BidBee

EBay-Amazon

Design Report
For Web Application

Version <1.1>

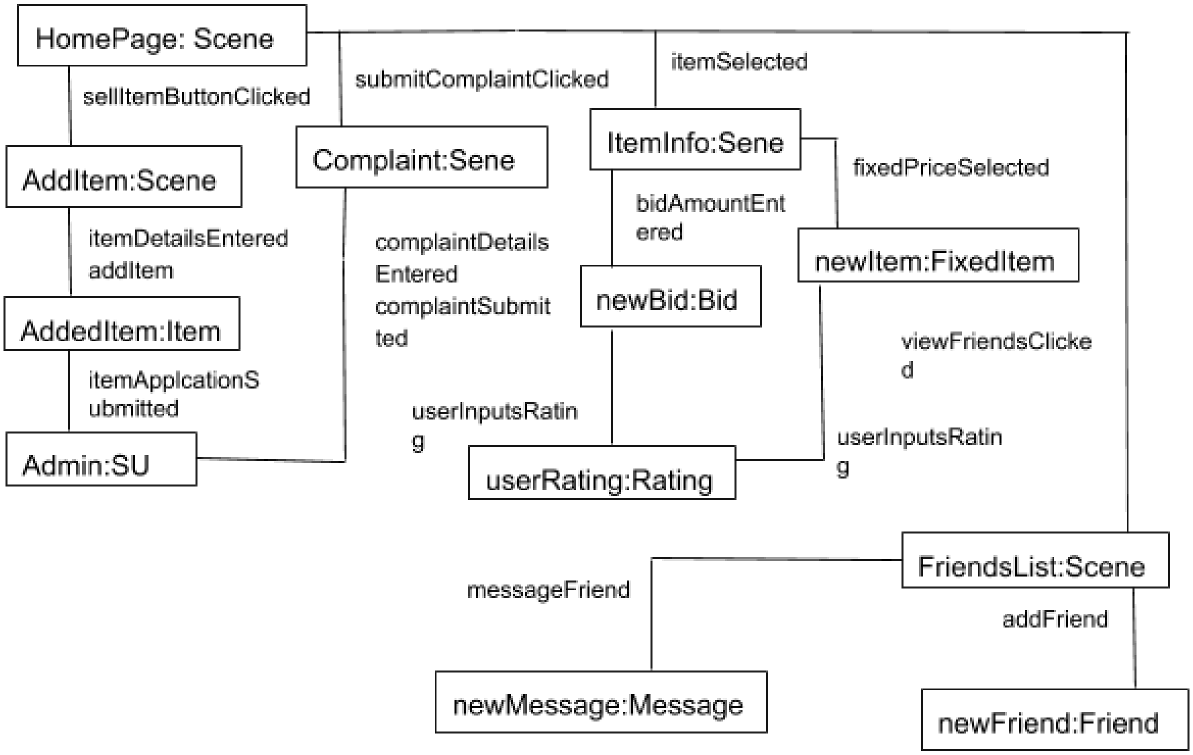
Revision History

Date	Version	Description	Author
25/MAR/19	<1.0>	Created the outline of the software requirement specifications	Aqsa Malik Yifang He Marvin Estime Andy Yeung
18/Apr/19	<1.1>	Create Design Report	Aqsa Malik Yifang He Marvin Estime Andy Yeung

TABLE OF CONTENT

1. Introduction

The purpose of this document is to provide the overview of design and functionality of entire Ebay-Amazon system.



2. Use Case Analysis

This section shows the detailed description of all the main use case that we mentioned in our software specification document. For each of the use case, we added the sequence diagram and petri-net.

List of the use case are:

1. Apply for Ordinary User
2. Login in.
3. Purchase item
4. Sell item
5. Rating.

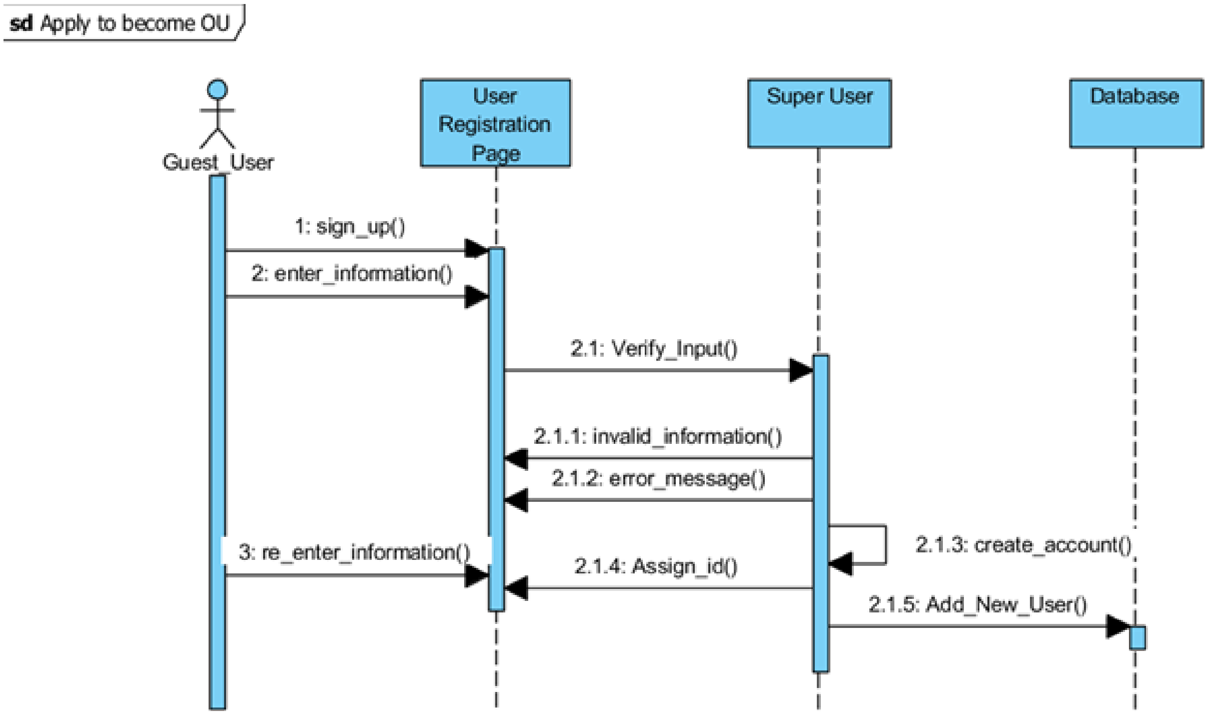
2.1:Apply for Ordinary User (OU):

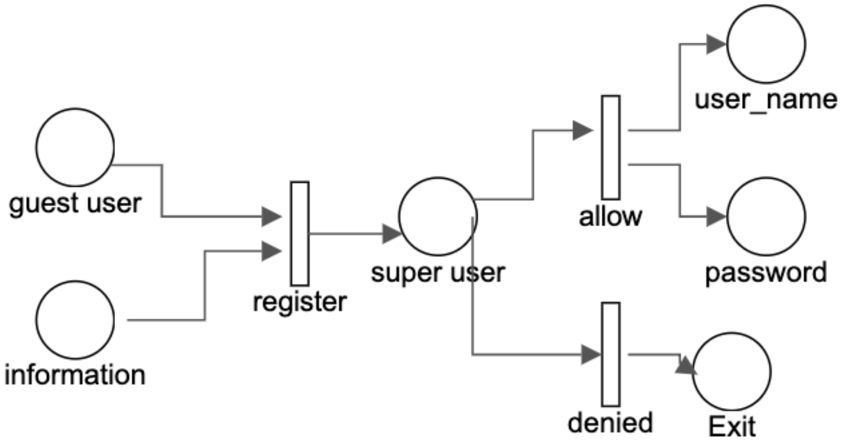
Normal Scenario:

In order to purchase or sell something, guest user need to register for account to become the ordinary user of the system. They fill up the registration form and then super user need to validate the information and create account for guest user. Super user assigns the unique id to every ordinary user.

Exceptional Scenario:

Customer didn't put valid information for example the put the information of expired credit card. System will ask the ordinary user to re-enter the information.





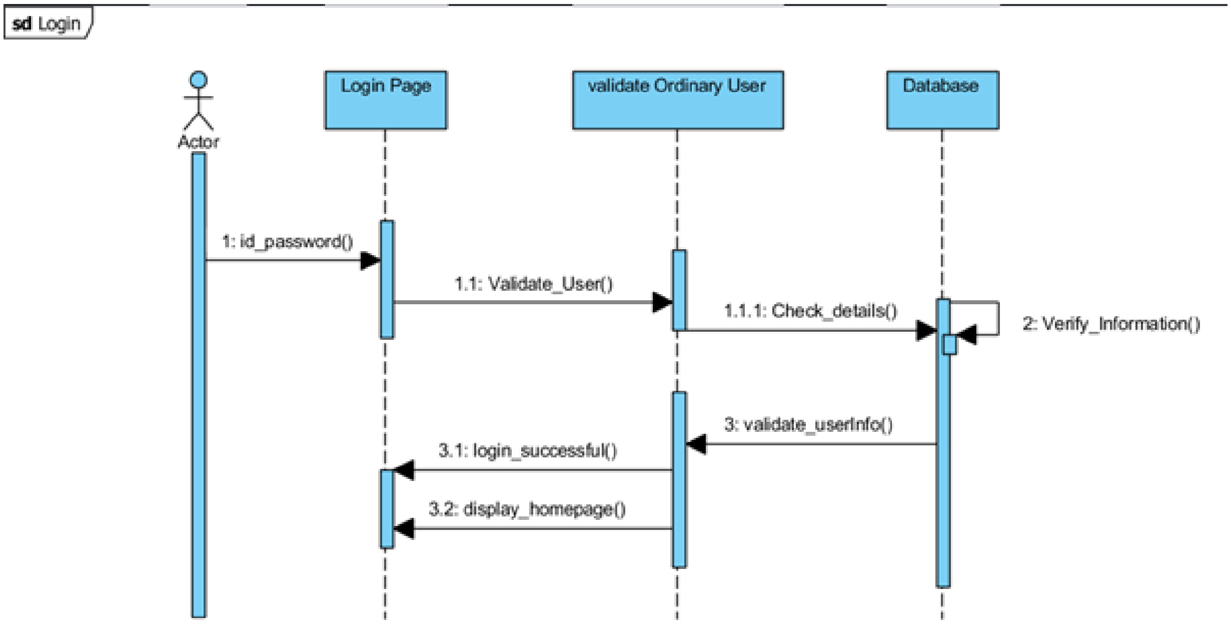
2.2: Login

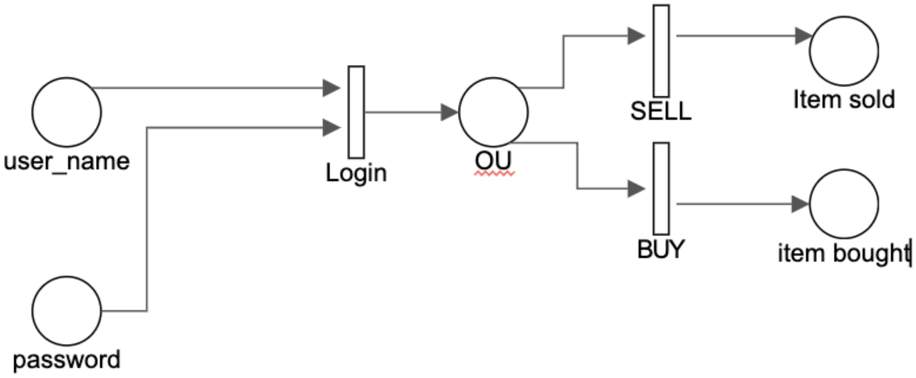
Normal Scenario:

The ordinary user will be prompted to input their User Id and their password. When user enter their information, it will be validated from database. Once customer login in they will access their account and will be able to sell or purchase items.

Exceptional Scenario:

If the user hasn't sign up yet it will show the screen to ask the customer to sign up first. Or if the customer is in the block list it will notify the customer.





2.3: Purchase item:

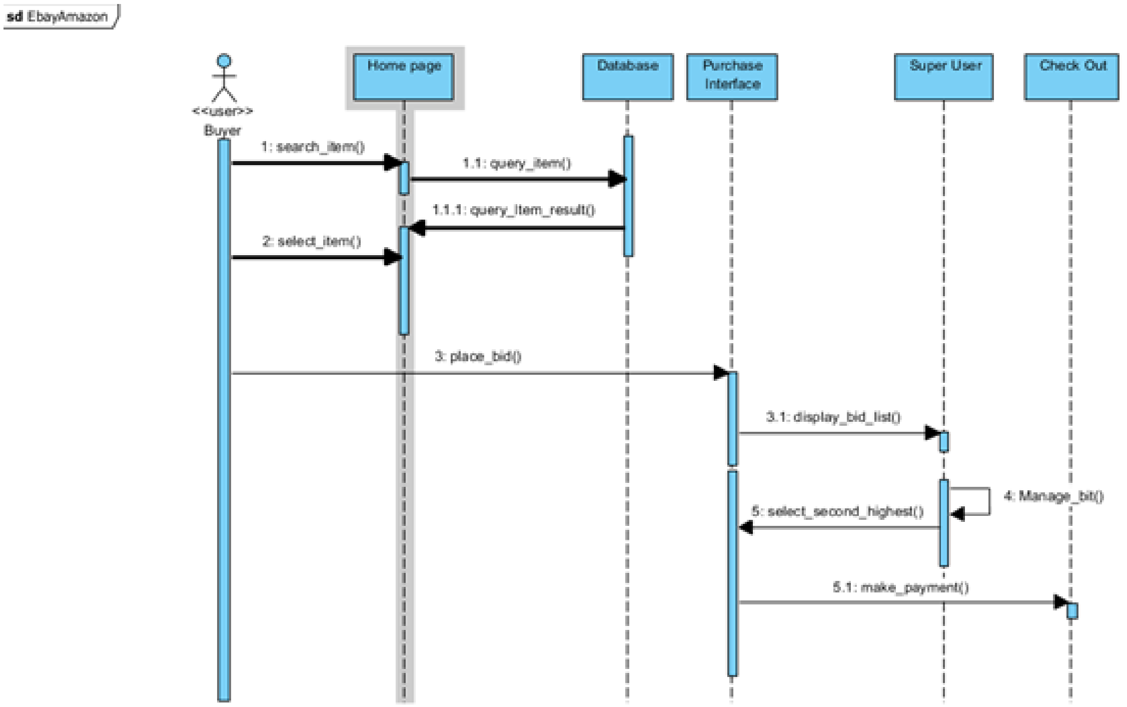
a) Bidding

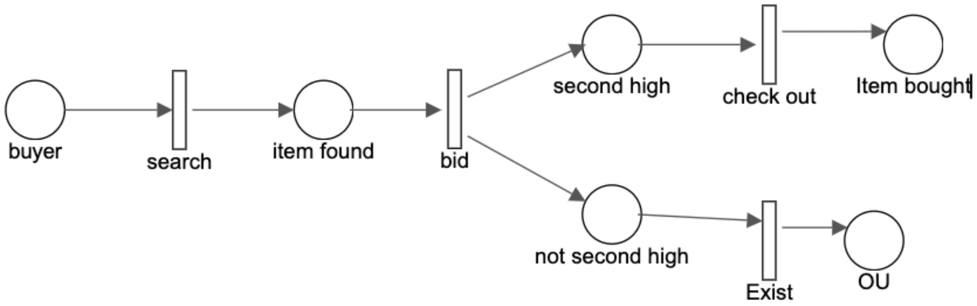
Normal Scenario:

Ordinary users will search for items, system will look for item in database, and it will show all the related items in the database. Ordinary user will select the desired item and place the bid for required item. Super user manages bids and select the second highest bid.

Exceptional Scenario:

If credit card or debit card has insufficient balance, order will not procced and customer will have to re-enter another credit or debit card.





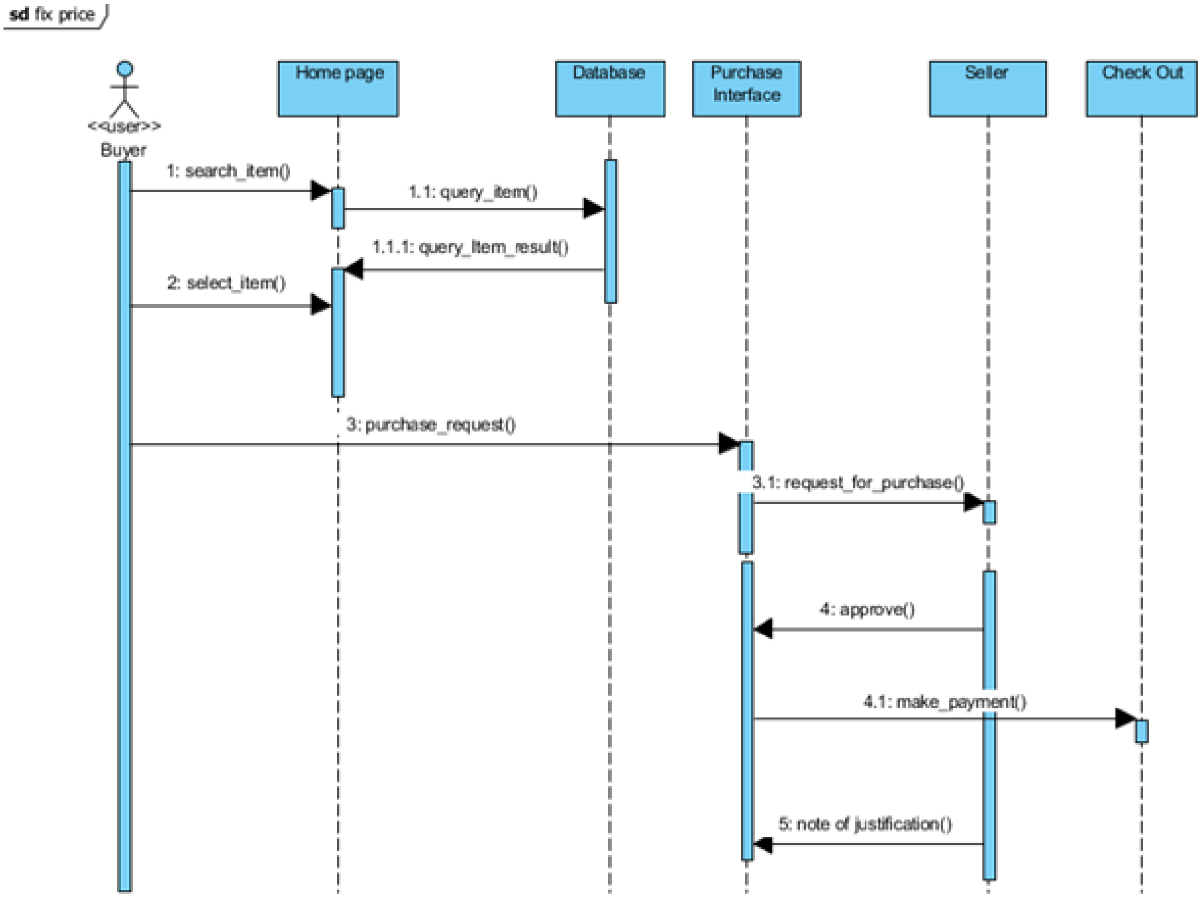
b) Fixed Price

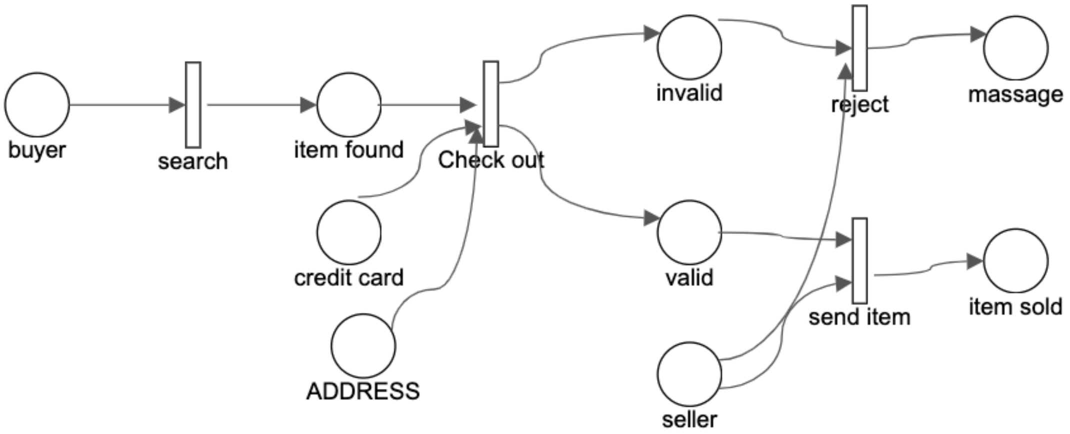
Normal Scenario:

Ordinary users will search for items, system will look for item in database, and it will show all the related items in the database. Ordinary user will select the desired item and request for purchase. Seller will approve the request for item or reject the request and provide the note of justification why he is not chosen.

Exceptional Scenario:

If credit card or debit card has insufficient balance, order will not proceed and customer will have to re-enter another credit or debit ca





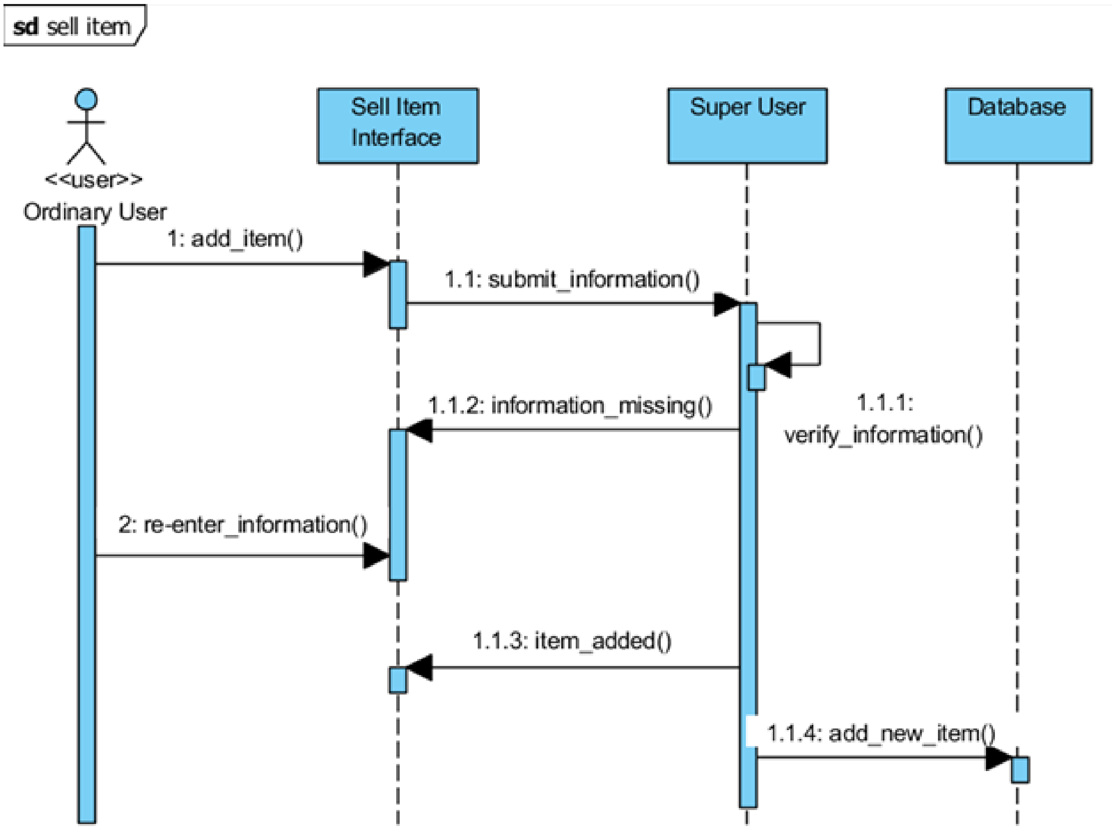
2.4: Sell Item

Normal Scenario:

When ordinary user wants to sell an item, they submit the information (title, keywords, price and picture to super user. And then all information of item is stored into database.

Exceptional Scenario:

Ordinary user enters incomplete information of item they want to sell, or they enter wrong information related to item, super user requests the ordinary user to re-enter the information.



-

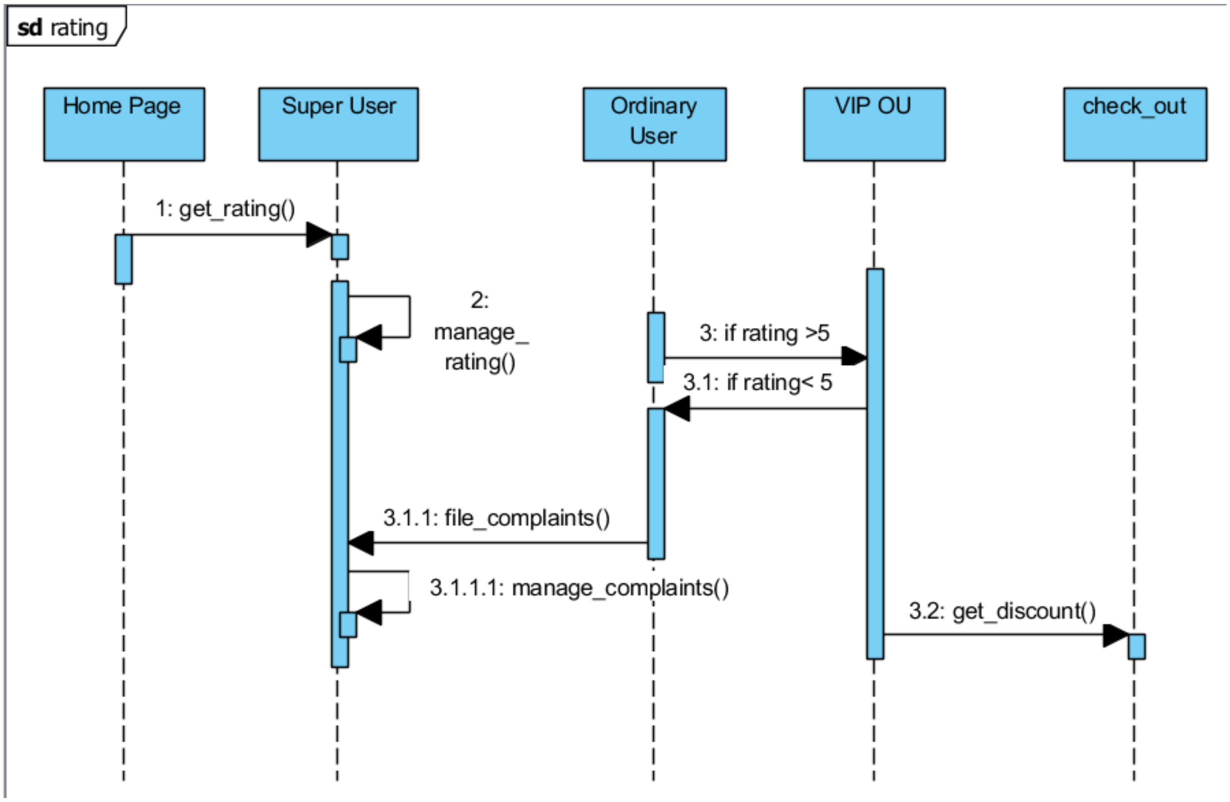
2.5: Rating:

Normal Scenario:

After the user purchase item, both buyer and seller can rate each other based on the service. The rating is based on 0(worst) to 5(best) scale.

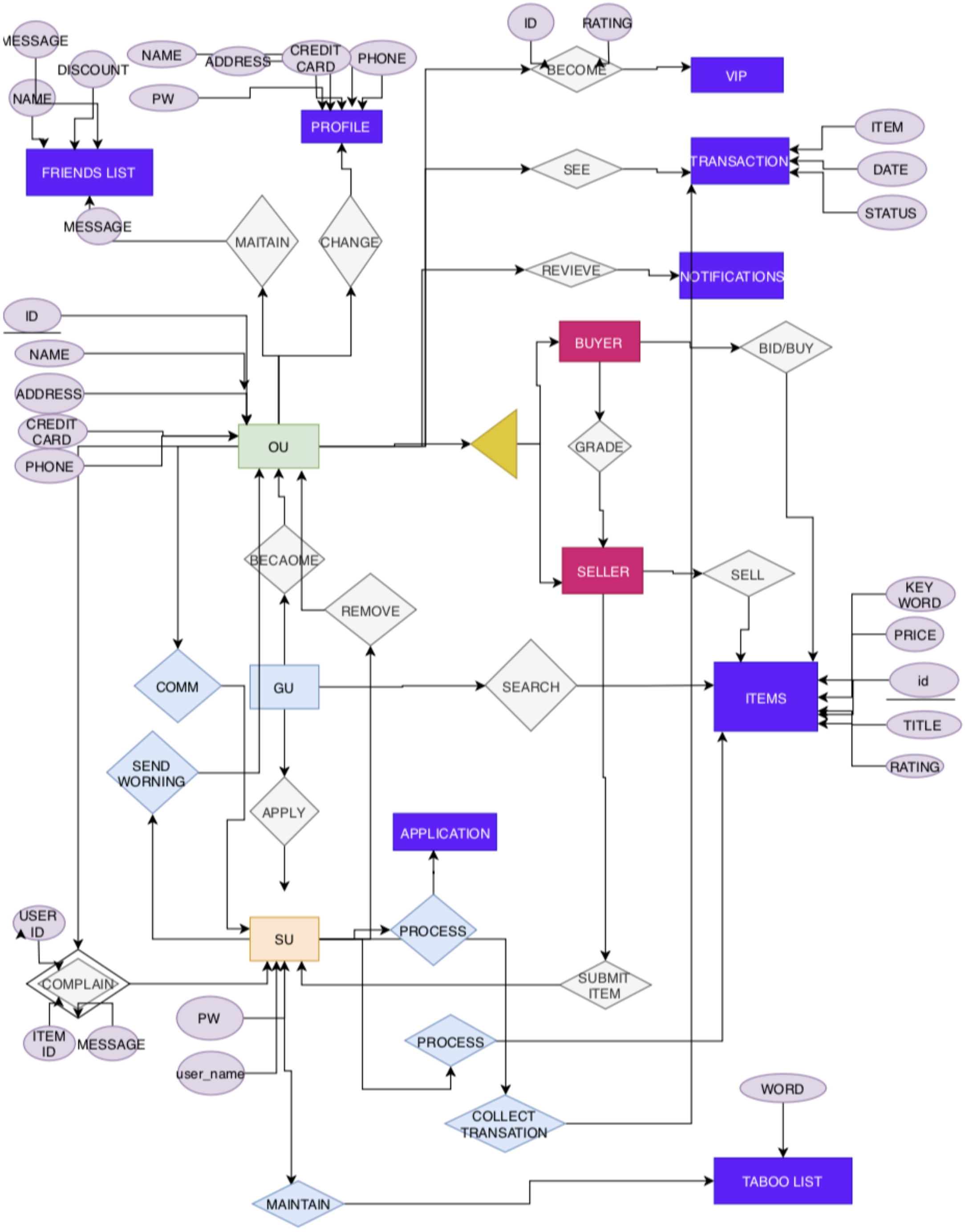
Exceptional Scenario:

If rating is greater than 4 ordinary user become VIP OU. VIP OU will get 5% discount on check out. If ordinary user submitted more then three low rating or three high rating, super user will warm it as reckless grader. A VIP OU is moved to ordinary user if his rating is below 4 and send him warning.


sd rating
Home Page
Super User
Ordinary User
VIP OU
check_out
1: get_rating()
2: manage_ rating()
3: if rating >5
3.1: if rating< 5
3.1.1: file_complaints()
3.1.1.1: manage_complaints()
3.2: get_discount()

3. E-R Diagram

An entity-relationship diagram of our system is shown in figure below, it outlines our database and shows the relationship between object, place and event with in entire system.



4. Detailed design: for EVERY method use pseudo-code to delineate the input/output and main functionalities

- makeBidItem(String itemName, String/float bidTime, String/Image imageFileLocation, String serllerUsername, String keywords)
    - Executes update query as prepared statement: INSERT INTO Item(name,seller, bidTime,…) Values(? ? ? ...)
- makeFixedItem(String itemName, String/float price, String/Image imageFileLocation, String sellerUsername, String keywords)
    - Executes update query as prepared statement: INSERT INTO Item(name,seller, price,...)
- isBidItem(itemID) : returns bool
    - Checks the itemID in the database to see if value for bidItem is true/false, and returns that value
- getItemPrice(itemID):String/float
    - Executes query as prepared statement: SELECT price FROM Item WHERE id= ?
    - Returns the fied price of the item
- getItemImage(itemID): String/Image
    - Executes query as prepared statement: SELECT image FROM Item WHERE id= ?
    - Returns the address of the image or an Image instance
- getItemHighestBid(itemID): String/float
    - Executes query as prepared statement: SELECT amount FROM Bid WHERE item= ?
    - Iterate through the result set and find bid with largest amount
    - Return largest amount (or return [largest amount, username])
- getItemKeywords(itemID): String
    - Executes query as prepared statement: SELECT keywords FROM Item WHERE item=?
    - Returns list of keywords seperated by a comma in a String
- addBid(username, itemID, bidAmount)
    - Executes query update as prepared statement: INSERT INTO Bid(user,item,amount) VALUES(?,?,?)
    - Adds the bid of the user for the item to the database
- addPurchase(buyerUsername, sellerUsername, itemID, price)
    - Executes query update as prepared statement: INSERT INTO Purchase (buyer,seller,item,price) VALUES(?,?,?,?)
    - Adds purchase to the database
- addRating(itemID, seller, buyer,rating)
    - Executes query update as prepared statement: INSERT INTO Rating(item, seller, buyer, rating) VALUES(?,?,?,?)
    - Adds rating to the database

- addKeywords(itemID, keywords)
    - Executes query update as prepared statement: INSERT INTO Item(keywords) VALUES(?) WHERE id=?
        - Might need an UPDATE statement
- getNotifications(username): ArrayList
    - Retrieve a list of notification ids for provided user
- getNotificationTitle(id):String
    - Returns the title of the notification associated with the id provided from the database
- getNotificationMessage(id):String
    - Returns the message of the notification associated with the id provided from the database
- makeComplaint(title,message,user)
    - Creates a complaint message by the user that is sent to the SUs. Complaint is added to the Complaint(title,message,user) table. The title and message are provided in the GUI
- getItemsSold(username): ArrayList
    - Selects ids of all rows from Item where seller=username
    - Collects all itemIDs into an arraylist and return it
- getItemsBought(username):ArrayList
    - Select ids of all rows from Item where buyer=username
    - Collect all itemIDs into an arraylist and return it
- makeCancelationRequest(itemID, cancelingUser, cancelingMessage)
    - Adds a request to the CancellationRequest table
- getFriends(username)
    - Looks up usernames of friends of associated user
- addFriend(user,friend)
    - Adds Values(?,?) into Friend(username,friendUsername)
- deleteFriend(user,friend)
    - Removes (username,friendUsername) from Friend table

5. System screens:

Here,we demonstrate major GUI screens of the system and showcase our Sign up and Login functionality in our system.

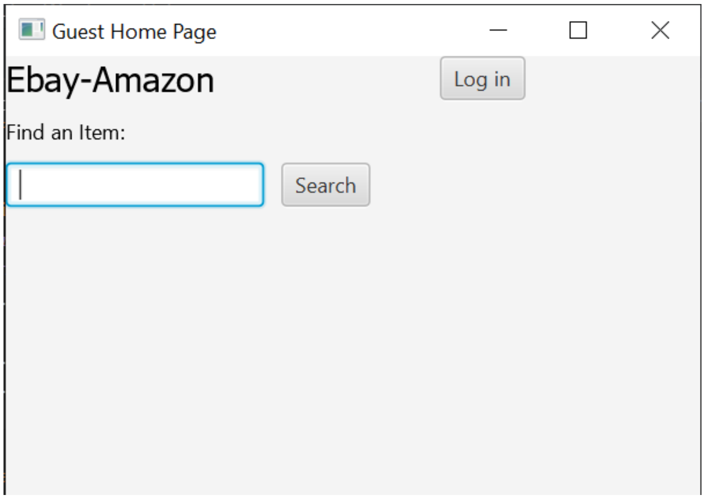


Figure 1: Guest Home Page with only Search bar. Login button to login or signup.

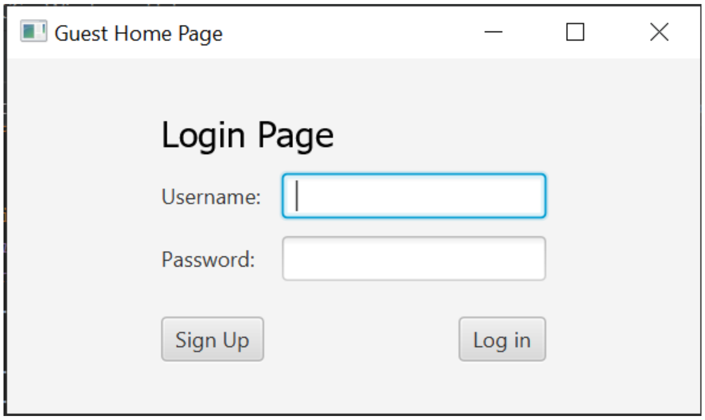


Figure 2: Login Page where user can login if they have an account or sign up if not.

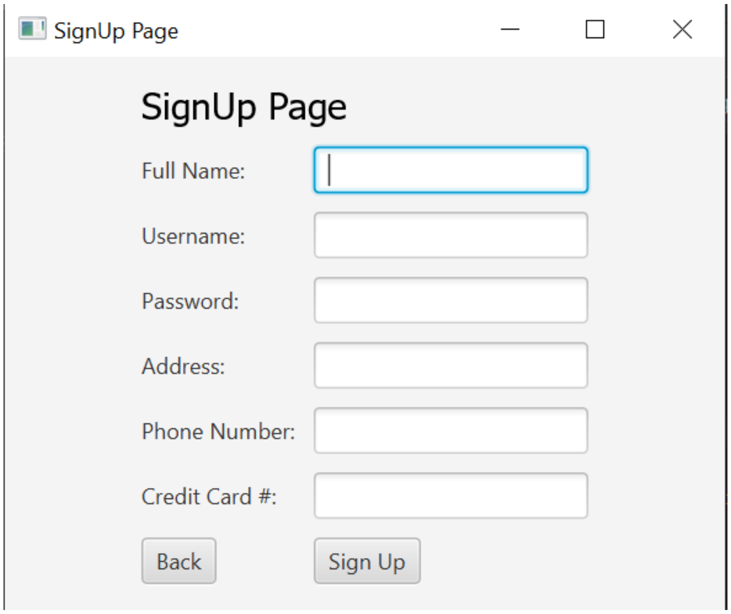
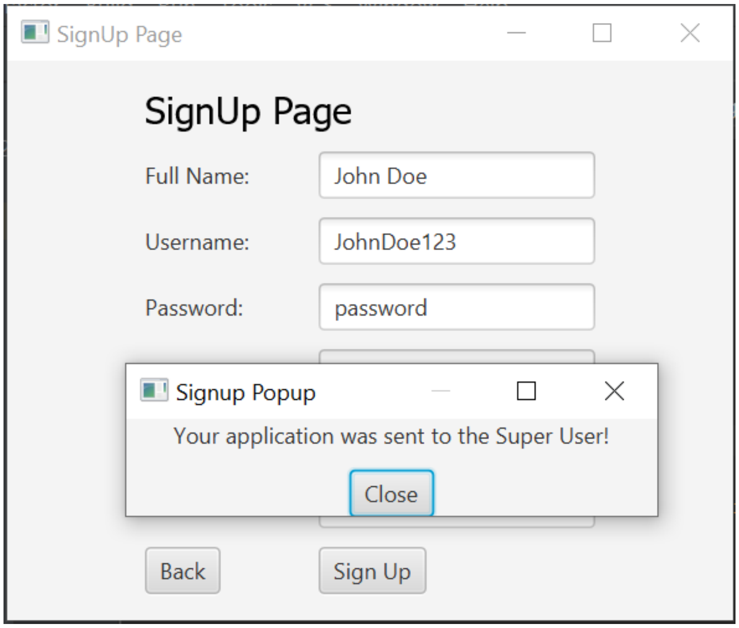


Figure 3: Sign up page



Full Name: John Doe

Username: JohnDoe123

Password: password

Your application was sent to the Super User!

Figure 4: Successful Sign up with a pop up.

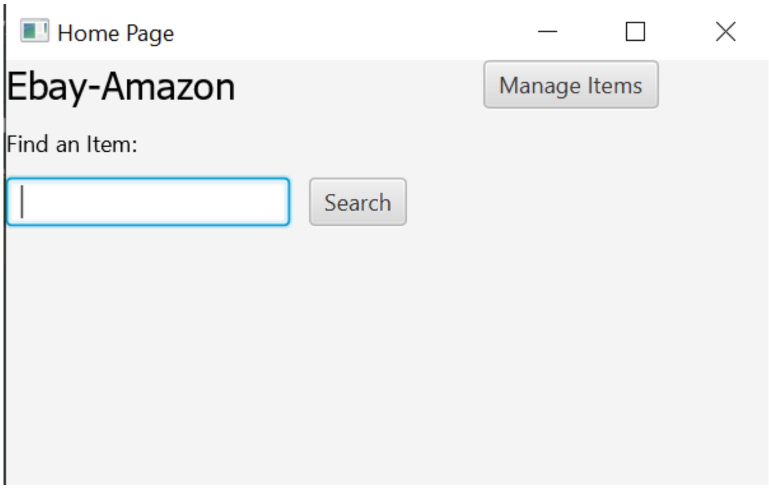


Figure 5: Home Page for Ordinary User, should have more buttons for managing an account.

6. Minutes of group meetings

- ~3 hours of coding
- ~10 hours of designing and planning.
  - Meeting 1: Split up who is in charge of what (Frontend and backend) read the requirements.
  - Meeting 2: Start drawing low level design for the layout of how the program should look like.
  - Meeting 3: Continuation of low level design and some coding.

7. Address of the git repo (github, gitlab, bitbucket, etc) of your team's work so far - put all materials including this report there

https://github.com/andyyeung122/EbayAmazon