

---

**Team 13**

---

**OsakiTenka  
Design Report  
For eBusiness System**

**Version 2.0**

OsakiTenka	Version: 2.0
Design Report	Date: 4/19/2019
Spring 2019	

## Revision History

Date	Version	Description	Author
3/23/2019	1.0	Specifications and Planning	Mingzhi Xu, Yu Bin Liu, William Chung, Jia Peng ,Zhen
4/19/2019	2.0	Design Report	Mingzhi Xu, Yu Bin Liu, William Chung, Jia Peng Zhen



OsakiTenka	Version: 2.0
Design Report	Date: 4/19/2019
Spring 2019	

## Table of Contents

1. Introduction
  - 1.1 Collaboration Diagram of whole system
2. All use cases
  - 2.1 Collaboration and Petri-net for each case
  - 2.2 E-R Diagram for the entire system
3. Detailed design
  - 3.1 Pseudo-code
  - 3.2 GUI Prototype
4. Supporting Information
  - 4.1 Group Meeting Details
  - 4.2 Github



OsakiTenka	Version: 2.0
Design Report	Date: 4/19/2019
Spring 2019	

# Design Report

Confidential

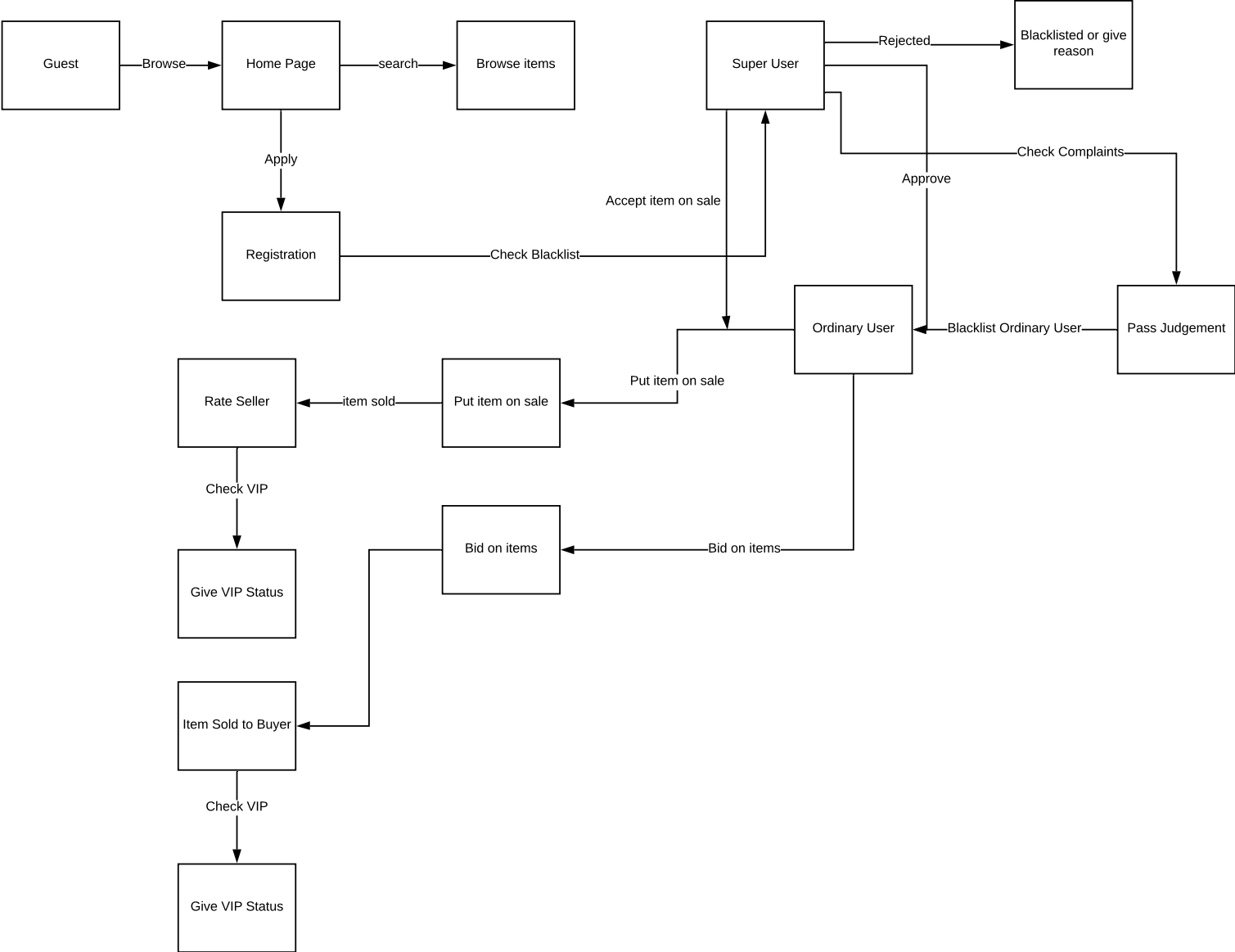
©Team 13, 2019

Page 4

## 1. 1 Introduction

OsakiTenka will be an eBusiness platform system that will work similarly to eBay and Amazon, which allows users to bid, buy, and sell items. There will be three types of users for this system: superusers, ordinary users and guests. The superusers will be able to manage the activities of ordinary users and guests. The ordinary users will be able to bid, buy, and sell items that are listed in this platform, file complains and rate items, check transaction history and change user information, enjoy VIP status and friend list discounts. The guests are only able to search items and apply to become an ordinary user. The purpose of this document is to provide more technical details on how the system would work.

The diagram below shows the overall picture of the system using a collaboration class diagram. It shows how a guest, Ordinary User, and Super User can interact with the website and their limitations . For instance, the guest can only access the website and browse the bids, but unlike the Ordinary User, guests cannot buy and sell items. The Super User on the other hand acts like an administrator and supervises the Super Users

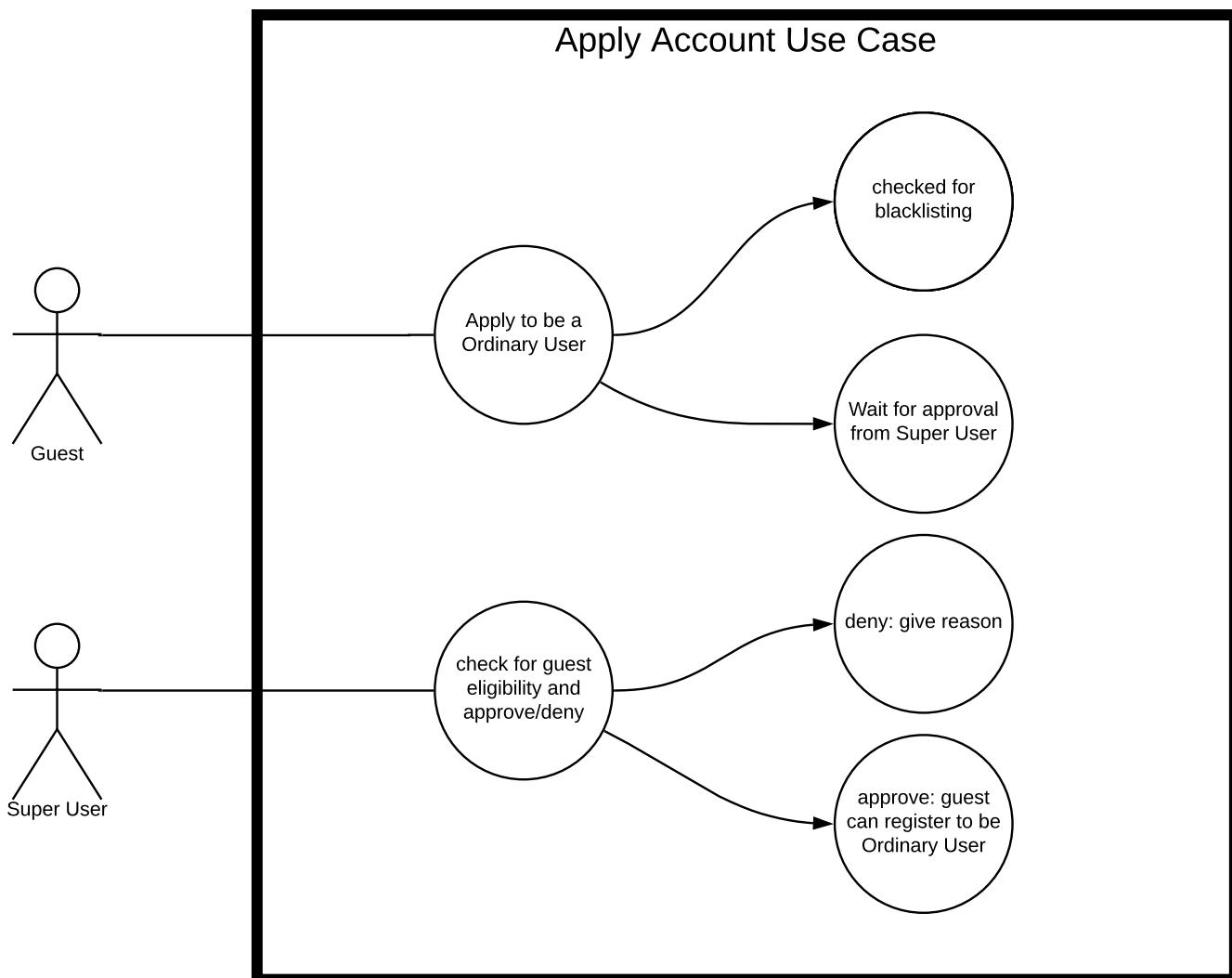


## 2. All Use Cases

This section will show scenarios on how each user would commit actions to the site. I will be using diagrams to illustrate the use cases and making it easier to follow. This section will be split into 3 parts: (1) Scenarios for each case, (2) Collaboration diagram for each use case, and (3) Petri-net diagram for each use case.

### Scenarios for each use case

Use case: Applying for an account



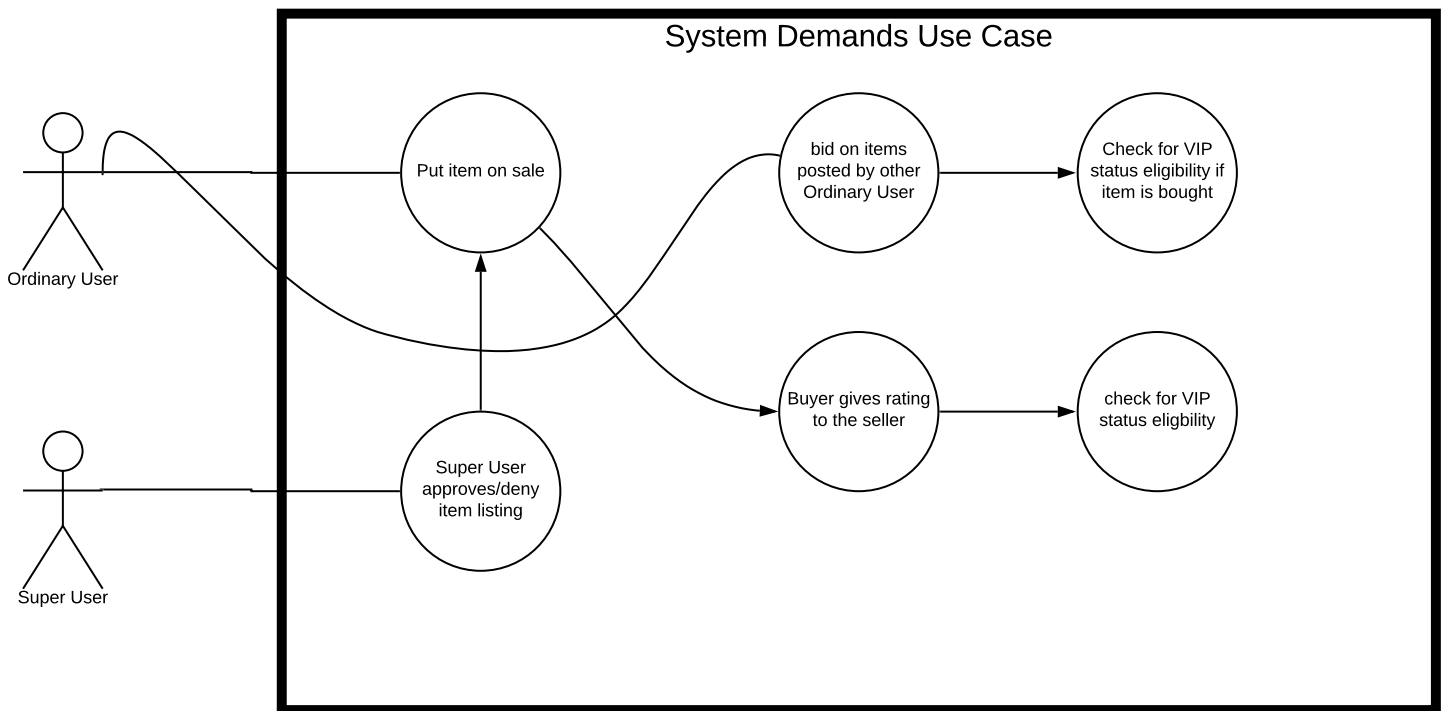


User: Guest

An unregistered Guest can apply to be a Ordinary user and will be checked on blacklist

1. A guest accessing the site can apply to be a Ordinary User
2. A guest has to wait for Super User to approve the application
3. Super User will check the guest's eligibility and approve/deny
  - A. If rejected the Guest cannot register to the website
  - B. If accepted, the Guest will have to provide registration information

Use-case: System Demands



Users: Ordinary Users/Super Users

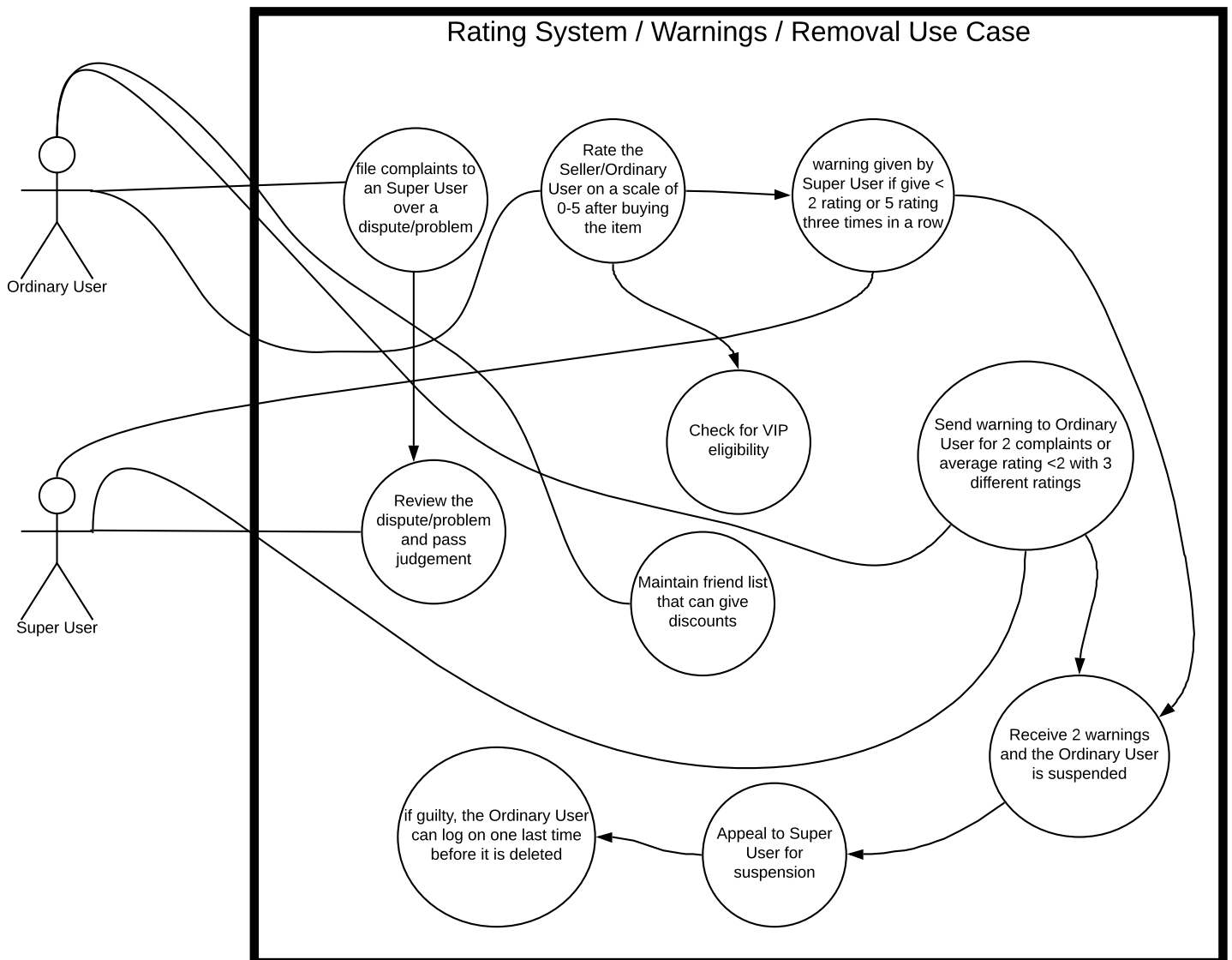
Ordinary Users can post an item to sell:

1. Ordinary User has to wait for the item listing to be approved by Super User
2. Ordinary User can bid on items posted by other Ordinary Users
3. The Ordinary User that bought the item can rate the seller on a sale of 0-5

The Super User is essentially an admin and oversees Ordinary Users:

1. Super User can approve/deny item listing

Use-case: Rating System/ Warning



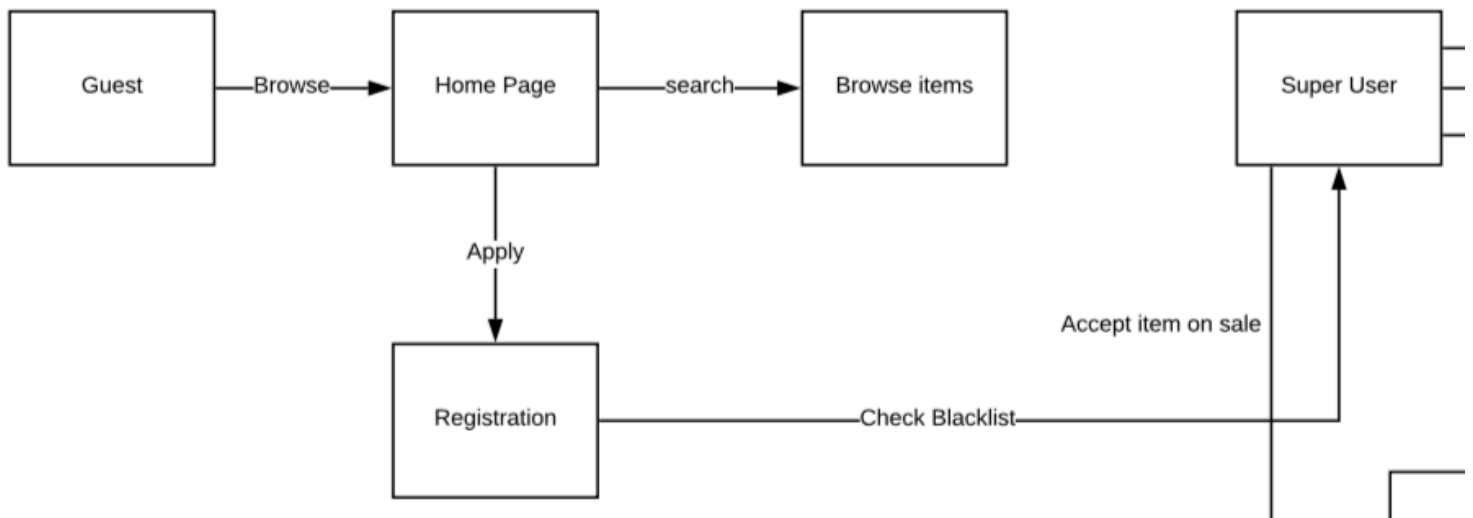
Users: Ordinary Users/ Super Users

A rating and warning system is put in place to ensure disputes are settled between Ordinary Users, with a Super User acting as adjudicator.

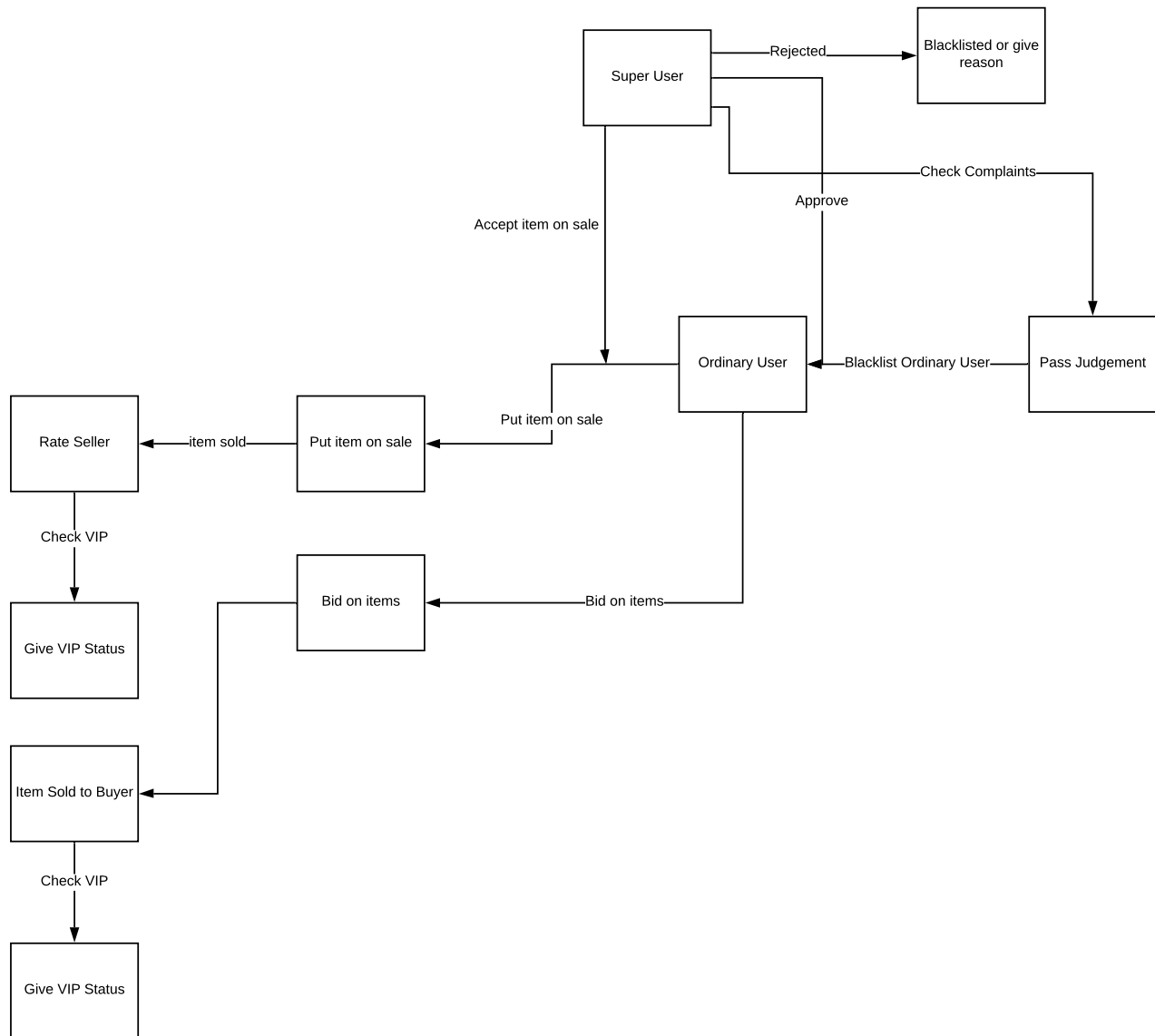
1. An Ordinary User will become a VIP if her/his rating is  $\geq 4$  (by 3 different Ordinary User) or spend more than \$500 and without any prior warnings.
2. The VIP status is removed if his/her rating is below 4 automatically or receives one warning.
3. A Ordinary User will received a warning if they use inaaprioate words in any context.
4. Super User will send a warning to Ordinary User who receives two complaints or average rating is  $< 2$  with three different ratings.
5. An Ordinary User that receives two warnings are suspended from the system and can apply to appeal
6. The Ordinary User account is deleted if the appeal is denied.

## Collaboration diagram for each use case

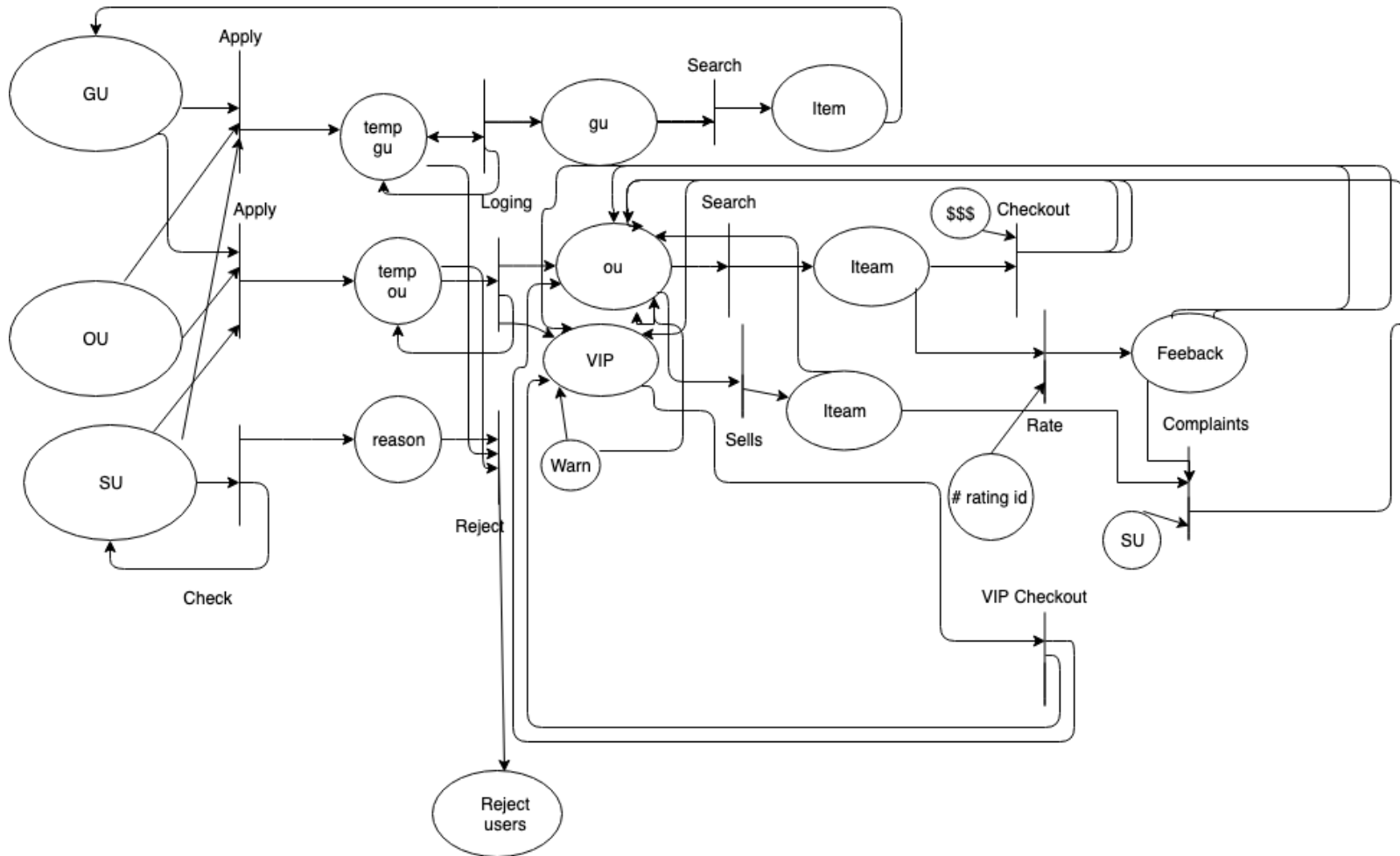
Use-Case: Applying for an account



## Use-case: System Demand/ Rating/ Warning



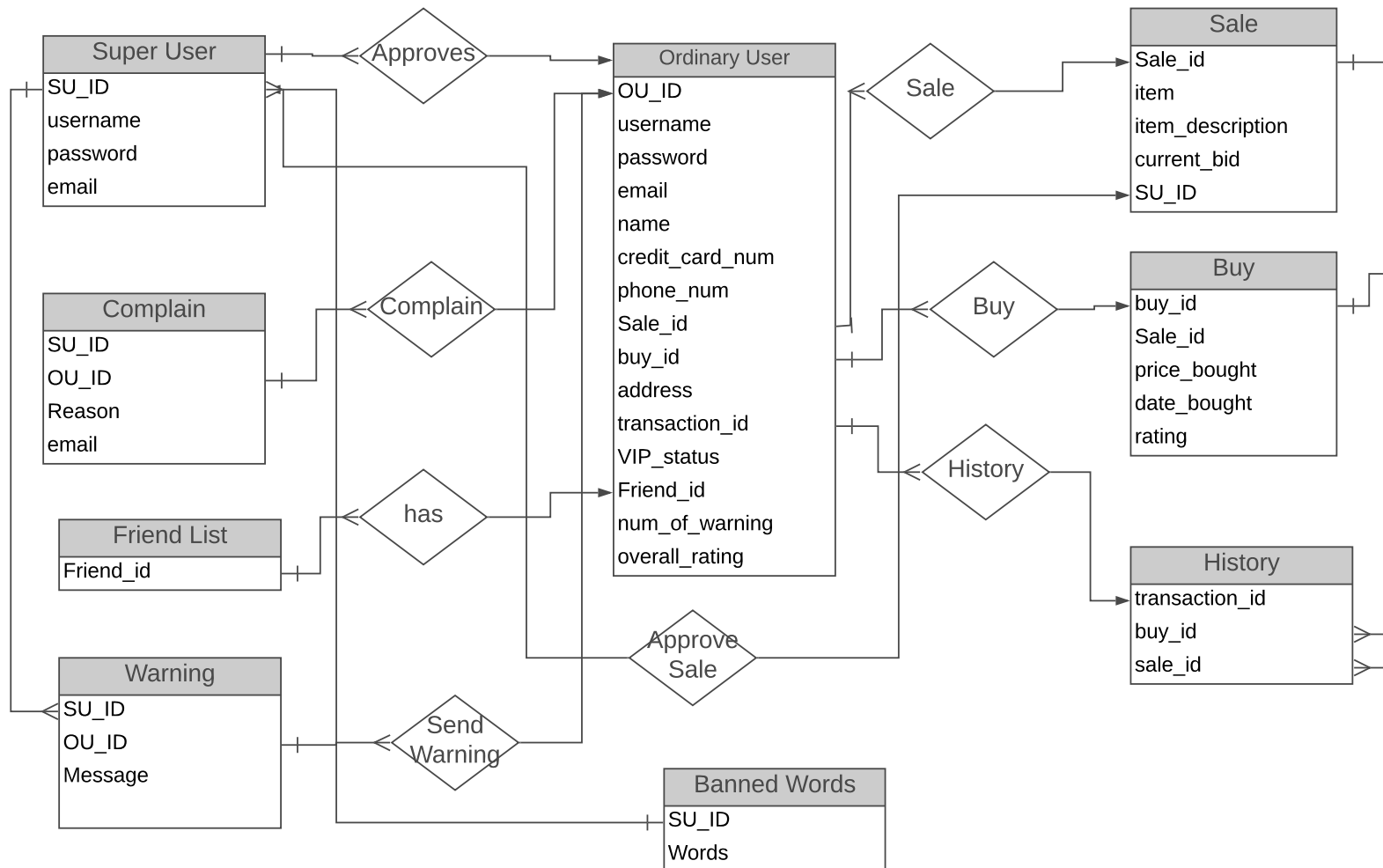
# Petri-net diagram for each case



### 3. E-R Diagram for the Entire System

This section explains the Entity Relationship diagram of the entire system. It shows how the database stores its data and how the data is used to interact with each other. Each shapes has its own meaning, rectangle is an entity, diamond is an action, connecting lines show the relationship between entities. In the diagram, each entity has an attribute.

For example, a Super User entity has four attributes namely, SU\_ID, username, password and email.



## 4. Detailed Design

The following are the core methods utilized in the system, along with the pseudocode that will be used as guidelines for hard coding them. The pseudocode is tentative and is more of a general concept of the method rather than set rules for development.

### **Registration(): {**

Prompt user for input for name, email, password, valid credit card, address, phone number

    If (!input.is\_valid()) { return to register\_page with error message }

    Else if ( user is blacklist ) { send user to home\_page with error message }

    Else if (user already registered) { send user to login\_page with message }

    Else:

        Create temp user profile

Insert temp profile into applications database

Send user to home page with success message

**}**

### **Login(): {**

    Prompt user for username and password

    Triggered by button press

    If (user does not exist)

        { send user to register page }

    If ( username is ordinary\_user\_account )

        { if (password corresponds to username in database)

    { set session to user\_id

Set profile and user\_controls to user\_id

    Send user to home\_page }

    If ( username is super\_user\_account )

        { if (password corresponds to username in database)

            { set session to superuser\_id

            Set profile and user\_controls to superuser\_id

            Send user to home\_page

        } }

**}**

### **Logout(): {**

    Triggered by button input

    Prompt user for confirmation

    If (user selects yes )

```

        { set to guest session
          Send user to home_page }
    If (user selects no )
        { exit logout screen }
}

```

#### **Browse\_items(): {**

```

    Prompt user for search input
    Queries opened items based on name in items database
    If ( item is part of query search )
        { Find specifics of item and store in queryset
          Display items
        }
    User navigation including view bids, view details, etc.
}

```

#### **View\_item\_details(): {**

```

    Send user to item page
    Query items database for item details
    { item_id matches id in database }
    Display details on page
}

```

#### **View\_user\_profile(): {**

```

    Query user_profile from users database
    if (user == user_id in database) {
        Display user information
        Query transaction database where user == user_id
        Display user's transaction history along with opened bids
    }
}

```

#### **Post\_item(): {**

```

    Prompt user for title, price, keywords, and pics
    Infor are send to SU
}

```

#### **Buy\_Item():{**

```

    If buy:
        Submit buy
    If bid:
        Submit price
}

```



```

Choose_Buyer():{
    OU choose which bidder gets item
    If fixed price:
        Choose first one posted will be on top of list
    If bid price:
        Choose second highest bid
    Else:
        OU justified why first one not chosen

Report():{
    Submit a email/message to SU about problems or explain complaint
    Should ask for order#, buyer userID, seller userID
}

Rate_Items():{
    After successfully purchasing item OU have the right to rate item
    If OU submit 0 or 1 > 3 in a row or 5 > 3 in a row:
        Warning as reckless grader
}

Check_History():{
    Display user transaction history of successful purchases.
}

Change_Information():{
    User has the right to change password, name, address, credit card # but not ID
}

VIP_Status:(
    Check rating
    If average >= 4 (by >= 3 different other OU) or > $500 and no warning:
        Promote to VIP Status
        Purchasing item 5% discount
    Else:
        Remove VIP Status
}

Friend_List():{
    OU can add other OU into friend list
    OU can set discounts for friends
    OU are able to receive and send friend-only message
}

```

```

    OU is free to add/remove friends
}

Approve_Reg():{
    SU looks over register infor
    If approve:
        Account add into DB
}

Approve_Item():{
    SU looks over submitted item info
    If approve:
        Item add into DB
}

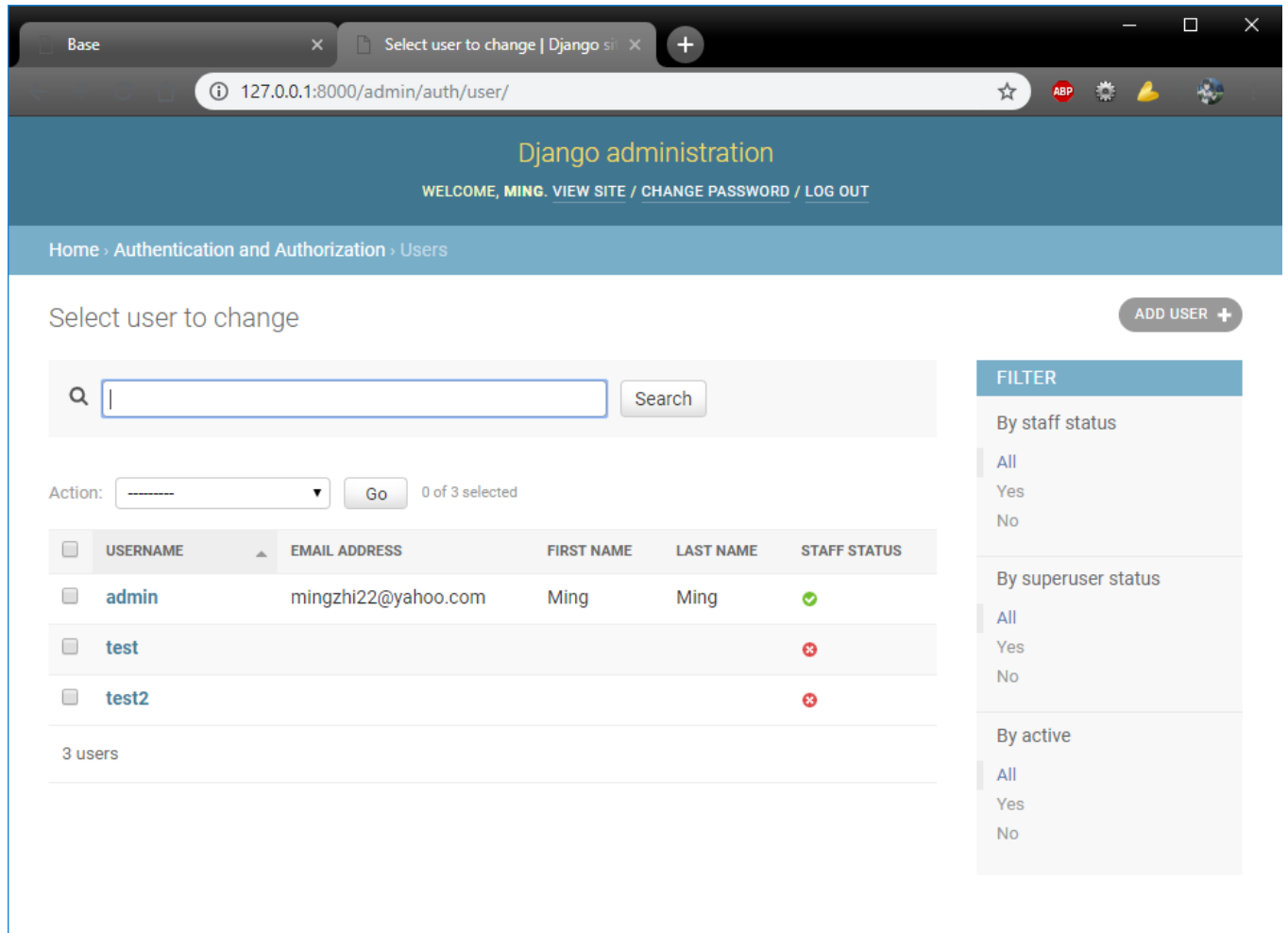
SU_Warning():{
    If complaints justified > 2 or rate < 2 (by >= 3 graders):
        Big_warning+=1
    If big_warning >=2:
        Notified next login
        If appeal:
            If approve:
            Else:
                denied
                User is removed
        Else:
            Resign
}

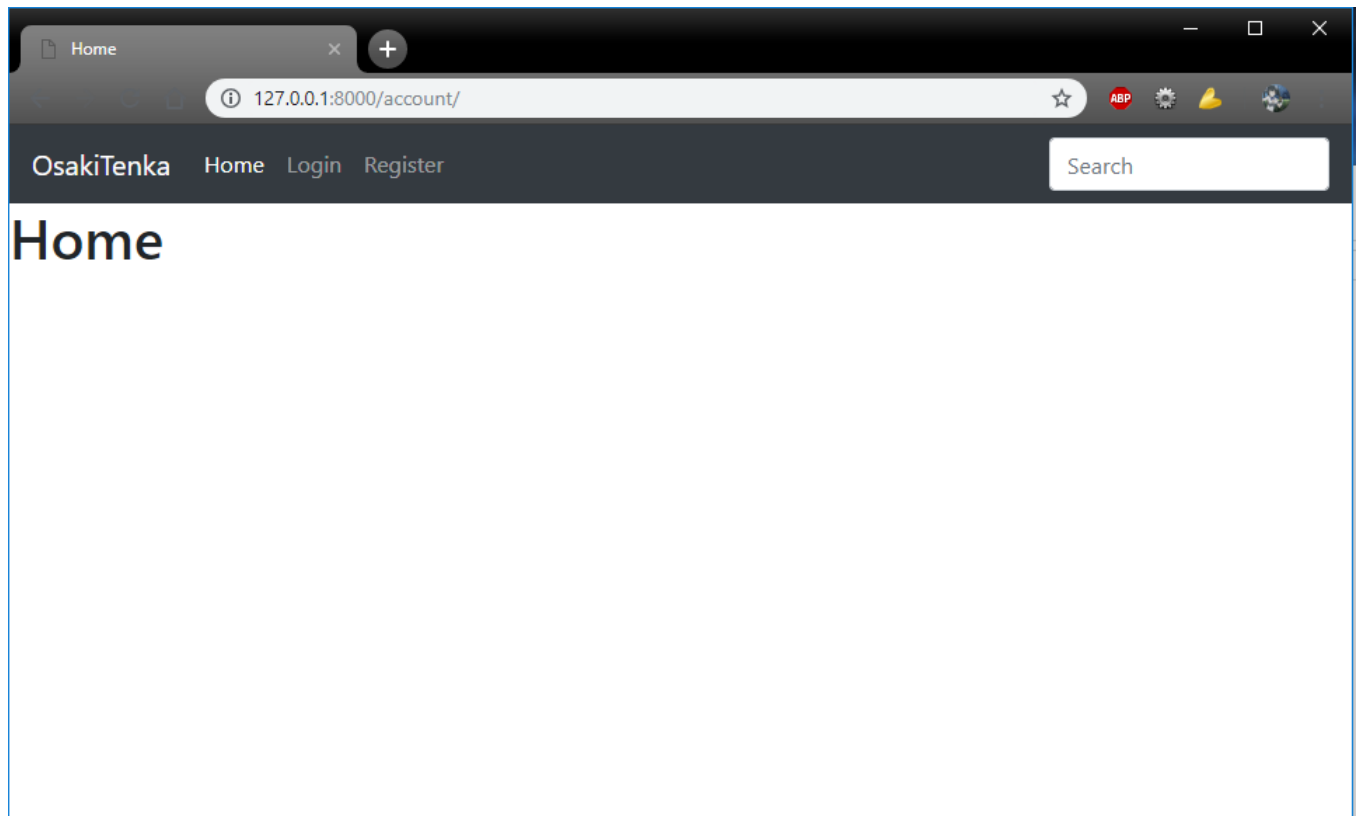
Remove_OU():{
    SU can remove any OU based on judgement
    Removed OU can login last time before account is deleted and remove from DB
}

```

## 5. System Screen

These are some of the prototype pages that are currently in our web application. These are our initial designs and there will be more improvements in the future. This section also shows one of our website functionality and we will feature the steps on how a client creates a posting.





A screenshot of the 'Register' page in the 'OsakiTenka' application. The browser's address bar shows the URL '127.0.0.1:8000/account/register/'. The page has a dark navigation bar with the 'OsakiTenka' logo, links for 'Home', 'Login', and 'Register', and a search bar. The main content area is titled 'Register' and contains the following form elements:

- Username:** A text input field. To its right, the text reads: 'Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.'
- Password:** A text input field.
- Password requirements:** A bulleted list of four requirements:
  - Your password can't be too similar to your other personal information.
  - Your password must contain at least 8 characters.
  - Your password can't be a commonly used password.
  - Your password can't be entirely numeric.
- Password confirmation:** A text input field. To its right, the text reads: 'Enter the same password as before, for verification.'
- Submit:** A button at the bottom of the form.

Base

127.0.0.1:8000/account/login/

☆ ABP ⚙️ 🔔 🌐

OsakiTenka

Home Login Register

Search

# Welcome

You can log in here!

## Login

Username:

Password:

Log in

The screenshot shows a web browser window with two tabs: 'Base' and 'Change user profile | Django site'. The address bar displays the URL '127.0.0.1:8000/admin/accounts/userprofile/1/change/'. The page header is 'Django administration' with a welcome message 'WELCOME, MING.' and links for 'VIEW SITE', 'CHANGE PASSWORD', and 'LOG OUT'. The breadcrumb trail is 'Home > Accounts > User profiles > UserProfile object (1)'. The main heading is 'Change user profile' with a 'HISTORY' button. The form contains the following fields: 'User:' with a dropdown menu showing 'test' and edit/add icons; 'Description:', 'City:', and 'Website:' each with a text input field; and 'Phone:' with a text input field containing '0'. At the bottom, there are four buttons: 'Delete' (red), 'Save and add another' (blue), 'Save and continue editing' (blue), and 'SAVE' (blue).

## 6. Minutes Of Group Meetings

Every Monday and Wednesday our group comes together for 30 minutes to discuss our plans for OsakiTenka. So far, our total group meeting accumulate to 4 hours.