CS 319 - Object-Oriented Software Engineering

Analysis Report

BestTrade

Group 3M

Berk Türk - 21302570

Ezgi Çakır - 21402334

Irmak Tural - 21301099

İbrahim Berker Kırdök - 21502870

İlhami Özer - 21300828

Contents

1. Introduction

BestTrade is a desktop application for trading stuff like books, lecture notes in between students. The main purpose of this project is to give a platform to students from various universities to exchange their stuff by wasting less effort. The users can enter the system via their e-mail address and their password. For selling items, students must enter their products' details into the system. For buying items, users can use various filtering options like search engine and check buttons for types of items. Also, the existing items' details can be edited and deleted by the user.

The main part of this trading system will be implemented by OOP and it is going to feature different topics from the course, like the models. Also, this application is going to use a connection from a server to Java, for using the database for storing the information of items.

The first design of the application will be described briefly in this report. The report will look over to the application about its features, and then it will separate into different details like functional, non-functional and pseudo requirements of the project. After that, the report will show various system models of this application, scenarios, and a look to the how user interface is going to be.

2. Proposed System

2.1 Overview

Java and MySQL will be used for implementing BestTrade, and the aim of the program is to trade items between users. At the beginning, users have to sign up to the application by using their specified username and password. For entering the application the users have to enter that username and password. After entering system, the users can sell their items based on the attributes and search their desired item by filtering those attributes.

The application will store the attributes of items and users. The users' attributes will be stored as name, surname, username, password, e-mail and university name. The items' attributes are book, furniture, lecture notes, technological product. After adding attributes, the users can edit their attributes from their profile pages.

2.2 Functional Requirements

2.2.1 Sign Up

The user has to sign up the system in order to use the program. In the screen, the program asks for name, surname, a unique username, password, e-mail, and university name. For example, a user whom chooses a username called bilkent06 cannot be chosen by anyone else as

a username and cannot use the same e-mail for signing up for different accounts.

2.2.2 E-Mail Activation Code

After signing up, the program will send an activation code to given e-mail automatically. The user who wants to sign up must enter the given code.

2.2.3 Sign In

For signing in, there are two choices: Admin or User. The admin can reach every aspect of the application. For entering the system, the matching username and password is needed.

2.2.4 Add Item

The user can add item and have to choose the item type. The user is given a combobox for choosing the item type, a checkbox for indicating whether it is second hand or not, a button for uploading the item's photo, and a textbox for entering the price and a description about the item. For example, a user could add a book which is second hand and 20 TL. As description it could be written "It is an original copy, no damage."

2.2.5 Delete Item

If the user enters his/her own profile, he/she can be able to delete the desired item by clicking "Delete" button that is next to the item list.

2.2.6 Edit Item

By clicking the “Edit” button that is next to “Delete” button, the user can be able to edit the information of their item. For instance, the user can modify the product’s photo and price.

2.2.7 Choose Item

For buying an item, the user has to choose the item by selecting the desired attribute. For example, a user can find a book priced by 20 liras by checking the book checkbox and entering the price range of 10-30 liras.

2.2.8 Send Message

While choosing an item, the user can be able to message the seller for asking further information about the items.

2.2.9 Look Profile

Users can look their own profile by using “My Profile” button on various screens like buying and choosing items and users can edit their own profiles. Also, admin can see all users’ profiles.

2.3. Non-Functional Requirements

2.3.1. Internet connection does not lead to data lose

Since one of our computers work as a server and keeps all database information on the disk of the server computer. Thus, internet connection cannot cause any information on the database.

2.3.2. User Friendly Design

All buttons and actions could be interacted with user will be described on their title as much as possible so that user could not confused about how he/she uses the program. Furthermore, by using simple and clear design and interface prevent any confuse of the user.

2.3.3. Protection of the data

During coding part, encapsulation will be used properly to prevent any private data by who has not permission to see them. Moreover, precautions will be on board to prevent SQL injection which is so popular to access database without permission.

2.3.4. Reusability

Each part of the system will be modular so that, almost all class could be reusable for future projects. Especially, database and connection to it could be used for other projects. By coding classes as much as independent than others, reusability of the code will be improved. Thus, we will use dependency injection concept which led us control class dependencies from outside class so that no change will be required in a specific class which could be used for future projects.

2.4. Pseudo Requirements:

- The code will be written in Java.

- Queries of database will be MySQL.
- The program will be developed for desktop only.

3. System Design

3.1. Scenarios

3.1.1 Use Case Name: Sign Up

Primary Actor: User

Stakeholders and Interests:
-User need to sign up in order to sell/buy an item via BestTrade.

Pre-conditions: -
Post-condition: The enter activation code. After that, the system adds the user's account into the database.

Entry Condition: The user clicks on "Sign Up" button.
Exit Condition: The user click "<" button to return main page.

Success Scenario Event Flow:

1.The user is enabled to enter his/her name, surname, ,username, password, e-mail address, and university name.
2. The user is enabled to enter verification code which is sent to his/her e-mail address.
3. The user is able to sign in.

Alternative Flows:
A. If the user remembers that he/her already has an account, he/her can return to main page by clicking "<" button.

3.1.2. Use Case Name: Send Message to a Seller

Primary Actor: User

Stakeholders and Interests:
-The user wants to send a message to a seller to buy his/her product.
-System enables users to send messages each other.

Pre-conditions: The user need to log in.
Post-condition: The message is sent to the seller.

Entry Condition: The selects and views the item that he/she want to purchase, then he/she clicks the “send a message” button.
Exit Condition: The user click “<” button to return the page that sold items are displayed.

Success Scenario Event Flow:
1.The user click “send a message” button.
2.He/she writes her message into the text field.
3.He/she clicks “Send” button.
4. The message is sent to the seller.

Alternative Flows:
A. If the user wants to go back, he/her can return by clicking “<” button.

3.1.3. Use Case Name: Delete the Sold Item

Primary Actor: User

Stakeholders and Interests:
-The user wants to delete his/her sold item.
-The system deletes the item from the user’s profile.

Pre-conditions: The user need to log in and view his/her profile.
Post-condition: The selected item is deleted from the sold item list of the user.

Entry Condition: The user views his/her profile and clicks on “edit my profile” button, then “delete item” button.
Exit Condition: The user clicks “save” button, the “<” button to return his/her profile page.

Success Scenario Event Flow:
1.The user wants to delete his/her sold item.
2.The user enters his/her profile page.
3.The user clicks “edit my profile” button, then “delete item” button.
4.The item is deleted from the user’s sold item list by the system.

Alternative Flows:
A. If the user wants to go back, he/her can return by clicking “<” button.
B.If the user wants to edit the item, he/she can clicks on “edit the item” button.

3.1.4. Use Case Name: View an Item

Primary Actor: User

Stakeholders and Interests:
-The user wants to view an item which he/she wants to display its detailed description.

Pre-conditions: The user need to log in and enter a page that item are displayed.
Post-condition: -

Entry Condition: The user clicks on “search item” button.
Exit Condition: The user clicks on “<” button to return the search page.

Success Scenario Event Flow:
1.The user wants to display detailed description of an item which he/she may want to buy.
2.The user clicks on “view the item” button.
3.The detailed description of the item is displayed.

Alternative Flows:
A. If the user wants to go back, he/her can return by clicking “<” button.
B.If the user wants to send a message to the seller he/she may clicks on “send a message” button.
C. If the user wants comment on the item, she/he may clicks on “comment” button.

3.1.5. Use Case Name: Trade Up

Primary Actor: User

Interest: The user wants to add product which will be sold in the program.

Pre-condition: User has to be log in.

Post Condition: The feature of the product which will be sold by user will be added into the database system.

Entry Condition: User clicks on “Sell Products” in the main menu which will be appear when user log in the system.

Exit Condition: User clicks on “Save” in the trade up screens.

Successful Scenario Event Flow:

1. User wants to sell item
2. Application returns trade up screen
3. User types the feature of products
4. The product will be added into database system.

Alternative Flows:

When the user click on the save button, system renders the **“My Profile”** menu screen.

3.1.6. Use Case Name: Edit Item

Primary Actor: User

Interest: The user wants to edit the features of products which are already in the database system.

Pre-Condition: User has to be log in.

Post Condition: Changes will be updated on the database system.

Entry Condition: User clicks on “Edit Item” in the “My Profile” screen.

Exit Condition: User clicks on “Back” in the My profile screens.

Successful Scenario Event Flow:

1. User wants to change the feature of products.
2. User clicks on “Edit Item” in the “my profile” menu.
3. User will be able to change features of item which are type, price, name and category.
4. The changes will be updated on the database system.

3.1.7. Use Case Name: Sign In

Primary Actor: User

Interests: User wants to login the system.
Pre Condition: User has to be in Login screen, enter username and password.
Post Condition:. User has to click on Sign In button.

Entry Condition: User must be in the program.
Exit Condition: -

Successful Scenario Event:

1. User wants to login the system.
2. User enters username and password.
3. User clicks on Sign In button.

3.1.8. Use Case Name: Search for Item

Primary Actor: Admin/User

Interests: User want to search for items.
Pre Condition: User has to be logged in and must be in search screen.
Post Condition:

Entry Condition: User has to click on Search Item button in Trade screen.
Exit Condition: User has to click on back button.

Successful Scenario Event:

1. User wants to search for item.
2. User clicks on Search Item button in Trade screen.
3. User enters name of item to Search Box.
4. User can also make filters by choosing Filters from check boxes.
5. User presses Enter.
6. The items are listed.

3.1.9. Use Case Name: View Profile

Primary Actor: User

Interests: User want to display his/her profile that includes his/her information and the items that he/she is selling.
Pre Condition: User has to be signed in.
Post Condition: -

Entry Condition: User has to click on My Profile button.
Exit Condition: User has to click on back button.

Successful Scenario Event:

1. User wants to display his/her information
2. User clicks on My Profile button.
3. The information is displayed.

3.1.10. Use Case Name: Edit Profile

Primary Actor: User

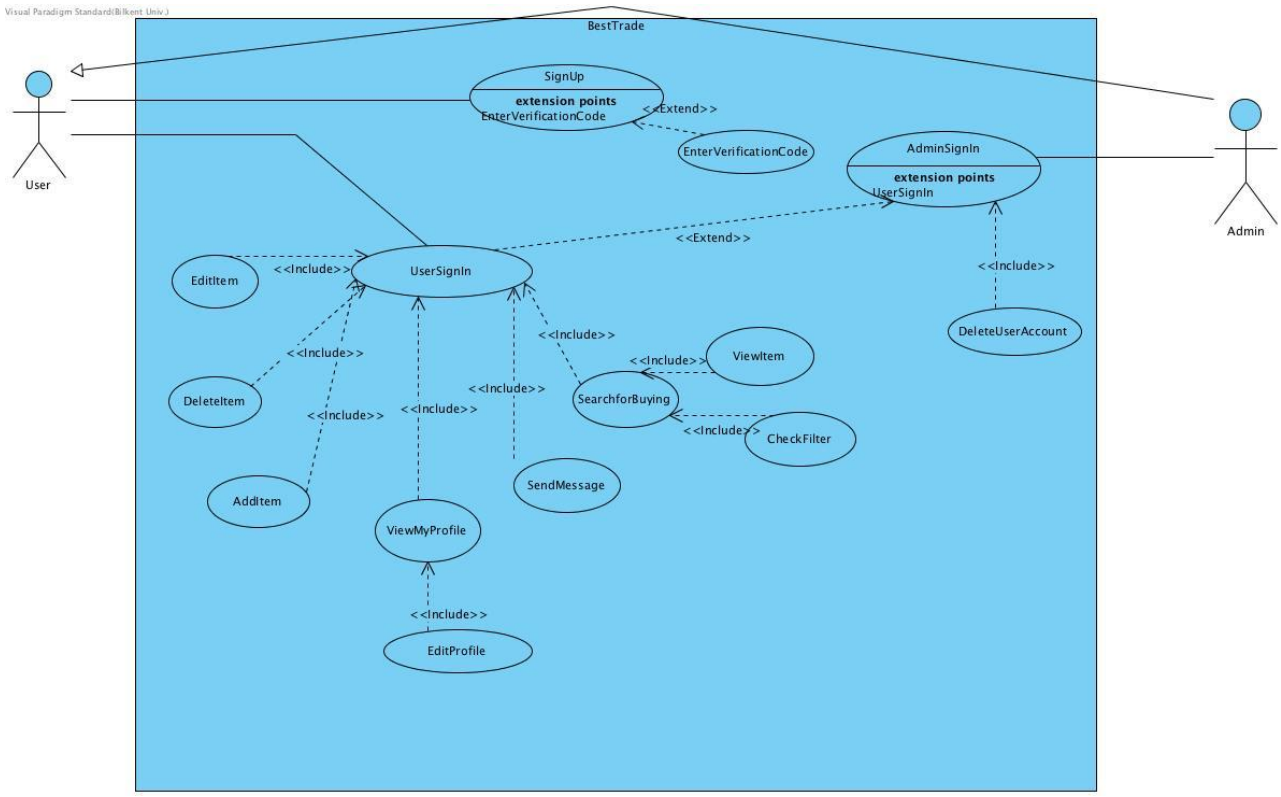
Interests: User wants to change his/her information.
Pre Condition: User has to be in My Profile screen
Post Condition: The changed information is updated by the system.

Entry Condition: User must click on Edit Profile button in My Profile screen.
Exit Condition: User must click on back button.

Successful Scenario Event:

1. User wants to edit his information.
2. User clicks on Edit Profile button in My Profile screen.
3. The Edit Profile screen is displayed.
4. Users can change their information by entering their name, username, University Name, E-mail Address or Password.
5. User clicks on Save Changes button.

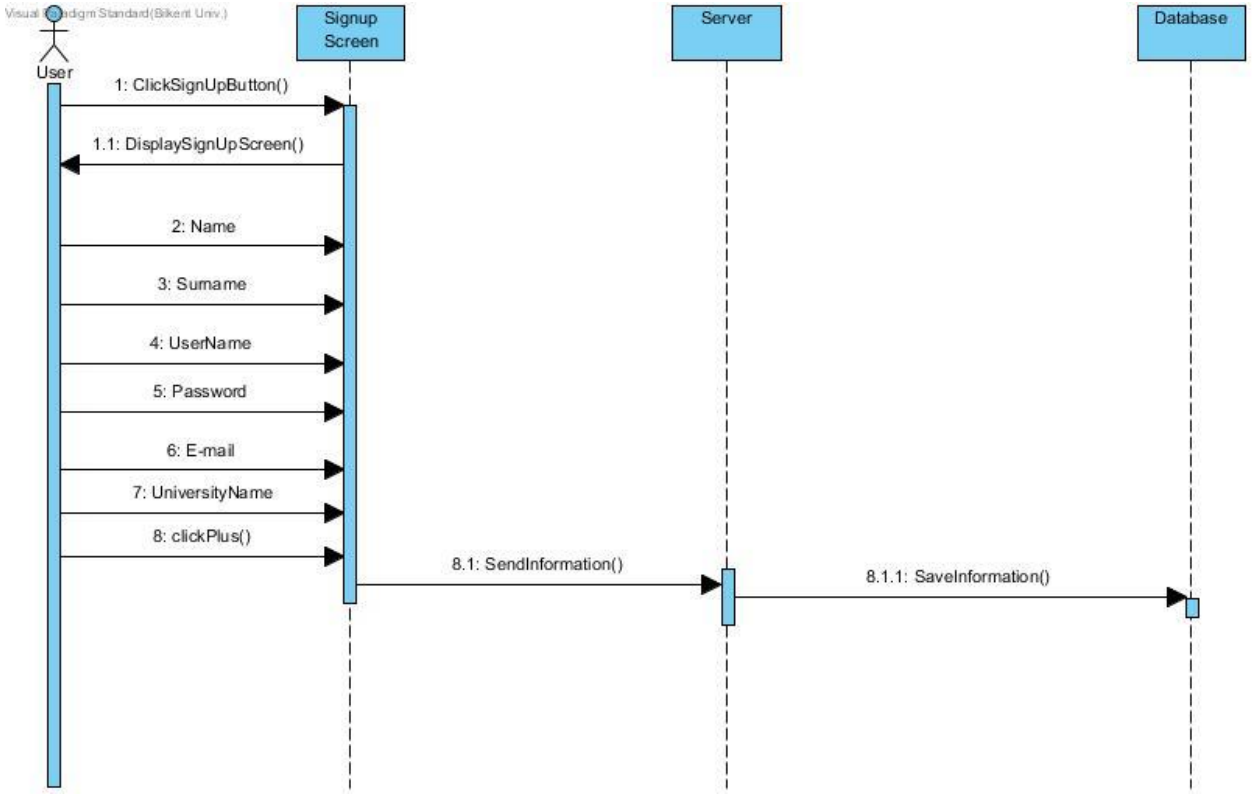
3.2. Use Case Model



3.3. Dynamic Models

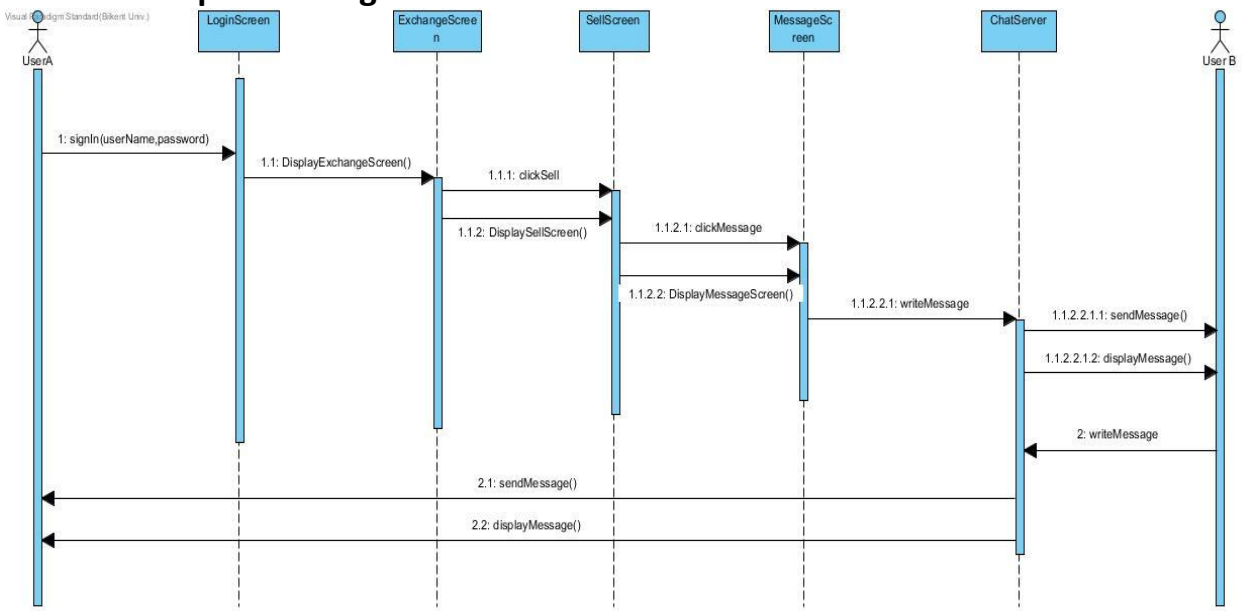
In this section, we intend to use the “Scenarios” part of the report in order to use the sequence diagrams.

3.3.1. Sequence Diagram 1



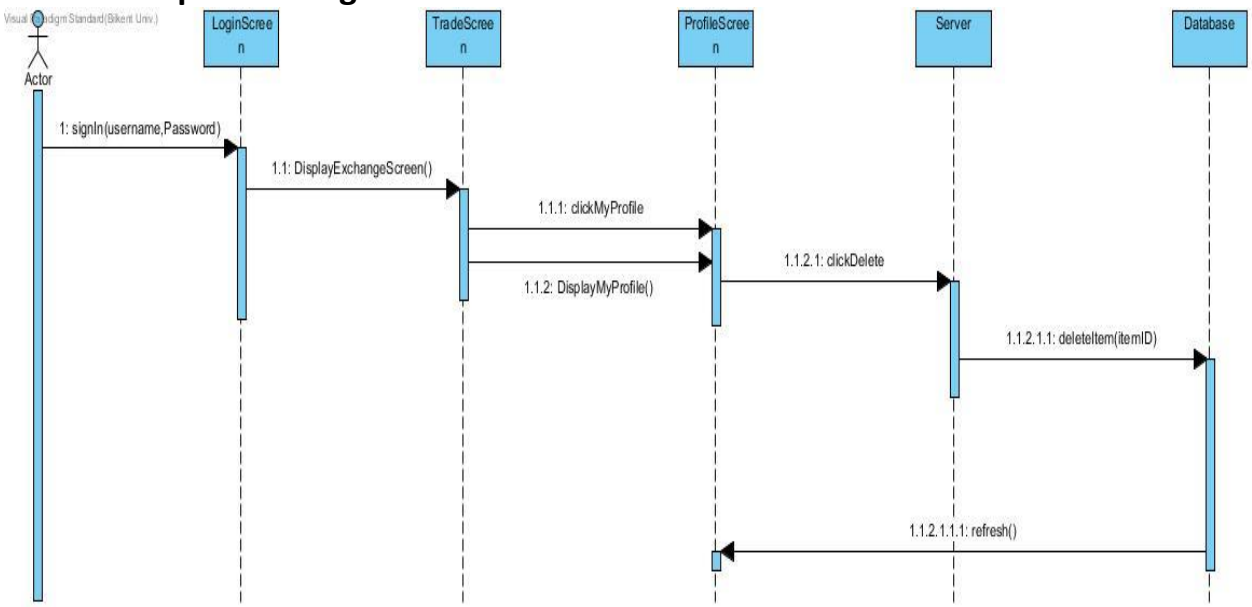
In this part, the user first enters the signup section of the application, and enters the information that enables user to sign. After that, the information is going to store in database.

3.3.2. Sequence Diagram 2



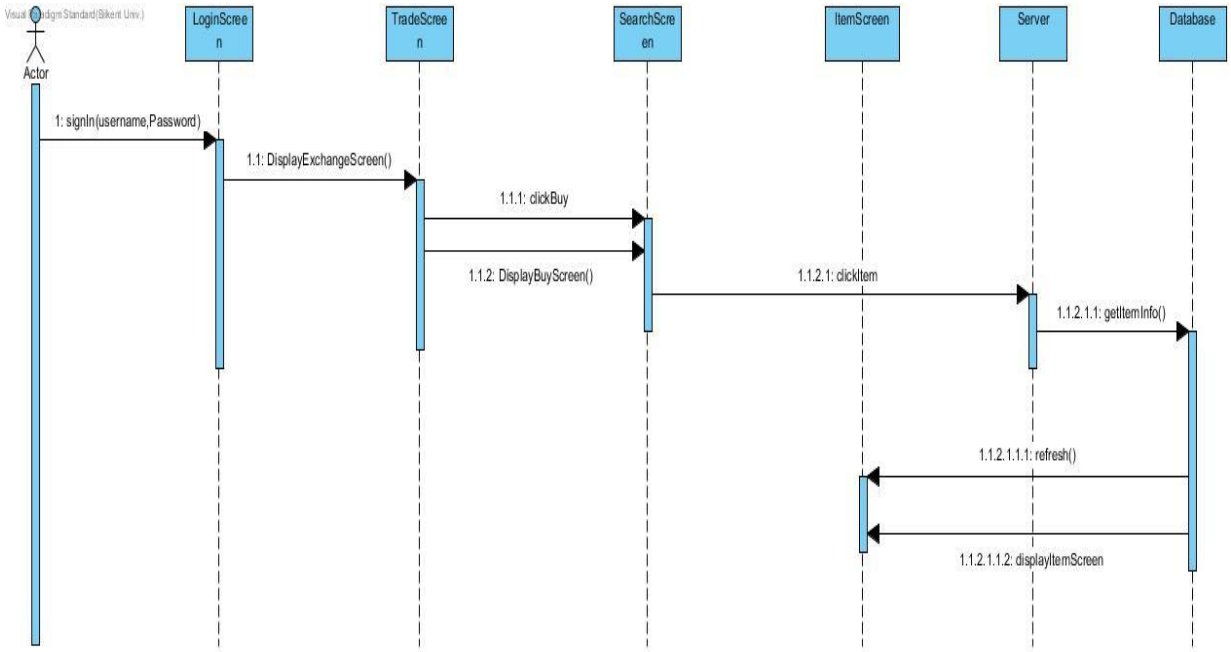
After the sign in process, the users can enter the buying option to see the items. In that section, the user can choose “Send Message” option to send the user a message, where they can chat in a new message server.

3.3.3. Sequence Diagram 3



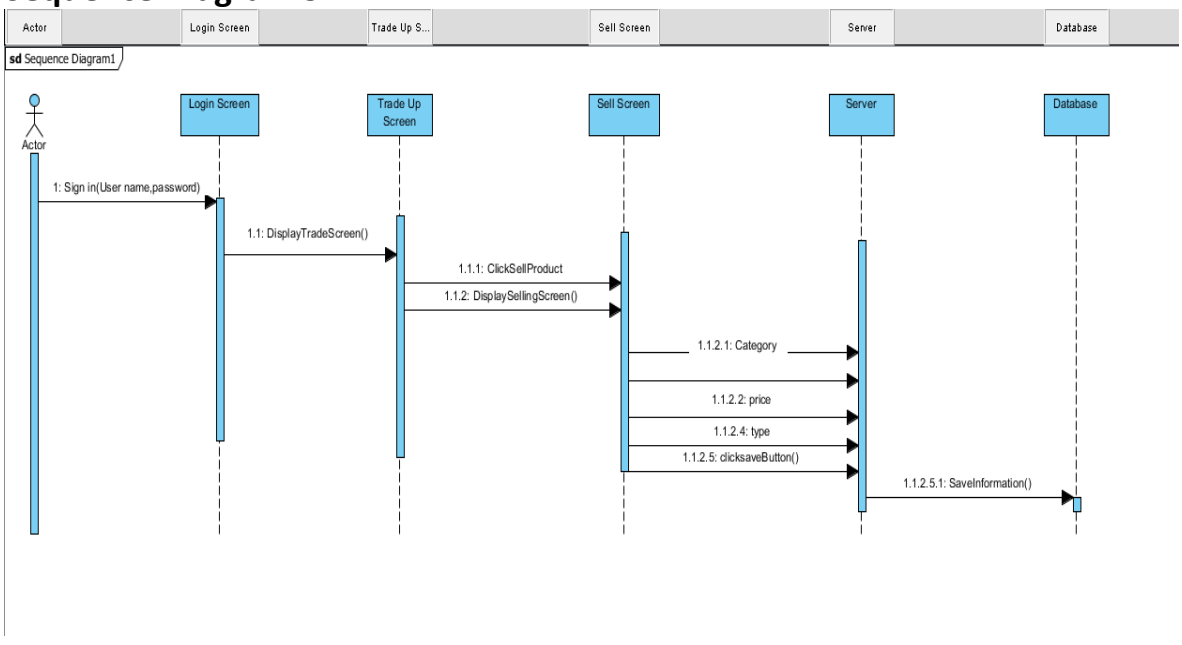
In this scenario, users enter to the system and it directs them to the main trade screen. In the main trading screen and all other screens other than sign in and sign up screen, the users can click their “My Profile” button on upper right, and it directs them to their profile page. In the profile page, the users can delete their items from the database by clicking the “Delete” button that is next to the item names. After the click, the information will be deleted from the database by a given query and the database will refresh itself.

3.3.4. Sequence Diagram 4



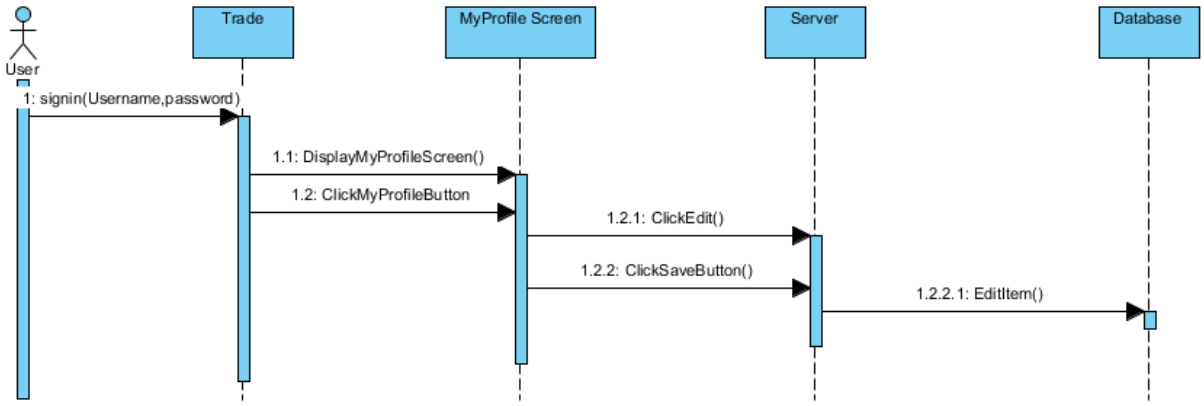
In this scenario, the user signs in to the system, and the system automatically sends user to trade screen, where the user can choose buy option and sees the items. In this section, users can click the items and a new screen opens to show the item to all users. The item information is taken from the database, and each time it gets the information, the database will refresh itself.

3.3.5. Sequence Diagram 5



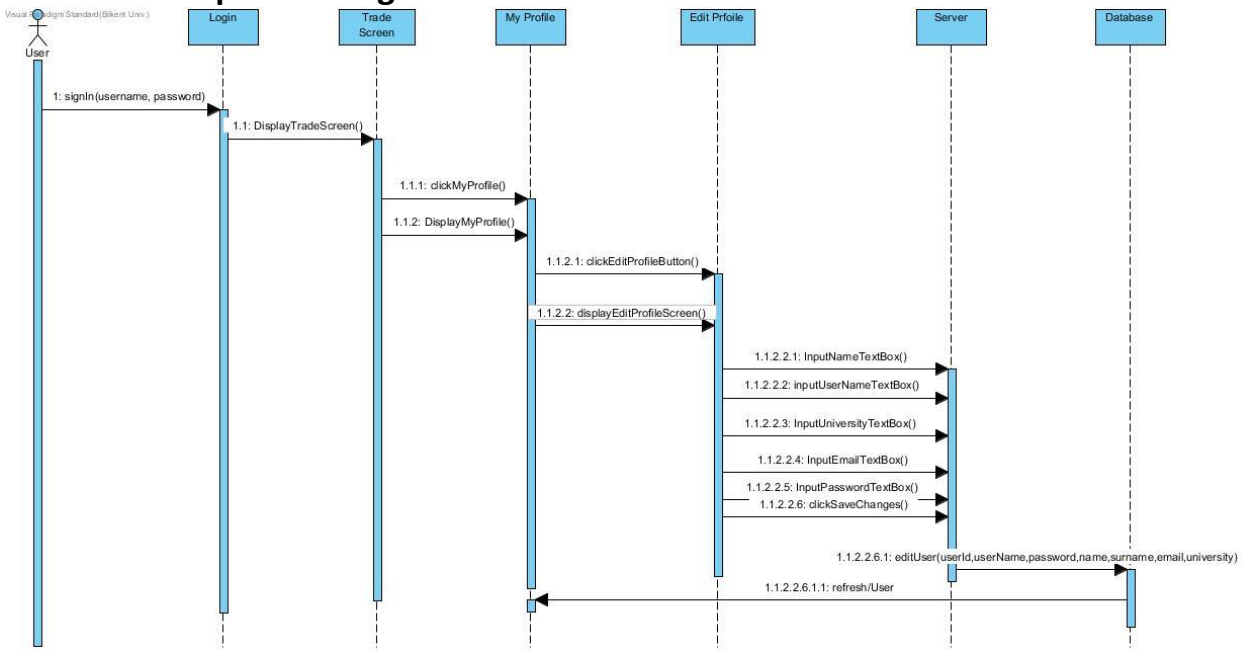
User decided to sell a product via BESTTRADER. First off the user enters the system with an account which is already created by himself/herself. After the log in process, the system directs the user to trade up screen and he/she clicks on sell button. After that the user types what are the features of the product and then the product will be added into the database system.

3.3.6. Sequence Diagram 6



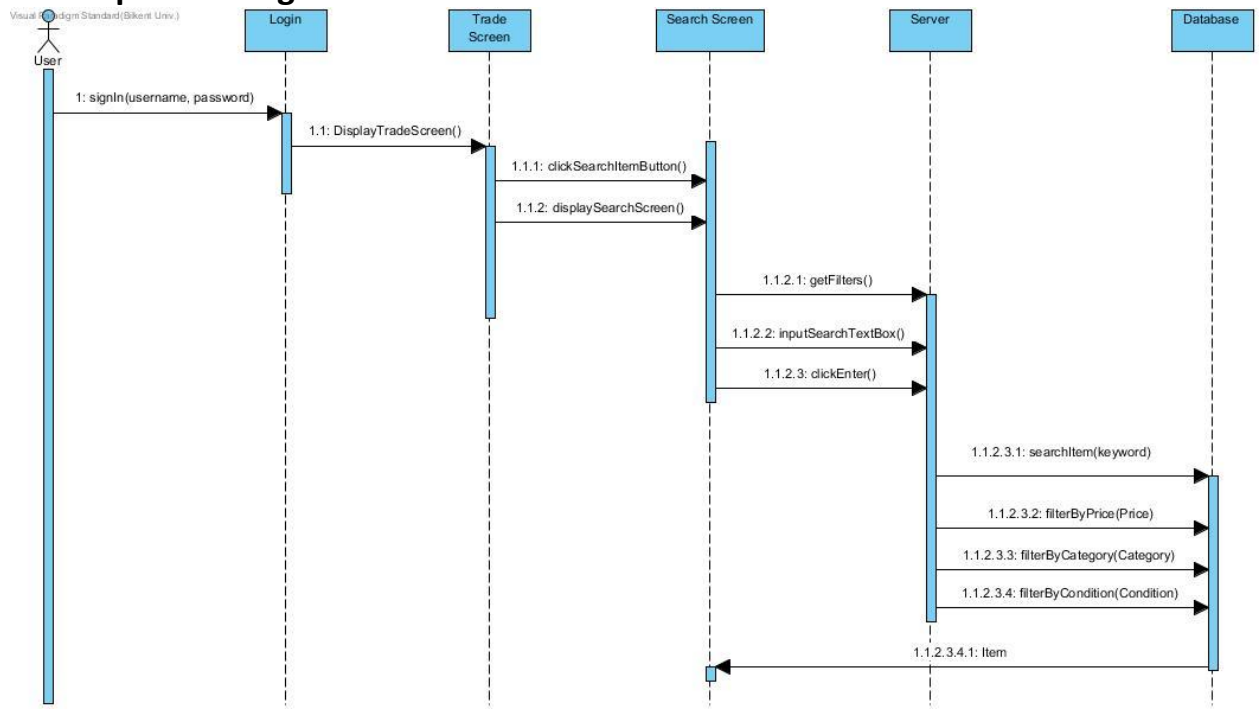
User decided to change his/her product price on the application. After he/she entered the system she viewed his/her profile and clicked the edit item button. And he/she decreased the price of his/her product.

3.3.7. Sequence Diagram 7



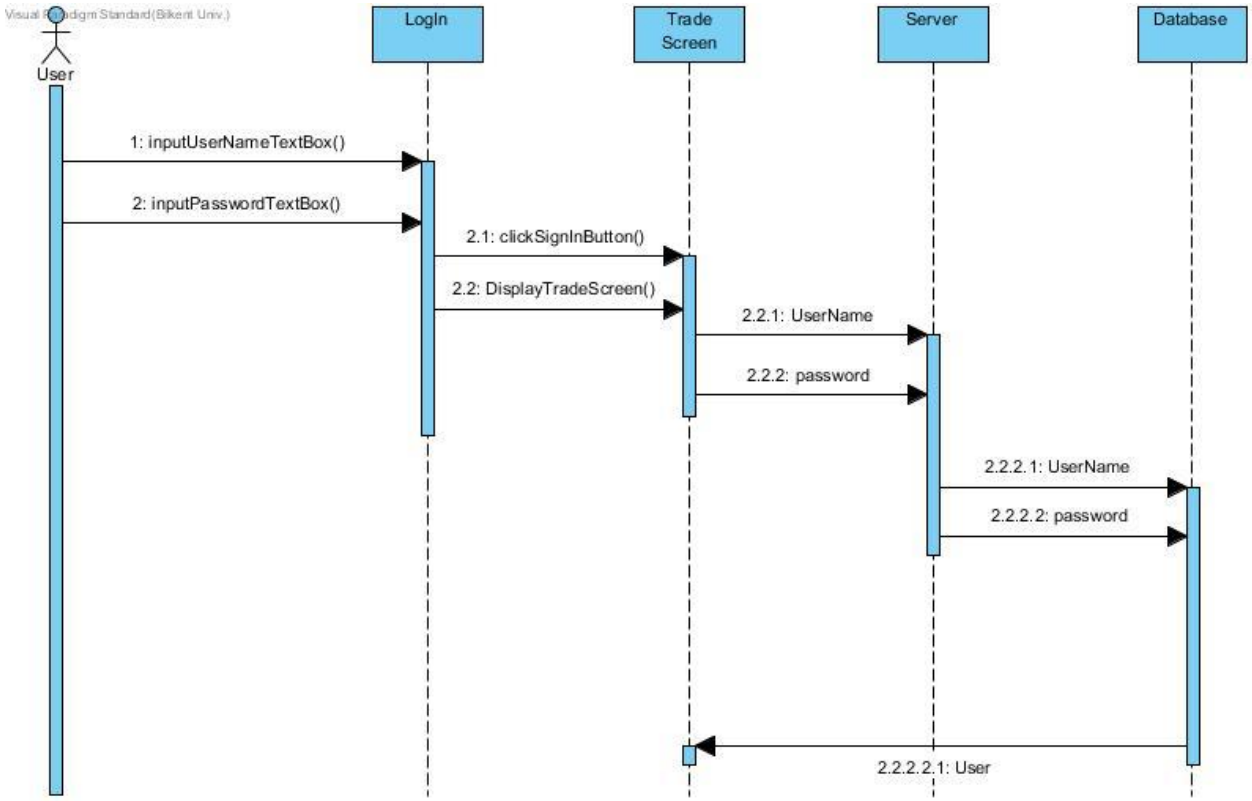
User wants to change his/her information. After the user signed in, DisplayTradeScreen() method is called. Trade screen is displayed.User clicks on My Profile button and My Profile screen is displayed. By clicking Edit Profile button displayEditProfileScreen() is called and the screen is displayed. User enter the changes that he/she wants to make by inputTextBox() methods for the different features of textboxes. Then user clicks Save Changes button. By the clickSaveChangesButton() the changes go through to server and by the method editUser(userId,userName,password,name,surname,email,university), user information is updated in database according to taken parameters. It is refreshed and User object returns to My Profile screen. The information had changed and it is displayed.

3.3.8. Sequence Diagram 8



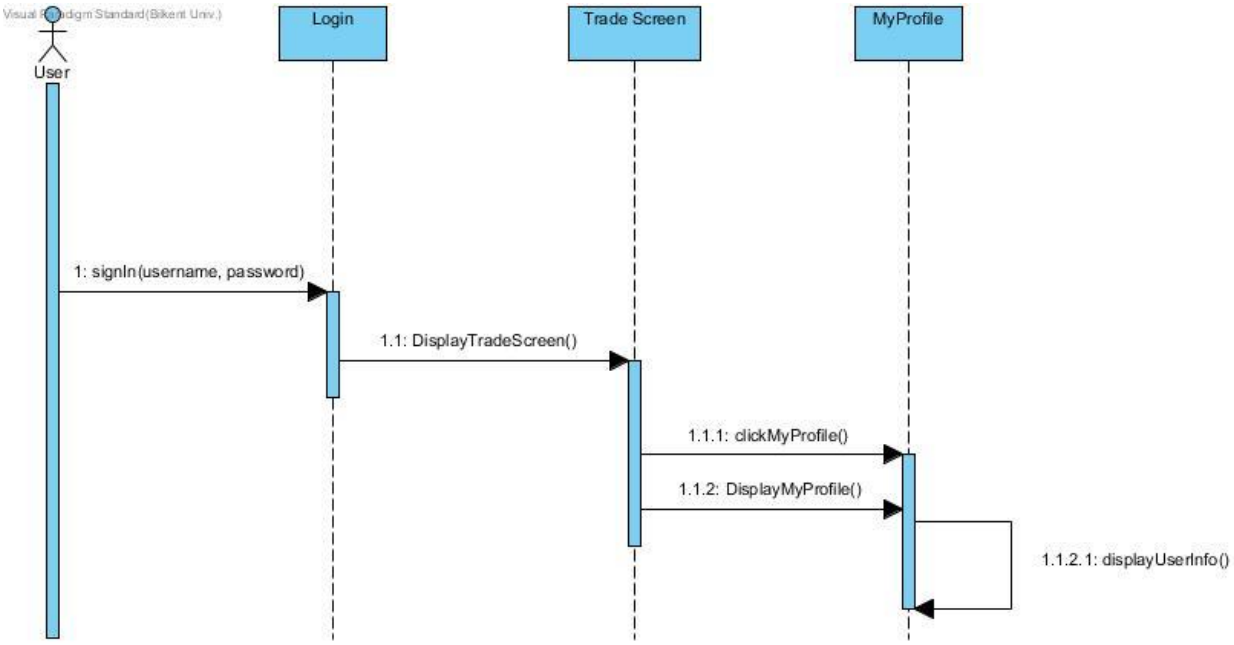
After the user signed in, DisplayTradeScreen() method is called. Trade screen is displayed. Trade screen includes Search Item button. User wants to search item according to his demand. When the user clicks displaySearchScreen() is called and the Search Screen is displayedwhich provides user to search for a item with filters. When the user tick the checkboxes the getFilters() method is called and by InputSearchTextBox() method the system receives the name/keyword of the Item which is entered by user. Then user clicks Enter. The information goes to server and the methods searchItem(keyword), filterByPrice(Price), filterByCondition(Condition), filterByCategory(Category) are called according to parameters that comes from User. The items and features that matches with that parameters in database, returns as an Item objects to Search screen and they are displayed. User can see the items that matches with user's inputs.

3.3.9. Sequence Diagram 9



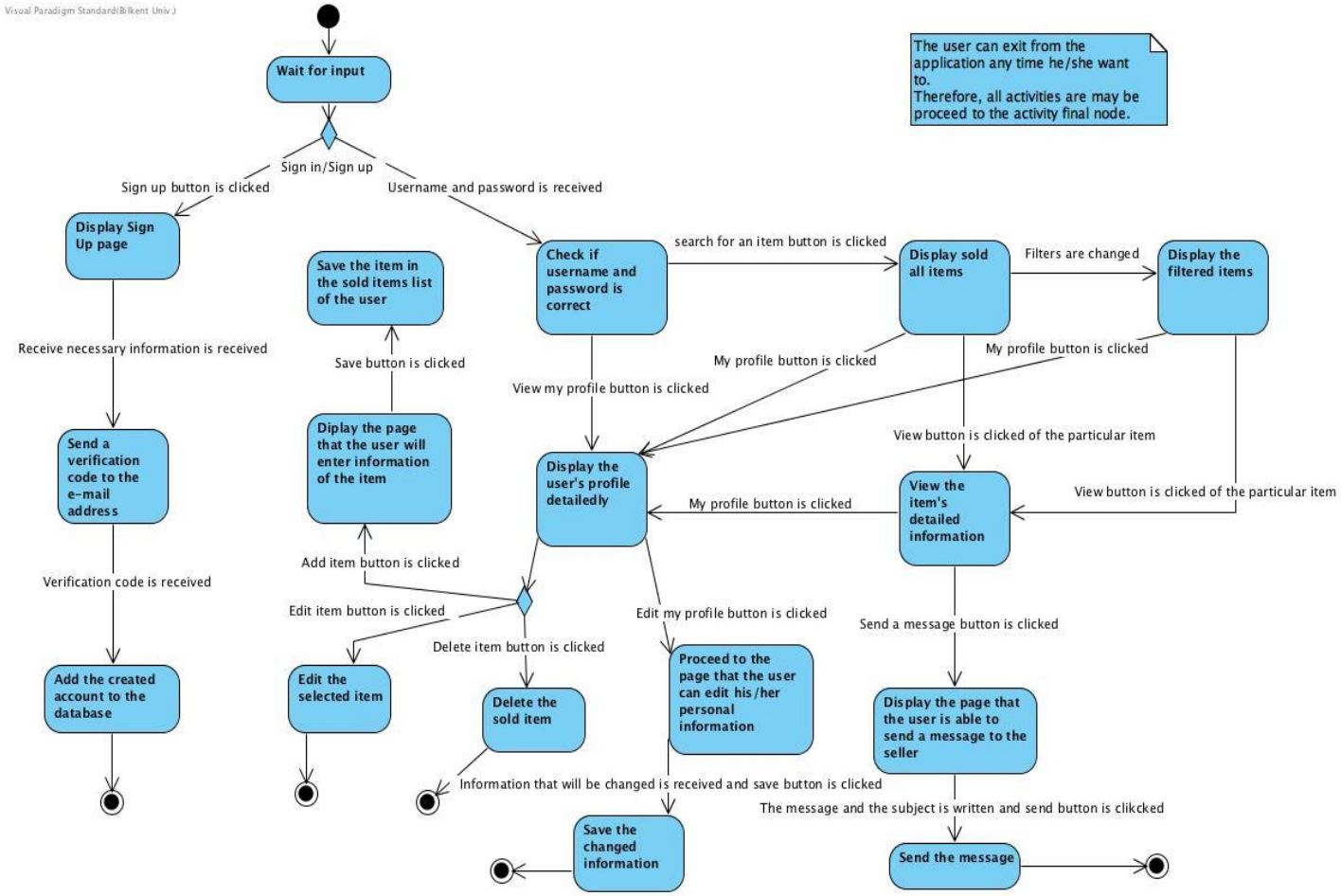
When user is in program, firstly, Log in screen is displayed. By the methods inputUserNameTextBox() and inputPasswordTextBox, user enter his username and password. When user click Sign In button, DisplayTradeScreen() method is called and the Trade Screen is displayed. The input username and password are sent to server and checked in database if they are included or if they are matched with each other. If the conditions are provided User object returns.

3.3.10. Sequence Diagram 10



After the user signed in, DisplayTradeScreen() method is called. Trade screen is displayed. User wants to see his/her profile that includes information and the items that are being sold. User clicks on My Profile button on Trade Screen. DisplayMyProfileMethod() is called and MyProfile is displayed. Then displayUserInfo() is called and the information is displayed. (At that stage database is not required, because when the user signed in, all information was received.)

3.3.11. Activity Diagram



When the user opens the application, the system waits for input from the user. The user may either sign in –if he/she has an account- or sign up –if she/he does not have an account-.

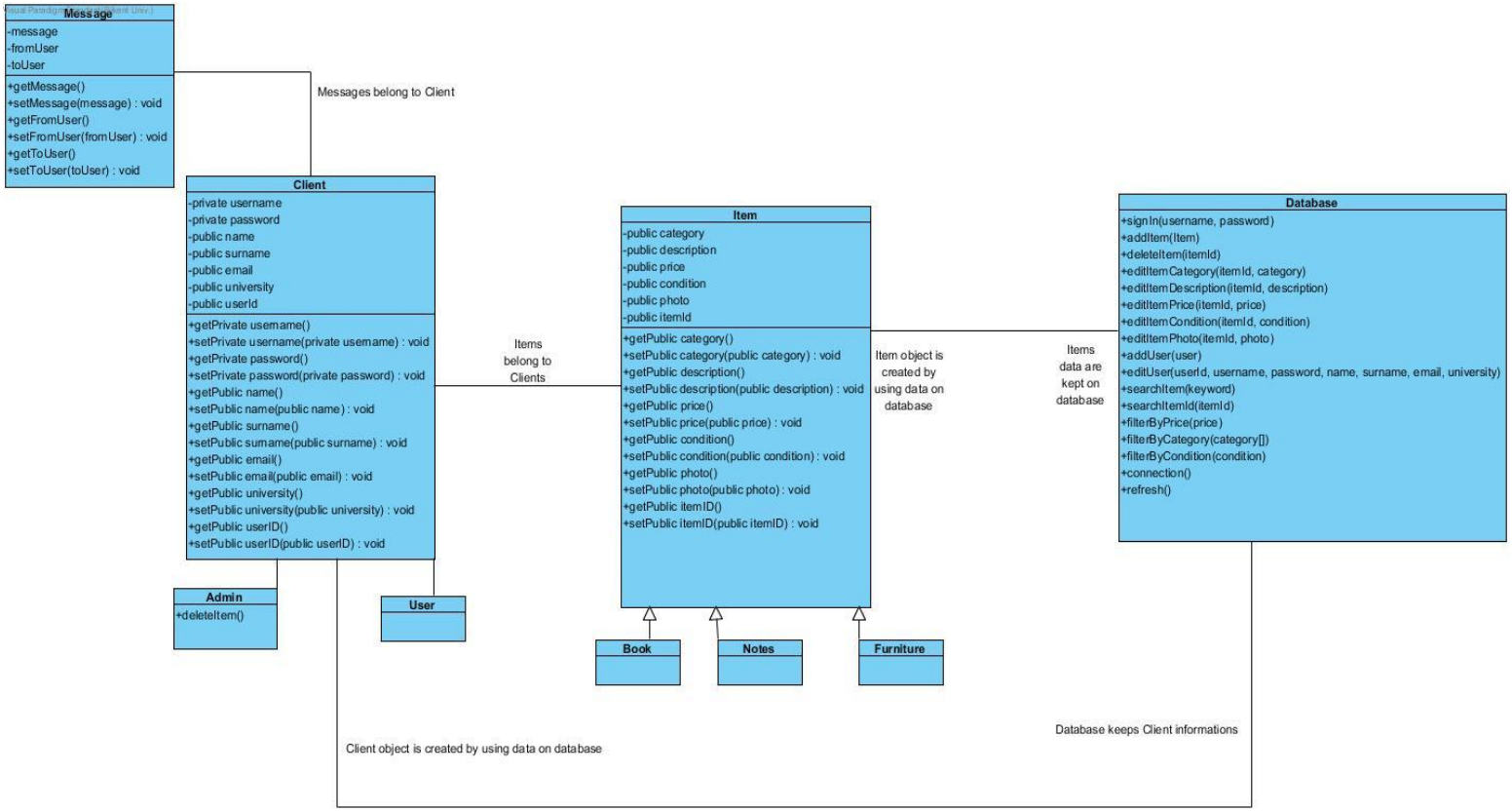
If the user clicks on “sign up” button, sign up page is displayed. After the user submits her/his personal information to create an account, the verification code is sent to his/her e-mail address. After he/she enters the verification code, the account is added to the system.

If the user has an account, she/he is able to enter the system by submitting his/her username and password. After entering system, the user may either view the items that are being sold or view his/her own profile.

If he/she chooses to display sold items, he/she may or may not filter them. If he/she decides to look at an item detailedly he/she will click on “view item” button. After examining the item, she/he may send a message to its seller in order to buy or ask anything about the item. After

clicking “envelope” button, she/he is able to fill the message and the subject area, then send the message to the seller.
The user is able to go to her/his profile anytime he/she wants. If she/he clicks on “my profile” button. In his/her own profile he/she is able to edit his personal information, add an item to sell, delete a sold item, or edit an item which is being sold.

3.4. Object and Class Models



The object model of the BestTrade program is illustrated above. This Class Diagram of BestTrade consist of 9 classes.

Anyone who login to the program called as Client and separates as Admin and User. This class keeps attributes of the client. Any client could have items to sell, messages that is sent or received. Messages class consists of the content of the message, sender and receiver.

Item class has all attributes of an item could have. It has three sub classes called as book, notes and furniture. An item could be any of them.

Database class hold and transfer almost all data. It keeps data for all clients and items. The information that will be shown on GUI part of the program will be taken from database by using database class.

3.5. User Interface

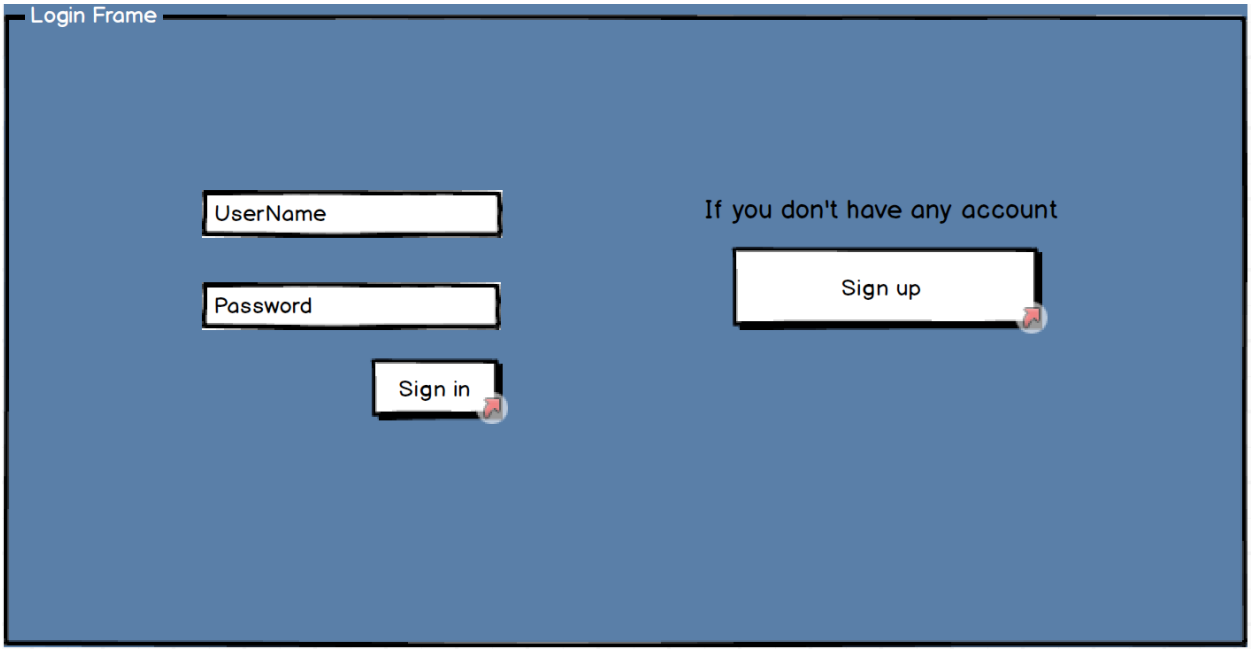


Figure 1: Login Screen

This screen displays itself when the application starts first. In this screen, users that had an account can write their unique username and password to enter the screen. The ones who has not an account can click on the “Sign Up” button to create an account.

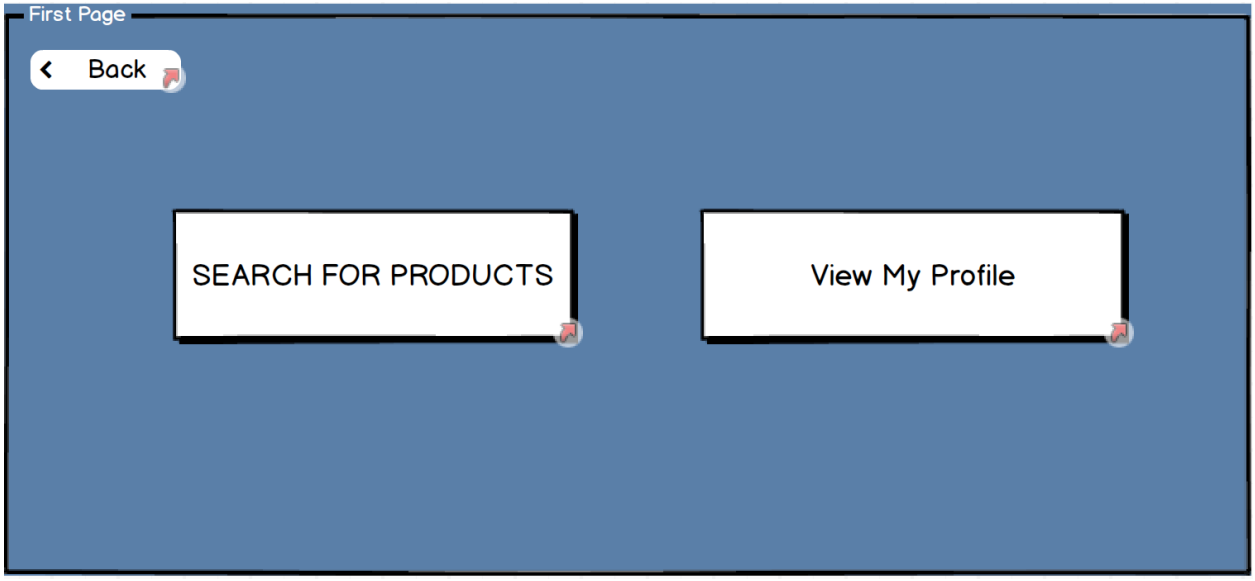


Figure 2: Trade Screen

This screen displays itself if the sign in process is successful. Users can either search for items that other user sells in system or check their profiles.

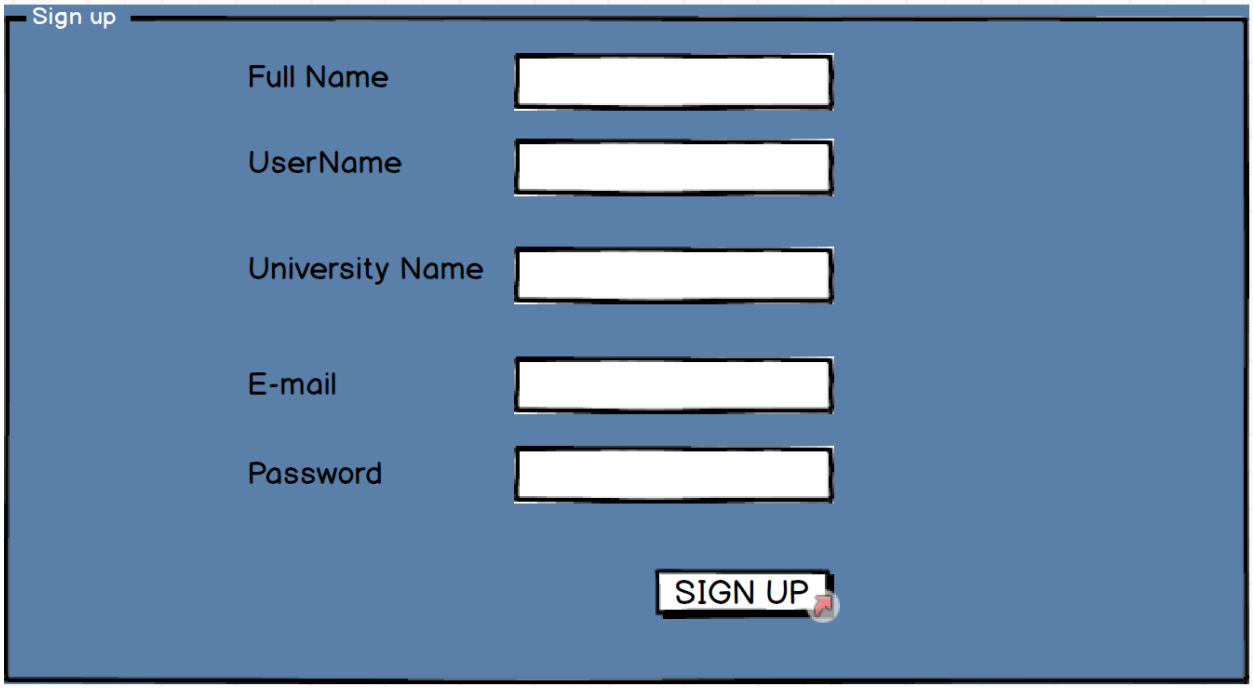


Figure 3: Sign Up Screen

In this screen, users have to enter the information that is needed to sign up to the system. However, the username and e-mail address must be unique in order to successfully signing up to the system.

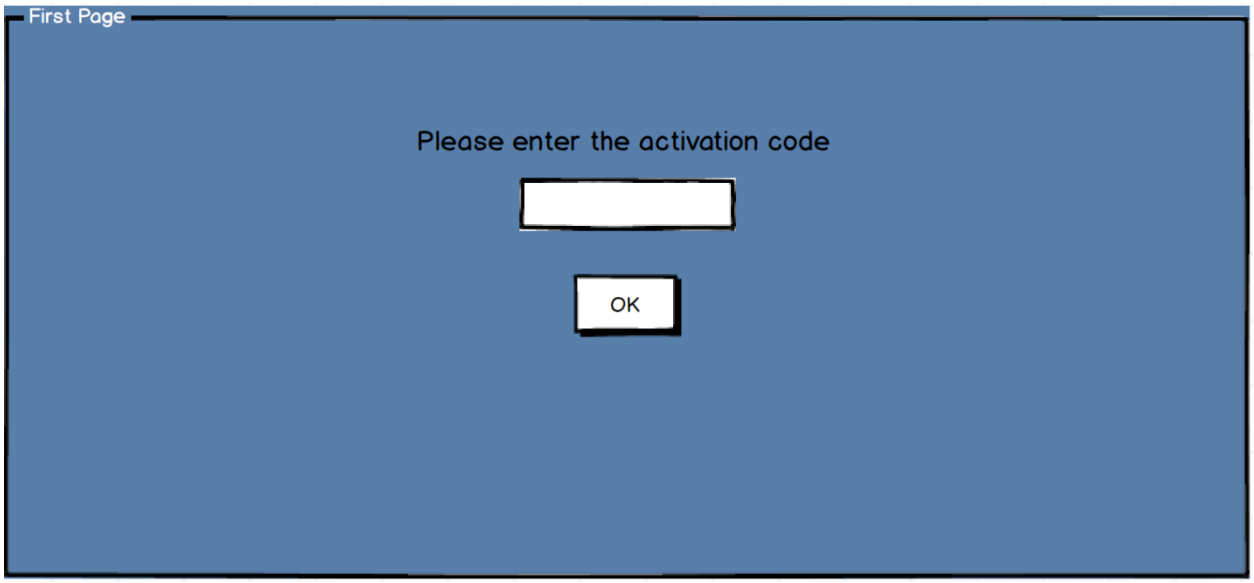


Figure 4: Activation Code Screen

After a successful sign up period, this screen automatically shows up and asks for a activation code that the system sends to the given e-mail address.

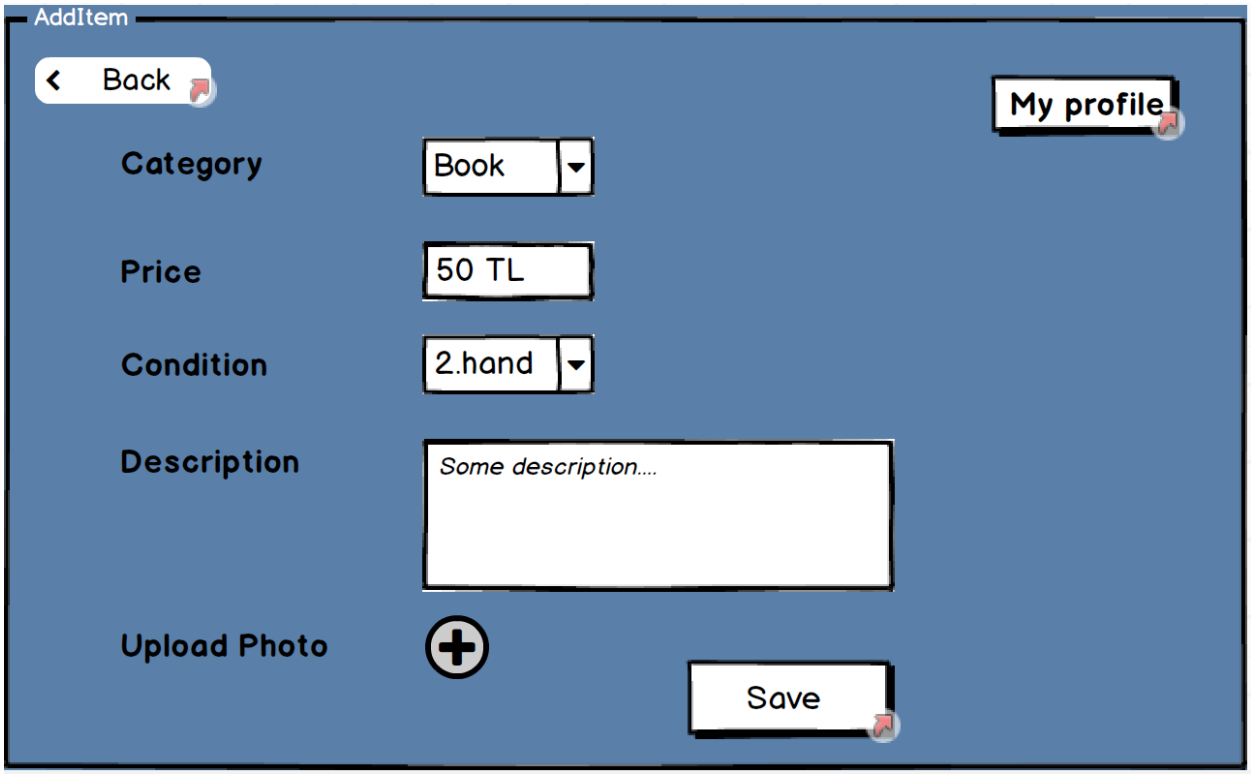


Figure 5: Add Item Screen

This screen enables users to add an item based on the attributes they give to the system. After clicking save, the attributes goes to the system and system refreshes itself.

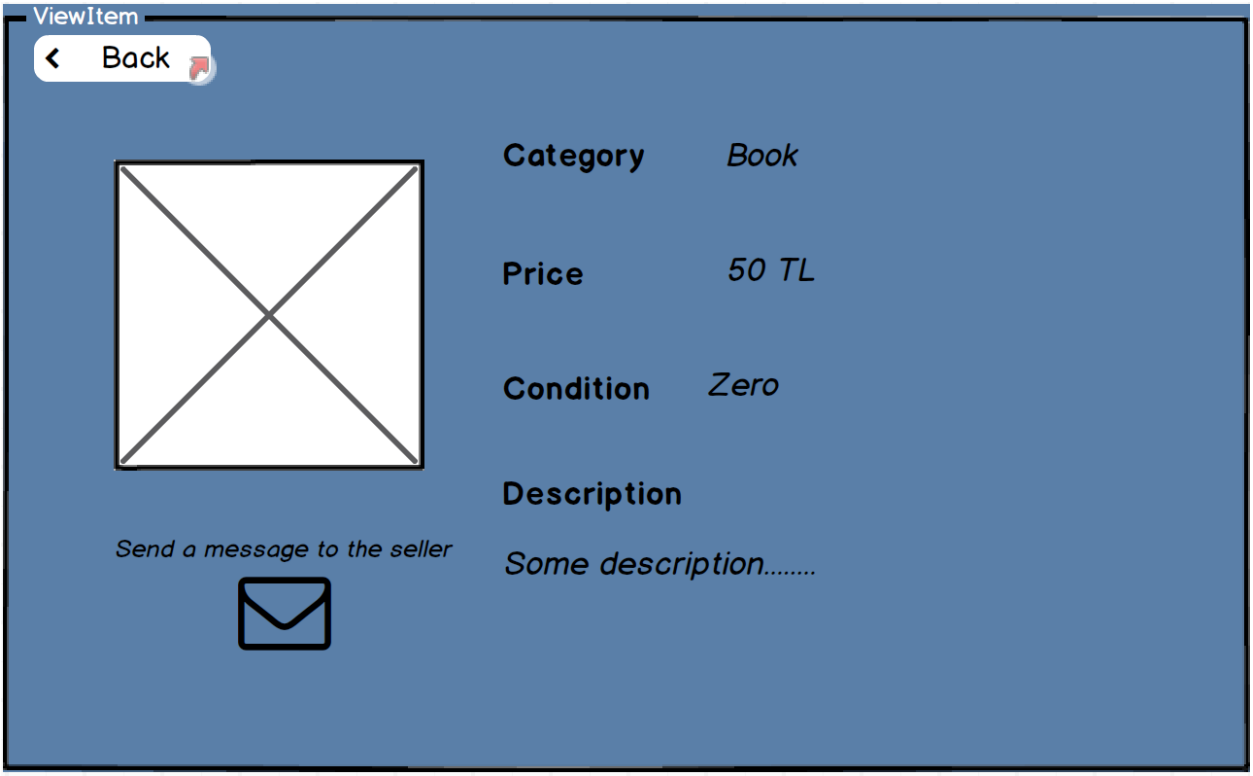


Figure 6: View Item Screen

In search item screen, users can check any item that they see. The item has a photo, a category, a price and condition whether it is second hand or not.

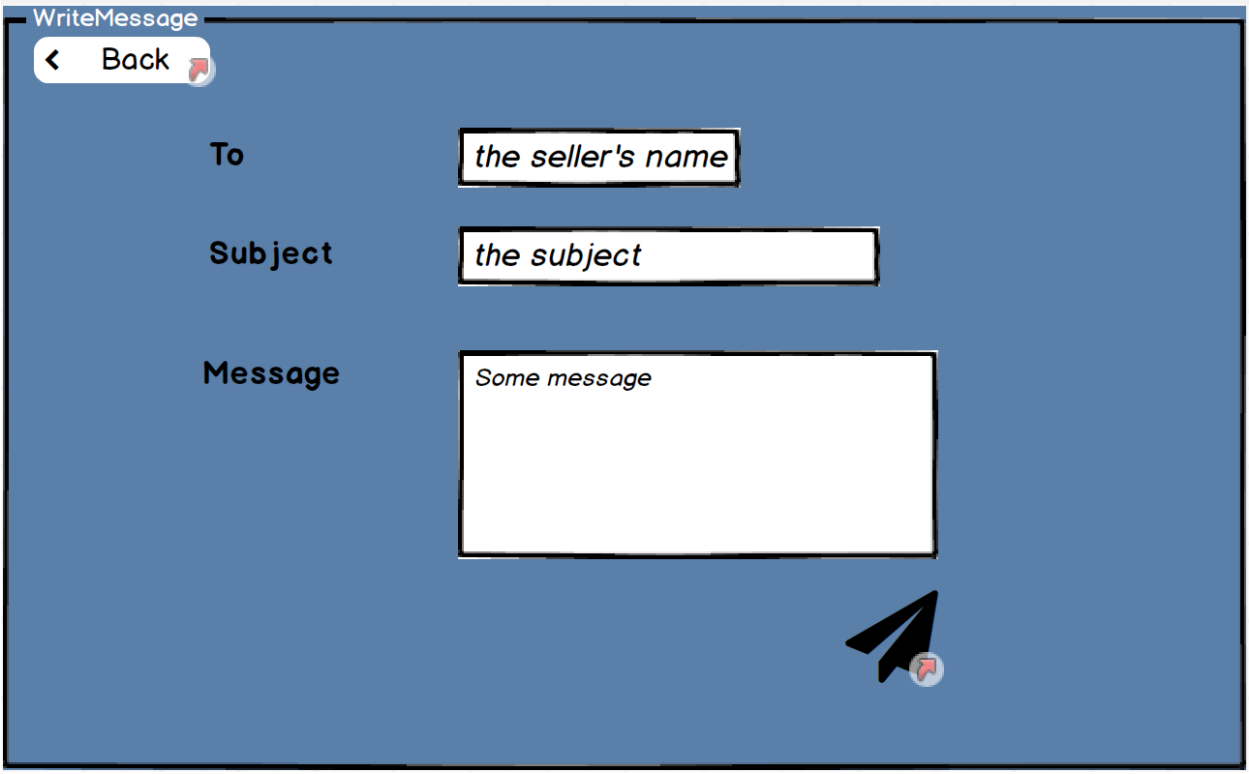


Figure 7: Send Message Screen

In send message screen, the users can send messages to a user that sells items. To send a message, the user has to write the other user's name, what will the subject be, and some message to clarify the subject.

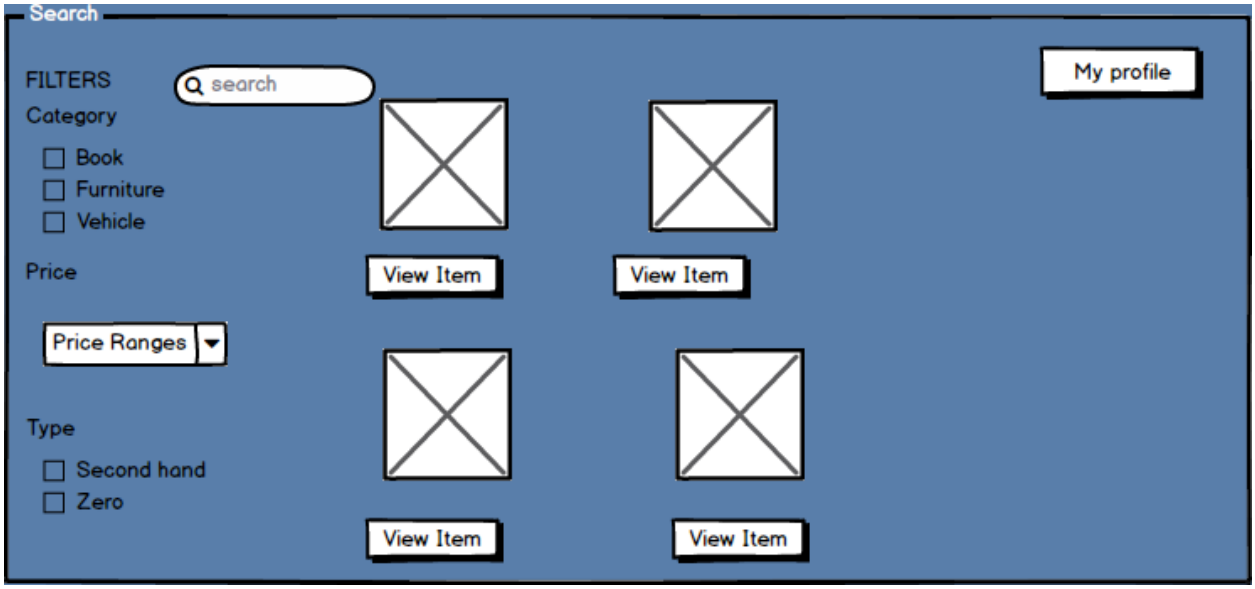


Figure 8: Search Item Screen

In this screen, users can search for any item they wanted via the filters on left. The items which corrects those filters will be presented on the middle of screen.

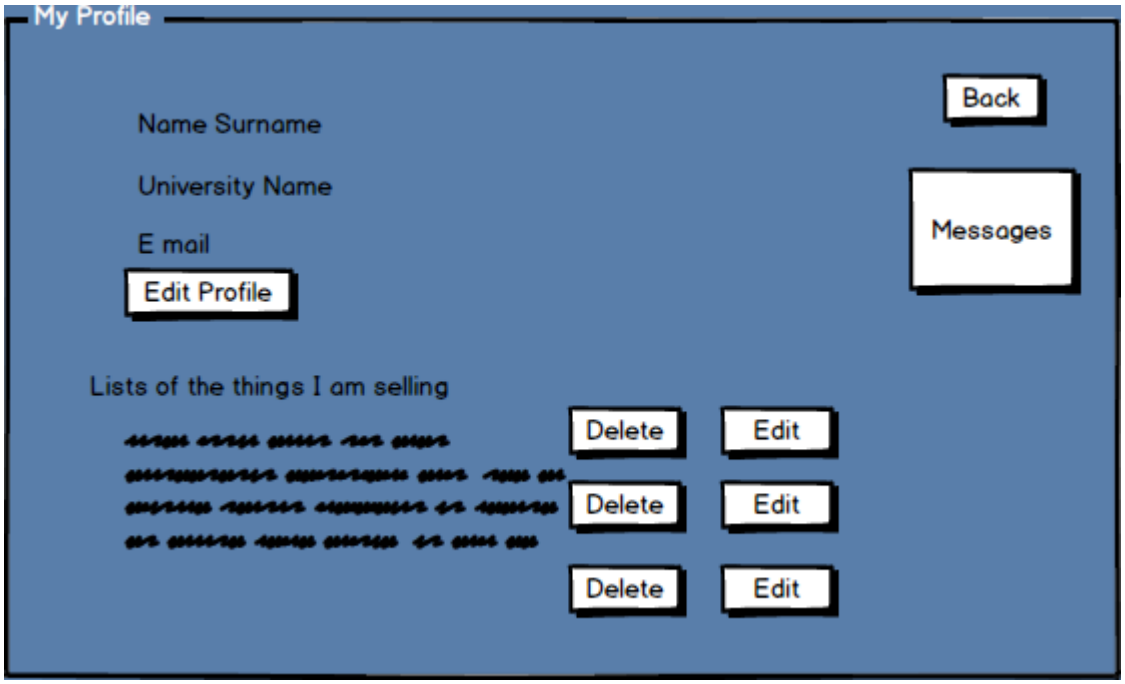


Figure 9: My Profile Screen

This screen contains information about the user. In this screen, any user can edit their existing profiles, delete or edit their item information, take a look at the items the user is selling at the moment, or look at their inbox via “Messages” button.

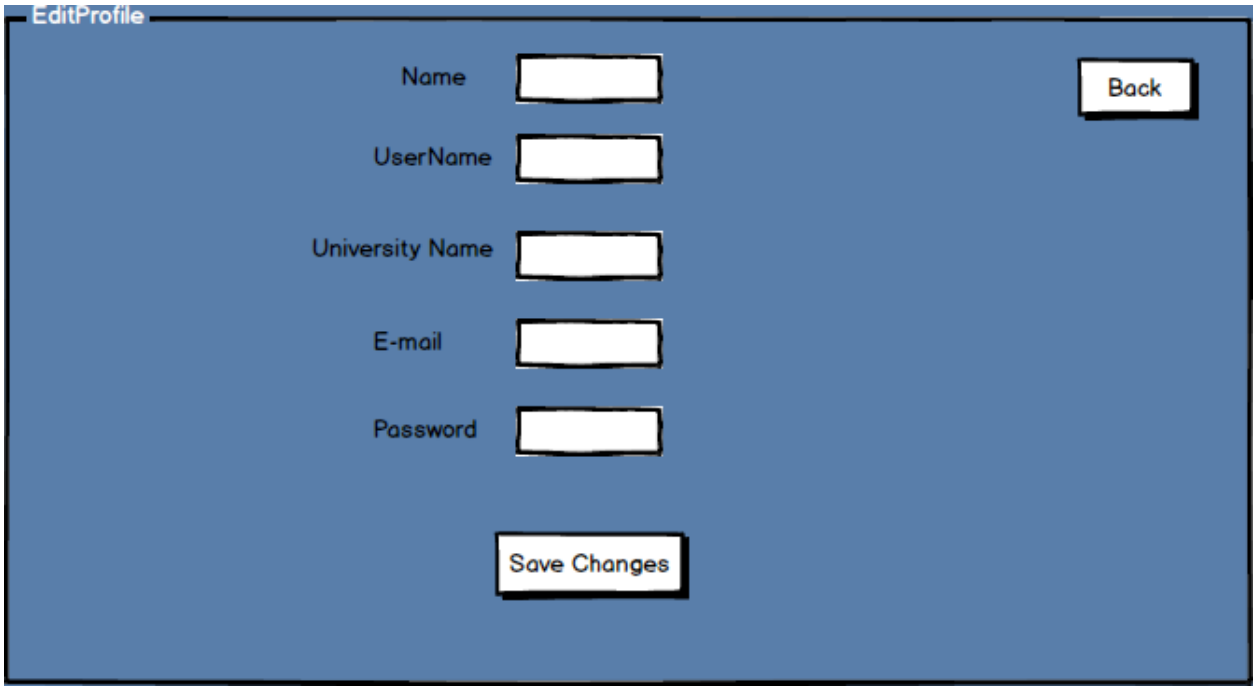


Figure 10: Edit Profile Screen

This screen has similarities with the sign up screen, however in this screen, users can change their profile information, which will be changed in database by using a query.

4. Conclusion

In this report, we created our analysis report in order to design and implement a desktop application called as “BestTrade” which helps students to buy and sell their staff easier. This report includes basically two parts. First part explains all requirements while second part is about all models and scenarios.

In requirements part, we try to finalize what we need to implement. This part basically showed us way to create second part. We tried to think all possibilities according to that we listed our functional and non-functional requirements. We planned that these requirements will help and show how when we move on implementation part.

Second part basically consists of four parts which are:

1. Use case model
2. Activity diagram
3. Class diagram
4. User interface
5. Sequence diagrams

At this part we thought about all scenarios and try to show them on use case and activity diagrams. We created mockup designs for our user interface and according to these three diagram (use case model, activity diagram, user interface) we created our class diagram. At this stage we tried to think about all possible states and activities that user could do. So, sequence diagrams basically show all possible scenarios. Furthermore, we tried to create our class diagram as close as possible to its last version so that in implementation part we will not need any other classes different than stated on the diagram. On user interface part we used mockup designs so that we could easily find what we would be needed when a user tries to use program. This part helped us a lot to create other diagrams and models. We tried to create our user interface as much as possible so that it will be user-friendly software.

As conclusion, we tried to think all possibilities about our project so that nothing unexpected will be pop-up later parts of the project. This report will guide us during design and implementation parts of the project. We tried our best to make analysis very well so that there will be no surprises during later parts of the project.