CS 319 - Object-Oriented Software Engineering

Design Report

BestTrade

Group 3M

Berk Türk - 21302570

Ezgi Çakır - 21402334

Irmak Tural - 21301099

İbrahim Berker Kırdök - 21502870

İlhami Özer – 21300828

İçindekiler

1. Introduction

1.1 Purpose of the system

BestTrade is a software for university students to buy/sell items such as book, notes and furniture. Nowadays, a lot of student try to get books and notes for lectures easier and cheaper way. Furthermore, since most of them move from their hometown they need furniture for their apartment flat or dorm room. Thus, Best Trader wants to help them to reach who sell it. Moreover, by selling their books, notes and furniture students get extra money to provide their other needs.

1.2 Design Goals

The main point of the design period is to illustrate our solutions for requirement to our software. The focus at this process we try to be sure about what we really need and is there any missing points on our requirements for whole software. Thus, at this stage we proceed one step forward and shape our software more so that we could easily find out what we need. As we come closer to implementation part, necessary parts show up which makes whole design more complex. Therefore, the main goals are clear documentation of design parts and being as much as flexible to adapt changes and surpass and problems we could face to.

- ## Tradeoffs

Efficiency – Reusability: Reusability is the main concern of this project, because it can be integrated for the future applications of trading or upgrading this system, so

classes must be adjusted properly. However, the program has to be efficient also, because a trading system must have at least a little performance in order to make the page shifts faster and efficient.

Memory – Performance: In this project, we will use database for storage space, because of that, memory will not be an issue. So, we will focus more on the speed of the program rather than the memory.

Customizability – Robustness: In this program, we will implement it such a way that users rather than admin can not change the functionality of the program by using a settings option. The reason for this is most customers and sellers may not have technical knowledge, therefore playing with settings may give some errors to their application. Because of that, not having a settings option will indicate that the program will perform in a predefined way.

2. Software Architecture

2.1. Overview

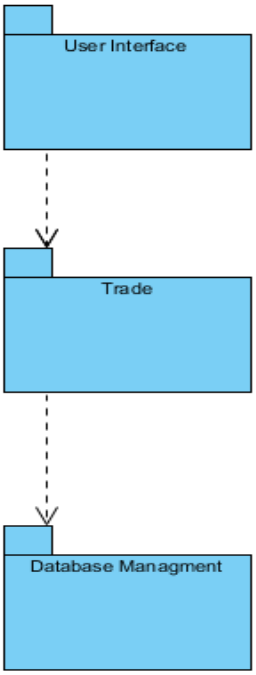
The software architecture that is the most suitable design for our system is three tier. In three tier architecture, the subsystems are organized into three layers. Three-tier architecture includes user interface/presentation, functional process/functional process logic and data storage/access/database layers.

In the interface layer, there are boundary objects which is dealing with user and provide interaction between user and the system.

In application logic layer, the commands and datas are processed, functions of system, logical decisions are done.It includes entity and control objects.

Database layer the information responsible retrieveing data and passsed back to user or logic layer to process. Also it realizes query of persistent objects. By investigating these organization structure our application is fit by three tier system.

Our system is organized into 3 layers, User Interface Subsystem, Trade Subsystem, Database Layer.



The classes that provides interaction between the user and the system are in the User Interface Subsystem. According to user's input and choices, the interaction between Layer 1 (UI Subsystem) and Layer 2 (Trade Subsystem) is provided. In the trade subsystem, the input that user give is processed based on the functionalities of classes in Trade System. We have classes responsible for providing an interface between the user and the system. Also the Layer 2 and Layer 3 interacts with each other. The change in objects are updated in Database by passing objects between them.

Maintenance Criteria:

Usability: Most user has low tolerance to complex and difficult desgins. So our program is user friendly by its User Interface Design. User can perform the task that he aimed easily. The

screens are designed according to user's view, by buttons and texts, program leads user to do the prescribed tasks easily. User can interact with the system easily, it is not complex and it is designed to provide the need of user.

Extendibility : Based on demand and need of users the program can be added. The program will allow to make these changes in terms of its design, it is accessible to be improved. New components, entities and features can be included according to what user would need and demand.

Portability: Portability is an important in respects of capability to reach every user that use different platforms. Our project is implemented Java. It has ability to run on differen machines. It does not require recompilation. It has source code, CPU architecture,OS/GUI portability. By JVM it provides platform independency.

Reliability: Our system will guarantee user to not to experience failure. All the cases which may lead to an system crash will be considered carefully. By these considerations at all of the stages in the program, it will be checked. This criteria will provide user a bug-free program.

2.3. Hardware / Software Mapping

BestTrader will be implemented in Java and it requires Java Development Kit (JDK) environment for writing Java. For the hardware requirements, BestTrader requires keyboard to type username, password of user and type, price of product and also it requires mouse to click buttons. Furthermore it requires computer to upload the application.

For the storage, BestTrader will store the information of user and information of items on database system. For the database connection and storage, we will need extra computer to use it as a server.

2.4. Persistent Data Management

he datas belonging to our system will be stored in database. Information of user (attributes of client class) is pushed in database when user signed up. Information of item (attributes of item) is pushed in database when it is added to system. When the information of user and item would be changed, the information is updated in database, refreshed and returned the system. Also the item that would be removed from the system will be deleted from database.

2.5. Access Control and Security

BestTrader will require network connection. This connection is required for the database connection between user and server. For the user account BestTrader has some restrictions. For instance user doesn't view the password of other users and also they don't view the profile of the other user. However for the admin account, there is no any restriction. Furthermore there will be security issues to prevent the SQL injections.

2.6. Boundary Conditions

Initialization

BestTrader doesn’t require any kind of setup process.

Termination

For the termination of BestTrader clicking close (X) button at any screen of application will enable to user to terminate. BestTrader doesn’t have specific log out button, clicking the X button user will log out automatically.

Error

If user doesn’t remember username or password they will not be able to sig in.

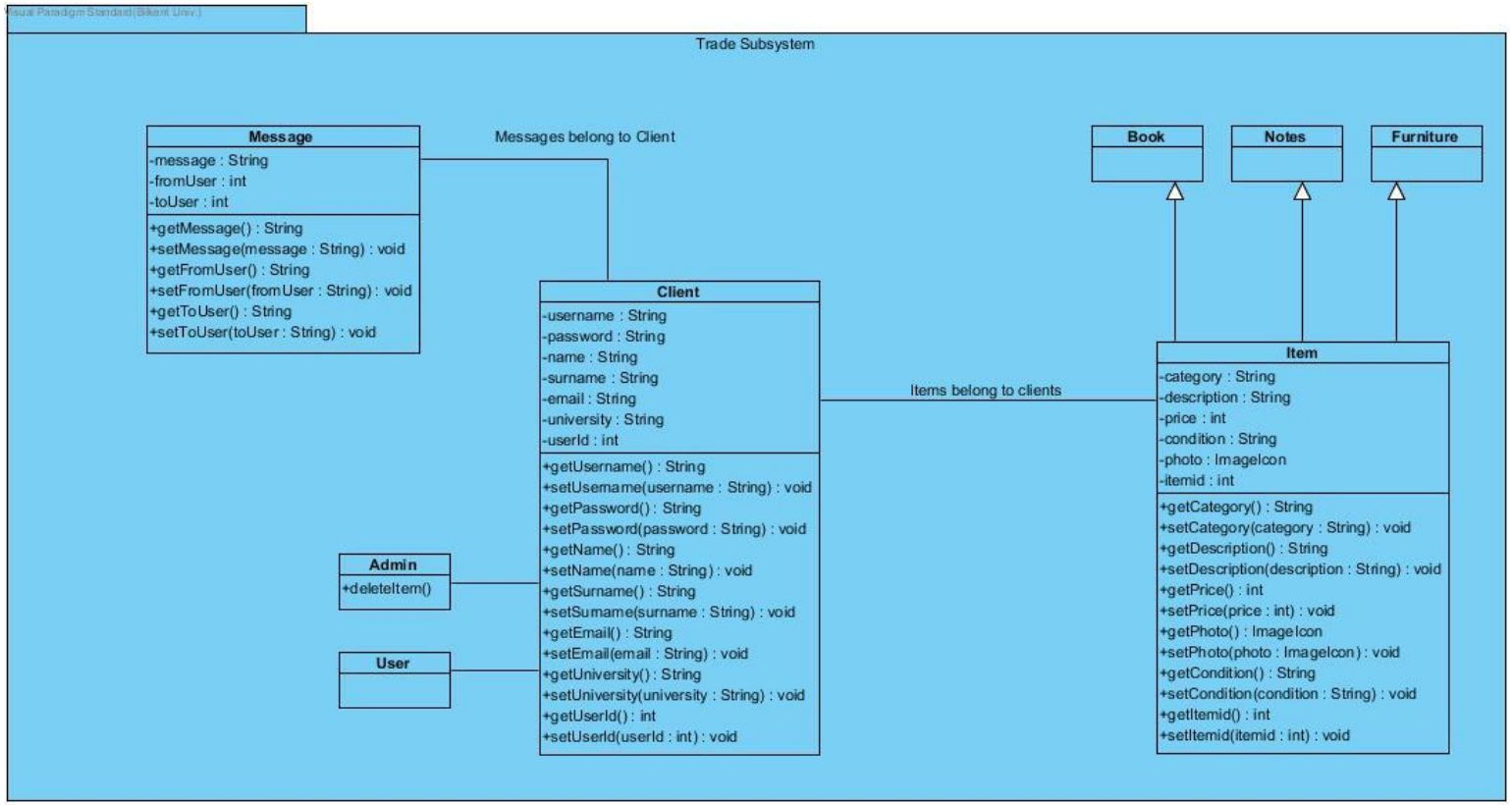
If the users don’t connect to the server they will not be able to sig in and also they will not be able to access item.

For error situation there will be error pop ups.

3. Subsystem Services

3.1. Trade Subsystem

In this subsystem, we have our entity objects which are Message, Client, Admin, User, Item, Book, Furniture, and Notes. The class diagram of this subsystem can be seen in the Figure 1. Detailed description of all classes are as following.

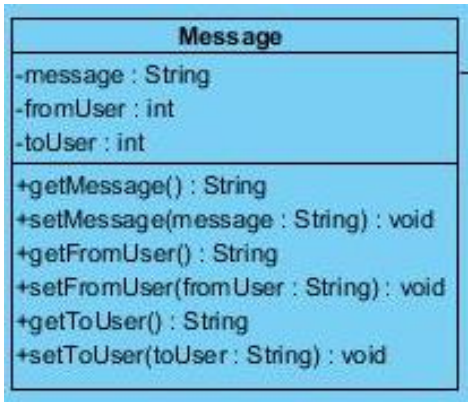

Trade Subsystem
Message
-message : String
-fromUser : int
-toUser : int
+getMessage() : String
+setMessage(message : String) : void
+getFromUser() : String
+setFromUser(fromUser : String) : void
+getToUser() : String
+setToUser(toUser : String) : void
Messages belong to Client
Client
-username : String
-password : String
-name : String
-surname : String
-email : String
-university : String
-userId : int
+getUsername() : String
+setUsername(username : String) : void
+getPassword() : String
+setPassword(password : String) : void
+getName() : String
+setName(name : String) : void
+getSurname() : String
+setSurname(surname : String) : void
+getEmail() : String
+setEmail(email : String) : void
+getUniversity() : String
+setUniversity(university : String) : void
+getUserId() : int
+setUserId(userId : int) : void
Admin
+deleteItem()
User
Items belong to clients
Book
Notes
Furniture
Item
-category : String
-description : String
-price : int
-condition : String
-photo : ImageIcon
-itemid : int
+getCategory() : String
+setCategory(category : String) : void
+getDescription() : String
+setDescription(description : String) : void
+getPrice() : int
+setPrice(price : int) : void
+getPhoto() : ImageIcon
+setPhoto(photo : ImageIcon) : void
+getCondition() : String
+setCondition(condition : String) : void
+getItemid() : int
+setItemid(itemid : int) : void

Message Class

Message class stands for when the user wants to send a message to the seller. This class has 3 attributes.

- message: The message type is string. This attribute includes the message that will be sent to the seller by the user.
- FromUser: This attribute stands for the id of the user who sends the message.
- toUser: This attribute stands for the id of the user who will receive a message.

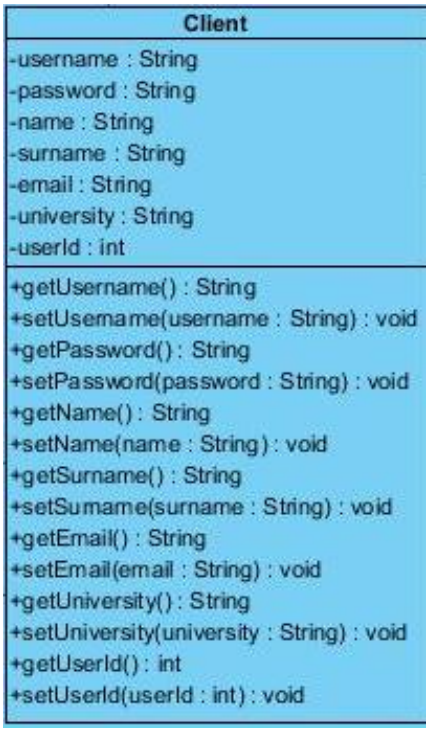
Message class consists of 6 methods. Those are getter and setter methods for all of its attributes.



Client Class

Client class implements the User interface. Client class has 7 attributes which are username, password, name, surname, email, university, userId and an array of messages. Only the typle of userId is int, others' are String. This class has plenty of methods that are getter and setter methods for all of its attributes.

- Username, password, name, surname stand for the client's usernamei password, name and surname, respectivelyç
- Email stands for the e-mail address of the user.
- University stands for university name of the user.
- userId will be used for database operations. The features of the user will be called by its ID from the database.
- Array of message stand for the messages that are received or sent by the system.



Admin Class

Admin class extends Client and implements User. Addition to features of Client class it has deleteItem() method that will enable Admin to delete an item that is inappropriate.

Item Class

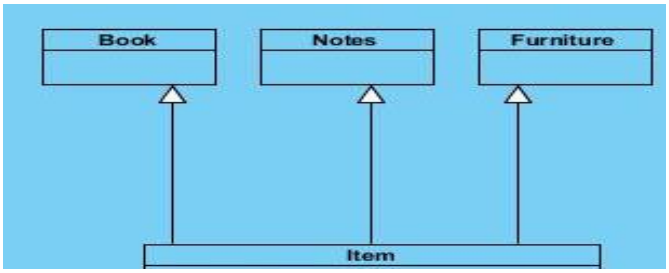
Item class has 6 attributes that are category, description, price, condition, photo, and itemId.

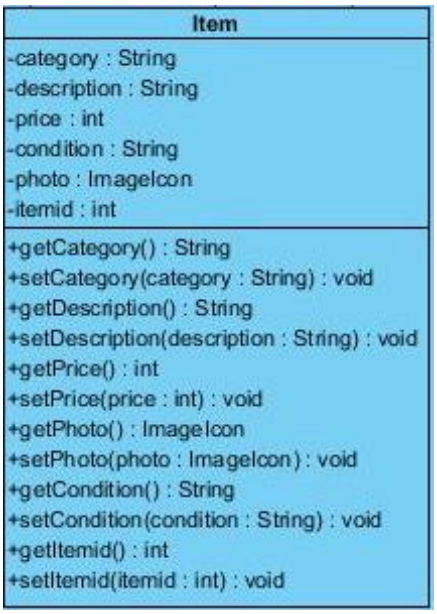
- Category stands for category of the item, for example book, notebook, or furniture.
- Description stands for detailed description of the item that is written by its seller.
- Price stans for the price of the item.
- Conditions states that the item is etiher zero or second hand.
- Photo is an ImageIcon that stands for the photograf of the item. It is added by its seller.
- itemId will be used for database operations. Operations will be done by using the ID of the item.

This class has getter and setter methods for all its attributes.

Book, Notes, Furniture Classes

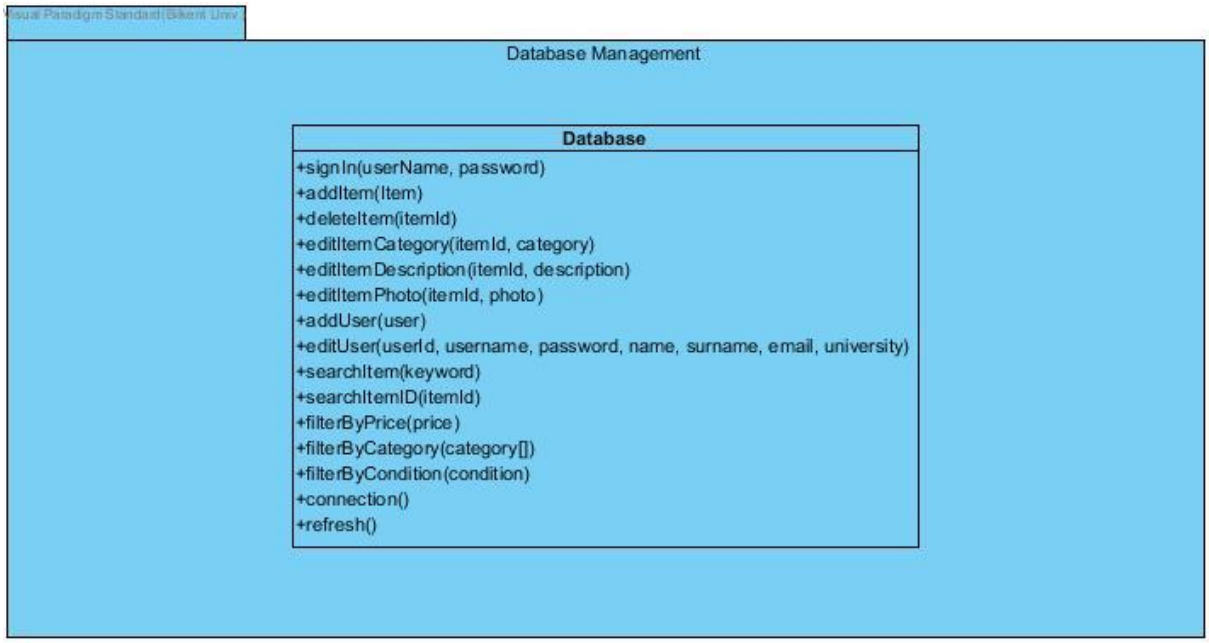
Those classes extends Item Class. The inherits all of features of Item.




Item
-category : String
-description : String
-price : int
-condition : String
-photo : ImageIcon
-itemid : int
+getCategory() : String
+setCategory(category : String) : void
+getDescription() : String
+setDescription(description : String) : void
+getPrice() : int
+setPrice(price : int) : void
+getPhoto() : ImageIcon
+setPhoto(photo : ImageIcon) : void
+getCondition() : String
+setCondition(condition : String) : void
+getItemid() : int
+setItemid(itemid : int) : void

3.2. Database Subsystem

In this subsystem, we have only one class called as “Database”. This class diagram of this



shown at figure below. Detailed descriptions are given also.

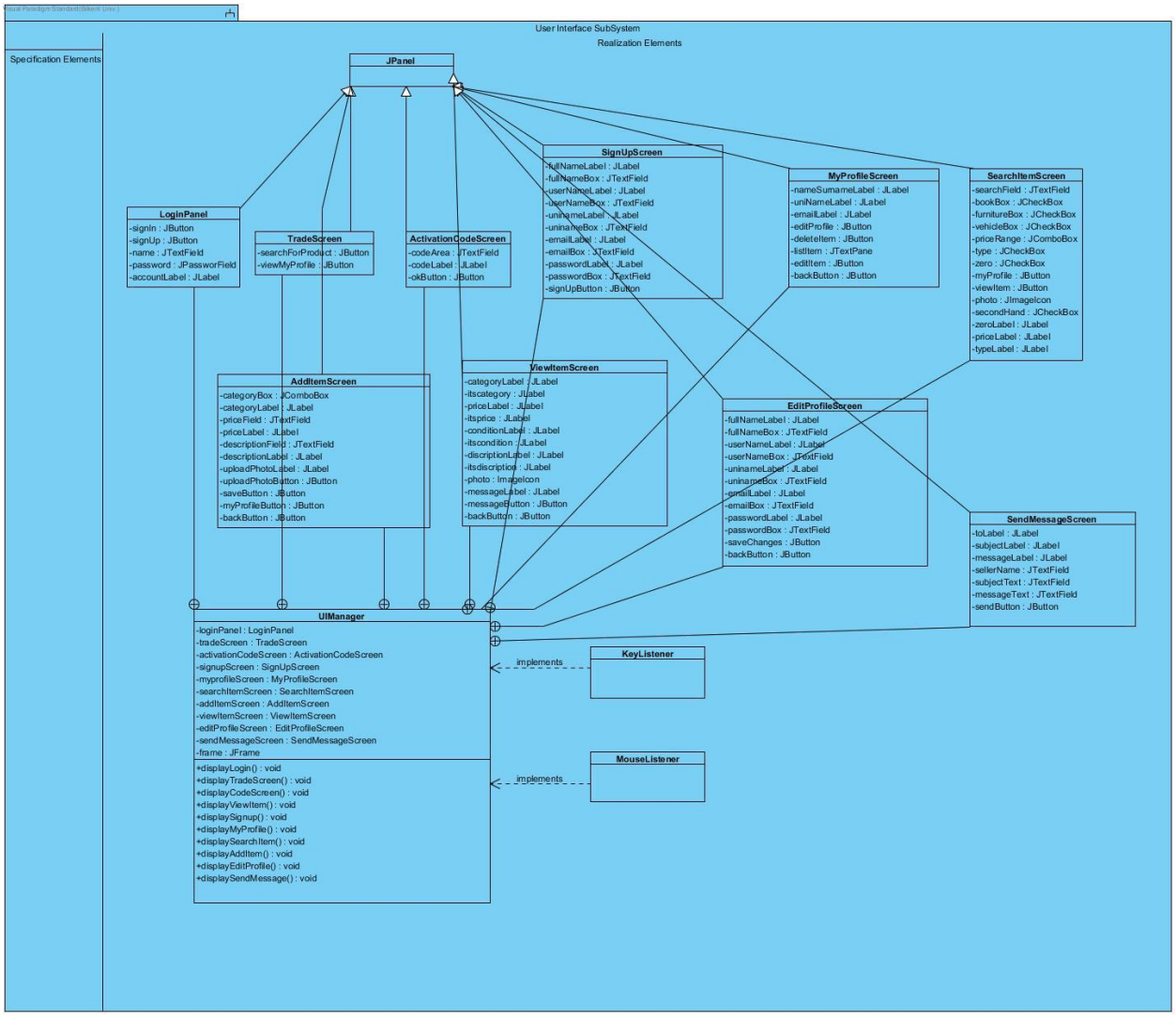
This class contains only methods and all attributes ae just temporary ones to use these methods. All methods are explained below.

- signIn: It takes userName and password that is given by the user and search for these data on database to find if given information is correct or not. If it is, then this method takes all necessary data from database to create proper objects on backend part of the project.
- addItem: This method takes Item object and by using the item’s attributes creates proper MySQL query to send database. By this way the data of the item object is transferred to database.

- deleteItem: It takes an item and create proper MySQL query to find the item on database and delete it.
- editItemCategory, editItemDescription, editItemPhoto: These three methods are getting proper attribute/s and change the wanted property of the item.
- addUser: This method is invoked when a new user wants to sign up. It takes the user object and by preparing appropriated MySQL query to forward given data to database.
- editUser: This part takes all information about the user and update them on database.
- searhItem, searchItemId, filterByCategory, filterByCondition: These methods are used to make searching and finding wanted items easier and practice way.
- connection: This method is used to create connection between code part of the project and the database.
- refresh: It is used to refresh all data that is taken from the database. By this way any change on database by a user/admin could be seen on his/her program or any other program that is used by any other user/admin.

3.3. User Interface Subsystem

This subsystem gives the program visual components. It will only be used for the various design attributes for the program. Also, it can enable users for the shifts between panels. The UIManager class will be used for the main component of the subsystem that handles the shifts between panels and holds information about other panels as attributes.



KeyListener and MouseListener Classes

Those classes are used for the interaction between keyboard and mouse and the program.

LoginPanel, TradeScreen, ActivationCodeScreen, SignUpScreen, MyProfileScreen, SearchItemScreen, AddItemScreen, ViewItemScreen, EditProfileScreen, SendMessageScreen Classes

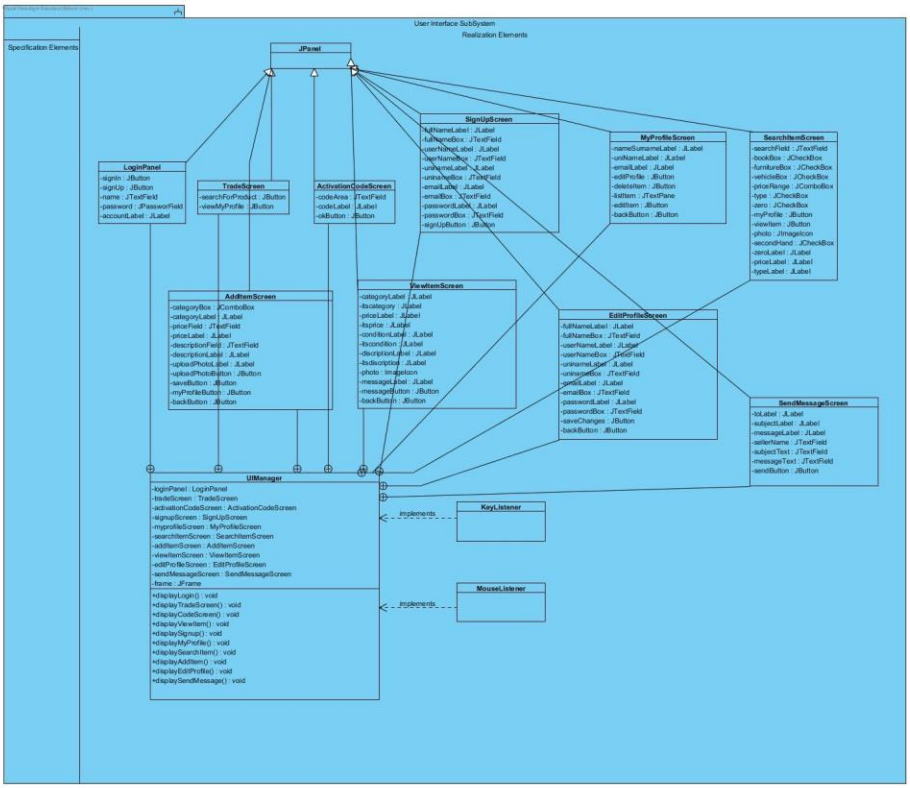
Those classes will not be described in detail. However, those classes will hold different information about how the panels would be. Each of these classes will hold a panel and when it is clicked or written via KeyListener and MouseListener classes, they will be shown via their visibility attributes.

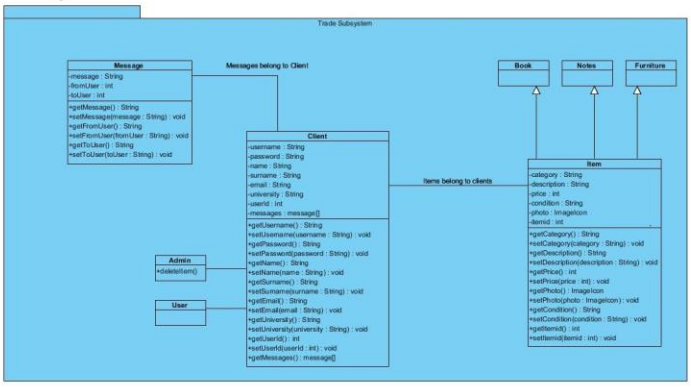
UIManager Class

- Display methods: Those methods are going to get used for shifts between panels. When the click happens via listener classes, the visibility of all pages except the wanted page will set to false.
- frame: This will be the main frame for all visual content. They will be shown on one main frame.

- The …panel attributes: Those attributes will hold information about panel classes, which are made by different attributes and visual design.

3.4. Detailed System Design


User Interface SubSystem
Realization Elements
Specification Elements
JPanel
LoginPanel
TradeScreen
ActivationCodeScreen
SignUpScreen
MyProfileScreen
SearchItemScreen
AddItemScreen
ViewItemScreen
EditProfileScreen
SendMessageScreen
UIMenager
KeyListener
MouseListener
implements
implements


Trade Subsystem
Message
Messages belong to Client
Book
Notes
Furniture
Client
Items belong to clients
Item
Admin
User


Database Management
Database