
The A-Team

Akram Uddin, Fatoumata Drammeh, Israel Pina, Desiree Caceres, Emmanuelle Padilla

Salutorg.com Mini E-Bidding System Design Report

Version <1.0>

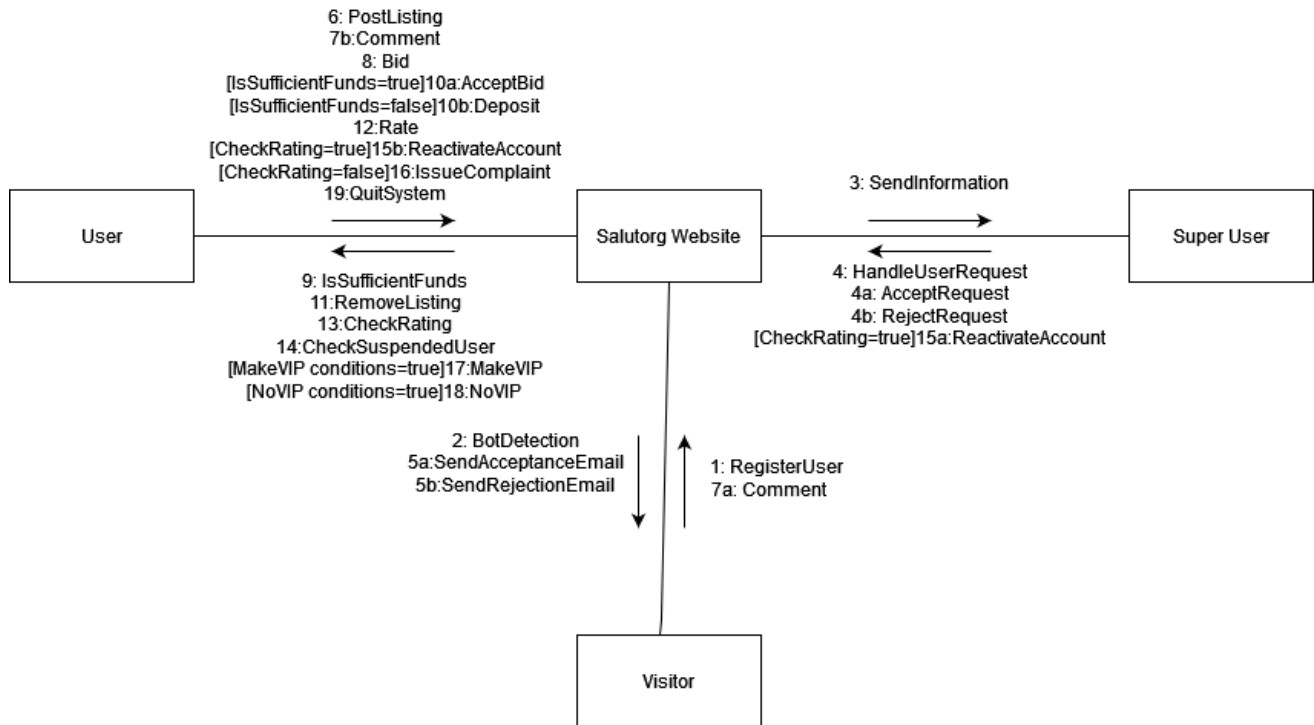
Table of Contents

1.	Introduction	<i>2</i>
2.	Use-Case Model	<i>3</i>
3.	Entity Relationship Diagram	<i>12</i>
4.	Detailed Design	<i>15</i>
5.	System Screens	<i>28</i>
6.	Group Meetings Memos	<i>37</i>
7.	Github Address	<i>37</i>

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

1. Introduction

Collaboration Class Diagram

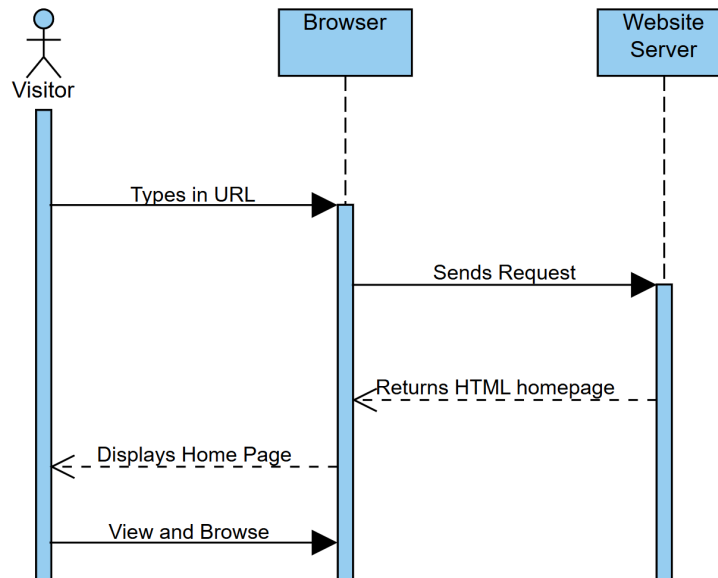


<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

2. Use-Case Model

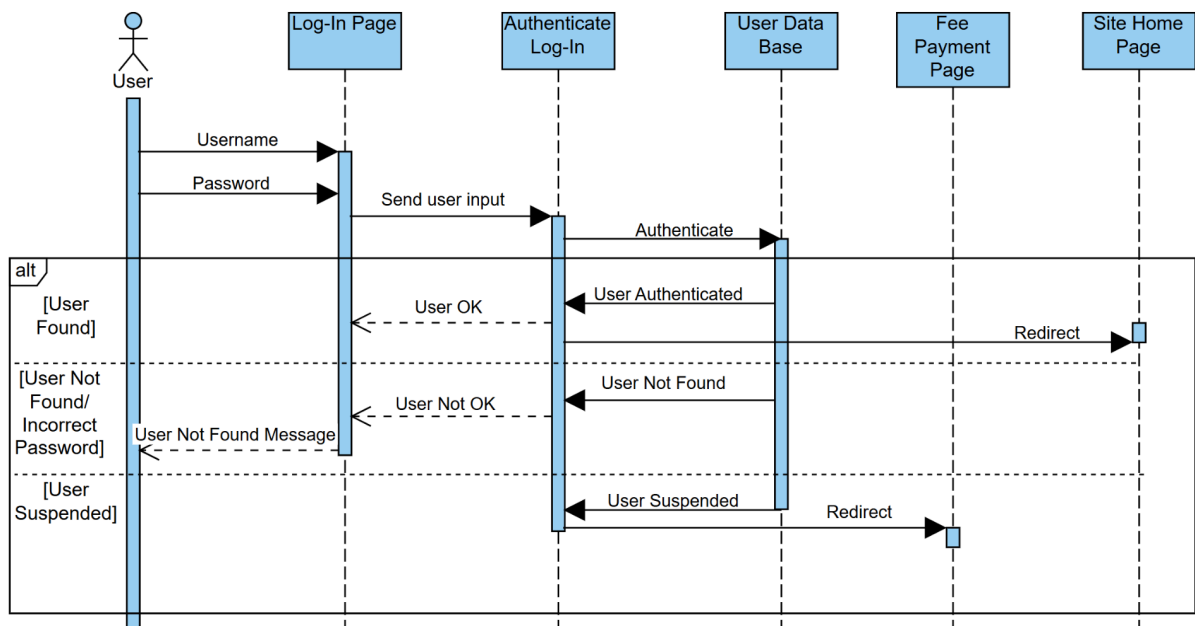
1. Viewing

Anyone who doesn't have an account with the bidding system is labeled a visitor. Visitors have access to the site and can view/browse all listings.



2. Logging In:

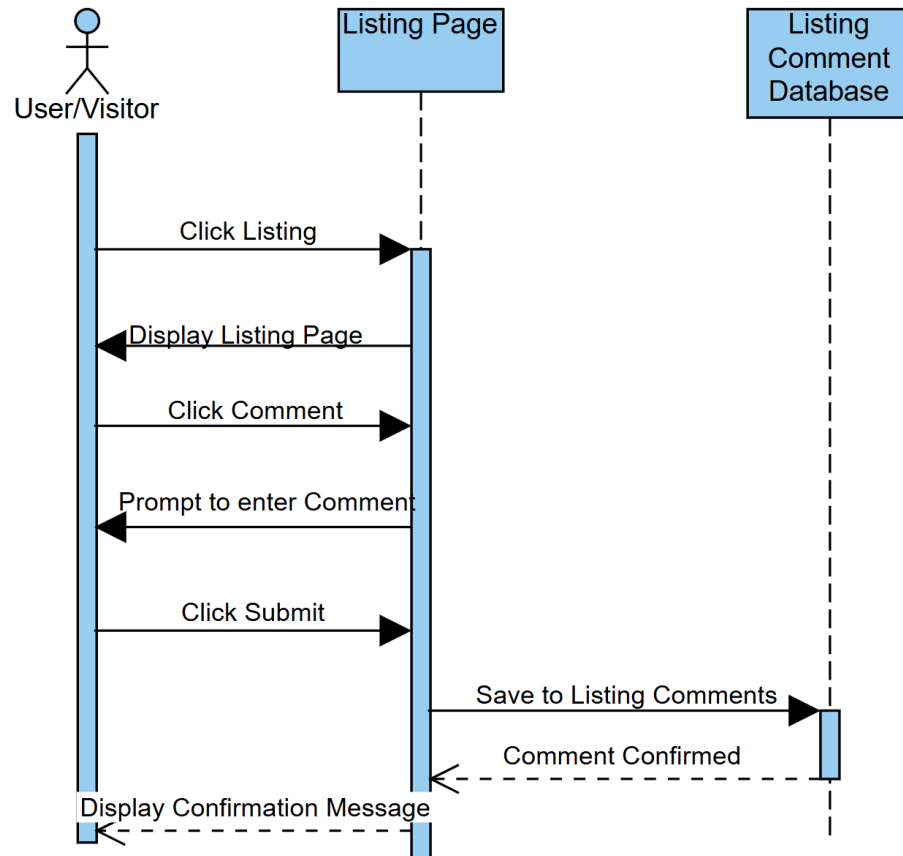
Users need to provide accurate login information (Username and Password) to gain access to their accounts. If a suspended user tries to log in, they'd be prompted to pay the \$50 fine. If the user is not found, a user not found message is displayed.



<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

3. **Commenting:**

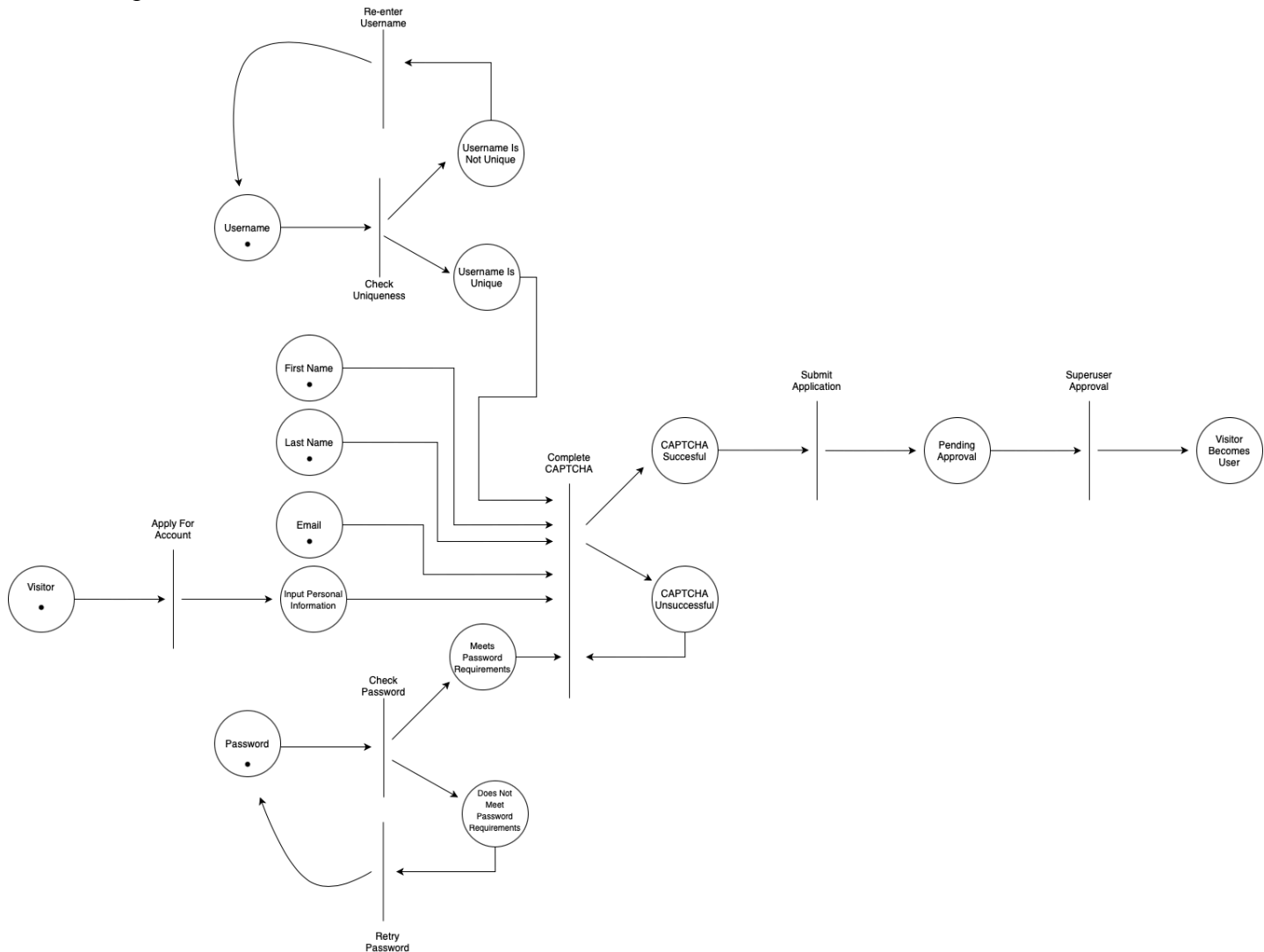
Visitors and Users are allowed to comment on all listings.



<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

4. Signing Up/Application Process:

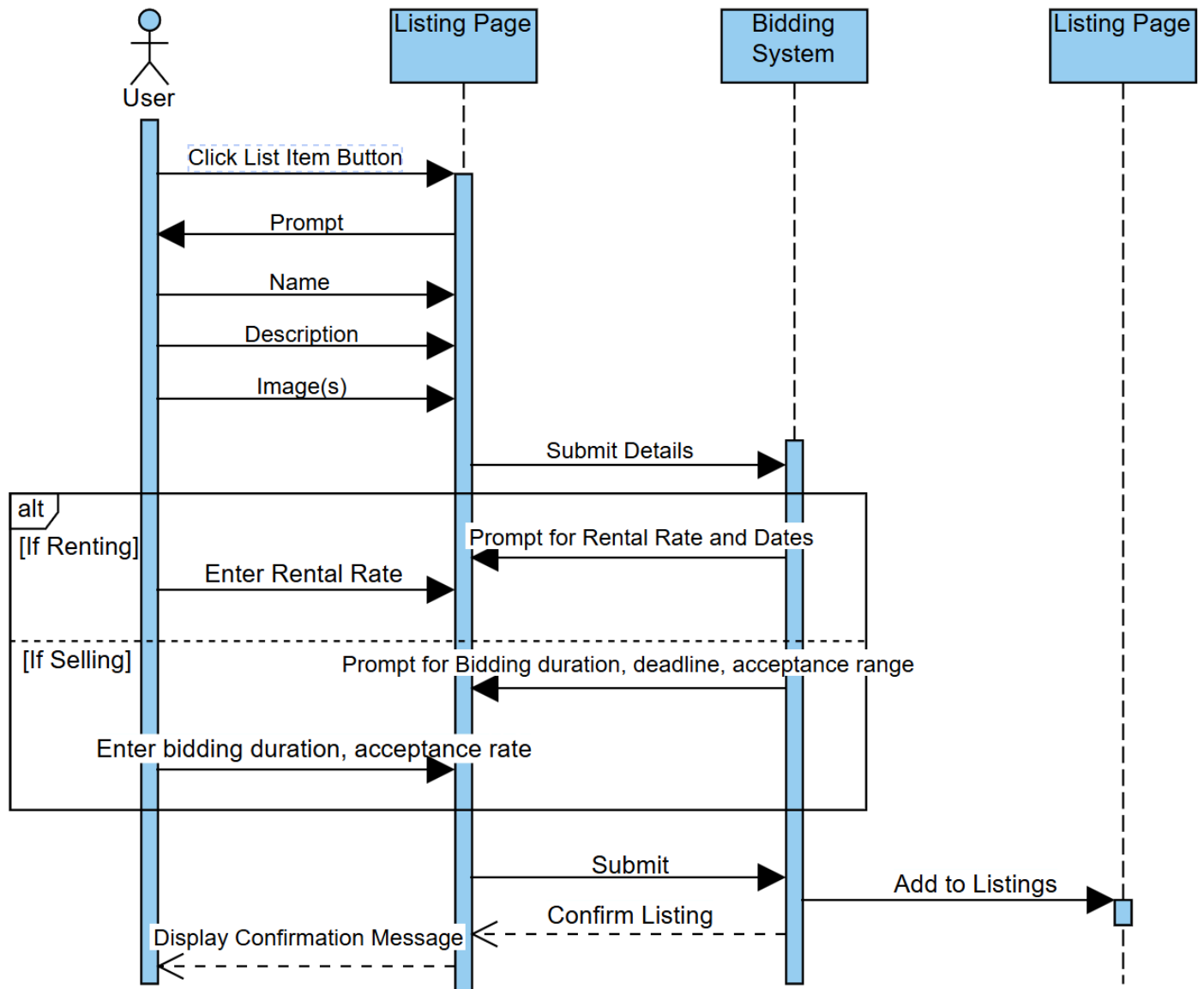
Visitors can apply to become users of the site. They must provide personal information (First/Last Name, Username, Email, and Password). Passwords must have 8 characters, at least 1 uppercase letter, and at least 1 number. When all application requirements are met, the visitor must complete a CAPTCHA. The application is submitted and only when approved by a superuser will the visitor become a User.



<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

5. Listing Items:

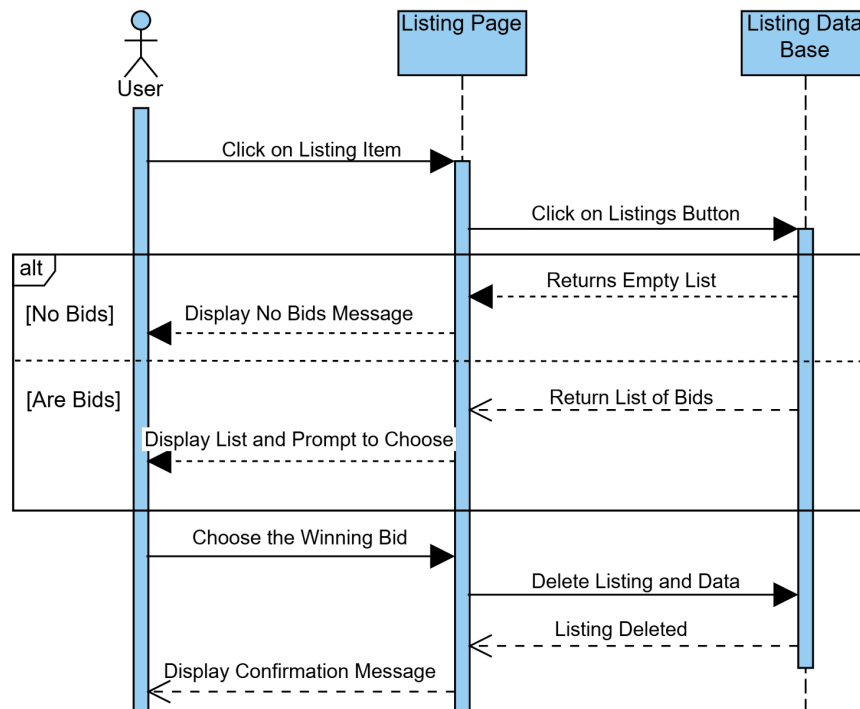
Users are the only ones who can list items. To list, the user has to choose whether the item is up for sale or rent. **For both:** The user has to input the item name, at least one image, the asking price, the acceptable price range, and a description of the item. **For Rent:** Provide renting range and rate. **For Sale:** Provide a bidding deadline.



<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

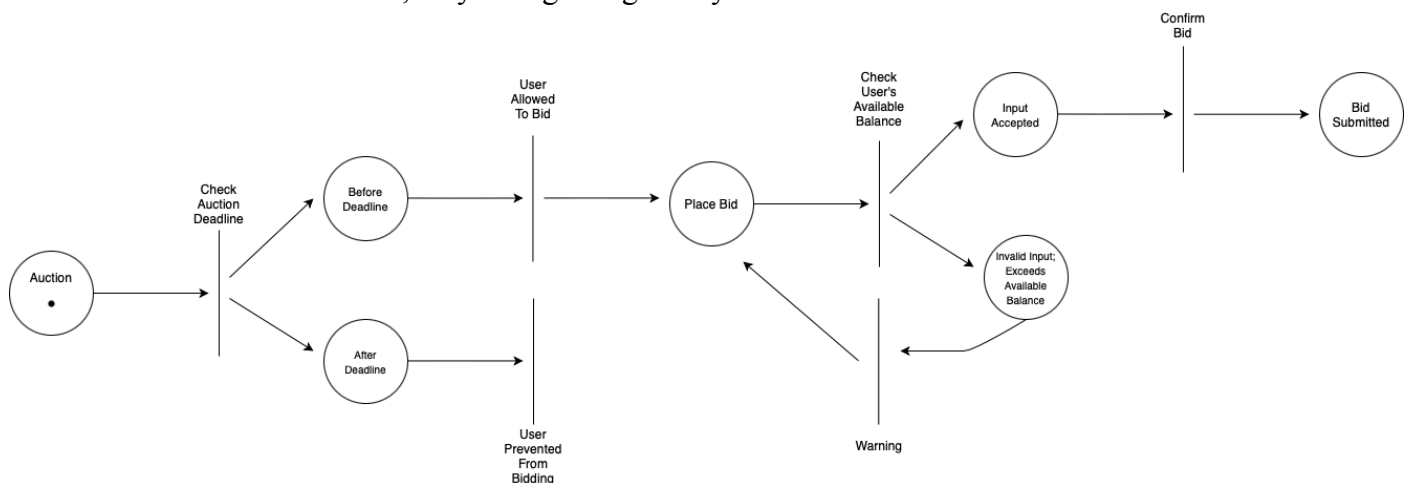
6. Selling/Choosing who to rent/sell to:

The User can choose which bid to accept. If there are no bids by the deadline, there are no bids to choose. Else, the user accepts whichever bid they want. The listing is taken down once the transaction is complete and the money is processed.



7. Bidding:

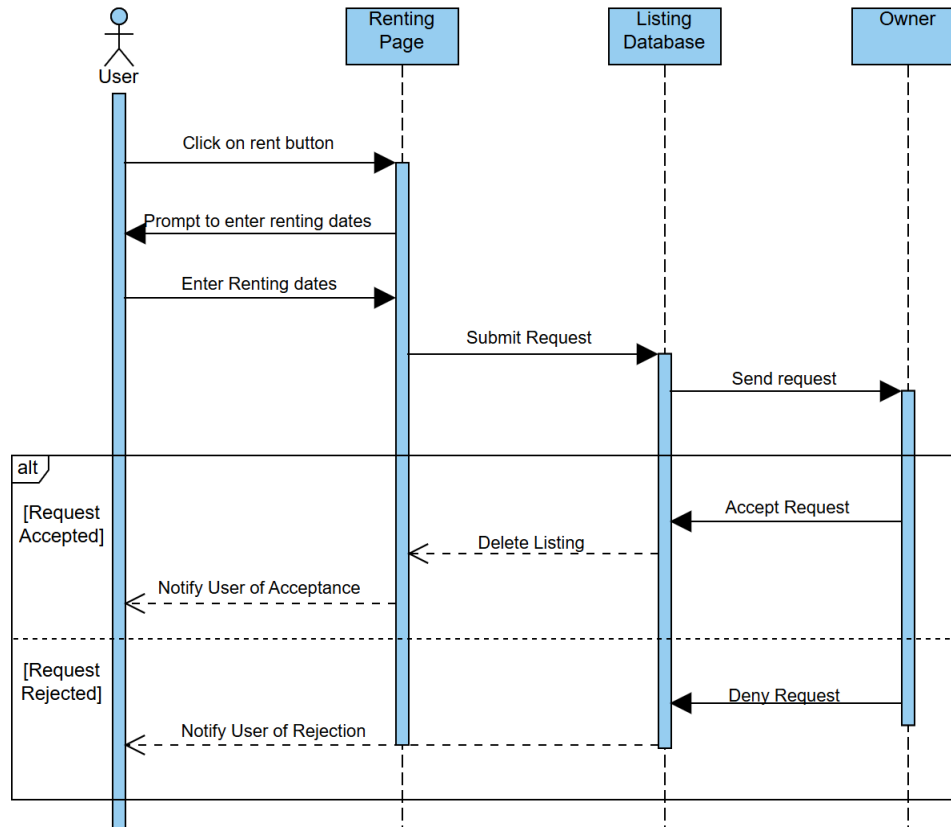
When a user decides to bid on an item, the user must have enough money in the account to bid. The user must enter an amount within the acceptance range. If after the deadline no bids can be taken. If an amount out of that range is entered the screen will output an error message and the user will be prompted to enter an amount again until an acceptable amount is entered. Once a user hits submit, they are agreeing to buy the item.



<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

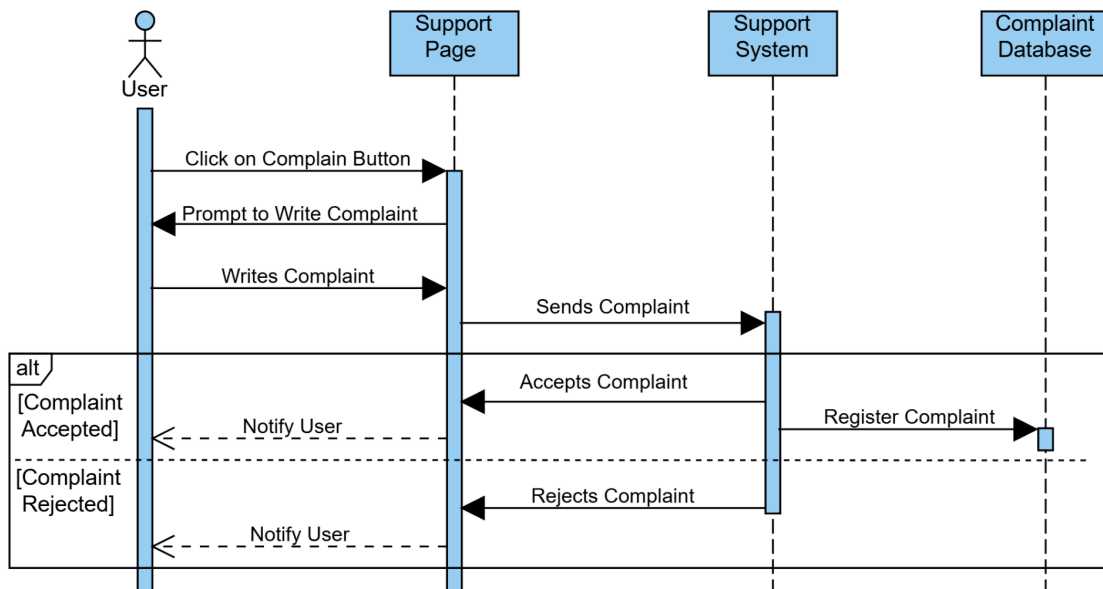
8. Renting:

Renting an item requires a user to specify start and end dates. When 'submit' is pressed, the user must agree and there must be adequate funds in the user's account. Once the user accepts and the money is processed, the listing is taken down.



9. Complaints

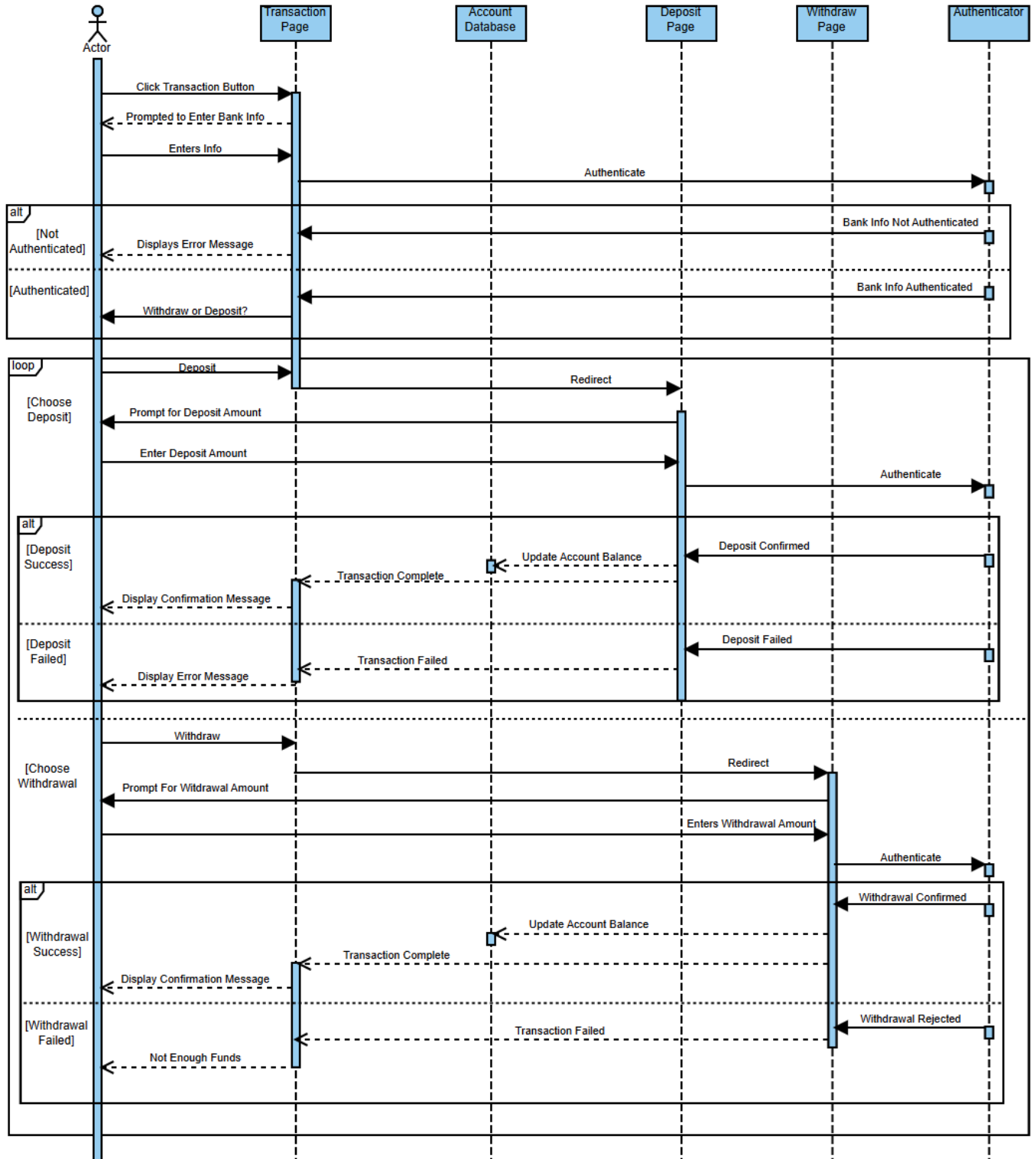
If a user complains about someone they haven't had a transaction with, the complaint will be rejected. The complaint can also be rejected if the superuser deems the complaint dismissable.



<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

10. Depositing/Withdrawing Money

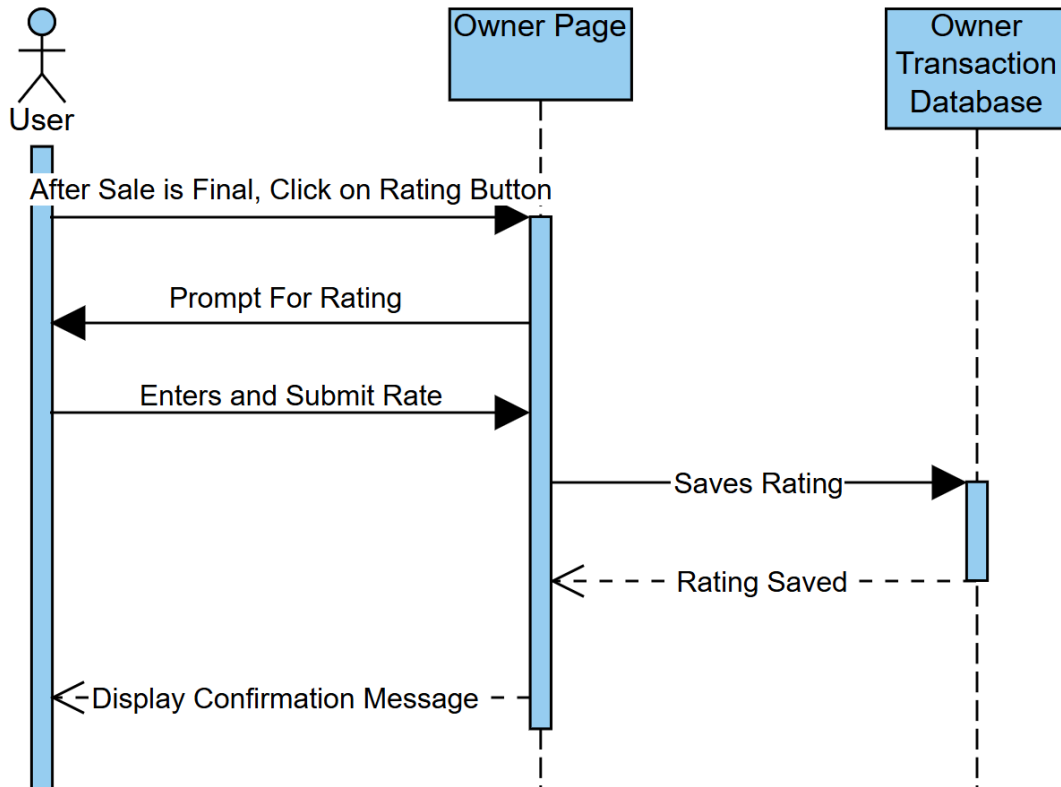
Users can Withdraw/Deposit money into the account. A user must have an authenticated account to transfer money. After the account is confirmed, the user can deposit or withdraw. Accounts must have sufficient funds to Withdraw/Deposit the entered amount.



<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

11. Rating

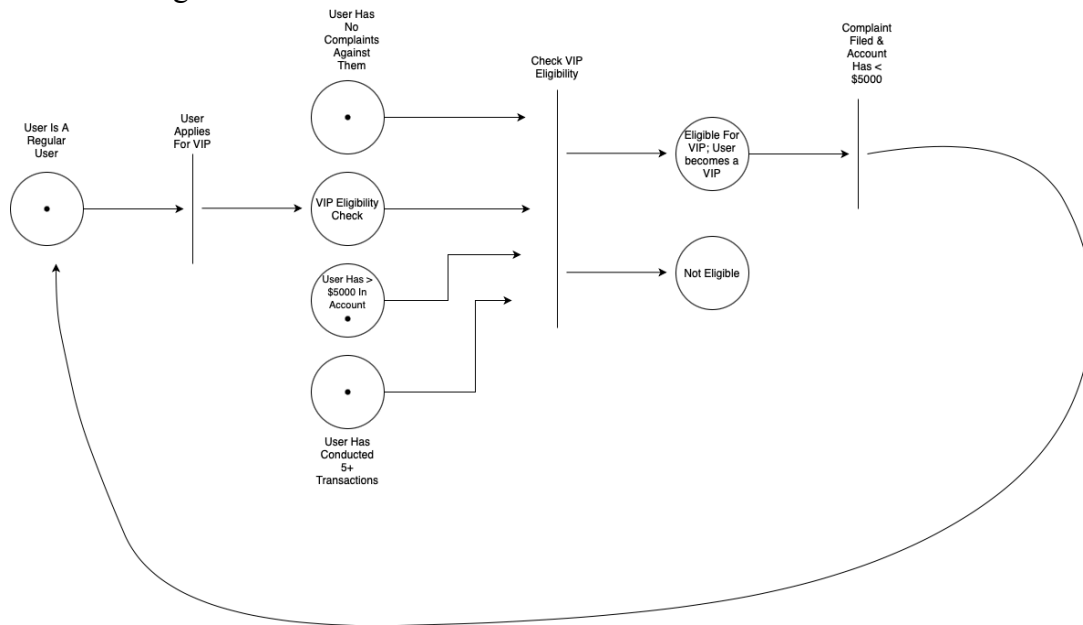
Rating a seller will only be available to users who have bought/rented something from that specific seller. Users can choose a rating from 1-5. Once submitted, the rating will be posted to the seller's rate page anonymously.



<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

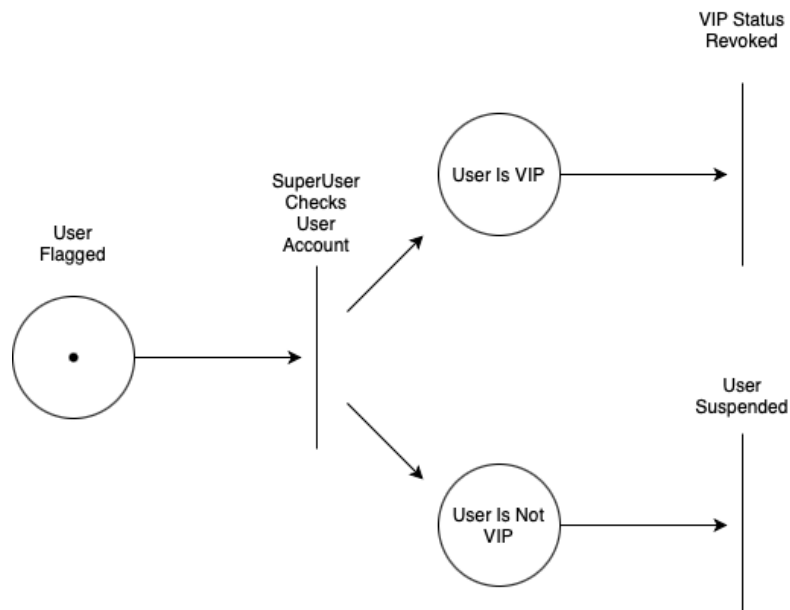
12. VIP

After each transaction and rating, a user is observed to see if they are qualified to become a VIP. A user must have more than \$5000 in the account, conducted 5+ transactions, and have no complaints. When requirements are met, the user becomes a VIP. After becoming a VIP, if a complaint is brought in and the user has less than \$5000 in their account, the VIP user becomes a regular user.



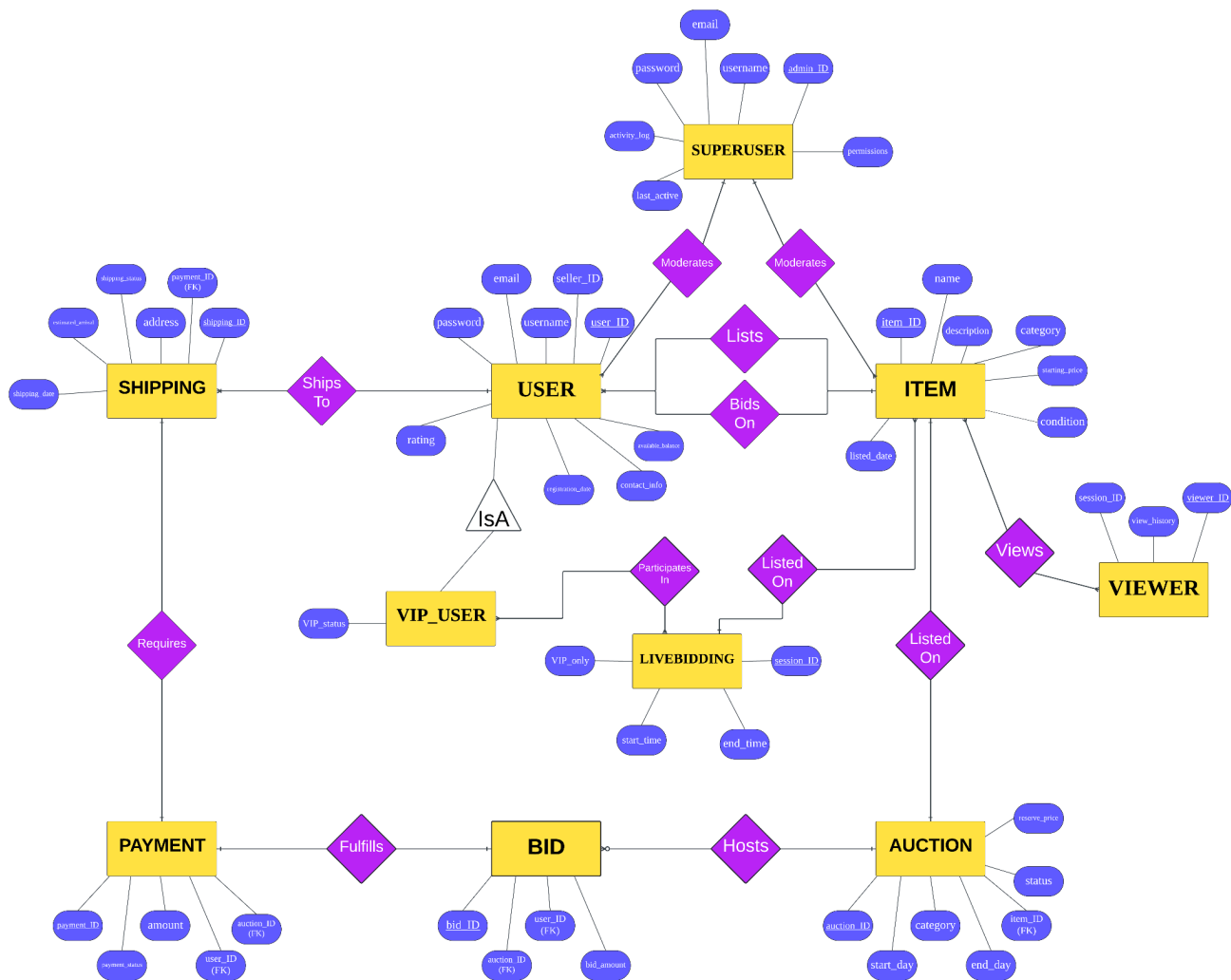
13. Suspension

If a user is suspended, the user's account must be checked first. If the user is a VIP, their VIP status is rescinded and they return to a regular user. Else, the user is suspended.



<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

3. Entity Relationship Diagram



Explanation of Diagram

- Viewer
 - Users who have yet to make an account/aren't logged in
 - Can only view listings! Cannot make or bid on auctions
 - Attributes:
 - viewer_ID (Primary Key)
 - view_history
 - session_ID
 - Cardinality
 - Viewer:Item → N:N
- SuperUser

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

- Functions like an administrator, being able to moderate and take down any inappropriate or fraudulent listings
- Able to moderate and ban users who violate the terms of the website
- Attributes:
 - admin_ID (Primary Key)
 - permissions
 - username
 - email
 - password
 - activity_log
 - last_active
- Cardinality
 - SuperUser:User → 1:N
 - SuperUser:Item → 1:N
- User
 - All users are able to bid on and create auctions, should they choose to do so
 - Users can leave reviews for sellers, rating their customer service, item quality, shipping, etc.
 - Attributes:
 - user_ID (Primary Key)
 - seller_ID
 - username
 - email
 - password
 - registration_date
 - rating
 - contact_info
 - Cardinality
 - User:Shipping → 1:N
 - User:Item → 1/N:1
- VIP User
 - Attributes
 - Subclass of User! Has all attributes User has, with the addition of:
 - VIP_Status (A Boolean statement, which checks whether the user is a VIP)
 - Cardinality
 - VIP User:LiveBidding → N:N
- Live Bidding
 - A live bidding session where only VIPs can auction and bid on items
 - Attributes
 - VIP_only (Boolean statement. Ensures that only VIP users are allowed to participate)
 - start_time
 - end_time
 - session_ID
 - Cardinality
 - LiveBidding:Item → 1:N

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

- Shipping
 - Attributes
 - shipping_ID (Primary Key)
 - payment_ID (Foreign Key, links to Payment)
 - address
 - shipping_date
 - shipping_status
 - estimated_arrival
 - Cardinality
 - Shipping:Payment → 1:1
- Payment
 - Attributes
 - payment_ID (Primary Key)
 - user_ID (Foreign Key, links to User)
 - auction_ID (Foreign Key, links to Auction)
 - payment_status
 - amount
 - Cardinality
 - Payment:Bid → 1:1
- Bid
 - Attributes
 - bid_ID (Primary Key)
 - user_ID (Foreign Key, links to User)
 - auction_ID (Foreign Key, links to Auction)
 - bid_amount
 - Cardinality
 - Bid:Auction → 0/N:1
 - Bid:Payment → 1:1
- Auction
 - Attributes
 - auction_ID (Primary Key)
 - item_ID (Foreign Key, links to Item)
 - category
 - start_day
 - end_day
 - reserve_price
 - status
 - Cardinality
 - Auction:Item → 1:1
- Item
 - Attributes
 - item_ID (Primary Key)
 - name
 - description
 - condition

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

- category
- starting_price
- listed_date

4. Detailed Design

Here is a list of all functions that we plan to implement into our project:

4.1 RegisterUser

```
define RegisterUser(username, password, age):
```

```
    # System gathers username, password, and age of the visitor.
```

```
    # Send information to super-user
```

```
    if BotDetection == True:
```

```
        SendInformation(username, password, age)
```

4.1.1 BotDetection

```
define BotDetection():
```

```
    # System prompts visitor to complete CAPTCHA.
```

```
    # Returns True or False value
```

```
    captcha = GenerateCaptcha()
```

```
    captchaResult = CaptchaAnswer(captcha)
```

```
    return captchaResult
```

4.2 HandleUserRequest

```
    # Super-user may reject or deny user application.
```

```
    # If accepted, the user is added to the Users database and acceptance email is sent.
```

```
    # Otherwise, the user is sent a rejection email.
```

```
define HandleUserRequest(email, username, password, age):
```


<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

if request denied:

RejectRequest()

else:

AcceptRequest()

4.2.1 AcceptRequest

define AcceptRequest(email, username, password, age):

AddToUsersDB(email, username, password, age)

SendAcceptanceEmail()

4.2.2 RejectRequest

define RejectRequest():

SendRejectionEmail()

4.3 WithdrawMoney

System transfers money from the user's bank account to their website account.

define WithdrawMoney(moneyToWithdraw):

userBalance -= moneyToWithdraw

4.4 DepositMoney

System transfers money from the user's account on the website to their bank account.

System first checks if the amount the user wants to transfer is permissible.

define DepositMoney(moneyToDeposit):

if IsSufficientFunds:

userBalance += moneyToDeposit

else:

print ERROR

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

4.4.1 IsSufficientFunds

```
define IsSufficientFunds(moneyToDeposit):
    if moneyToDeposit > userBalance:
        return ERROR
```

4.5 PostListing

```
# Adds listing to Listings database

# Subsequently posts the listing upon its addition to the database.

define PostListing(image, name, description, price):
    AddToListings(image, name, description, price)
```

4.6 Comment

```
# Uploads visitor/user comment into the comment's section of the item.
# If the commenter is not a user, the name displayed is "Anonymous"

# Initialize an in-memory dictionary to store listings and comments

listings = {}

function Comment(listingID, userID, commentText):

    # Check if user is a commenter

    if isCommenterUser(userID):

        # Simulate an API call to retrieve the username from userID

        user = getUsername(userID)

        comment = user.username + ": " + commentText + " - " +
getCurrentDateTime()

    else:

        comment = "Anonymous: " + commentText + " - " + getCurrentDateTime()
```

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

Check if the listing already exists in the dictionary

if listingID not in listings:

listings[listingID] = [] # Initialize a new list for comments if listing is new

Add the comment to the listing's comments list

listings[listingID].append(comment)

return "Comment added successfully."

function displayComment(item_ID):

if listingID in listings:

print("Comments for listing " + listingID + ":")

for comment in listings[listingID]:

print(comment)

else:

print("No comments for this listing.")

function isCommenterUser(userID):

Function to check if userID belongs to a valid commenter

For example, checking if the userID exists

return userID in validUserList

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

```
function getUsername(userID):
    # Function to retrieve the user's username based on userID
    # Simulated with a dictionary lookup or hardcoded data
    return userDatabase[userID]
```

```
function getCurrentDateTime():
    # Function to return the current date and time as a string
    return "2024-11-11 10:45 AM"
```

4.7 Bid

System records the user's bid and shares it to the website.

```
define Bid(priceBid):
    if IsSufficientFunds(priceBid):
        message = "User " + username + " bid " priceBid + " at " + dateAndTime
        DisplayMessage(message)
    else:
        print ERROR
```

4.7.1 IsSufficientFunds

Same idea as 4.4.1

```
define IsSufficientFunds(priceBid):
    if priceBid > userBalance:
        return False
    else:
        return True
```

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

4.8 RemoveListing

Removes specified listing from the Listings database, thereby removing it from the website.

```
define RemoveListing(listing):
```

```
    DeleteFromListings(listing)
```

4.8.1 PastDeadline

Removes listing if the bidding deadline has passed.

```
define PastDeadline(listing):
```

```
    if listing.deadline has passed:
```

```
        RemoveListing(listing)
```

4.9 AcceptBid

Function is invoked when a seller accepts a bid.

```
define AcceptBid(priceBid, listing, buyer, seller):
```

```
    seller.balance += priceBid
```

```
    buyer.balance -= priceBid
```

```
    seller.numOfTransactions += 1
```

```
    MakeVIP(seller)    # See 4.15
```

```
    RemoveListing(listing)
```

4.10 Rate

Function for when a user gives a rating to a user and vice-versa.

```
define Rate(rating, account):
```

```
    account.sumOfRatings = rating
```

```
    account.numOfRatings += 1
```

```
    account.rating = account.sumOfRatings / account.numOfRatings
```

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

CheckSuspendUser(account) # See 4.11

4.11 CheckSuspendUser

Checks if a user should be suspended, which happens when a user's average rating is less than 2 or greater than 4.

When a user is suspended, they are marked as suspended and thus cannot log in successfully into their account.

define SuspendUser(user):

 if CheckRating(user) == True:

 SendSuspensionEmail()

 user.isSuspended = True

4.11.1 CheckRating

Checks the user's rating.

define CheckRating(user):

 if user.rating < 2 or user.rating > 4:

 return True

 else:

 return False

4.12 ReactivateAccount

A suspended user may reactivate their account if a super-user does it or if the suspended user pays \$50.

define ReactiveAccount(user):

 user.isSuspended = False

4.12.1 PayFee

Function is invoked if the user pays their suspension fee.

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

define PayFee(user):

 user.balance -= 50

 ReactivateAccount(user)

4.13 QuitSystem

A user may ask to be removed from the system, which a super-user is able to do.

The user is deleted from the Users database and thus can no longer access their account.

define QuitSystem(user):

 DeleteFromUsersDB(user)

4.14 IssueComplaint

A user may issue a complaint to a super-user concerning an item bought.

define IssueComplaint(message, seller):

 SendMessage(message)

 seller.numOfComplaints += 1

4.15 MakeVIP

A user may become VIP if they meet the following criteria: if they have more than \$5000 in their account, if they have conducted more than five transactions, and if they have never received any complaints.

define MakeVIP(user):

 if user.balance > 5000 and user.numOfTransactions > 5 and user.numOfComplaints ==

 0:

 user.isVIP = True

4.16 NoVIP

A VIP user may lose their VIP status by either meeting the criteria to be suspended or by violating the original criteria for VIPs.

define NoVIP(user):

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

```

if user.isVIP = True:

    if not(user.balance > 5000 and user.numOfTransactions > 5 and
user.numOfComplaints == 0:)

        user.isVIP = False

    else if CheckRating(user):

        user.isVIP = False

```

4.17 PersonalizedGUI

```

define PersonalizedGUI():

    # Initialize variables to store user data

    username = null

    listings = []

    bids = []


    # Function to retrieve user data on component load

    onComponentLoad:

        retrieveUserData()


    # Function to retrieve user data

    # API call to fetch user data

    function retrieveUserData:

        userData = fetchUserDataFromAPI()


    # Set user data to local variables

```


<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

username = userData.username

listings = userData.listings

bids = userData.bids

Display UI elements

Display welcome message with username

display:

display "Welcome, " + (username or "Guest")

Section: User Listings

display "Your Listings"

if listings is not empty:

for each listing in listings:

display ListingCard with listing information

else:

display "No listings available."

Section: User Bids

display "Your Bids"

if bids is not empty:

for each bid in bids:

display BidCard with bid information

else:

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

display "No bids placed yet."

Component: ListingCard

function ListingCard:

input: listing data

display:

display listing.title

display listing.description

Component: BidCard

function BidCard:

input: bid data

display:

display "Bid Amount: " + bid.amount

display "Item: " + bid.item

4.18 LiveBidding

Function defines our live bidding functionality, where VIPs can participate in a “King of the Hill” style auction. Bidders have 10 seconds to outbid the highest bid, and if no new offer is made within that time frame, the item is given to the highest bidder.

define LiveBidding():

highestBid = 0

kingOfTheHill = None

timer = 10 # 10-second timer for bids

activeBidding = True

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

```

while activeBidding:
    if kingOfTheHill is not None:
        countdownTimer(timer)

    if timer == 0:
        announceWinner(kingOfTheHill)
        activeBidding = False
    else:
        wait for newBid()

    else:
        wait for firstBid()

```

```

function countdownTimer(seconds):
    while seconds > 0:
        sleep(1) # Sleep for 1 second
        seconds -= 1

```

```

function newBid():
    bid = getBidAmount()

    if bid > highestBid:

```

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

highestBid = bid

kingOfTheHill = getBidder()

resetTimer(10) # Reset timer to 10 seconds

function resetTimer(seconds):

timer = seconds

displayCountdown(timer)

function getBidAmount():

Retrieve bid amount from current user

return userInput

function getBidder():

Retrieve current user as the bidder

return currentUser

function announceWinner(bidder):

display("The winner is " + bidder + " with a bid of \$" + highestBid)

function displayCountdown(seconds):

display("Time remaining: " + seconds + " seconds")

function wait for firstBid():

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

```
display("Waiting for first bid...")
```

```
while kingOfTheHill is None:
```

```
    if getBidAmount() > highestBid:
```

```
        newBid()
```

```
function wait for newBid():
```

```
    display("Waiting for new bid... Time left: " + timer + " seconds")
```

```
    sleep(1)
```

5. System Screens

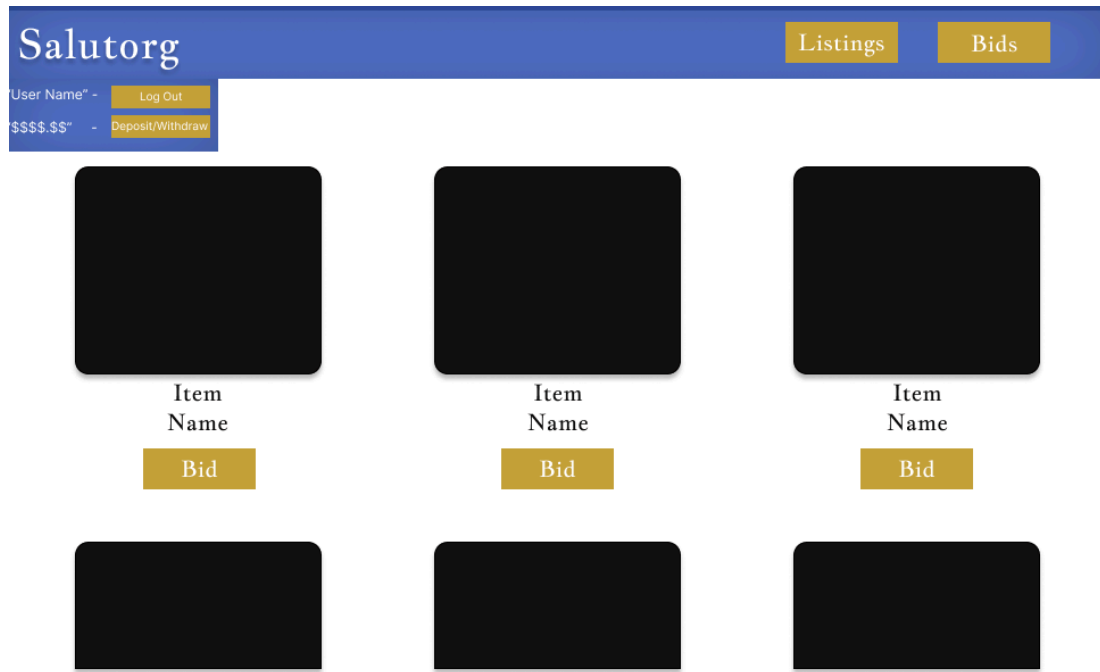
Visitor Home



The home page for visitors of the site, buttons to either login or register as a user, can also view listings.

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

User Home



The home page for users, shows their name and balance. Buttons to view their listings or bids and to log out or change their balance.

Item Listing

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

Salutorg

Sign In
Register

“Item Name”

“Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.”

Posted By: “User”

Deadline: “mm/dd/yy”

Bid

Add a comment...

Post

“User Name”

“Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud

“Visitor Name - (Anonymous)”

“Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud

This shows how an item would be listed, with its name, description, deadline to bid, etc. Also shows the comments posted by users and visitors below it.

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

User Bids

Salutorg

Listings

Bids

My Bids

Item Name

Current Price: "\$50.00"

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Top Bid: Yes

Deadline: "mm/dd/yy"

Remove Bid

Item Name

Current Price: "\$75.50"

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Top Bid: N/A

Deadline: "mm/dd/yy"

This shows the page that holds the information on bids the user has made. Includes if their currently the top bid, what the deadline is, and gives them the option to remove their bid.

User Listings

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

Salutorg
Listings
Bids

My Listings

Create Listing

Item Name

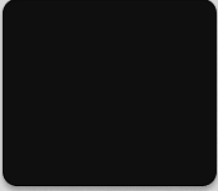

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Edit Information

Deadline: "mm/dd/yy"

Remove Listing

Current Price: "\$50.00"

Item Name


"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

Deadline: "mm/dd/yy"

Current Price: "\$75.50"

This shows listings made by the user. Allows the user to edit information like the price or deadline or just remove it entirely.

Super Admin Account Approvals

Salutorg
Accounts
Suspensions

"Super User Name" - Log Out

Account Approvals

"User Email"
Registration Date: mm/dd/yy

Approve
Reject

"User Email"
Registration Date: mm/dd/yy

Approve
Reject

"User Email"
Registration Date: mm/dd/yy

Approve
Reject

The page for super users to accept or reject new registrations. Shows the email someone registered with and the date they registered.

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

Super User Suspension Appeals

Salutorg
Accounts
Suspensions

Suspension Appeals

“User Name” Suspension Date: mm/dd/yy <div>Revoke Suspension</div> <div>Keep Suspension</div>	Reason For Suspension: <p>“Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.”</p>
---	---

This page shows the users who have been suspended, the reason for their suspension, and allows the super user to revoke or keep their suspension.

Register

Salutorg
Sign In
Register

Email

User Name

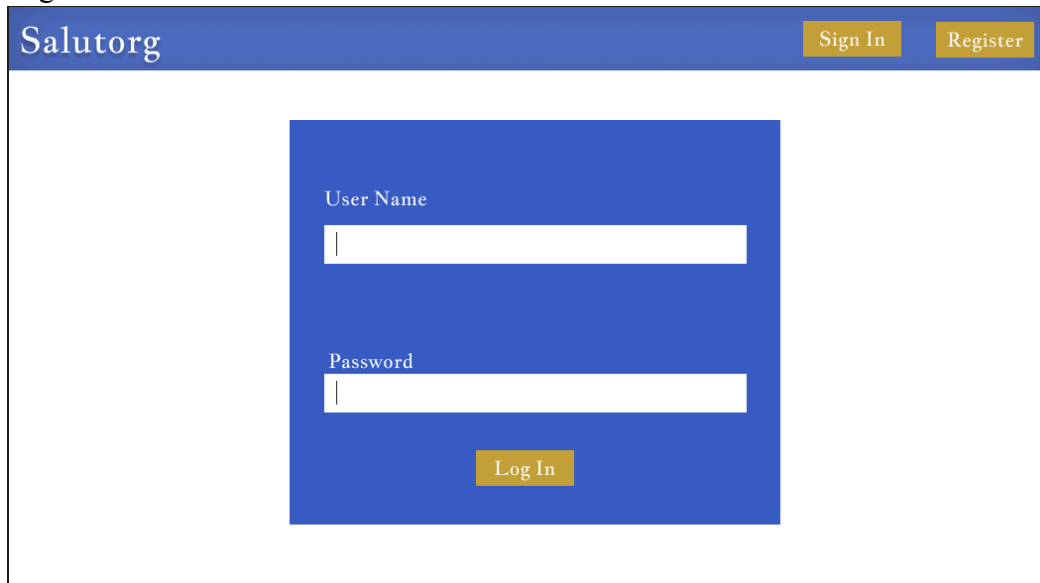
Password

Register

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

This is the register page with an input field for an email, username, and password.

Log In



The image shows a login page for 'Salutorg'. At the top, there is a blue header bar with the 'Salutorg' logo on the left and two yellow buttons labeled 'Sign In' and 'Register' on the right. Below the header, the main content area is white. In the center, there is a blue rectangular box containing the login form. Inside this box, there are two white input fields: the top one is labeled 'User Name' and the bottom one is labeled 'Password'. Below these fields is a yellow button labeled 'Log In'.

This is the login page whose only difference compared to register is that you don't need to input an email.

Prototype - (CreateListing)



The image shows a 'Create Listing' page for 'Salutorg'. It features a blue header bar with the 'Salutorg' logo on the left and two yellow buttons labeled 'Listings' and 'Bids' on the right. Below the header, the main content area is white. In the center, there is a yellow button labeled 'Create Listing'.

User is able to create a listing by pressing the "Create Listing" button on the "Listings page"

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

Salutorg

ListingsBids

Create Listing

Item Name

Nike Shoes

Item Price

\$99.99

Deadline

11/12/2024

Description

Very good condition

Upload Image



Post

Form appears for User to fill out with relevant information.

Salutorg

ListingsBids

My Listings

Create Listing

Nike Shoes



"Very Good"

Edit Information

Deadline: "11/12/2024"

Remove Listing

Current Price: "\$99.99"

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

Salutorg

ListingsBids

User Name" -

Log Out

"\$\$\$\$.\$\$" -

Deposit/Withdraw



Nike Shoes

Bid

Item Name

Bid

Item Name

Bid

Listing gets added to “Listings” page of user with information they added and to “Home” page

<Project Name: Salutorg.com>	Version: <1.0>
Software Requirements Specification	Date: <17/10/24>
Design Report	Date: <12/11/24>

6. Group Meeting Memos

Oct. 26, 2024

1. Created a github repository

Oct. 27, 2024

1. Created website homepages using Figma, showcasing the potential overall design of the website

Nov. 8, 2024

1. Further discussed the overall design and functionalities of the website
2. Gave clear responsibilities on who does what on the website
3. Further discussed the details of the report

Nov. 11, 2024

1. Completed section 5 of the report concerning major GUI screens and provided a prototype example of one of the functionalities. Completed the collaboration class diagram for section 1 using functions found in section 4.
2. Redesigned the ER diagram with more entities and attributes that better represent the project requirements
3. Further discussed the overall design and functionalities of the website
4. Began coding various functionalities, such as the overall UI and how a visitor can register to become a user

7. Github Address

https://github.com/israelpina004/Salutorg_SEFP.git