Phase II: Design Report

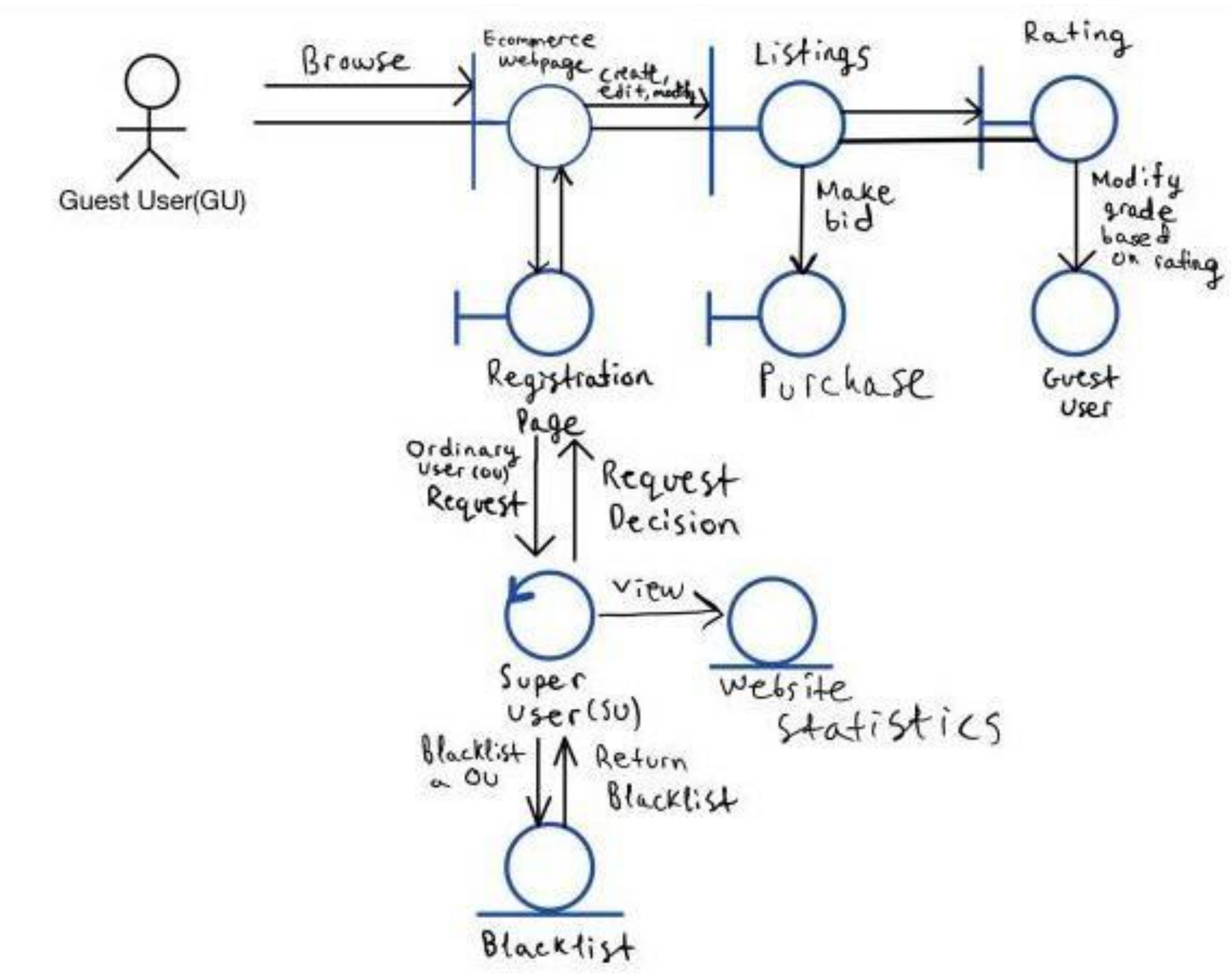
CSC 32200- Software Engineering Team D

Written by:
Mohammed Rahman, Mahfuzul Islam, Thierno Dicko, John Kim

Introduction

This specification document presents the design details and functionalities of our online ecommerce site . The main purpose of this report is to provide further details on understanding our system. The collaboration class diagram below describes an overall picture of our system and how the objects interact with the system and the main functionalities of the program.

Collaboration Diagram



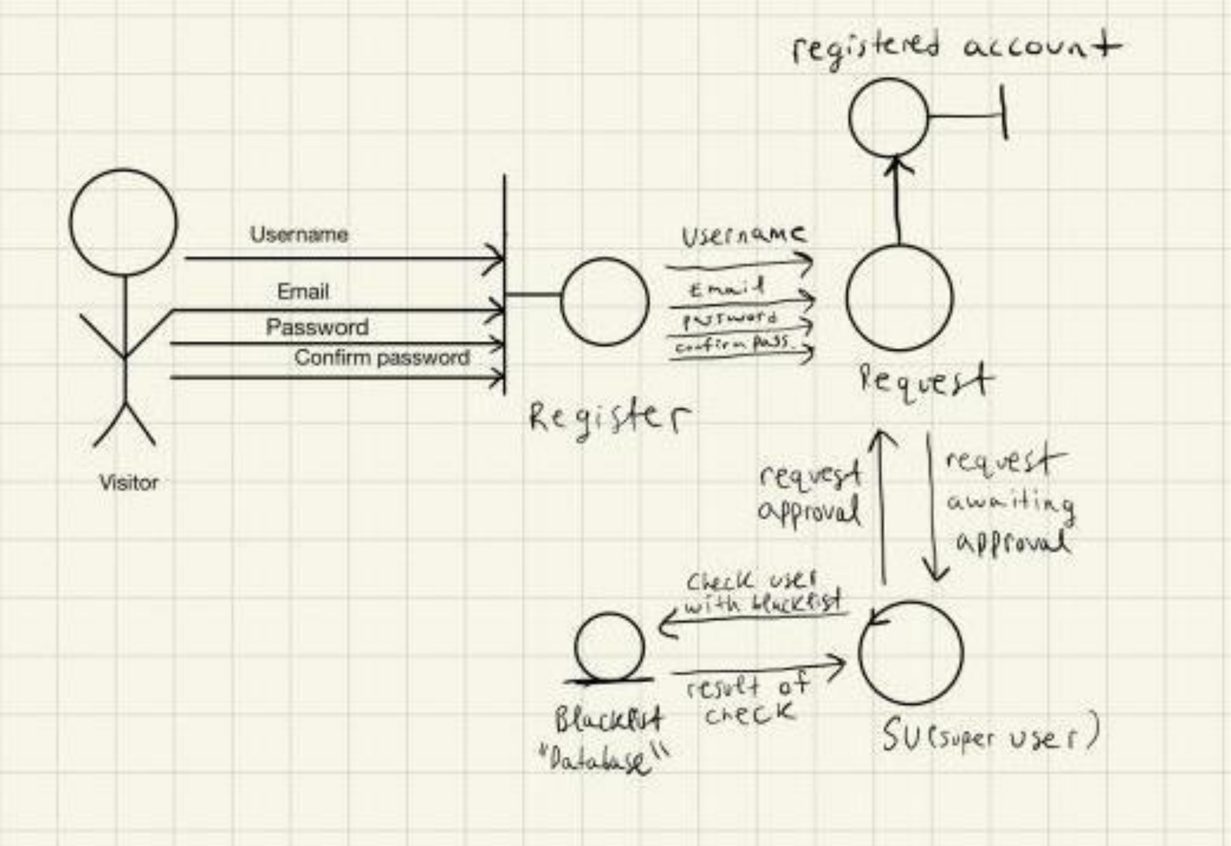
Use Case Analysis

In this section of the report, we provide more details on the main functionalities of the program that were introduced in the specification report. We provide collaboration class diagrams to describe the relationship between their classes and objects. We also provide petri-nets to gain a better understanding of the use case diagrams from the originating specification report.

Registering an Account on the Site

Normal Scenario: On the landing page, a guest user will have the option to sign up for an account if they have not already registered. The visitor must provide their username of choice, their email, password, and also their confirmation of their password(case-sensitive).

Exceptional Scenario: the information that will be provided by the guest user must go through a validation check. If the information given is currently present within the system's blacklist, then the guest user will not be allowed to register or apply to become an Ordinary User. A message will popup stating that the guest user's application to become an OU has been denied. As a result these users unable to register are only considered as Guest Users.

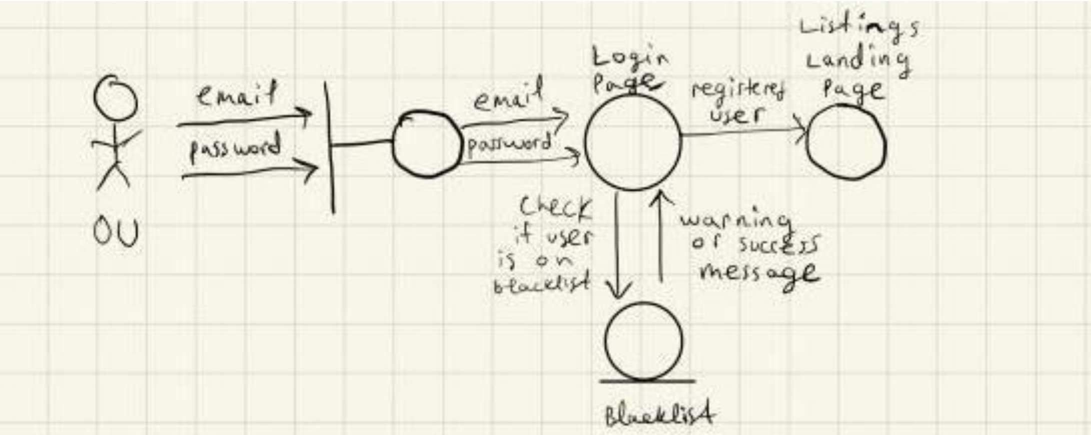


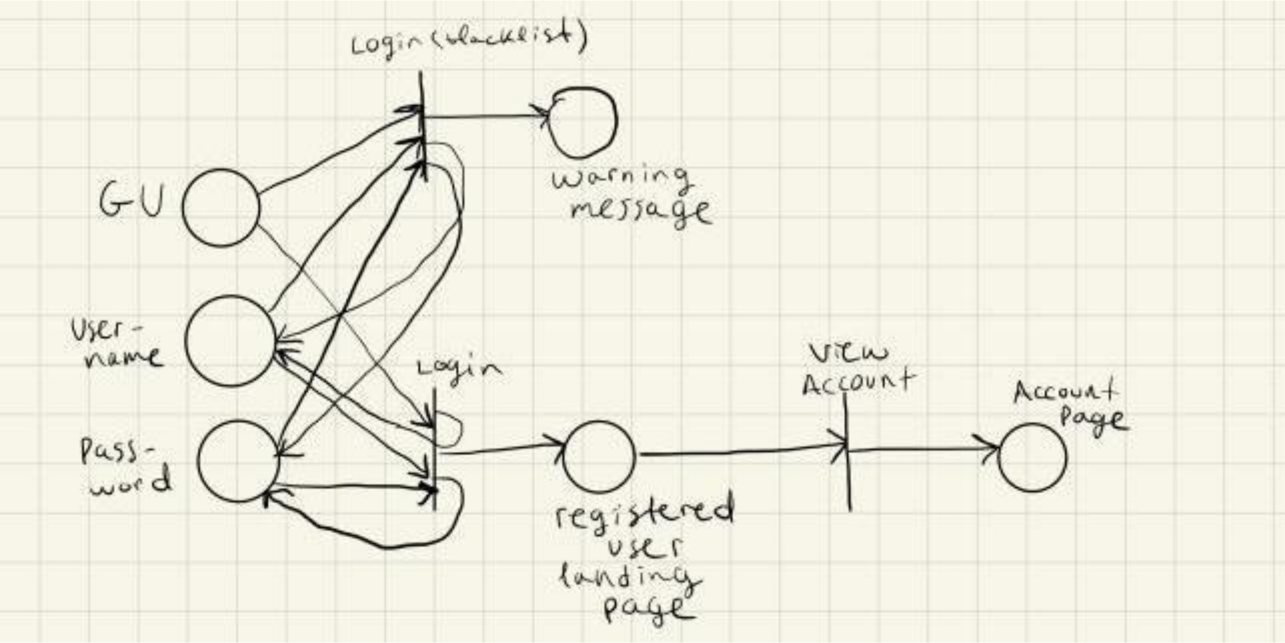
Logging onto the Site

Normal Scenario:

A guest user has the option to sign into their account. The sign in page will prompt the user to enter their username and password. On the condition that the login process is successful, the user will gain access to their account page and will be able to view the recommended listings on the site. At this point in time, the guest user is now considered as an Ordinary User within the confines of the system.

Exceptional Scenario: If a user has more than three bad ratings, a super-user will have the ability to place them on the blacklist. If this is the case, after the user then inputs their login information, they will see a warning message on the screen and will not be able to login to their account. If a guest user enters an incorrect username and password, they will be redirected to the login page and have the ability to try entering their information once more.

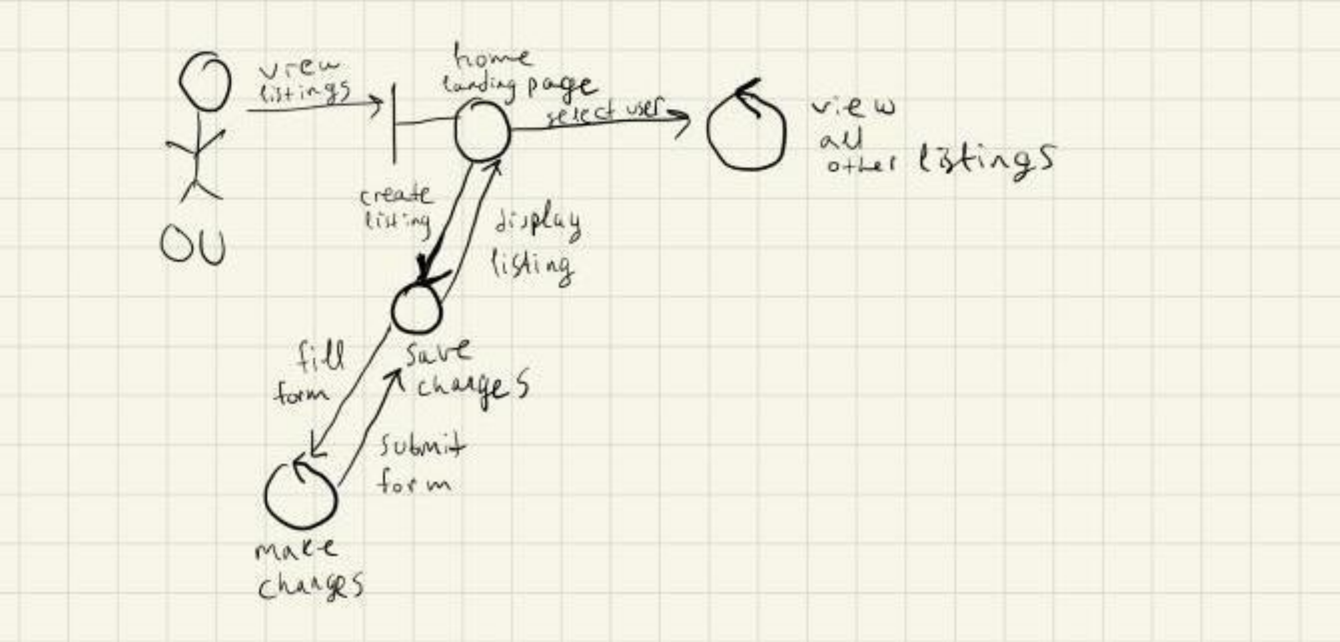




Item Listings and Bidding

Normal Scenario: Ordinary Users have the ability to view trending item listings on the home page once logging in. An OU can sell a posted item and submit a bid on an item of interest. They may search and browse through items based on their price and ratings.

Exceptional scenario: OUs may see a different screen of item listings depending on their interests.



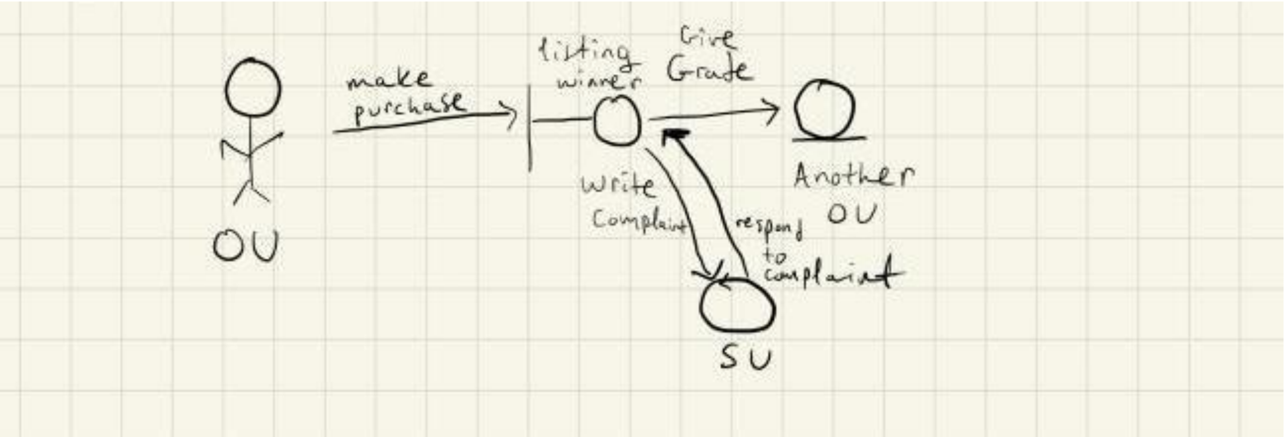
Grading and Complaint System

Normal Scenario:

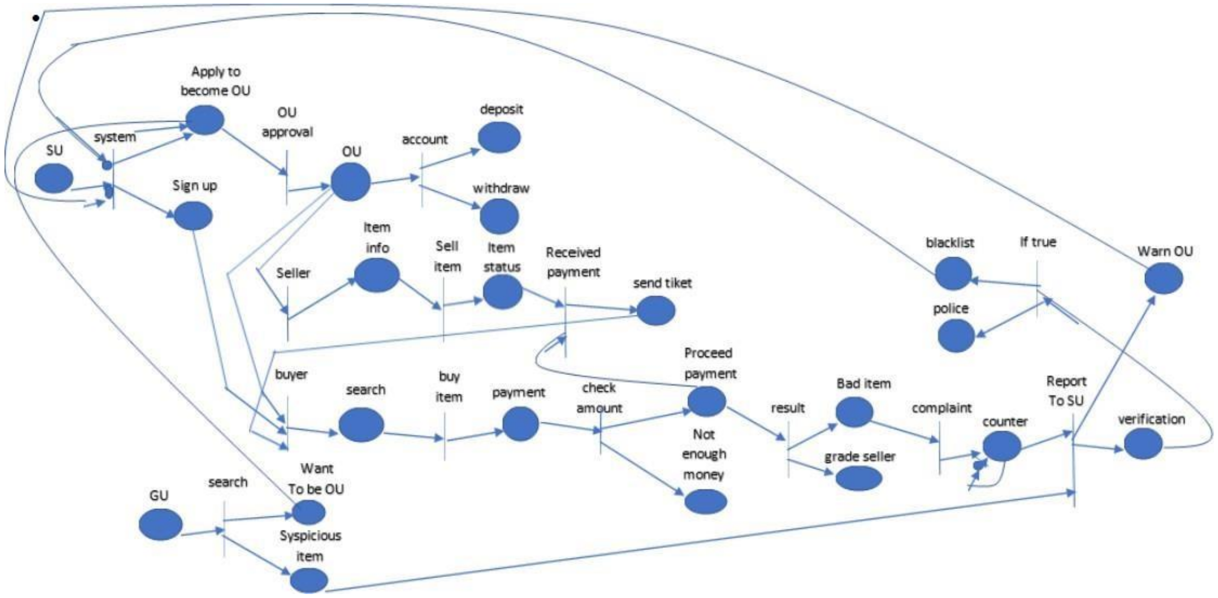
In normal cases, the Ordinary Users on the site will be able to rate one another after buying from an OU, on a scale of 1-5. They also will be able to file any complaints against an OU who sold an item to the Super User.

Exceptional Scenario:

The owner of an item decides whether transaction should proceed, and if denied, a grading or complaint won't ensue. The SU may further escalate the situation by responding to complaints, and suspicions of stolen items by both blacklisting an item and/ or user.

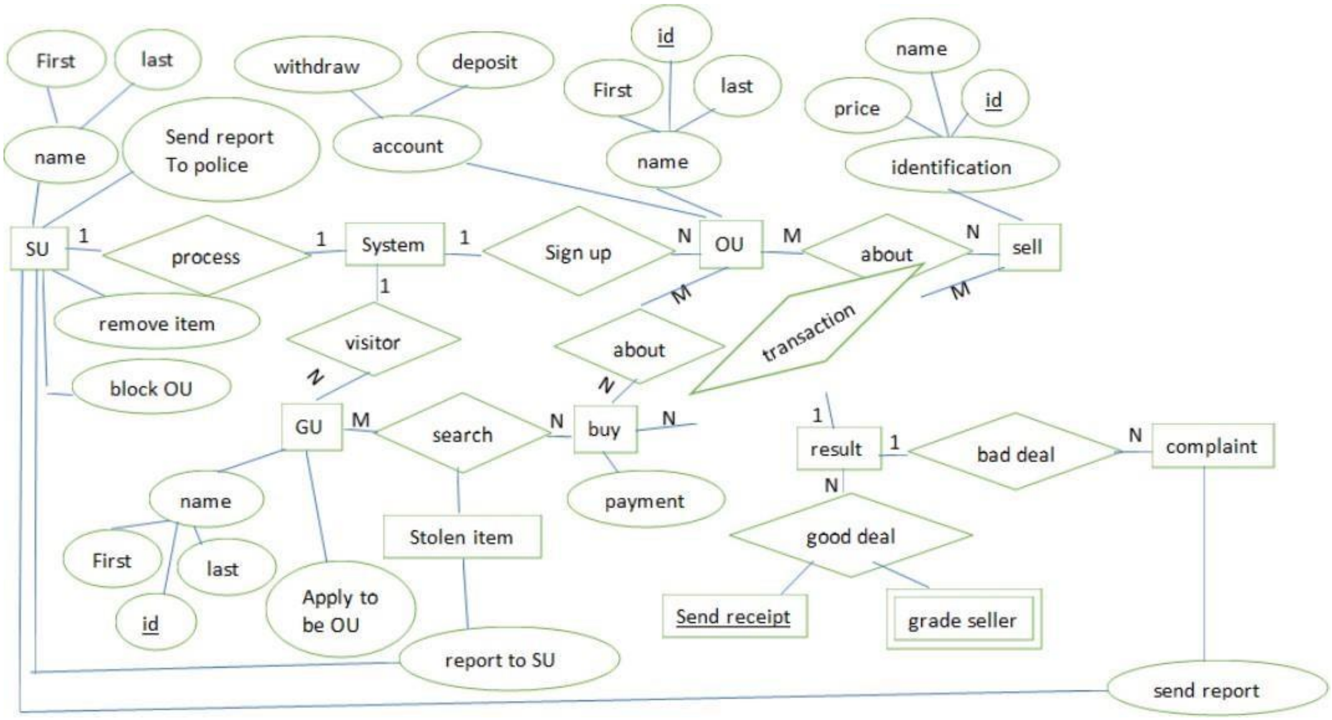


Petri-Nets



The ER diagram below describes the interrelations of the entities within the system.

E-R Diagram



Detailed Design of the Overall System

Register and log in:

```
// depending on the tab used to sign in or register
function parseForm()
get value of tab from HTML IF value
THEN registerUser()
ELSE
signInUser()
ENDIF
END

//if a user clicks the confirmation button while on the register tab call this function on button click
// user must enter email, username, password
function registerUser()
get email from text field
get username from text field get password from text field
get confirm password from text field

get database object index database for email IF email exists
THEN throw error ELSE
        IF isValidPassword() AND isValidUsername()
        THEN create new table in database with username, pass, email ELSE throw error
        ENDIF ENDIF
        close database
END

// if a user clicks the confirmation button while on the signin tab call this function on button click
function signInUser()
get username from text field get password from text field get user database object index database for username IF
username exists
THEN retrieve username table check table for password
IF password == password from database THEN go to home page
ELSE throw error ENDIF
ELSE throw error close database ENDIF
END

// checks database to see if email or username already exists
// also checks to see if email and username is in correct format
// throw error depending on which is incorrect
functionisValidUsername(email,username) get user database object
index for email
if email exists in database THEN throw error close database return false
ENDIF

    IF username exists in database THEN throw error
    close database return false
    ENDIF
    close database return true
END
```

```
//registerUser()
// takes the given password and the confirmed password as arguments
//check the password format
function        isValidPassword(pass,confirm) IF pass >= 8 AND pass == confirm
THEN return true
ELSE
return false ENDIF
END
```

Marketplace:

```
// loads the currently viewed items by retrieving them from the
// database and stores them in the table
function loadItems(component)
get database object for component get table HTML object
FOR amount of items in database for component
populate table with component name, image, price, description ENDFOR
close database
END
```

```
// opens a text field for the clerk
function promptClerk() overlay text field
END

// when the textfield is submitted


function parseClerkPrompt()
get text field for clerk prompt element IF text empty
THEN send warning to clerk notify manager
ENDIF
END
```

System Screens
Under construction

Group Meetings:

Sep 15th
We chose our team, held our first meeting the following week, and spoke about how to move the project further.

Sep 22th
We made group chat where we shared any relevent documents with each other throughout the semester

Oct 27th
We all decided to work with Python and Django after discussing which programming language we are most comfortable with in the group. We persisted with this concept even though we had no prior expertise with Django. We split up into two smaller groups to work on the front-end and back-end.

Nov 3rd
We suggested that the login and sign-up pages should be included in the website's first phase of development. It was argued that doing this would make the team consider the various user roles for the website and necessitate the creation of a simple database for user data.

Nov 24th
The development of each person in their role was the topic of discussion. The majority of the team was preoccupied with work from other classes and was unable to finish their parts. Each team member was given a role to play in order to complete the project's final phase.

Github Repository
Link: https://github.com/mrahman017/EBID-EBUY.git

Contributors:
Mohammed Rahman- mrahman017
Mahfuzul Islam- mahfuz6543
Thierno Dicko-
John Kim-