**<Pizza Delivey>**

**Analysis and Design Document**

**Student:Sapantan Gabriel**

**Group:30238**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**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**

I want to implement an application wich that we can put an order of pizza online .

I will be have o manager user ,a deliverer and a client .

The client can choose a pizza from the website, can write his name , addres , and can set up an hour between 2 and 4 ore in which he can pick-up the pizza.

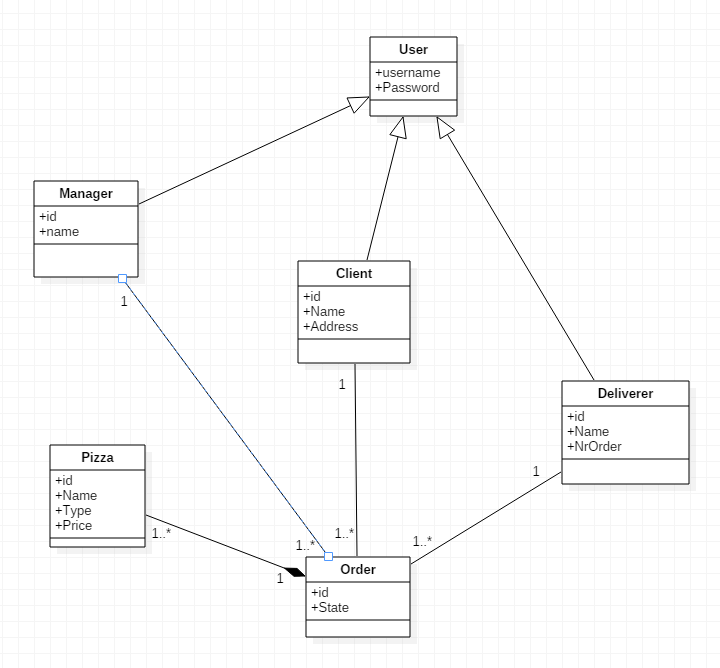
The client can see every state in which are pizza : if is ready to comming, or is in a cooking process .

The deliverer can take a pizza , can lead pizza, and is able to confirm a state pizza at every move .

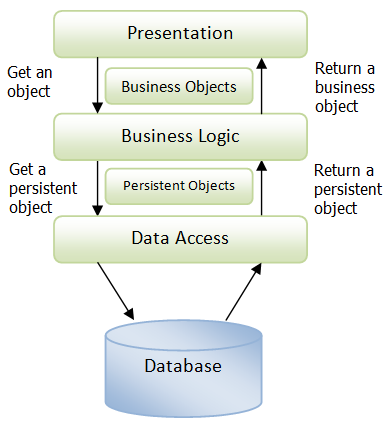
The manager can take a order , can say if some pizza is available , can to confirm cooking , actually can make the stae of pizza to be visible for client .

Every actor , if I could say so , can to login . The client and deliverer can create an account. The manager have an acoount from beginning , he doesn’t need to sign up .

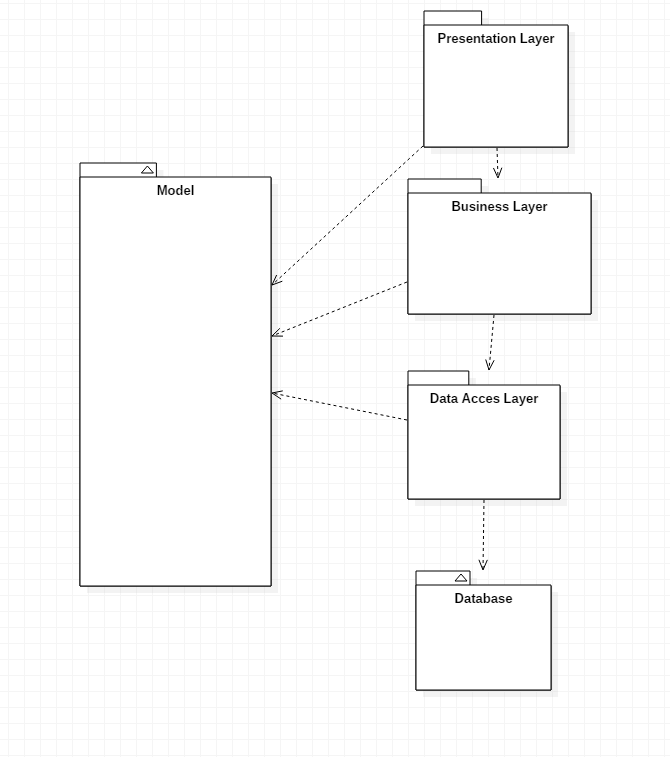
* **Elaboration – Iteration 1.1**
* **Domain Model**



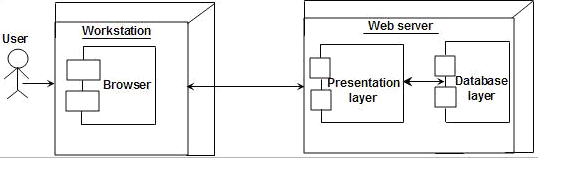
* **Architectural Design**
* **Conceptual Architecture**

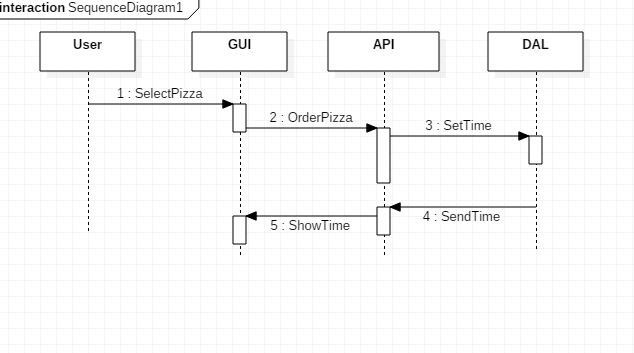


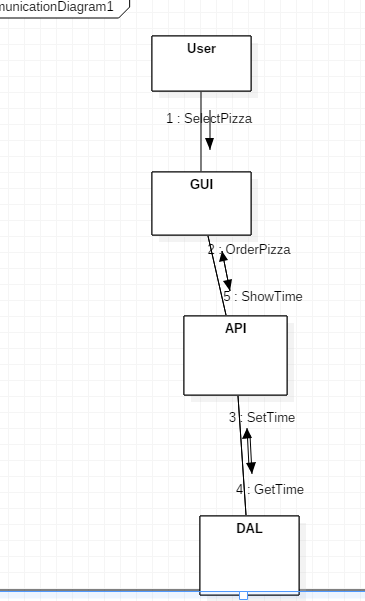
* **Package Design**



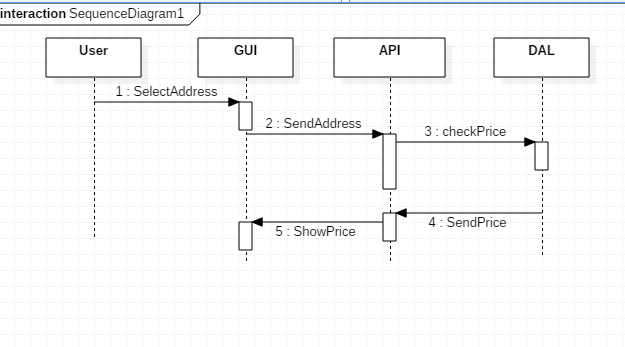
* **Component and Deployment Diagrams**

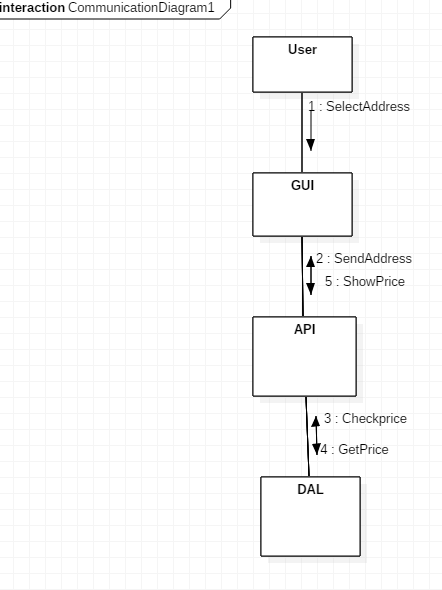


* **Elaboration – Iteration 1.2**
* **Design Model**
* **Dynamic Behavior**
* **Check the waiting time**
* 

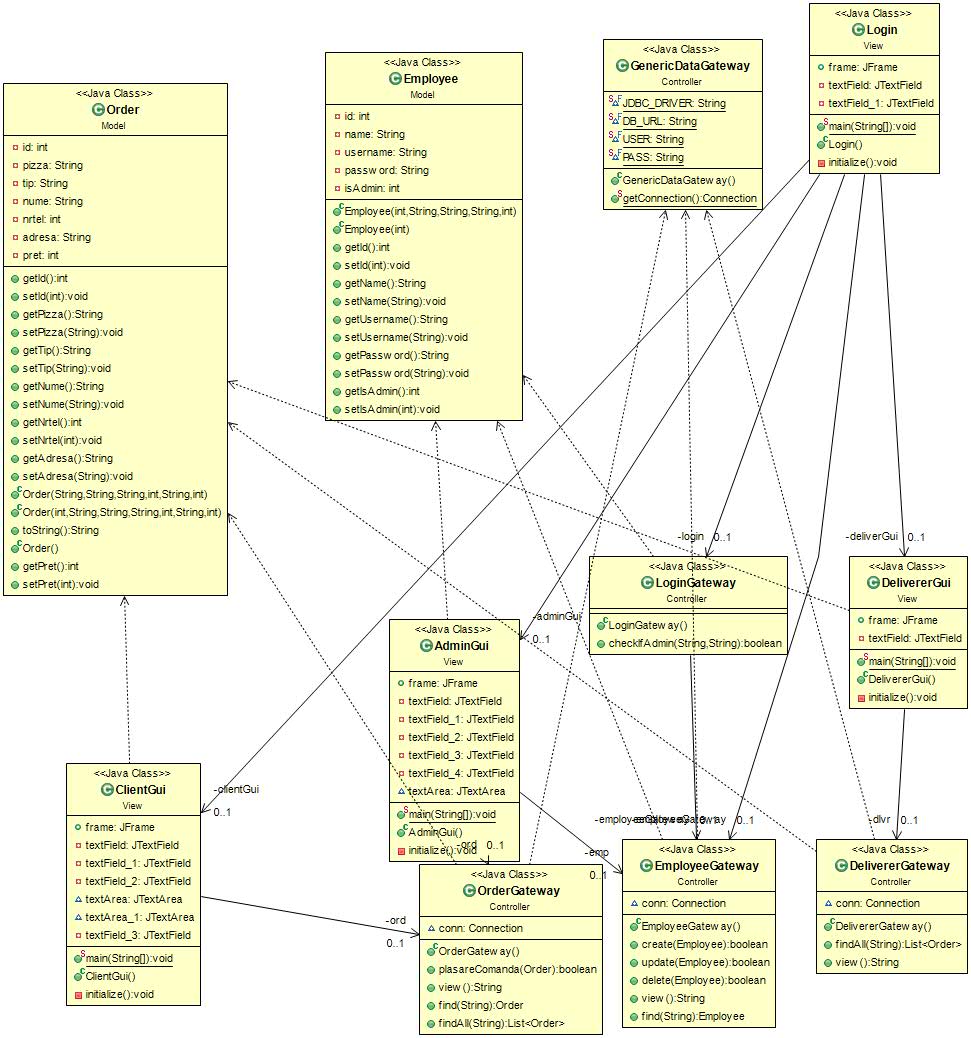


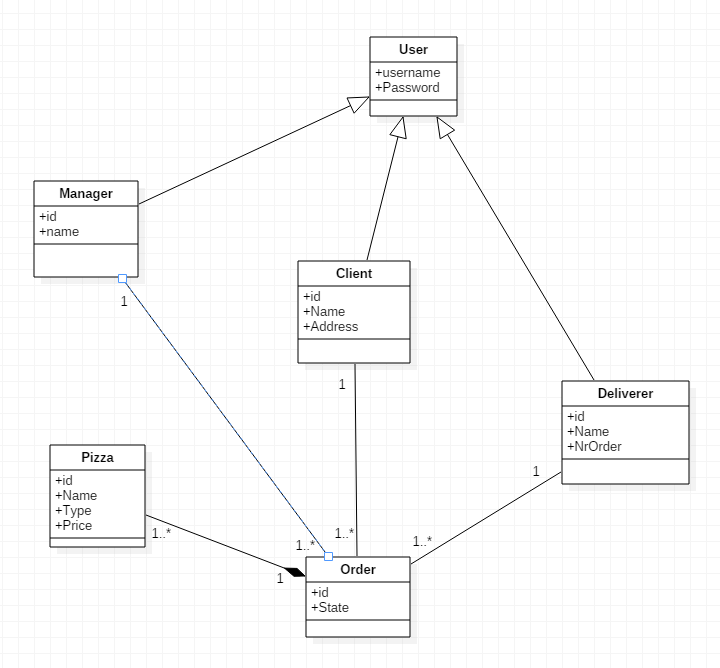
Check the price





* **Class Design**

****

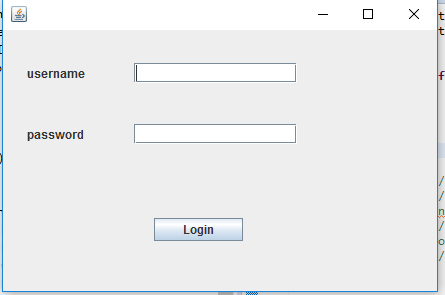
* **Data Model**
* **Unit Testing**

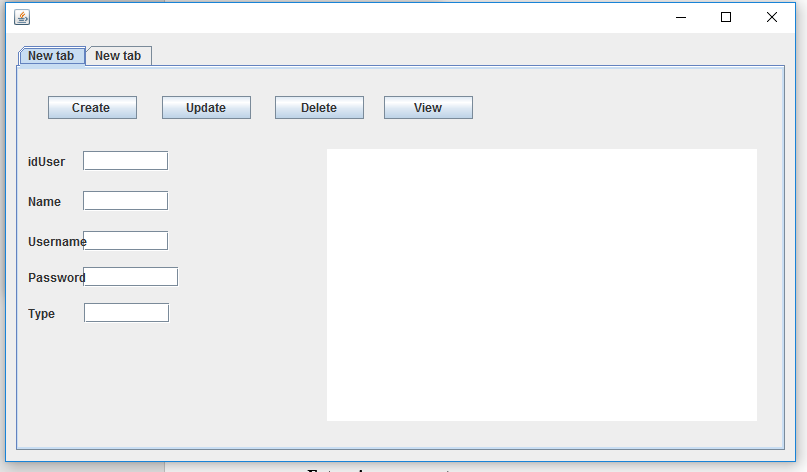
*[Present the used testing methods and the associated test case scenarios.]*

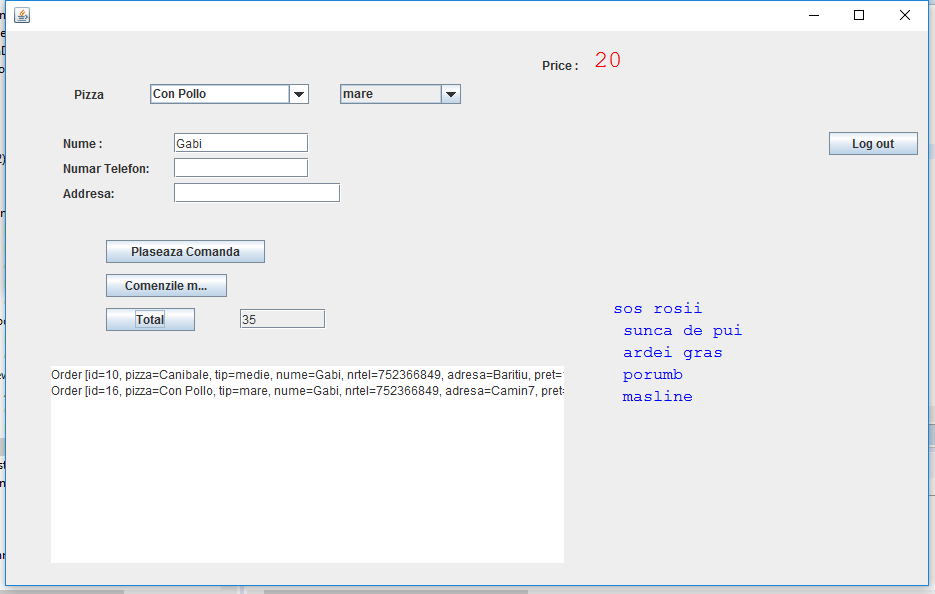
* **Elaboration – Iteration 2**
* **Architectural Design Refinement**

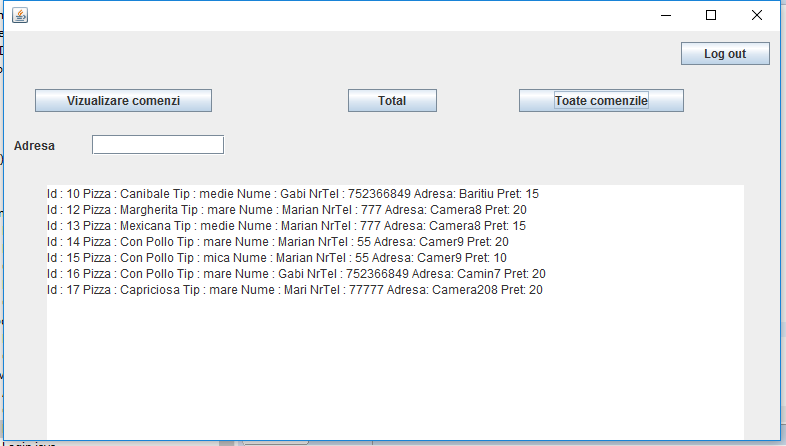
*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

* **Design Model Refinement**
* *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*
* **Construction and Transition**
* **System Testing**









* **Future improvements**
* **Aplicatia ar mai putea afisa timpul de asteptare**
* **Bibliography**

<http://docs.oracle.com/javase/tutorial/uiswing/>

<http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>

<http://www.saxproject.org/sax1-roadmap.html>

<http://www.roseindia.net/xml/dom/>

<https://msdn.microsoft.com/en-us/library/54xbah2z(v=vs.110).aspx>

<https://msdn.microsoft.com/en-us/library/e80y5yhx(v=vs.110).aspx>

<http://msdn.microsoft.com/en-us/library/system.xml.xmlreader.aspx>

<http://msdn.microsoft.com/en-us/library/system.xml.xmlwriter.aspx>

<http://msdn.microsoft.com/en-us/library/ms764730(VS.85).aspx>