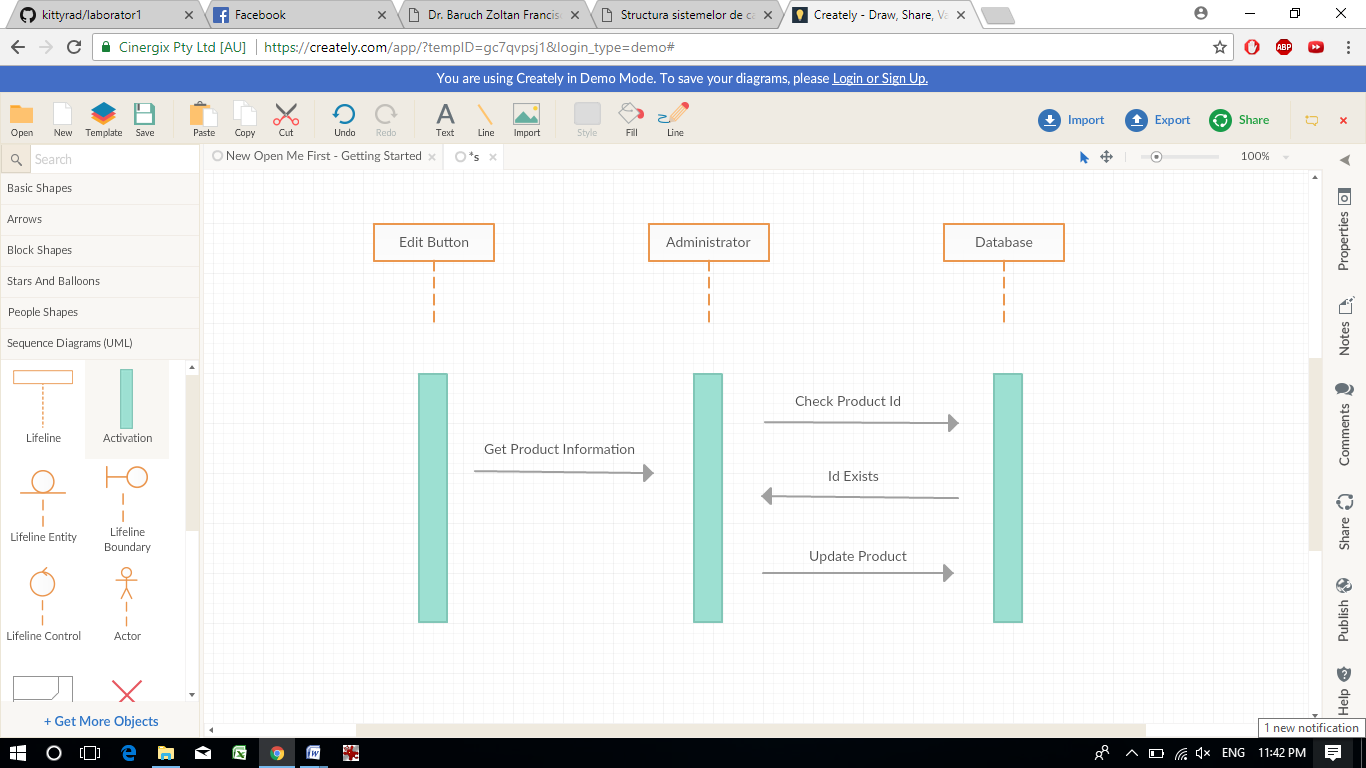
## Dynamic Behavior

Communication diagram

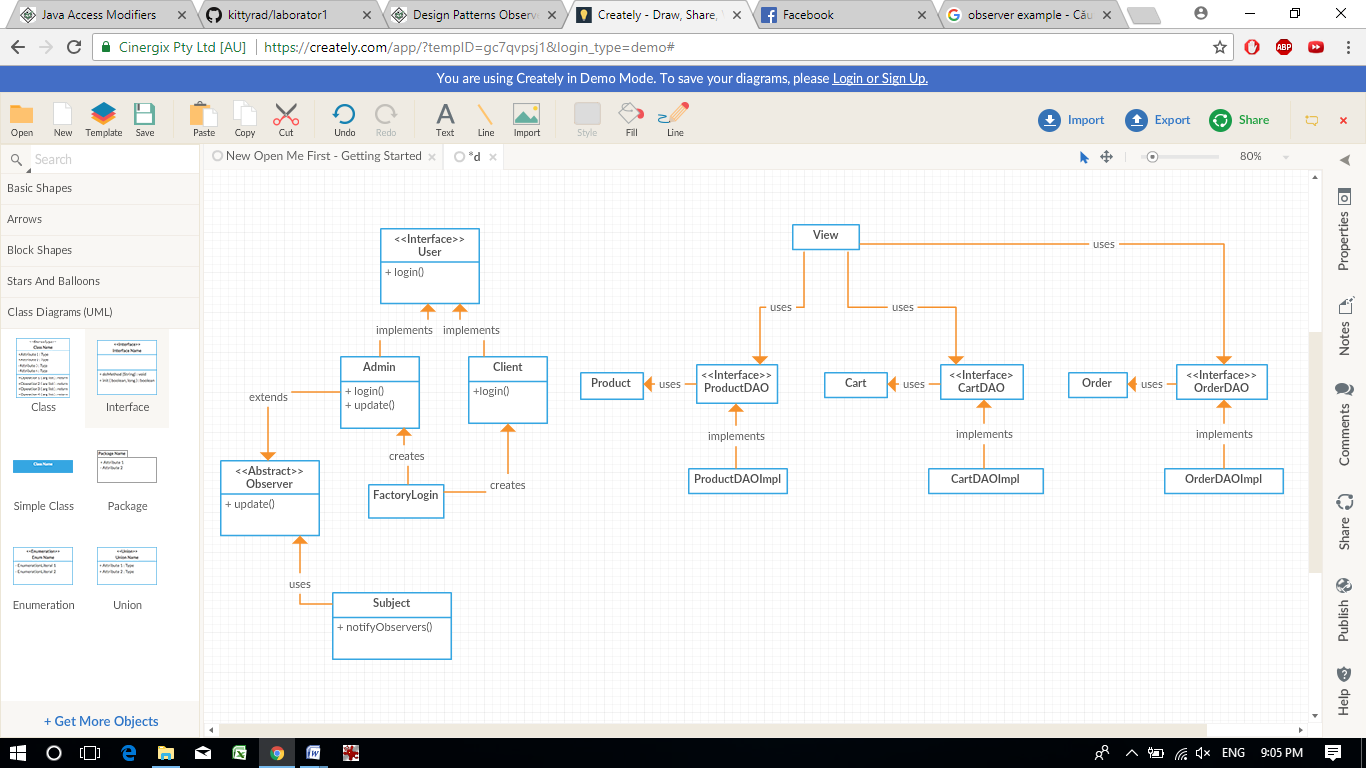
**

**

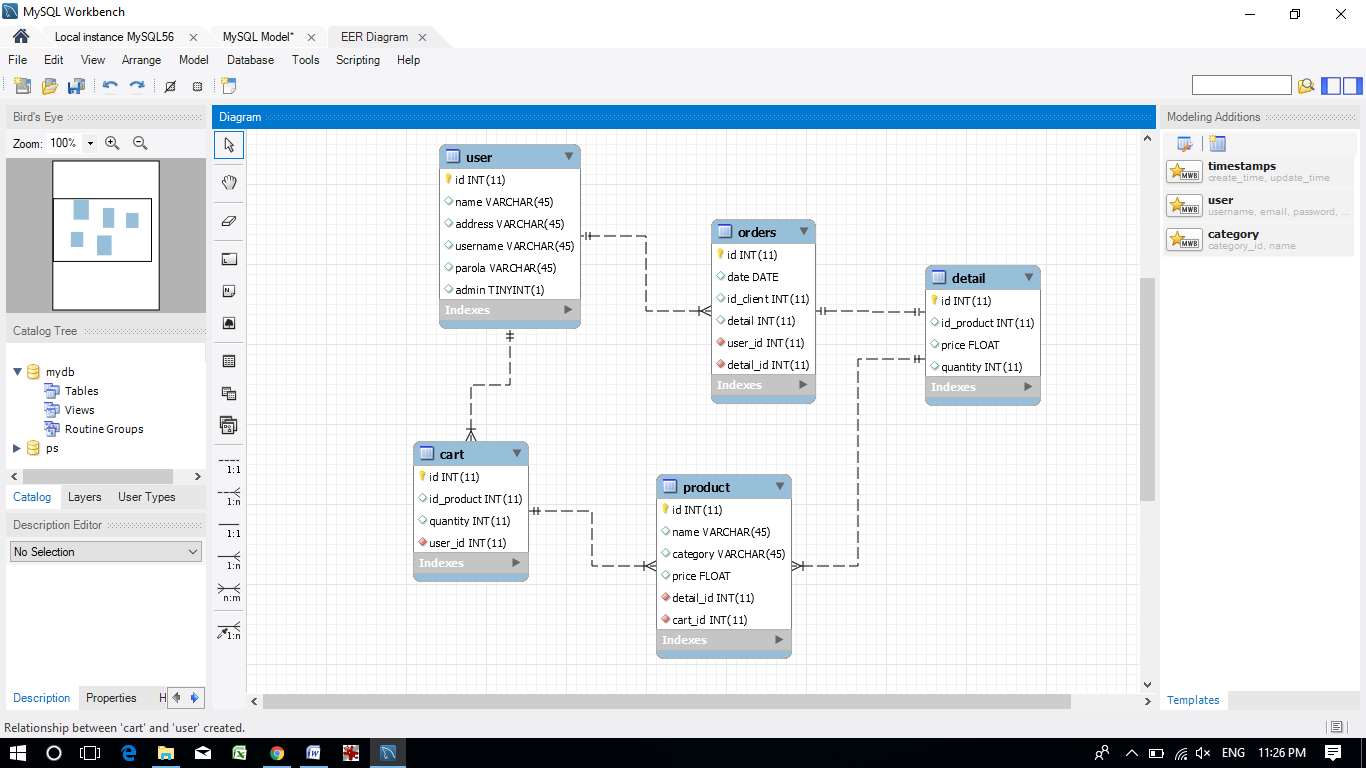
Sequence Diagram



## Class Design

**

# Data Model



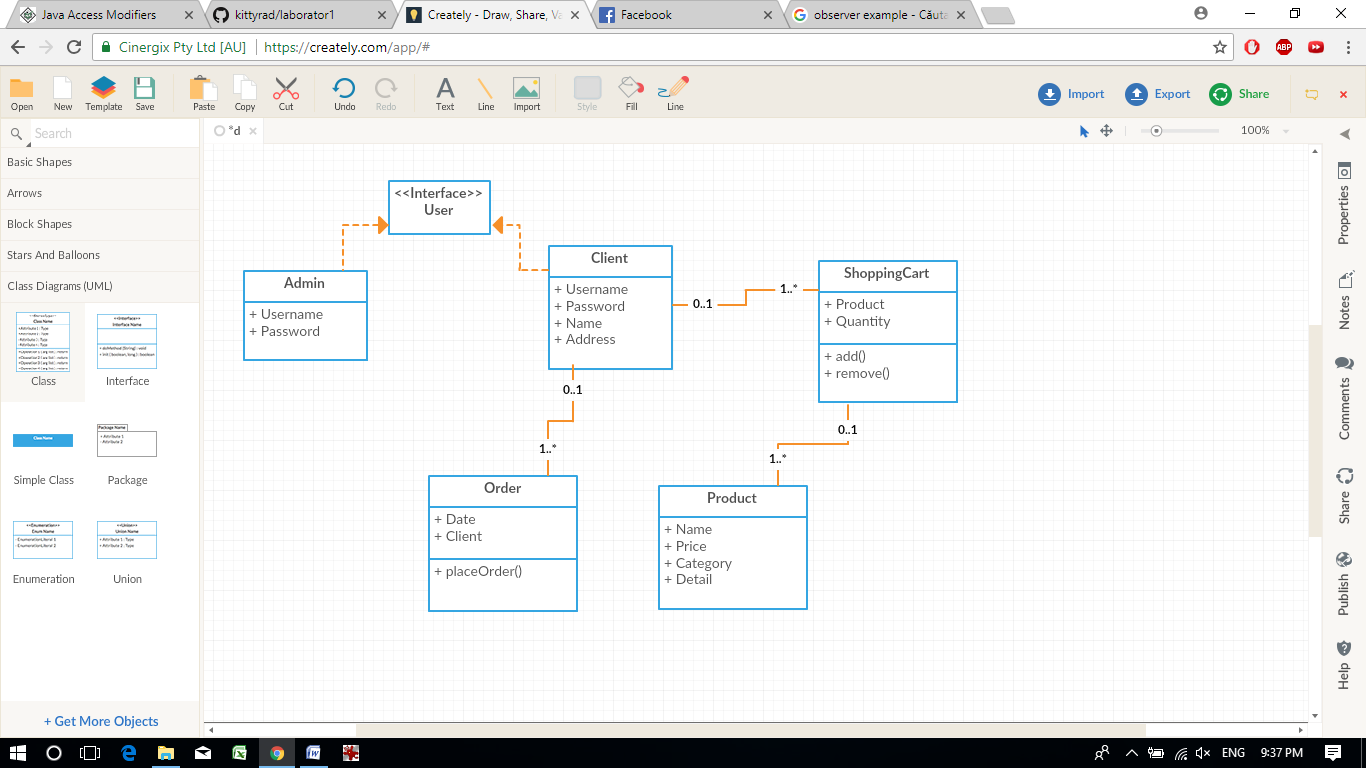
# Unit Testing

Pentru a verifica functionarea codului conform asteptarilor voi realiza Unit Tests.

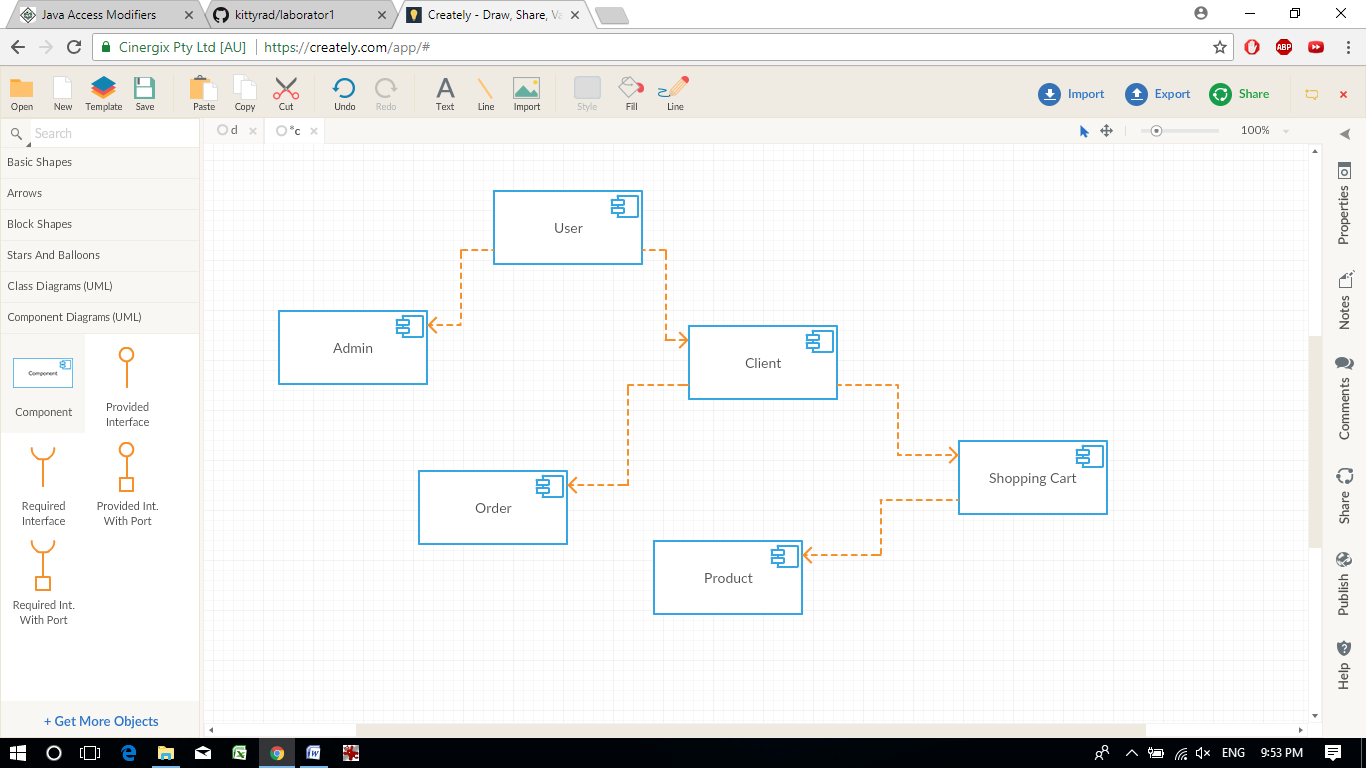
# Elaboration – Iteration 2

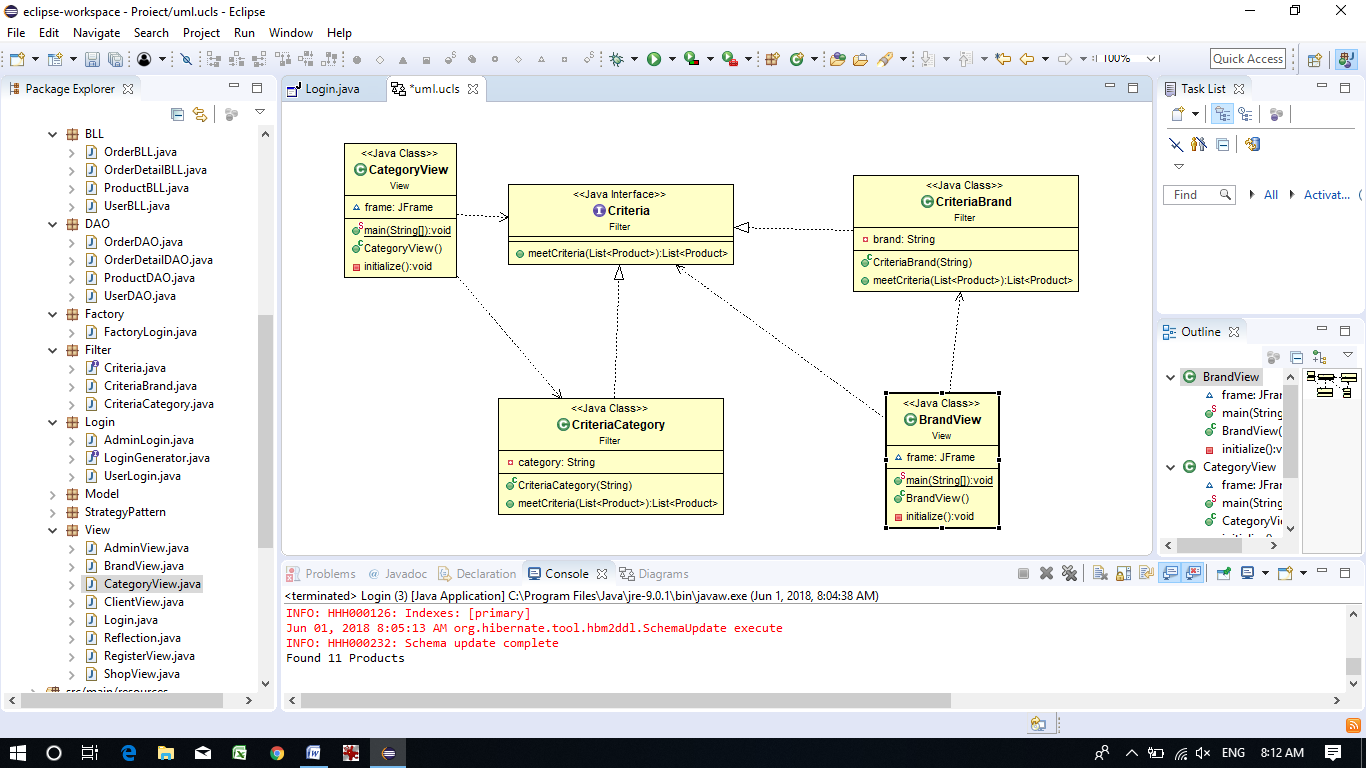
# Architectural Design Refinement

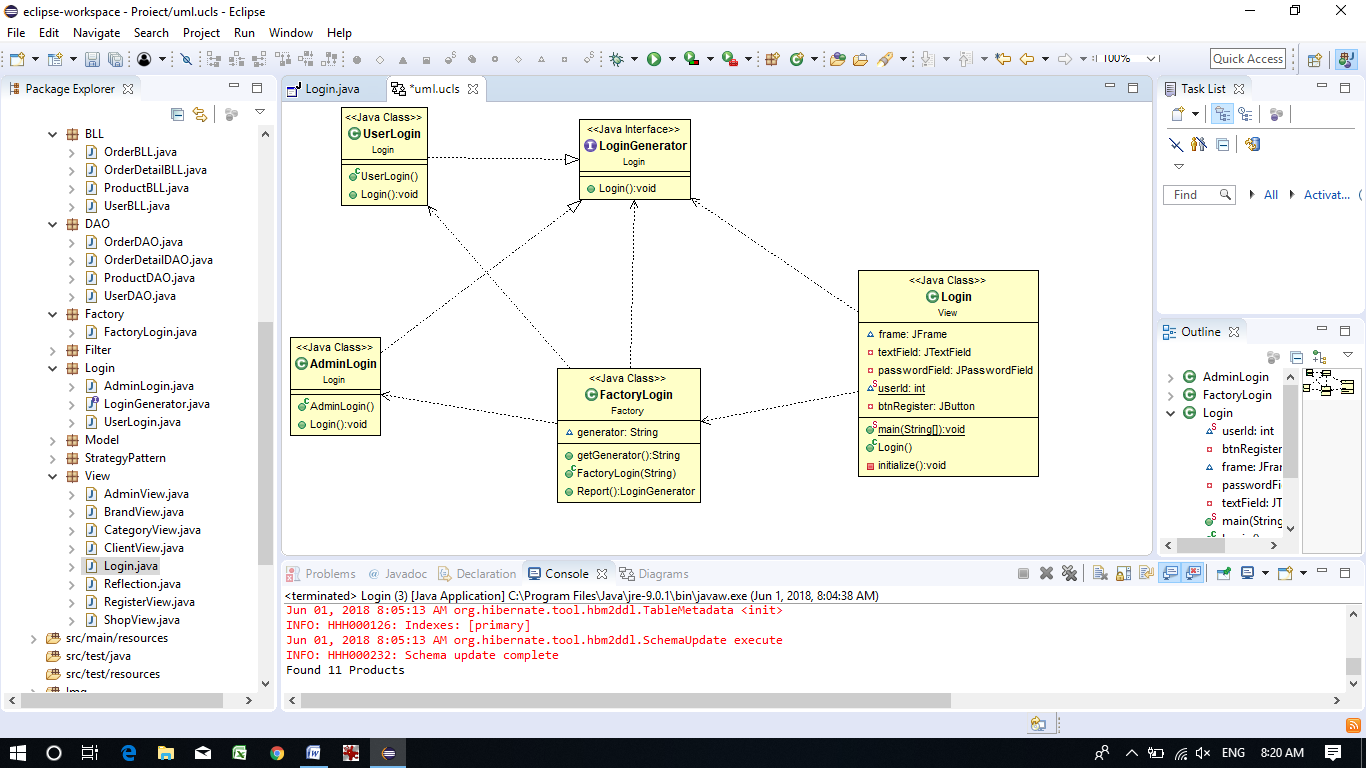
Domain Model

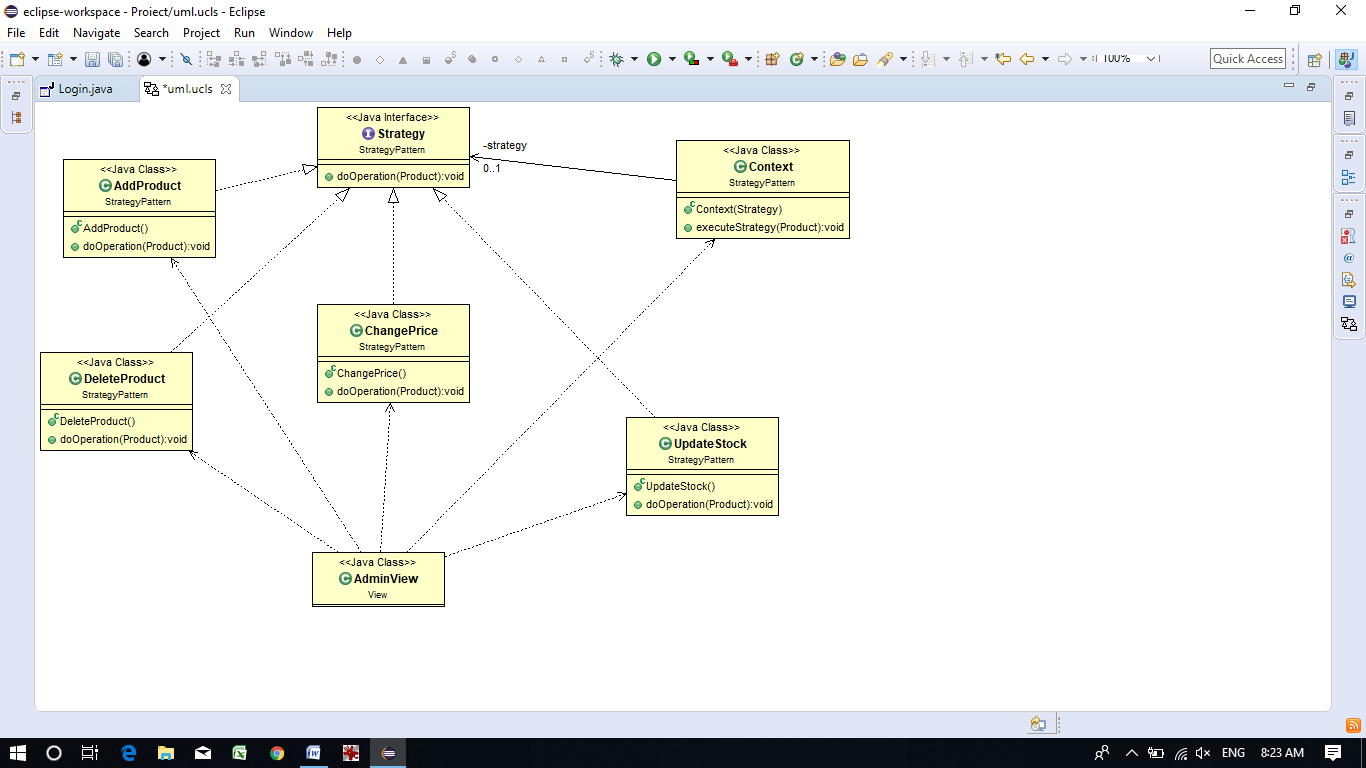


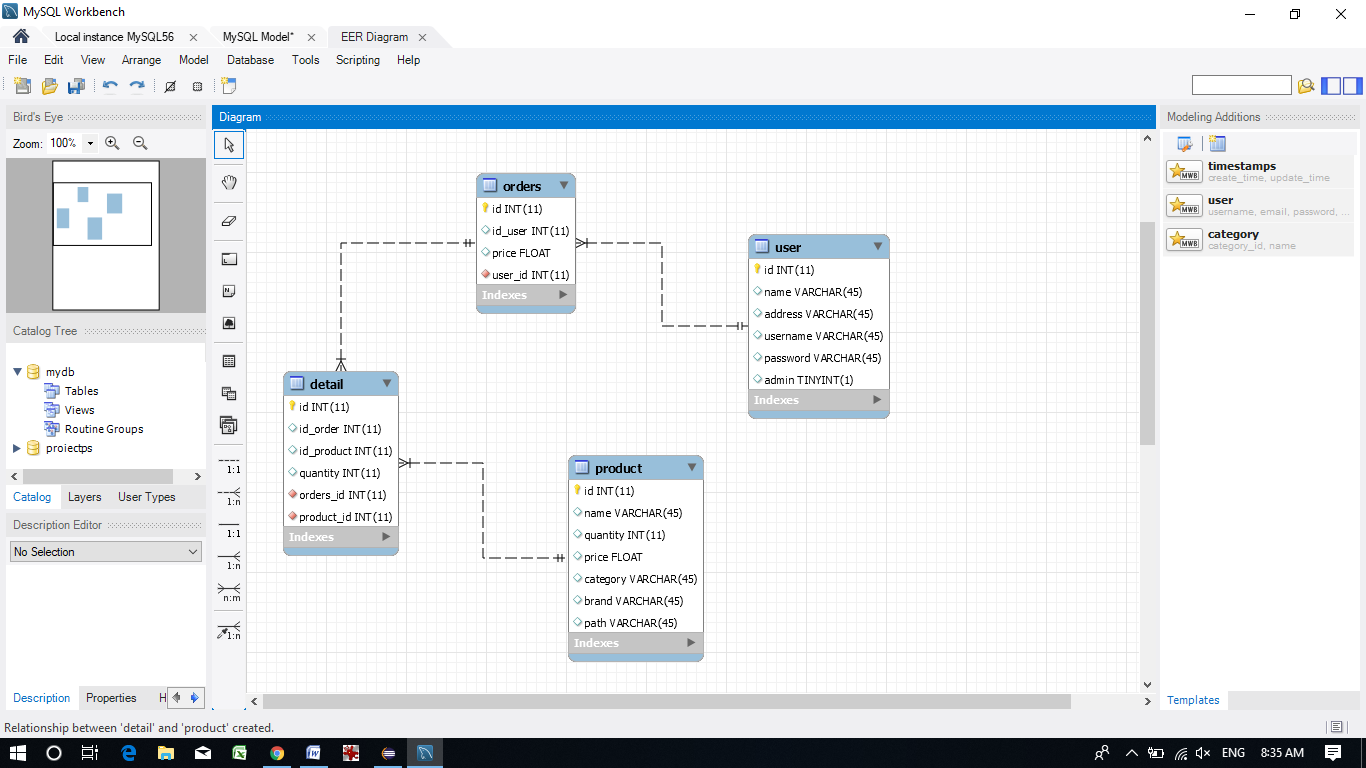
Component



**

**

**

**

Design Pattern-urile pe care le-am folosit sunt: Factory, Filter si Strategy. Pattern-ul Factory l-am folosit pentru a selecta care pagina sa o deschid atunci cand ma loghez, cea a administratorului sau cea a clientului. Pattern-ul Filter l-am folosit pentru a afisa produsele in functie de doua criteria: brandul sau categoria. Pattern-ul Strategy l-am folosit pentru a selecta operatiile pe care le realizeaza administratoul.

# Bibliography

https://www.tutorialspoint.com/design\_pattern/filter\_pattern.htm