

CS319-OBJECT ORIENTED SOFTWARE ENGINEERING

FINAL PROJECT REPORT

BESTTRADE



GROUP MEMBERS

- Ezgi Çakır
- İbrahim Berker Kırdök
 - İlhami Özer
 - Irmak Tural
 - Berk Türk

Table of Contents

1. Implementation	3
2. Challenging Functional Requirements.....	4
3. The Changes Made in Implementation Stage and Non-Implemented Parts	5
4. User's Manual	5
4.1 Introduction	5
4.2 Installation	6
4.3 Using the BestTrade.....	6
4.3.1 Sign In.....	6
4.3.2 Sign Up.....	7
4.3.3 Submit an Activation Code	7
4.3.4 After Sign In.....	8
4.3.5 Add Product to Sell.....	9
4.3.6 Search for Products	9
4.3.7 View Detailed Information of a Product	10
4.3.8 Send Message to a User.....	11
4.3.9 View the Profile.....	11
4.3.10 Edit the Profile	13
4.3.11 View All Products the User Puts on a Sale	13
4.3.12 Edit a Product the User Puts on a Sale	14
4.3.13 View All Chats of the User	14
4.3.14 View the Chat Between Two Users	15

1. Implementation

BestTrade is coded in Java programming language. Java Swing is used for user interface. MySQL and WampServer is used for database and JavaxMail API is used to connect host to send activation code to the email.

We started the implementation by considering our design and layers. The project is organized as three subsystems which are user interface, model classes, and database.

In this user interface subsystem, there are 15 classes. They are named as LoginPanel, SignUpPanel, ActivationCodeScreen, TradeScreen, AddItemScreen, EditItemPanel, SearchItemScreen, ViewItemScreen, MyProfileScreen, MyProductsPanel, SendMessageScreen, MessageScreen, ChatScreen, and UIManager. UIManager stands for the frame of the BestTrade application. The other classes stand for panels for user interface. Every panel creates panels that are displayed after clicking on some components on the current panel. The detailed information about panels is given in User Manual section.

In database subsystem, there is one class. In this class all connection, retrieving, updating and inserting methods are implemented. A database object is created in login screen and transferred to other screens. Since more than one user could use the program from different computers at the same time, refresh method was needed so that database class could retrieve data from database when it is needed again. In this class there are bunch of methods and attributes. Moreover, getter and setter methods are implemented so that some private attributes could be get or set from different classes. Connection method is used to connect database. In login method, necessary data such as client information, all items and get items that login client has are retrieved from database. Some other methods such as getItems, getMessages methods help for login method to do that. On the other hand, some methods such as addItem, addClient methods, insert new data to database whereas editItem and editUser methods update data on database. Furthermore, some methods are needed to modify data that is already retrieved from database and transfer them to proper classes when these methods are invoked.

We also have SendMail class which send activation codes to new users so that the program understand that the given email address belongs to the user. In this class, Gmail host is used and internet connection is needed to send mail.

In model subsystem, 7 classes are implemented. They are named as Admin, Book, Client, Furniture, Item, Message, and Notes. Notes, Book, Furniture are child classes of Item. Admin extends Client. Item and its child classes have attributes named as title, category, description, price, condition, photo, itemId, userId. Title is the attribute is used for the short description of the item to be shown when items are searched. Category indicates that whether the item is book, notes, furniture, etc. Descriptions stands for detailed description of the item which is shown when detailed information of the item is displayed. Price stands for the price of the item which is determined by its seller. Condition indicates that the item is either zero or second hand. Photo stands for the photograph of the items which is added by its seller. If the photograph is not added, then the default image is displayed. "No photo" is written of the image. itemId is used for database related functions. userId is the ID of its seller, which is also used for database related functions. Client and Admin classes have attributes name, surname, username, userID, email, universityname, password, messages, items, activationCode. Messages are an ArrayList which stores all messages that the user sent or received. Items are an ArrayList which stores all items that the user added to the system in order to sell. activationCode is the code that is sent to activate the account of the user.

All attributes of model classes are private. They own getter and setter methods which provides data encapsulation. The parent-child hierarchy of model classes enables the system to present inheritance.

2. Challenging Functional Requirements

2.1. Sign Up / E-Mail Activation Code

For this functional requirement, connection a mail host system and send mail was a necessity. At this part finding proper API was the hard part, after that using proper libraries had challenged us. At first although we wrote the correct code, because of library problem, we could not able to run the program correctly. We solved this problem by searching for correct and updated library.

2.2. Add Item / Edit Item / Choose Item

Implementation of add / edit item and search/choose item was challenging for us because of inserting into and retrieving images from database. We used blob type to store image on database and when we retrieve it we transform it to the ImageIcon to setIcon of the buttons or labels as items photo. This part was challenging because of the finding proper type to store images on database and use them on the application.

2.3. Send Message

This part was challenging for us because, messages should be sorted and separated according to users. In other words, a message should have from, to, message and data attributes to make this feature working properly. So we used timestamp to store date. Challenging part was sorting and distinguish messages user by user.

3. The Changes Made in Implementation Stage and Non-Implemented Parts

Admin class is not implemented and used in our project. We did not have time to manage the requirements of this class.

Additional Attribute

New attributes are added to Item class which is names as title and userID. Title was useful when users search for an product and userID was useful when users send messages to sellers directly through the panel that displays detailed information of items.

Additional Classes

We added a new class called SendMail that provides users to send a message via the application. The connection between the application and database is implemented in that class.

Additional Screen Implementations

We added new screens that do not exist in our analysis stage. EditItemPanel to modify the features of the item that is added before, MyProductsPanel that displays use the item that is added by that user, MessageScreen that displays all chats that the user is participated. ChatPanel is added to display a chat between two users.

4. User's Manual

4.1 Introduction

BestTrade is a desktop application that enables university student to sell or buy product, such as books, notes, furnitures, electronics, etc. Users have an account with unique username and password to login the system. Users can add items to sell or search for an item to buy. Users are able to chat thorough BestTrade. Students need to have a e-mail addresses which end with "edu.tr", which means their university related e-mail addresses in order to sign up. However after they sign up with e-mail addresses which end with "edu.tr", they can change their mail addresses

with what they want in edit profile screen.

4.2 Installation

System Requirements:

It requires Java Runtime Environment and WampServer for database. Furthermore, JvaxMail API is used to send activation code. Moreover, Apache Commons Collections library is used for its some useful methods such as union method.

4.3 Using the BestTrade

4.3.1 Sign In

When BestTrade is initiated, LoginPanel(Figure 1) is displayed. By submitting a username and the password of an existing account, the user is able to make an entrance to the application.

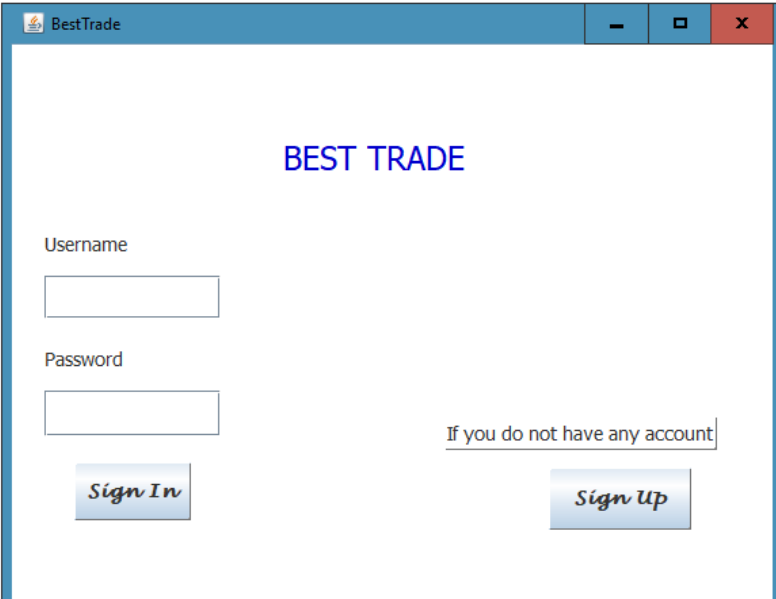


Figure 1

4.3.2 Sign Up

If the application is initiated and at LoginPanel, by clicking on the button where “Sign Up” written on, users are led to SignUpPanel(Figure 2) that enables to sign up. Users need to enter their personal informations such as their name, surname, desired username, e-mail address, the name of their universities, desired password. E-mail addresses must end with “edu.tr” since the application’s target domain is university student. When the user clicks on “Sign Up” button, an activation code is sent to his/her e-mail address to activate the account. Simultaneously, ActivationCodeScreen.(Figure 3) is displayed. If the user clicks on “Back” button while filling the information, LoginPanel is displayed and nothing is stored.

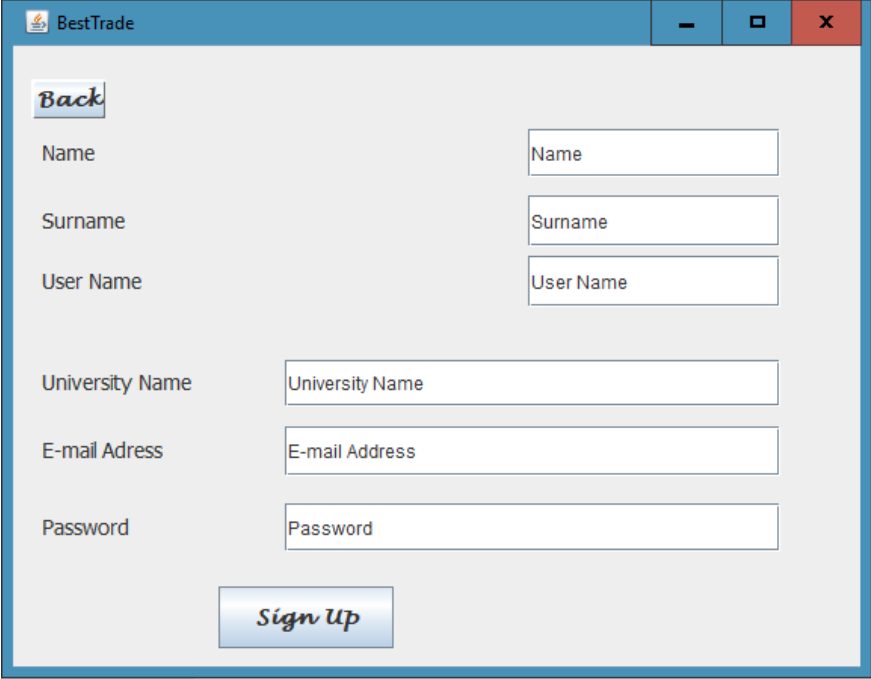


Figure 2

4.3.3 Submit an Activation Code

After the user submits his/her personal information and click on “Sign Up” button, ActivationCodeScreen is displayed. By submitting the code that is sent to the user’s e-mail address and clicking on “OK” button, the account is added to database, the account is automatically signed in and TradeScreen(Figure 4) is displayed. If the user clicks on “Back” button without submitting the activation code, the account is not created and added to the database.

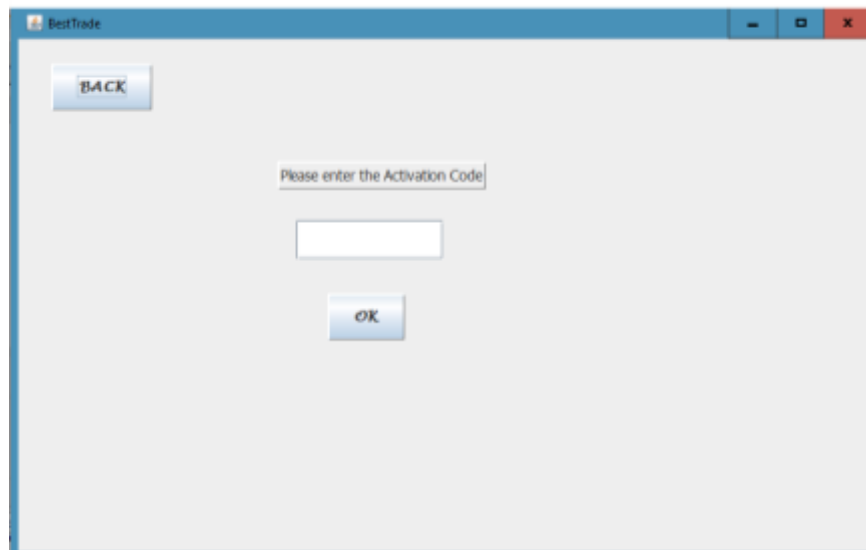


Figure 3

4.3.4 After Sign In

After signing in, TradeScreen is displayed. The user can logout by clicking on "Logout" button and LoginPanel is displayed. If the user clicks on "Add Product to Sell" button, AddItemScreen(Figure 5) is displayed. If the user clicks on "Search for Products" button, SearchItemScreen(Figure 6) is displayed. If the user clicks on "View My Profile" button, MyProfilePanel(Figure 7) is displayed.

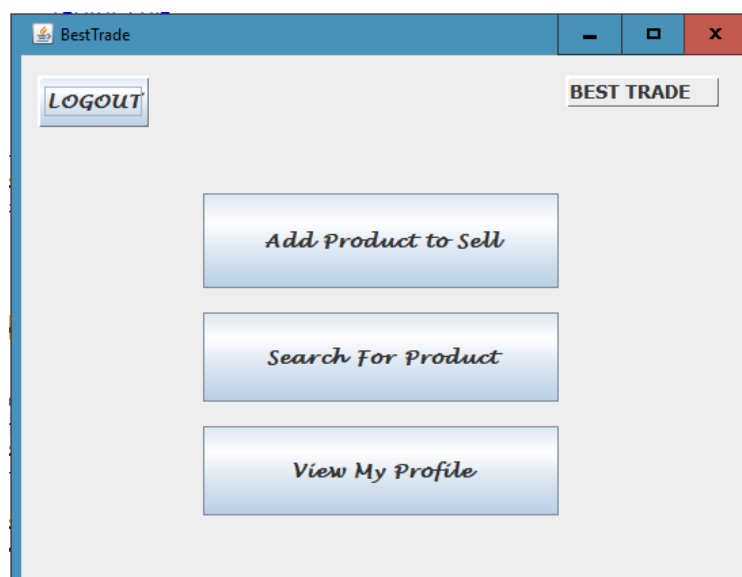


Figure 4

4.3.5 Add Product to Sell

After clicking on “Add Product to Sell” button when the application is at TradeScreen, users are led to AddItemScreen is displayed. The user is able to add a product to sell by submitting category, price, condition, description, title, and photograph of the product. If photo is not submitted, default image for products without photo is displayed while displaying the product’s information. After clicking on “Save” button, the product is added to the database. If the user clicks on “Back” button while submitting the information, nothing is stored and TradeScreen is displayed.

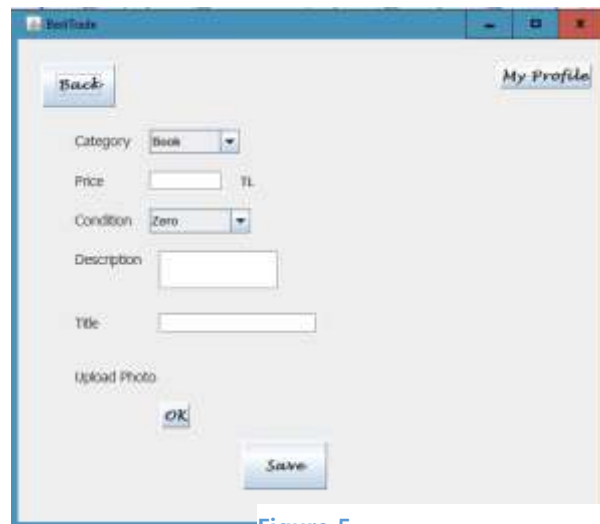


Figure 5

4.3.6 Search for Products

After clicking on “Search for Products” button when the application is at TradeScreen, users are led to SearchItemScreen is displayed. Initially, all products that are in the database is displayed. In each page, 8 products are displayed. Their titles, price, and photos are displayed. The user can filter items by choosing the desired category, price range, or type and clicking on “Apply” button. S/he can also filter products with using keywords with exist in titles of products by writing on search label and clicking on “search” button. S/he is able to discard filters by clicking on “Clear Filters”. If the user wants to view detailed information of a product, s/he is able to do that by clicking on the photo of the desired product. After clicking of the photo, ViewItemScreen(Figure 8) is displayed. If the user clicks

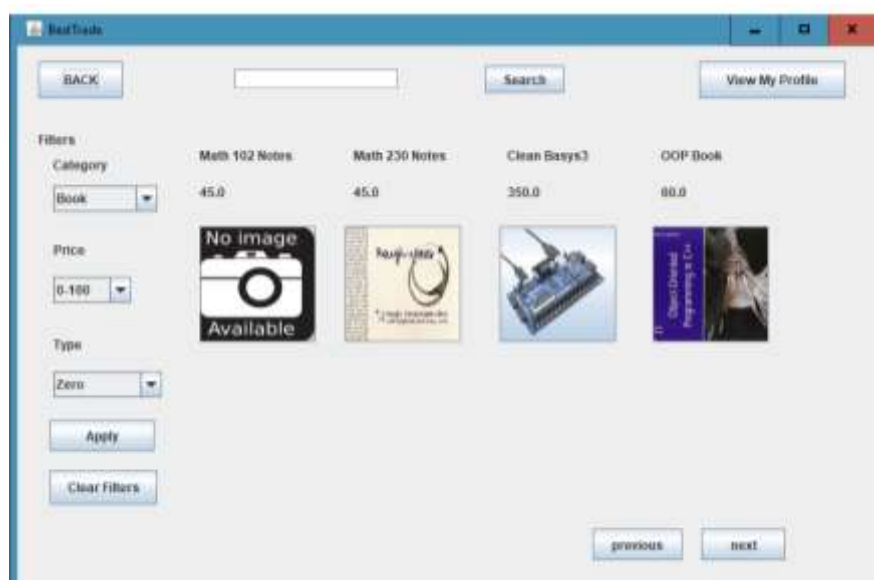


Figure 6

on “Back” button, TradeScreen is displayed. If the user clicks on “View My Profile” button, MyProfilePanel is displayed.

4.3.7 View Detailed Information of a Product

After clicking on the photo of a product when the application displays SearchItemScreen, detailed information of the product is displayed. These information are following: photo, title, category, price, condition, description of the products and name, e-mail address and university name of its seller is displayed. If the user wants to send message to the seller of the product, s/he needs to click on “Send Message” button, then SendMessageScreen(Figure 9) is displayed. If the user clicks on “Back” button, SearchItemScreen is displayed with discarding all filters that are applied earlier.



Figure 7

4.3.8 Send Message to a User

After clicking on “Send Message” button when detailed information of a product is displayed, the user is led to SendMessageScreen. The textfield which displays the username of the seller of the selected product is filled automatically. After typing the message and clicking of “Send” button, the message is sent to the seller of the product. Then, viewItemScreen which views the detailed information of the products is shown. If the user clicks on “Back” button without submitting the message, ViewItemScreen is displayed, either.

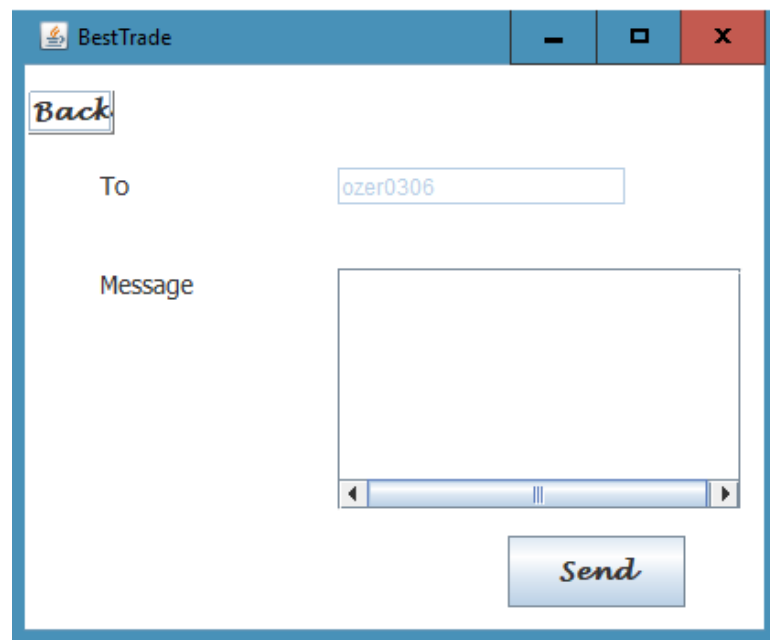


Figure 8

4.3.9 View the Profile

The profile of the user can be viewed by clicking on “View My Profile” buttons which are in TradeScreen and SearchItemScreen. After clicking, MyProfilePanel is displayed. In this panel, the user can view his/her personal information, edit the information by clicking on “Edit Profile” button, display all chats that he/she participated by clicking on “Messages” button, and display all products that s/he put up for sale by clicking on “Sold Products” button.

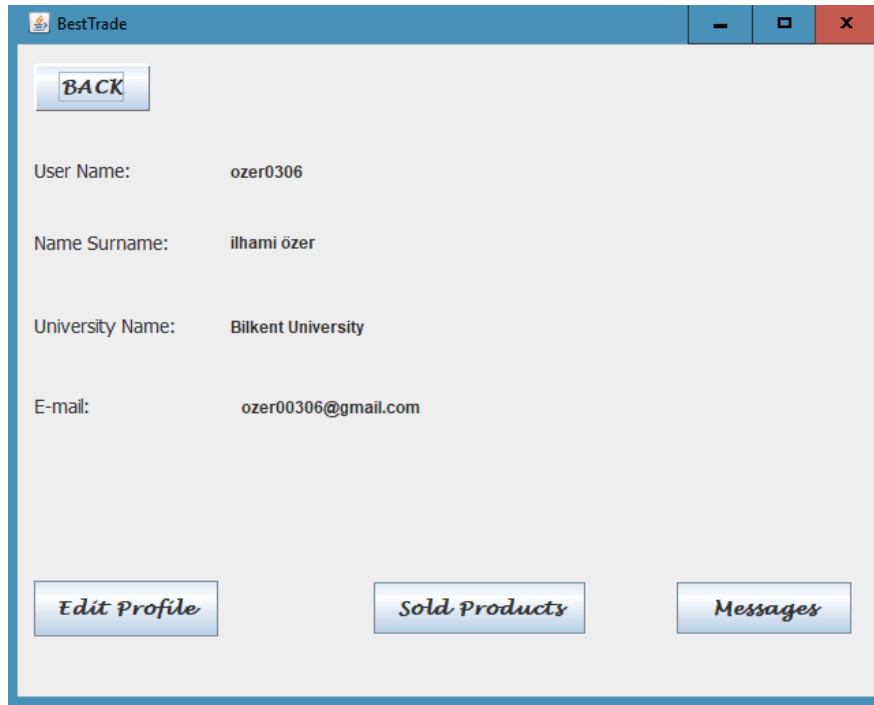


Figure 9

4.3.10 Edit the Profile

After clicking on “Edit Profile” button, when the profile of the user is shown, EditProfilePanel(Figure 10) is displayed. In this panel, the user is able to edit any personal information of his/her account. After clicking on “Save Changes” button, information is updated. After clicking “Back” and “Save Changes”, buttons, MyProfilePanel is displayed.



Figure 10

4.3.11 View All Products the User Puts on a Sale

While the application displayed the profile of the user, if s/he clicks on “Sold Products” button, MyProductsPanel(Figure 11) is shown. The panel displays titles of all products that the user puts on a sale currently. In a page, 7 products are shown. The user is able to view the remaining list by clicking of “previous” and “next” buttons which indicate “go to previous page” and “go to next page”, respectively. If the user clicks on one of the title of the products, EditItemPanel (Figure 12) is displayed. If the user clicks on “Back” button, the profile of the user is displayed.

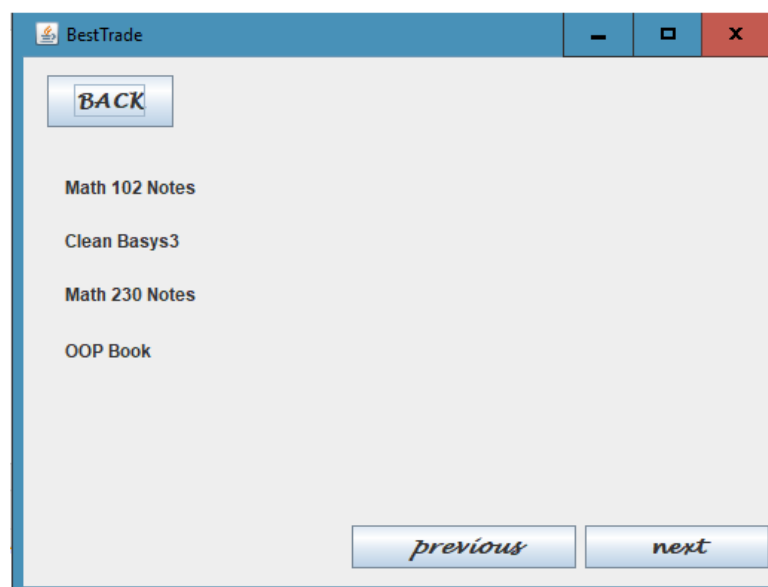


Figure 11

4.3.12 Edit a Product the User Puts on a Sale

After clicking on one of the title of the products when the application displays MyProductsPanel, EditItemPanel is displayed. In this panel, the user is able to edit any information of the selected product. After clicking on “Save” button, information of the product is updated. The user is also able to delete the item by clicking on “Delete” button.

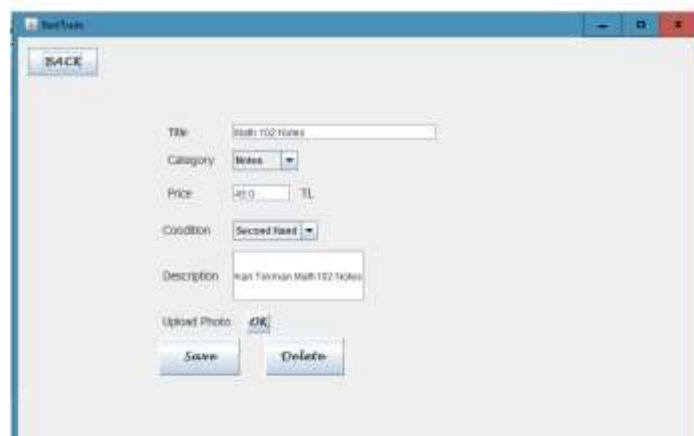
The screenshot shows a window titled "EditItem" with a "BACK" button in the top left. The form contains the following fields: "Title" with the text "Math 102 Notes", "Category" with a dropdown menu showing "Notes", "Price" with a text box containing "45.0" and a "TL" label, "Condition" with a dropdown menu showing "Second Hand", and "Description" with the text "Math 102 Notes". Below these fields is an "Upload Photo" button and an "OK" button. At the bottom of the form are "Save" and "Delete" buttons.

Figure 12

After clicking “Back”, “Save”, and “Delete” buttons, MyProductsPanel is displayed.

4.3.13 View All Chats of the User

While the application displayed the profile of the user, if s/he clicks on “Messages” button, MessageScreen(Figure 13) is shown. The panel displays the last messages of chats between the user and the other users. In a page, 8 products are shown. The user is able to view the remaining chats by clicking of “previous” and “next” buttons which indicate “go to previous page” and “go to next page”, respectively. If the user clicks on one of the messages, ChatScreen (Figure 14) is displayed which views all the messages that exists in a selected chat. If the user clicks on “Back” button, the profile of the user is displayed.

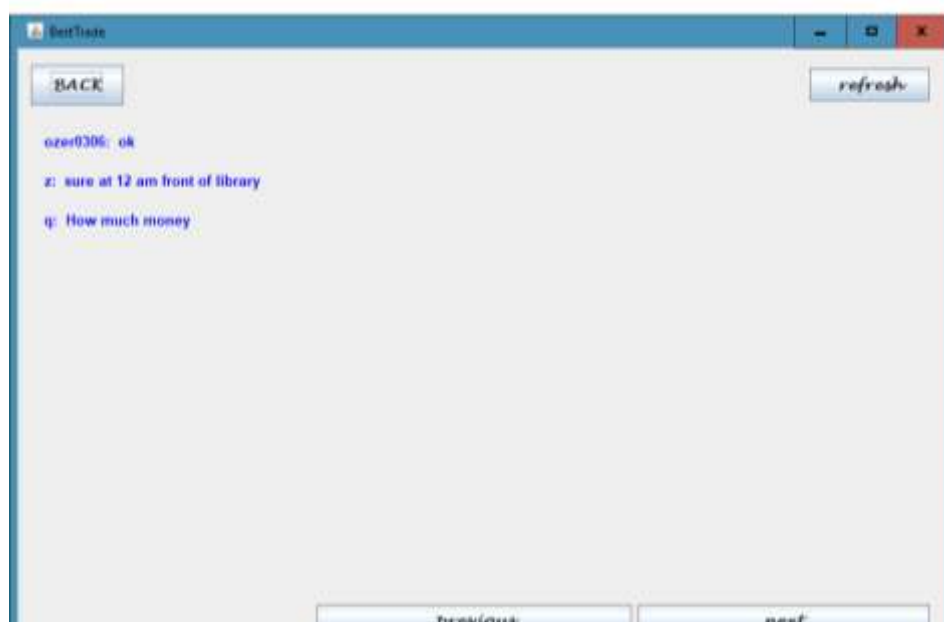
The screenshot shows a window titled "MessageScreen" with a "BACK" button in the top left and a "refresh" button in the top right. The main area displays a chat conversation with the following messages: "ozer0306: ok", "z: sure at 12 am front of library", and "q: How much money". At the bottom of the window are "previous" and "next" buttons.

Figure 13

4.3.14 View the Chat Between Two Users

After clicking on one of the messages when the application displays the list of the chats that the user participates in, ChatScreen is displayed. The panel views all the messages that exists in a selected chat. The messages are sorted from the last to first message that is sent. The current user's messages are displayed in red, the opposite chatter's messages are displayed in blue. In a page, 10 messages are displayed. The user is able to view the

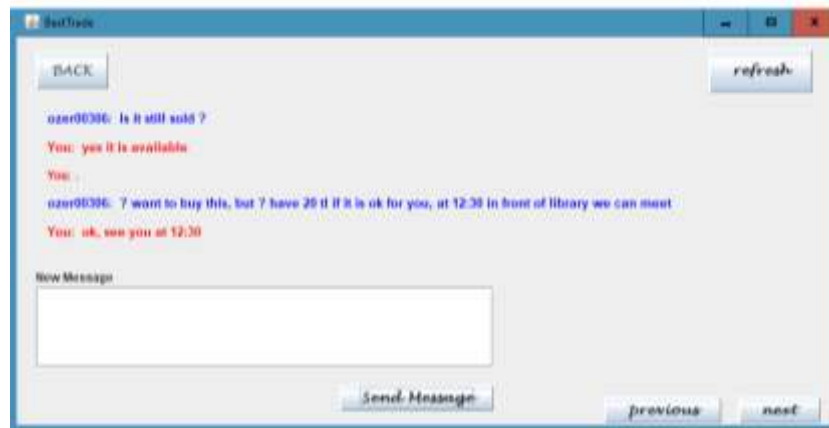


Figure 14

remaining messages by clicking on "next" and "previous" button, They stands for "go to next page" and "go to previous page", respectively. The user is also able to send a new message by typing the message and clicking on "Send Message" button. After sent, the page is refreshed. This occurs also when the user clicks on "refresh" button. If the user clicks on "Back" button, the panel that views all chats of the user is displayed.