Analysis and Design Document

Student:Brudan Oana

**Group:30234**

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 Deployment Diagram 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

4. Future improvements 5

IV. Bibliography 5

# Project Specification

**4. Hardware/Software Manager**

Design and implement a client-server application for the employees of a company selling the following types of products: hardware components (e.g. hard disks, memories, processors, etc.), peripherals (e.g. monitors, printers, keyboards, etc.) and software products (e.g. operating systems, office applications, etc.). Each type of product has the following information associated: identification number, technical features, producer, stock, and price. A customer may desire to buy individual products or may ask for a system that requires the configuration of multiple products. A system configuration may be composed of hardware, peripherals and software products.

The application should have two types of users (a regular user represented by the front desk 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:

* Sell products: involves initiating a sell, adding products to the sell, computing the total price, decreasing the products’ stocks, finalizing the sell and issuing an invoice.
* Configure a system for a customer.

The administrator user can perform the following operations:

* CRUD operations on user accounts.
* CRUD operations on products.

In case a product becomes out of stock, the system should automatically alert the administrator user.

# Elaboration – Iteration 1.1

# Domain Model

Domain model defineste patternul folosit in proiect.

* **Model**

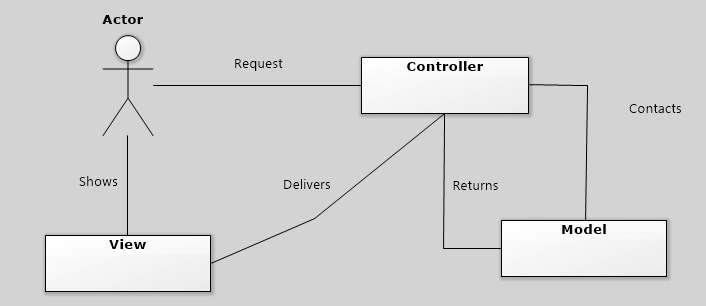
Această parte a controlatorului manipulează operațiunile logice și de utilizare de informație (trimisă dinainte de către rangul său superior) pentru a rezulta de o formă ușor de înțeles.

* **Viziune**

Acestui membru al familiei îi corespunde reprezentarea grafică, sau mai bine zis, exprimarea ultimei forme a datelor: interfața grafică ce interacționează cu utilizatorul final. Rolul său este de a evidenția informația obținută până ce ea ajunge la controlator.

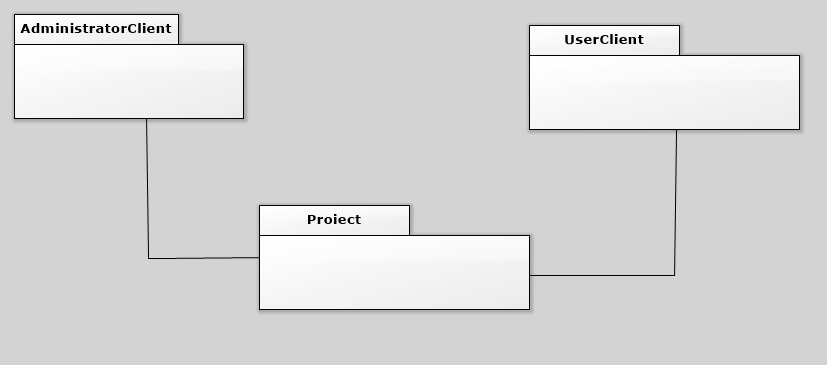
* **Controlator**

Cu acest element putem controla accesul la aplicația noastră. Pot fi fișiere, scripts sau programe, in general orice tip de informație permisă de interfață. În acest fel putem diversifica conținutul nostru de o formă dinamică și statică, în același timp.



# Architectural Design

## Package Design

**

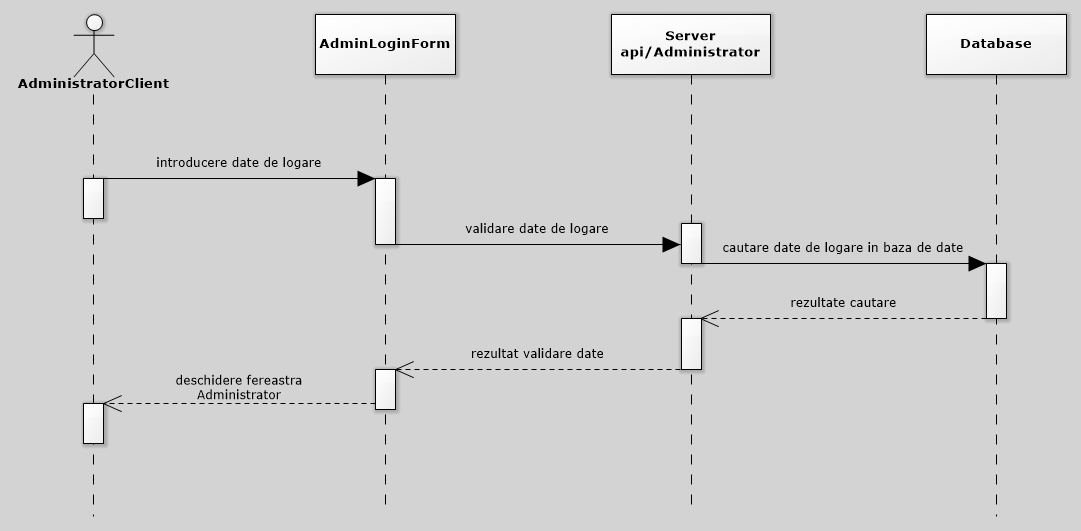
## C:\Users\Brudan-Oana\Desktop\diagrame\deployment.jpgDeployment Diagram

Deployment diagram arata o distributie a structurii proiectului.

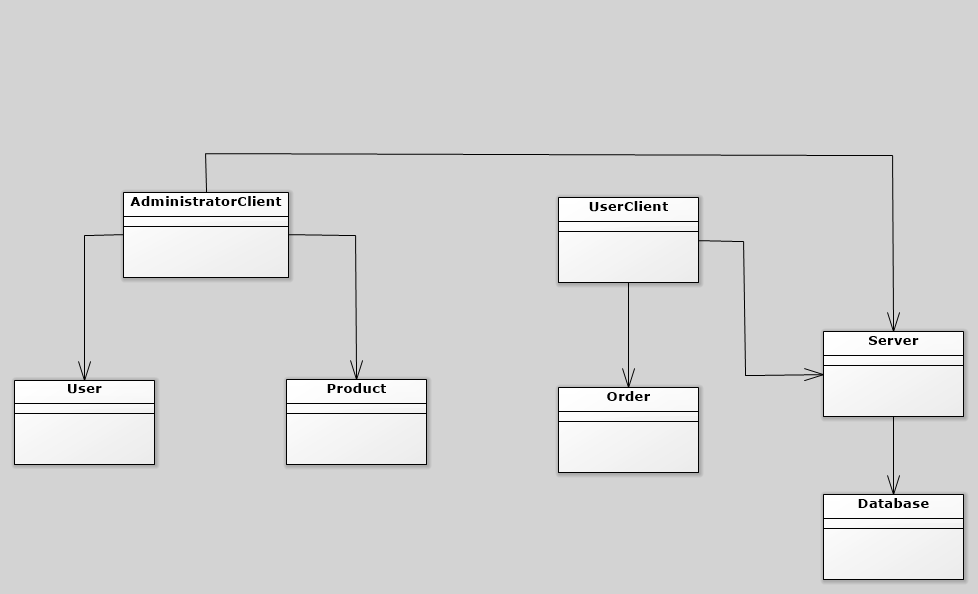
# Elaboration – Iteration 1.2

# Design Model

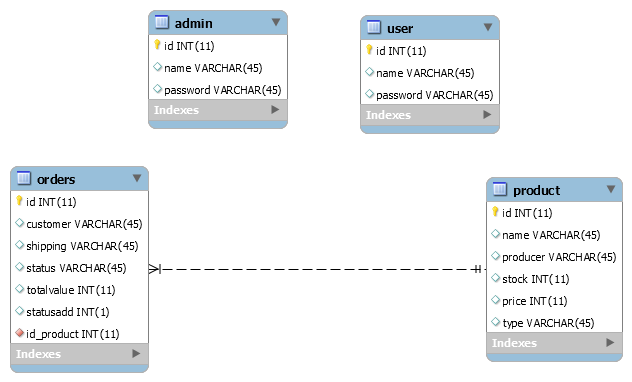
## Dynamic Behavior



## Class Design



# Data Model



# Future improvements

Imbunatatiri viitoare sunt implementarea operatiilor de actualizarea a stocului, calcularea a preturlui, eliberarea unei facturi de catre user.

# Bibliography

[1] „Patterns of Enterprise Application Architecture”, Book by Martin Fowler

[2] „Software architecture patterns”, Mark Richards