
**Team C (Johnson Chen, Dino Radoncic, Anova Sarkar,
Daphne Tang, and Konrad Zielinski)**

**TweakMart
Software Requirements Specification For
An Electronic Market System**

Version <2.0>

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Revision History

Date	Version	Description	Author
<27/10/2022>	<1.0>	<Written Purpose, Scope, Reference, Overview>	<Johnson Chen>
<28/10/2022>	<1.0>	<Written Definitions, Acronyms, and Abbreviations>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<29/10/2022>	<1.0>	<Written Use-Case Model Survey, Product Perspective, Product Functions, User Characteristics, Constraints>	<Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<30/10/2022>	<1.0>	<Written Assumptions and Dependencies>	<Johnson Chen>
<30/10/2022>	<1.0>	<Updated Use-Case Model Survey, Product Perspective, Product Functions, User Characteristics, Constraints>	<Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<31/10/2022>	<1.0>	<Written Use-Case Reports and Supporting Information>	<Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<31/10/2022>	<1.0>	<Written Supplementary Requirements>	<Johnson Chen>
<01/11/2022>	<1.0>	<Update Definitions, Acronyms, and Abbreviations>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<01/11/2022>	<1.0>	<Final Revision>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

<10/11/2022> - <10/15/2022>	<2.0>	<Updated the Report, Added Diagrams and Pseudo Codes>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<10/16/2022> - <10/24/2022>	<2.0>	<Edited GUI and Functionality Prototypes and Final Revision>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>

Table of Contents

1. Introduction	4
1.1. Purpose	4
1.2. Scope	4
1.3. Definitions, Acronyms, and Abbreviations	5
1.4. Reference	6
1.5. Overview	6
2. Overall Description	6
2.1. Use-Case Model Survey	6
2.1.1. Product Perspective	6
2.1.2. Product Functions	6
2.1.3. User Characteristics	9
2.1.4. Constraints	10
2.2. Assumptions and Dependencies	10
3. Specific Requirements	10
3.1. Use-Case Reports	10
3.2. Supplementary Requirements	11
4. Supporting Information	12
4.1. Diagrams	12
4.2. Pseudo-code	20
4.3. GUI and System Prototypes	26
4.4. Group meetings Memo	32
4.5. Github Repo Address	33

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Software Requirements Specification

1. Introduction

1.1. Purpose

- The purpose of this software requirements specification is to describe the scope, to provide a complete and comprehensive description of TweakMart, fully describe its external behavior, along with the nonfunctional functional requirements and design constraints. Additionally, it is designed and written for three types of users to this system, super users (SUs), ordinary users (OUs), and guests (GUs), each having their own respective tasks.

1.2. Scope

- The TweakMart is an implementation of an electronic trading system based on ebay which is to be used by the three types of users. The scope of the components of TweakMart is to simulate an electronic trading between users by maintaining the market system and enforcing the rules of the system. Each user of this system has their own respective ability to perform certain tasks. As the hierarchy goes, the SUs have more power over the OUs and GUs since they are the administrator of the system, basically having access to almost everything. The OUs unlock more features than the GUs because they are the registered members of the system such as changing their personal information, the ability to sell, bid, rate, do transactions, file complaints, etc. The GUs have limited features than the other users such as browsing, file reports to SUs, and apply to be OUs. In other words, the OUs and GUs don't have some of the power that SUs have to perform certain tasks and the GUs don't have some of the power that OUs have to perform certain tasks. Each user has their own restrictions and constraints as well. We are aiming to have a low cost of designing and maintaining the TweakMart and complete the basic functionalities by the end of the fall semester. We also want to make sure there are as no few issues present as possible and a high percentage of uptime in order for users to interact smoothly and enjoy their time here. Further, the use-case model it is associated with is the UML diagram that describes the functionalities of the system and the interactions between the users and the system. Refer to section 4.1 for the collaboration class diagram for the whole system.

1.3. Definitions, Acronyms, and Abbreviations

Acronym	Terms	Definition
TM	TweakMart	The online shopping site that GUs,

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

		OUs, and SUs will be using.
GU	Guest User	A user, with no account on the website, capable of viewing items and adding items to a shopping cart. Users are prompted to register when checkout is attempted.
OU	Ordinary User	A user, with an account registered on the website, capable of viewing items, and making purchases with a shopping cart and checkout. Registered users are also capable of listing items they want to sell and submitting reviews of products.
SU	Super User	An administrator user, with an account registered on the website, capable of processing applications and items for the OU, settle disputes between sellers and buyers, contact the OU about complaints, and collect transaction statistics on OUs.
GUI	Graphical User Interface	A method of interacting with the computer through visual means, like windows, icons, menus, etc.
HTML	Hypertext Markup Language	A markup language used for website creation and design (formats text and links on a webpage).
CSS	Cascading Style Sheets	Style sheet programming language which is used for web design; It's purpose is to add graphical components to an existing HTML file.
JS	JavaScript	Scripting language used for developing an interactive experience on webpages
PY	Python	General use object-oriented and high-level programming language.
SQL	Structured Query Language	Programming language used to communicate with databases.
CPU	Central Processing Unit	Also known as a microprocessor or processor. It is a vital piece of hardware for digital computing systems.

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

1.4. Reference

- spec_sample.pdf, located in the course website provided by Jie Wei
- proj_spec.docx, located in the course website provided by Jie Wei

1.5. Overview

- The second section of this document, overall description, will describe the general factors that affect the product and its requirements such as the use-case model which demonstrates the interactions between the system and the users of TweakMart and the assumptions and dependencies which contains any key technical feasibility, subsystem or component availability. The third section provides background information, use-case models, subsystems and features for these functional and software requirements which are defined in detail in order to make them easier to understand. The last section contains supporting information that makes the software requirements specification easier to use such as the table of content, index or appendix.

2. Overall Description

2.1. Use-Case Model Survey

2.1.1. Product Perspective

- TM is an eCommerce website aimed at vendors who want to reach out to potential customers through a user-friendly and easy-to-learn site. The project aims to provide a platform where moderation is linked to multiple steps of the selling/buying process, in order to provide a smooth experience for both customers and vendors. TM is intended to be stand-alone, thus it should not depend on any application or other form of software in order to function.

2.1.2. Product Functions

- Refer to the UML Diagram in section 4 for Actors and Use Cases

Actor Types:

Guest User (GU):

System distribution for the use cases for the Guest User can be found in Section 4.1

- Can search for items listed with the capability of browsing with certain parameters, such as price range, highest ratings, key words, etc.

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

- Can submit a report for suspicious item listings.
- Can apply for an account to become an Ordinary User and get access to other capabilities.

Ordinary User (OU):

System distribution for the use cases for the Ordinary User can be found in Section 4.1

- Can deposit into and withdraw funds from an account. Total balance can be used to buy items from sellers or place a bid.
- Can look at a record of previous transactions made from an account.
- Can submit a request to post an item up for sale or auction and provide information on said item. Sellers can choose which users they will sell to, but a reason must be given if not sold to the highest bidder.
- Can give a rating for a listed item after purchasing it.
- Can file a complaint against a vendor they have bought from.

Super User (SU):

System distribution for the use cases for the Super User can be found in Section 4.1

- Can review incoming account applications sent by guest users.
- Can review requests to post items up for sale or auction sent by ordinary users.
- Can collect and review the transaction history of any ordinary user's account, for statistical purposes.
- Can review complaints about sellers.
- Can give a warning or remove ordinary users if there is a complaint or several bad ratings on the user's listings.
- Settles disputes between sellers and buyers and gives appropriate functions, either warning or potential removal.

Use-Cases:

- **Browse/Search Items**
 - OUs and GUs have the option of filtering searches using keywords, price range, and ratings.
 - Using a search bar, users can browse and search for listed items.
- **Deposit/Withdraw Money**
 - Money can be withdrawn or deposited by OUs, from own accounts.
 - Designated page for OU accounts will allow for account balance to be manipulated through either withdrawals or deposits.
- **Submit Bid to Buy Items**
 - Any bids submitted by OUs can be considered by the vendor of the item, for whether or not the vendor decides the transaction should proceed.
 - OUs will be able to navigate to a page designated for the sale of the item they are interested in, where an option to place a bid is available. OUs

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

can input specific bid through a simple GUI, which cannot exceed the OUs current account balance

- **Submit Information of Item Being Sold**
 - OUs should include pictures or video of their item, as well as the title, keywords, time limit, and price range.
 - Requests to post items for sale can be submitted once the item information is filled out.
 - OUs will have access to an option connected to their account that brings them to a page where they can fill in the details of an item they want to submit for sale, and then request for the item to be approved for posting.
- **Sell the Item Posted by User**
 - OUs can choose who to sell the item to.
 - Reason must be given if not sold to the highest bidder.
 - Once the bidding process is over, vendor OU will have an option to sell the item, selling it by default to the highest bidder, unless the vendor specifically requests the item be sold to a different bidder.
- **File Complaints Against Vendors**
 - If there are problems with the item purchased, OUs can file complaints against the vendor.
 - SU reviews these complaints.
 - If a vendor receives more than 2 complaints, the vendor will receive a warning from SUs.
 - Upon a selling of an item, OUs will have an option to file a complaint about the seller to the SUs, if there are any issues with the sold item. The SUs will then deal with the situation and issue a warning if need be.
- **Grade Vendor After Buying Item From Vendor**
 - OUs rate the vendor based on purchase experience out of 5 (1 is the worst, 5 is the best).
 - Vendors with an average grade of 2 or less from at least 3 evaluations will receive a warning from SUs.
 - Once the item is confirmed to be shipped, OUs will be able to leave a review of the vendor on the vendor's profile page.
- **Check Transaction History**
 - OUs can check the transaction history of their account (purchased items and relating details).
 - SUs can access OUs' transaction history and obtain transactional data of OUs.
 - Option to access transaction history should be listed under the OUs account options.
- **Report Suspicious Items**
 - OUs and GUs can report items they deem suspicious.
 - SUs reviews and processes these reports.
 - Once an item is placed to be sold publicly, there will be an option to flag the item as suspicious.
- **Account Registration**
 - GU can apply to become an OU.

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

- SU will accept or deny applications from GU.
- GUs that access the website will have an option to register for an account through a GUI button on the homepage, and can also be prompted to register if they browse an item and attempt to make a purchase.
- **Settle Disputes**
 - If there are disputes between sellers and buyers, SUs deal with them.
 - SUs will apply warnings or removals if necessary.
 - In the case of disputes, SU will communicate with both seller and buyer and come to a conclusion about the situation. SU will then apply warning or removals if deemed necessary.
- **Collect Information on Transactions**
 - SUs obtains transactional data over a certain period of time and/or a certain OU(s).
 - OUs' submissions of information on the item being sold.
 - OUs' submissions of biddings on the items.
 - OUs' choice to sell items to a certain OU.
 - SUs will have access to an option to view a OUs transaction history when viewing the OUs account details.
- **Review Requests to Post Items for Sale**
 - SUs review requests to post an item for sale
 - If SU rejects the request, OUs are notified of this
 - If SU removes an item, it is blacklisted
- **Send warnings to OU**
 - If there are more than 2 complaints
 - If the average grade of vendor has an average of 2 or less (from at least 3 evaluations)
- **Remove OU**
 - If there are already warnings sent to the OU
 - If they have more than 2 warnings
 - If removed, OU is prohibited from reapplying
- **Application Review**
 - SU will accept or deny account applications from GUs
 - If accepted, the GU will become an OU

2.1.3. User Characteristics

- Users should be familiar with navigating through web pages.
- Users should also be familiar with terms used in online shopping sites, such as "checkout", "registration", "shopping cart", etc.
- Users are heavily advised to be familiar with terms of service before creating an account.
- Users require stable internet connection.
- Users can read English.

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

2.1.4. Constraints

- No support for languages other than English.
- No accessibility options.
- No theme customization.
- No refunds after a week.
- Reviewing complaints and reports is not automated.

2.2. Assumptions and Dependencies

- In order to run a proper web application, we must consider the hosting service, the front-end, the back-end, and the database. The layout, structure, and presentation component in the front-end will be created using HTML and CSS. The program operation behind most of the functionalities component of the back-end will be created using JavaScript and Python. The database we will be using is SQL. The SQL database should be correctly connected to our web application, where the correct data is stored in the tables setup with no other problem occurring.
- We are assuming that all users know how to use their computer, are familiar with internet browsing and have access to the internet. We are only considering people that have an average level of English literacy skill since our interfaces will consist only of English.
- We care about our user experiences. The web application should be user friendly, with as no or few issues present as possible. The users will be able to navigate to their destination shortly and smoothly as well. The web application should have the basic functionalities for the users' needs. The web application should be able to work across most browsers, especially popular used ones and windows/linux/macOS.

3. Specific Requirements

3.1. Use-Case Reports

Function	Actors/Description
View Item Listings	GUs and OUs can browse items
Sign Up	GUs can apply for an account
Login	OUs and SUs can log in
Sell Items	OUs can sell items
Buy Items	OUs can buy items

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Function	Actors/Description
Report Items	GUs and OUs can report suspicious items
View Transaction History	OUs can view own transaction history
Deposit/Withdraw Money	OUs can deposit or withdraw from account
File a Complaint	OUs can file a complaint against other OUs
Rate Sellers	OUs can rate other OUs
Update Account Details	OUs can update account details
Collect Information	SUs can collect transactional data
Settle Disputes	SUs can settle disputes between sellers and buyers
Review Requests to Sell	SUs can review requests to sell
Warn OUs	SUs can warn OUs
Remove OUs	SUs can remove OUs
Review Applications	SUs can accept/deny applications from GUs
Remove Items	SUs can remove items (blacklist)

3.2. Supplementary Requirements

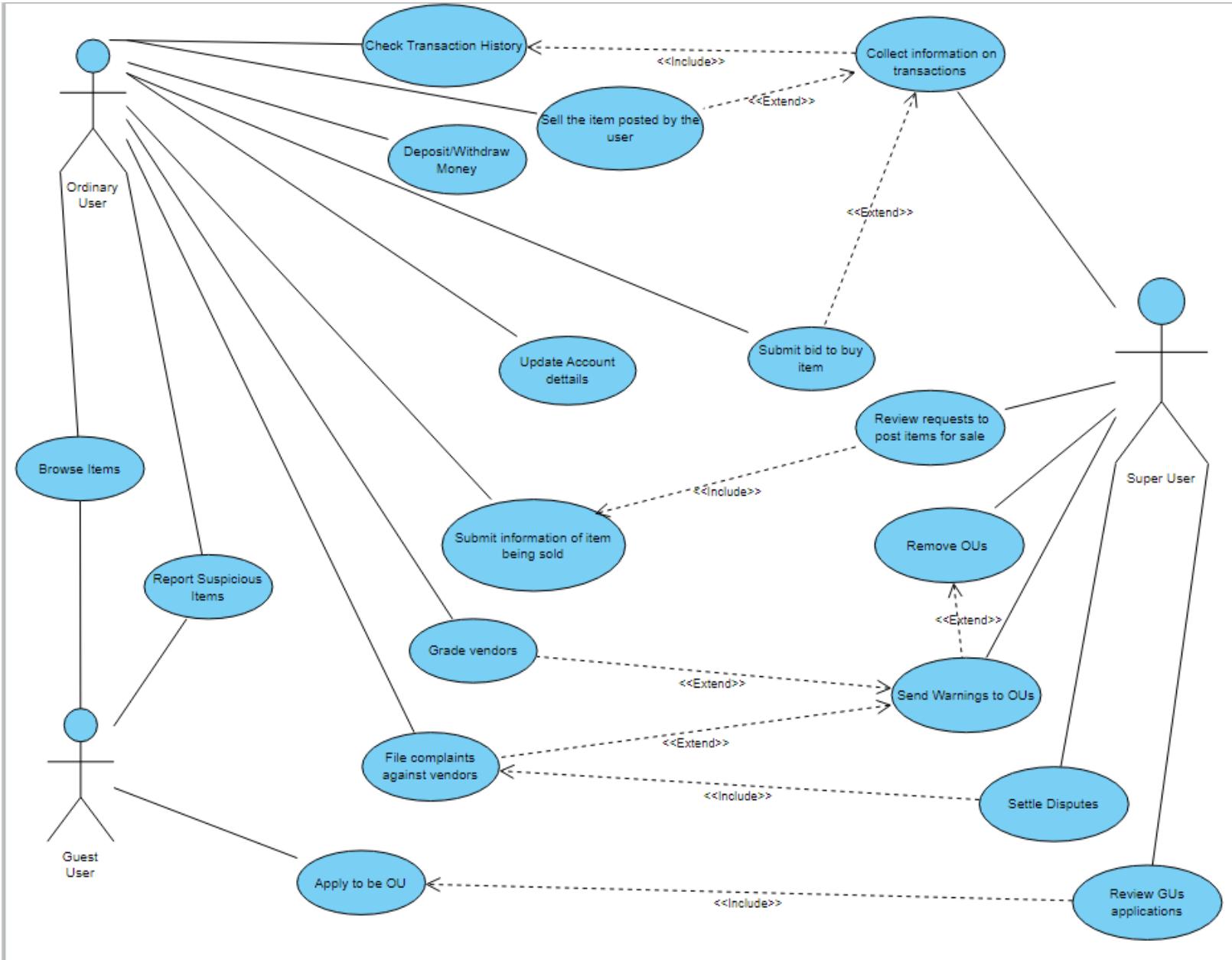
- There are requirements that are needed to design, host, and run this web application. Some of the requirements include having a recommended 64-bit processor/CPU since it can handle more data at once, and have one of the following operating systems like Windows 7/10/11, the recommended MacOS Sierra or higher (>10.12), or the popular Linux latest Ubuntu version or similar recommended LinuxOS version types. Programming languages used in the web application such as Python should be at version 3.10.4 or higher, HTML should be at version HTML5, CSS should be at version CSS3, and SQL should be at version 15 (SQL Server 2019).

4. Supporting Information

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

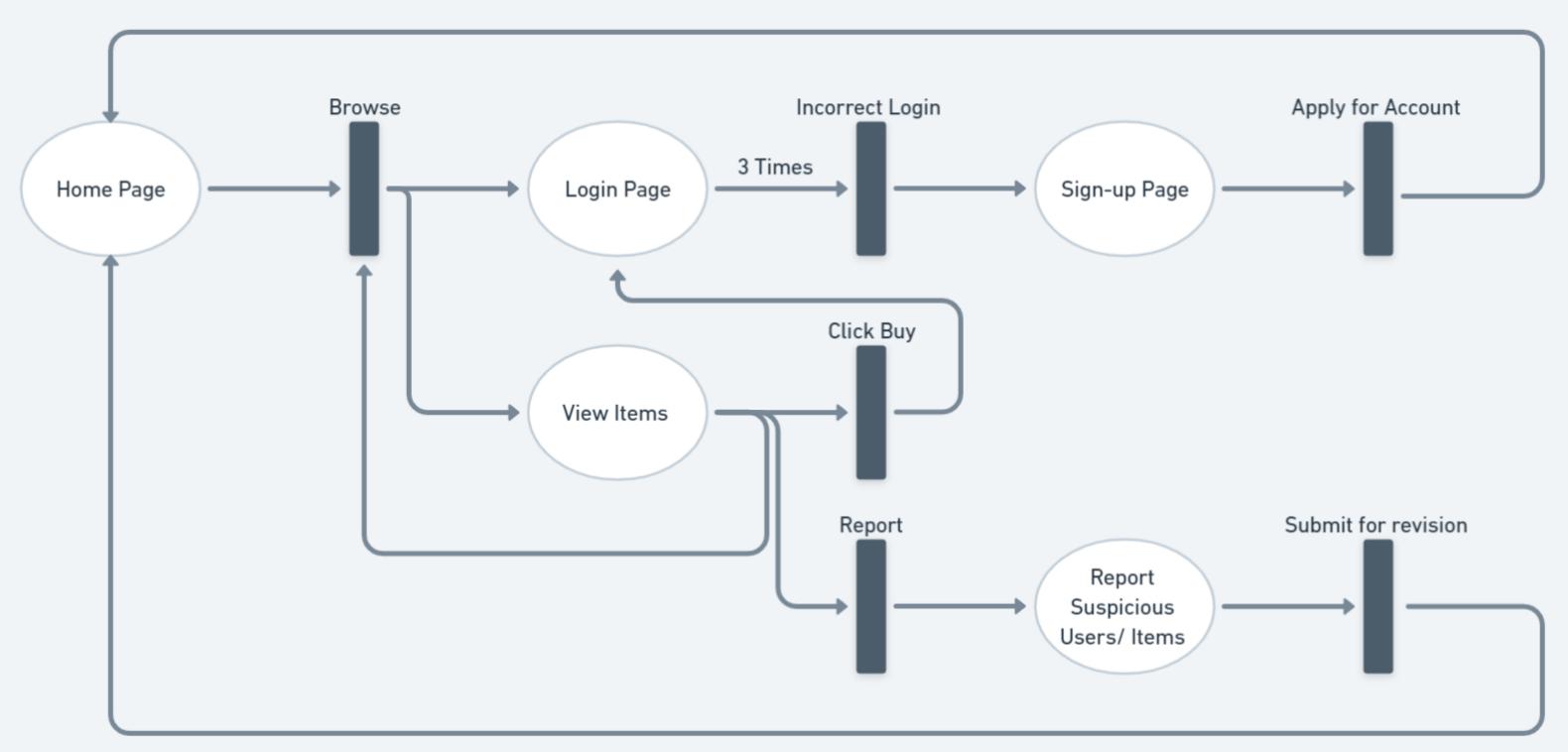
4.1. Diagrams

Overall Use Case Diagram



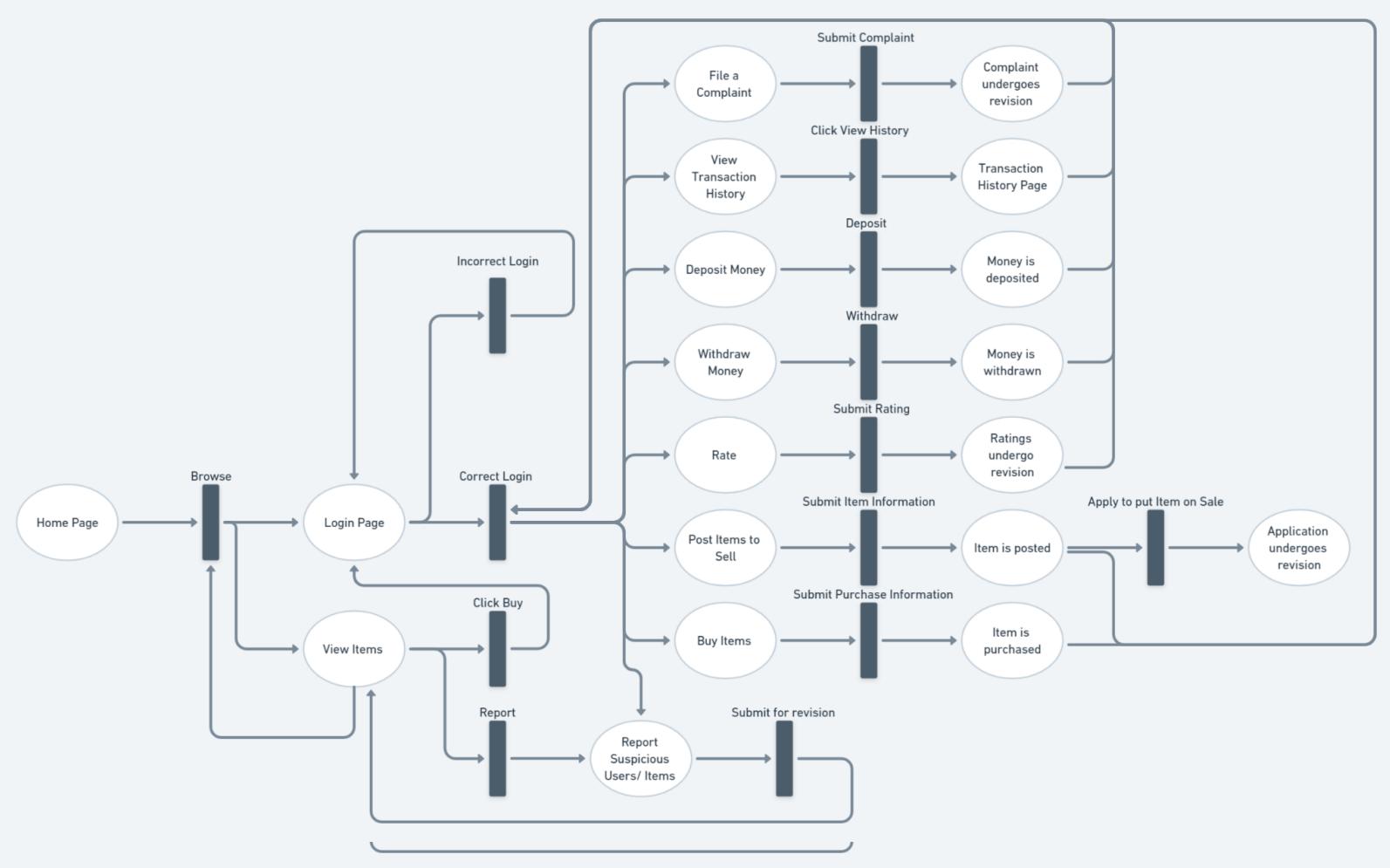
TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Guest Use Case Petri Net



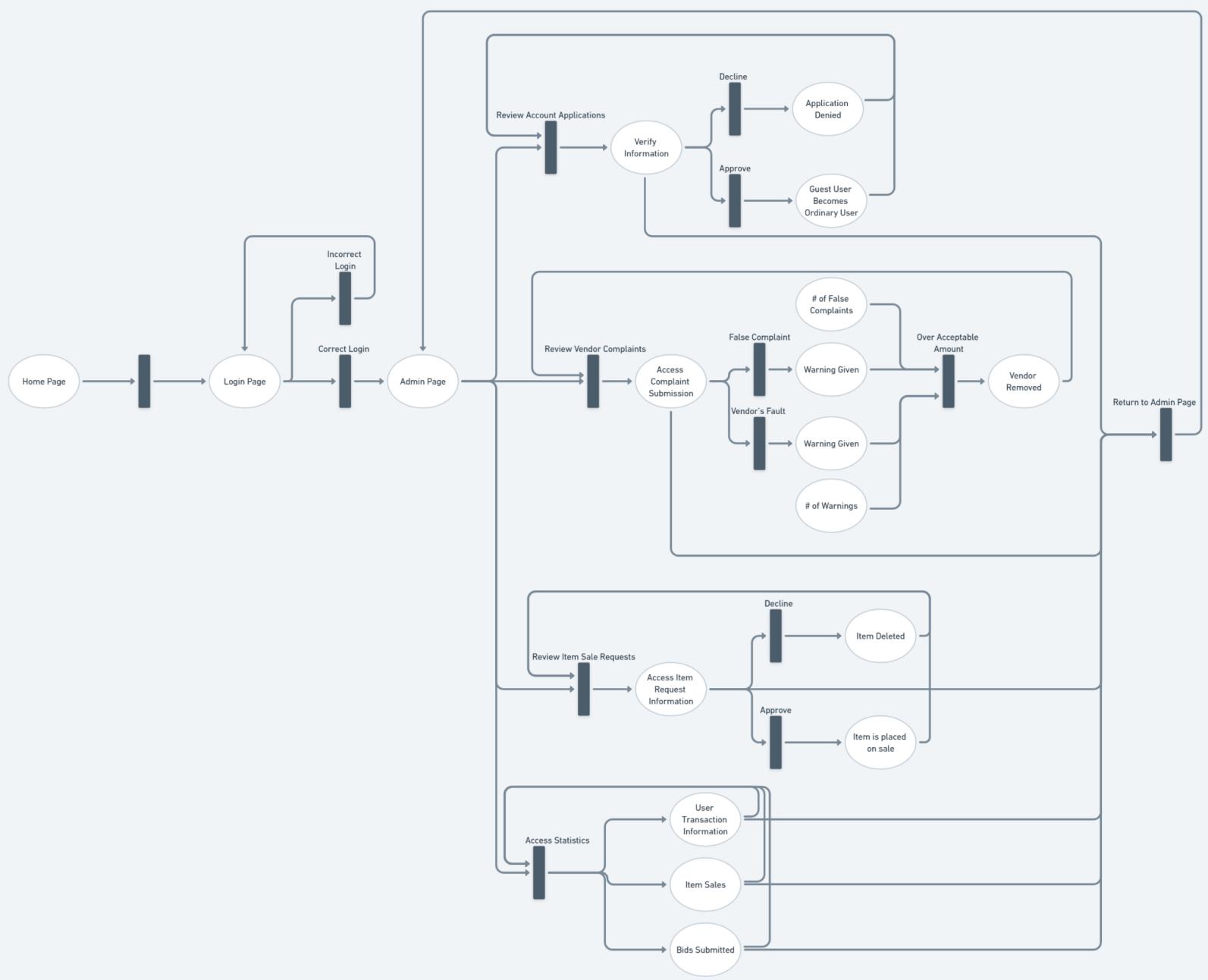
TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Ordinary Use Case Petri Net



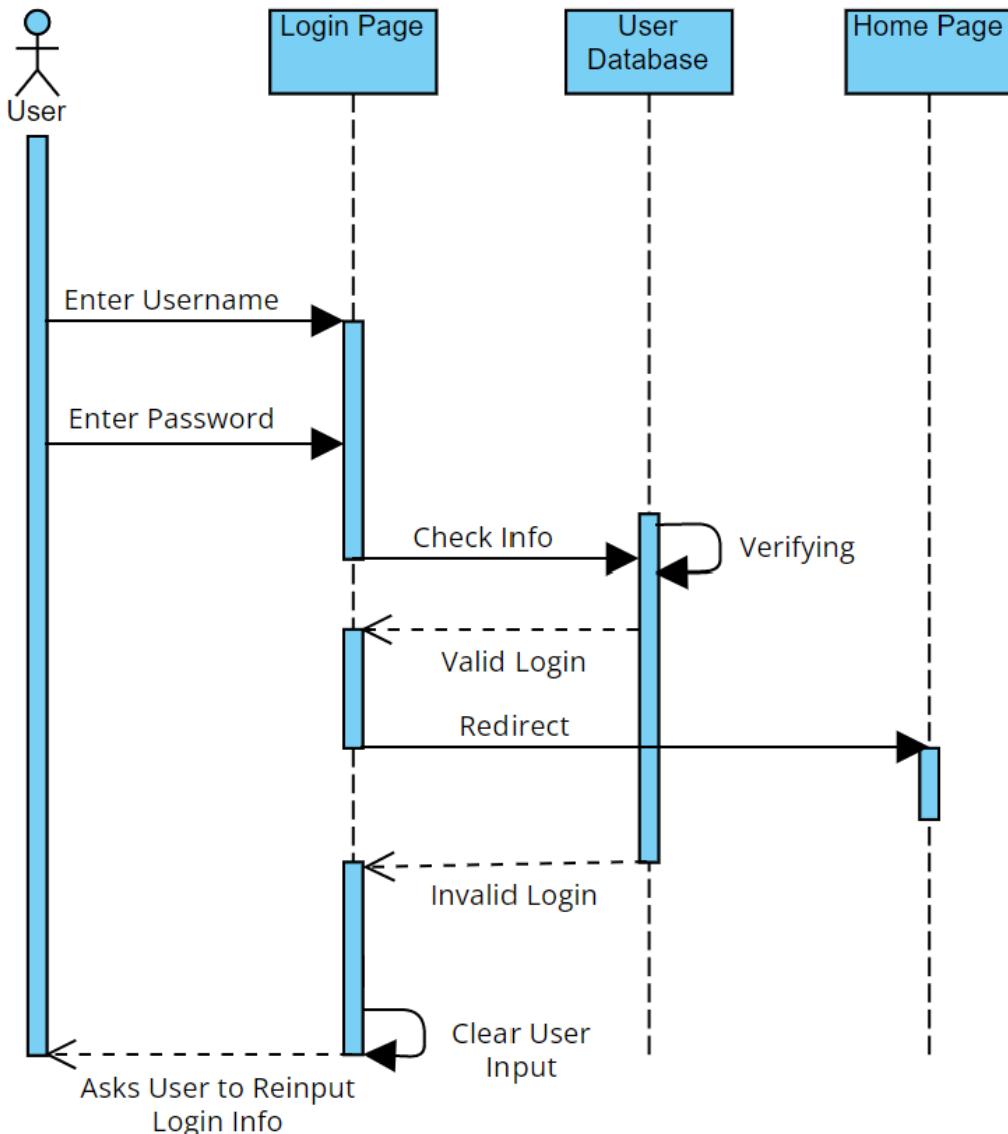
TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Super Use Case Petri Net



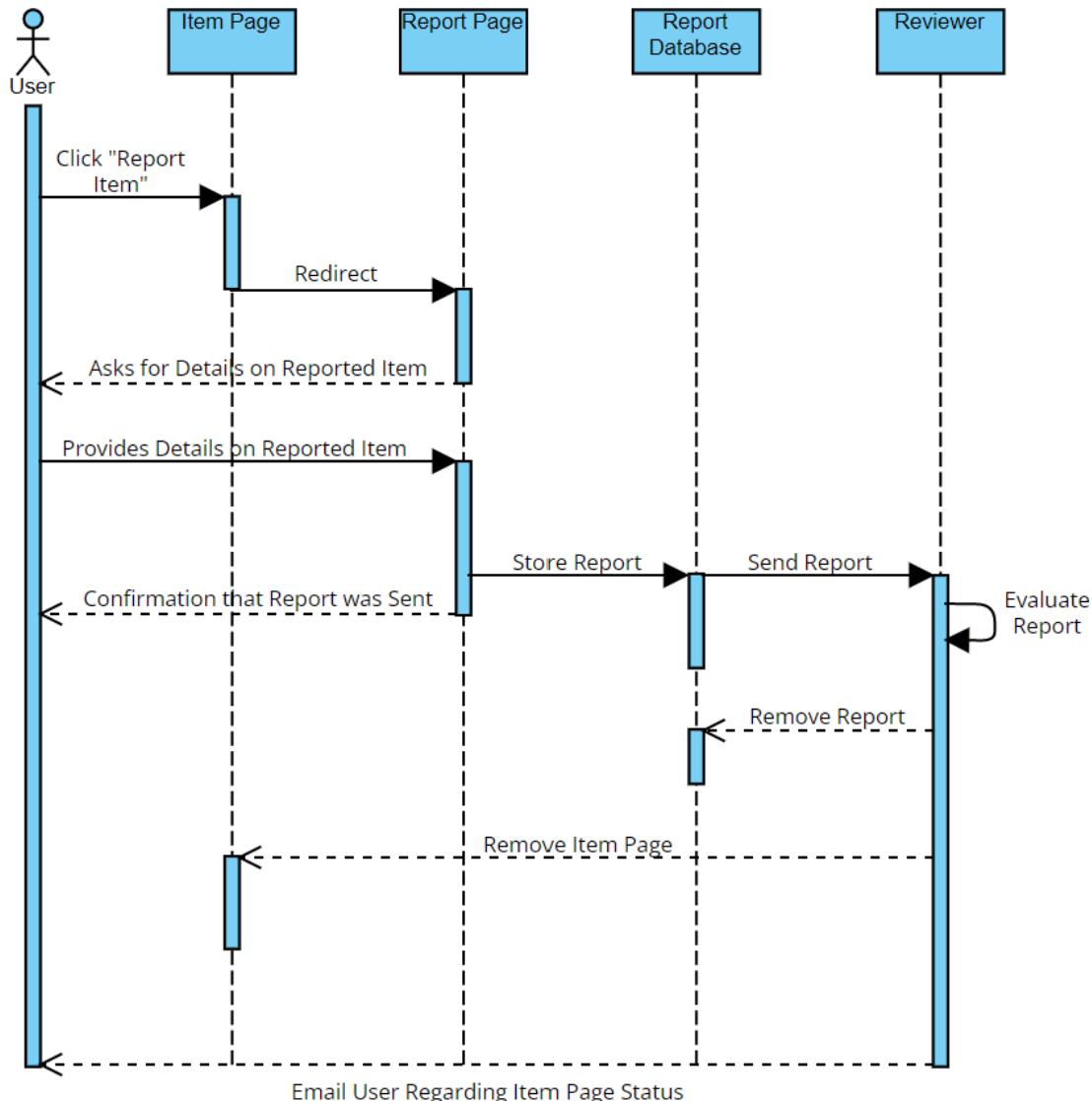
TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Login Page Sequence Diagram



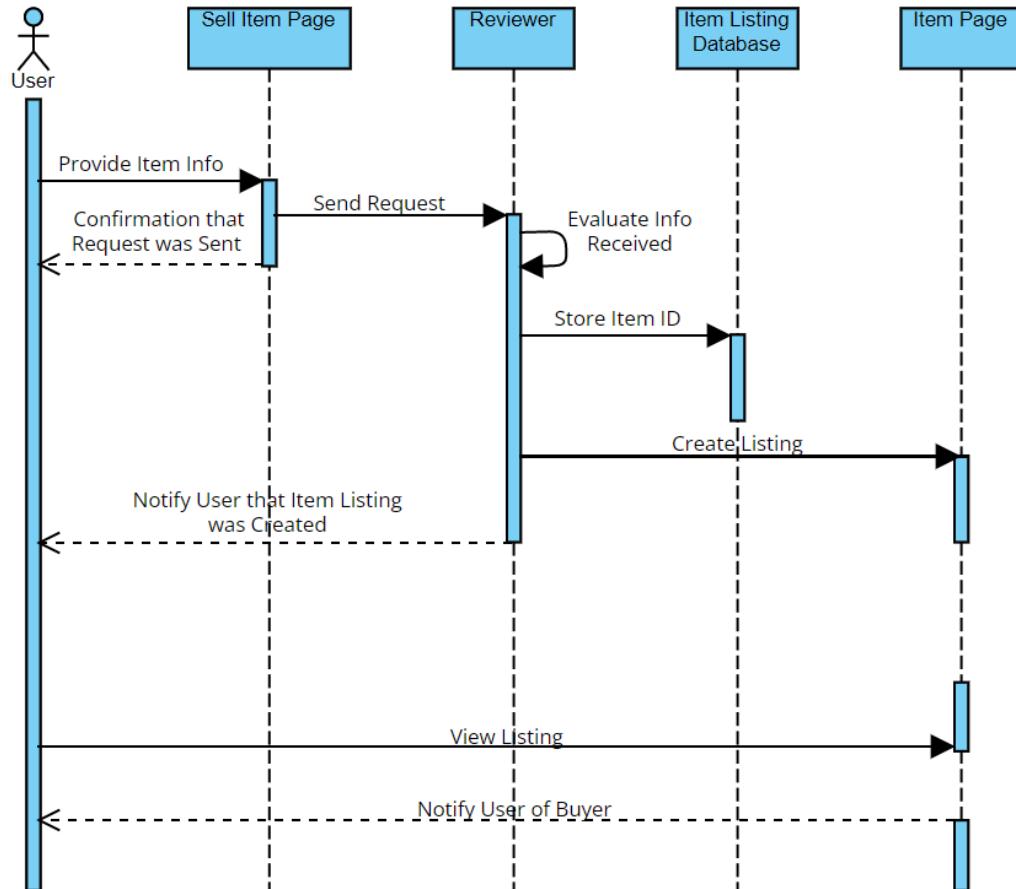
TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Reporting Item Sequence Diagram



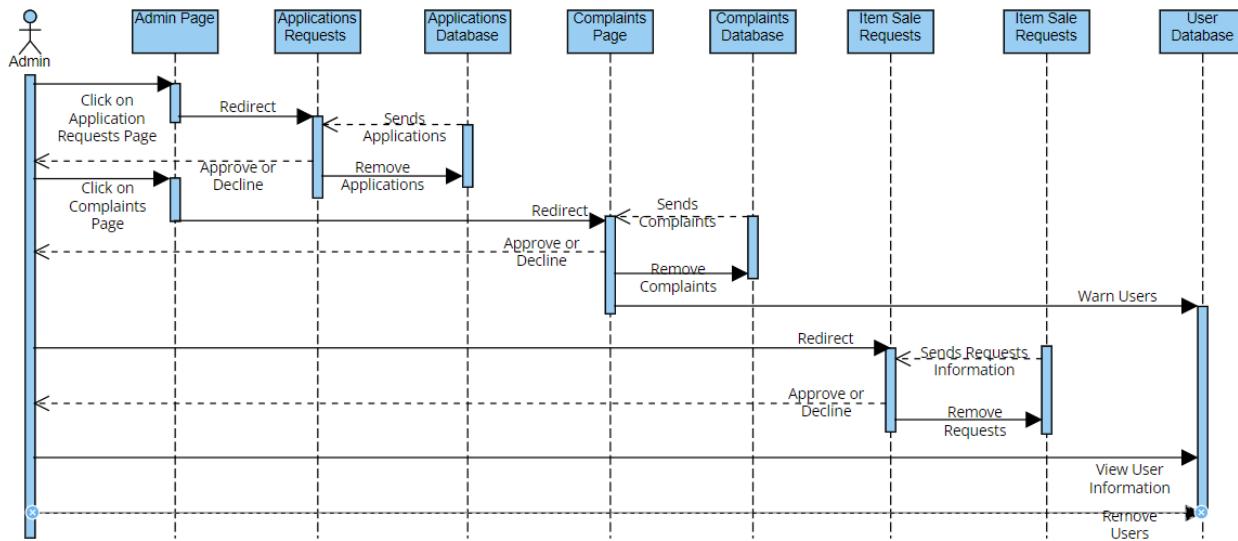
TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Request for Item Posting Sequence Diagram



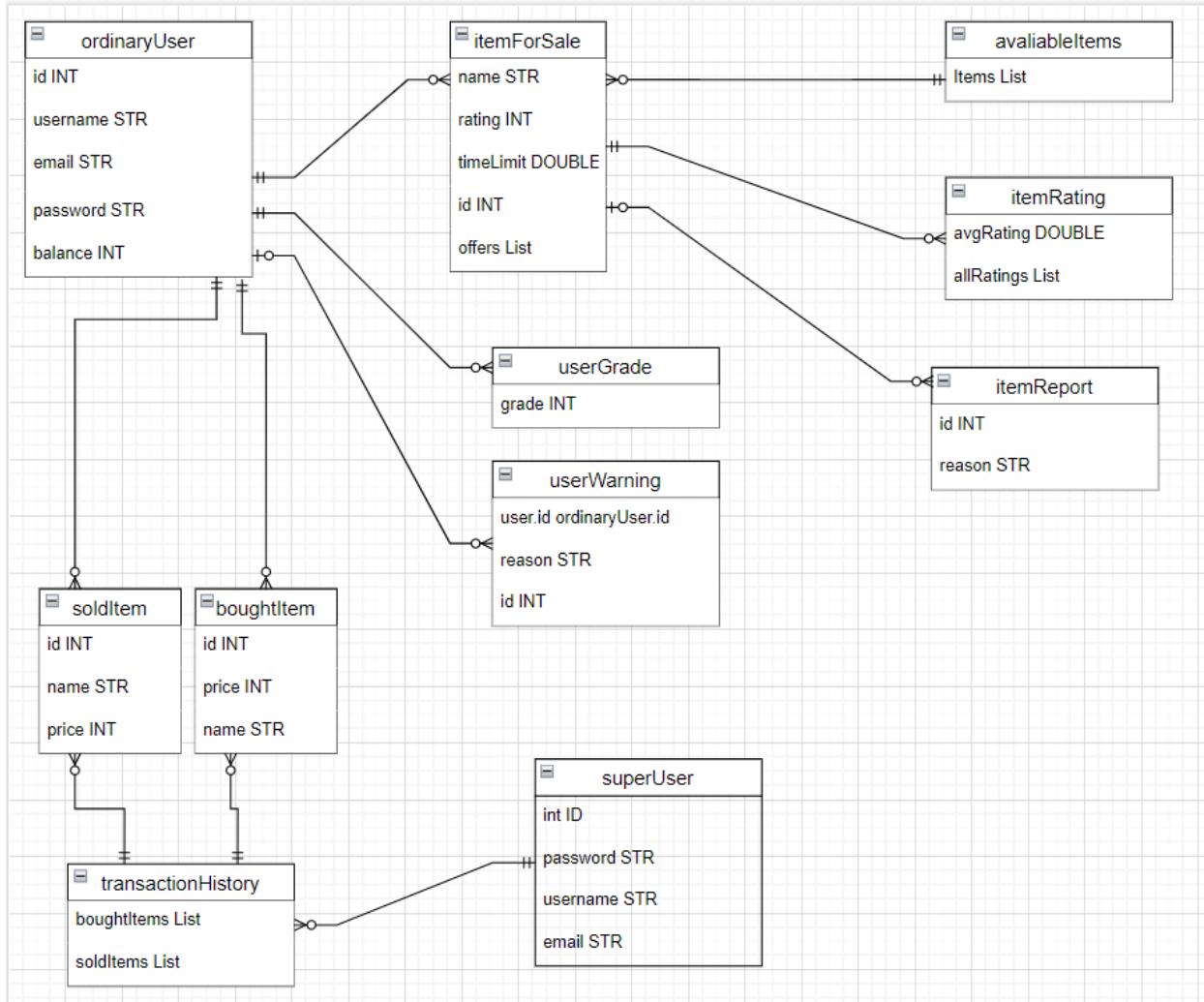
Admin Sequence Diagram

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	



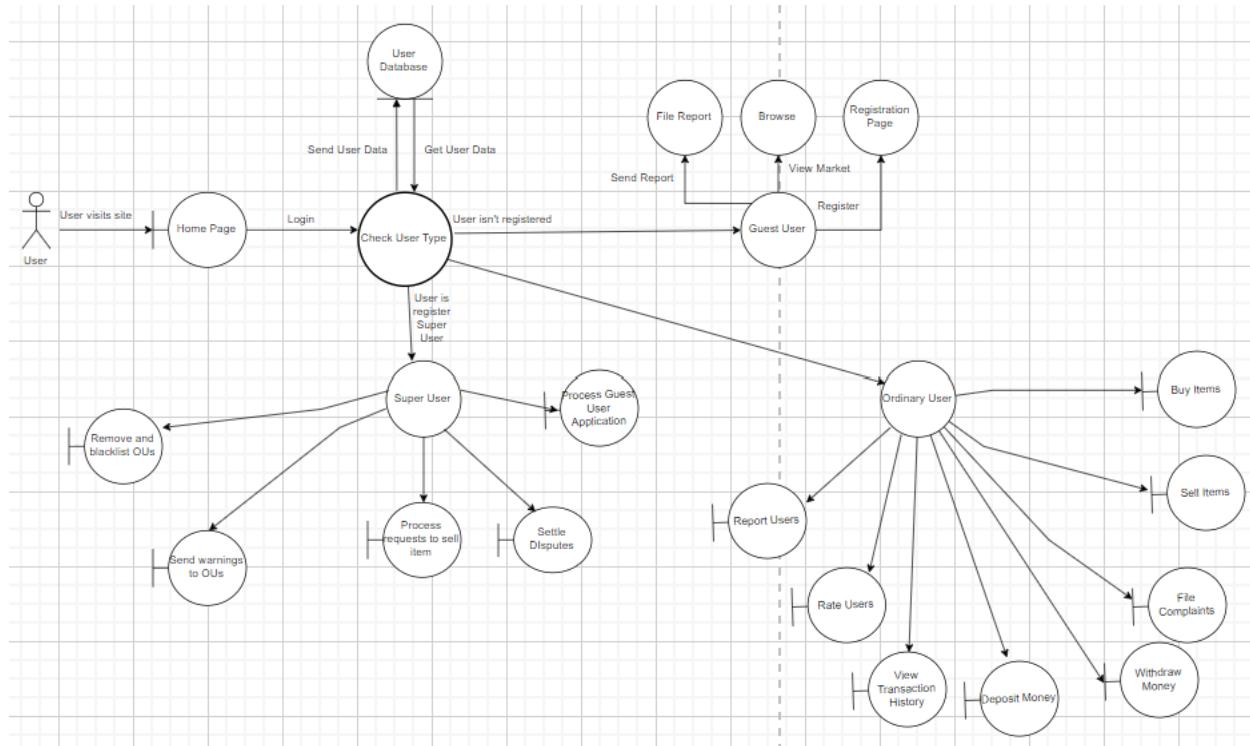
E-R Diagram

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	



Collaboration Class Diagram

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	



4.2. Pseudo-code

```
# Guest User Methods

# Browse method
# Exhausts an array of items back once a request is submitted in the
search bar
def browse(search) -> List[str]:
    search = search.split()
    List = []
    for column in cursor:
        # Checking the SQL database for the name of the listing items
        # If column[val] = any word search -> if the name of a listing
contains any part of the search
        List.append(column[val])

    return List
```

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

```

def login(email, password) -> bool:
    # Checking the column with the emails
    # Find the assigned id with the email
    # call function getPassbyID and compare the input password with the
output
    for column in cursor:
        if email == column[val]:
            id = getIDbyEmail()
            if getPassbyID() == password:
                return True
            else:
                counter += 1
                if counter == 3:
                    # go to apply page
                return False

# reportItem also applies for OU
def reportItem(item, textExplanation):
    # Item is an object which contains information about the item, such as
the id, name, price... which is stored in the class Item
    # A object called Report which contains information regarding the
report, and the id of the item reported will be returned
    reportedItem = itemReport(item, textExplanation)

    # List of reports in appended with this new report -> super user is
alerted when there is a new report
    reportedItems.append(reportedItem)

def accountApply(username, email, password):
    # Make a new instance of a class with the given credentials
    newApplication = application(username, email, password)

    # Append the array with the applications
    applications.append(newApplication)
    # A method where the super user will be alerted of a application
    superuser.alert_application()

```

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

```
# Ordinary User

def userComplaint(user, textExplanation):
    # userReport is a class which contains a class of user in it, as well
    as a field for the text explaining the reason behind the
    reportedUser = userReport(user, textExplanation)

    # List of reports in appended with this new report -> super user is
    alerted when there is a new report
    userReports.append(reportedUser)

def viewHistory(user):
    # we iterate through the transaction history and exhaust with each
    iteration
    for column in cursor:
        # exhaust each input to the screen

def depositMoney(user, amount):
    if(): # bank balance is enough
        # Check that there is a bank account/card is available
        assert(user.hasBank())
        user.balanceChange(amount)
        user.bankAdd(amount * -1)
    else:
        # Amount requested is larger than the bank balance

def withdrawMoney(user, amount):
    if (user.balance >= amount):
        # Check that there is a bank account/card is available
        assert(user.hasBank())
        user.bankAdd(amount)
        user.balanceChange(-1 * amount)
    else:
        # Amount requested is larger than the balance
```

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

```

def rate(item, rating) -> bool:
    # ratingRevision is checked by SUs; returns bool
    if ratingRevision(item, rating):
        item.rating = rating
        return True
    else:
        return False

def requestToPostSale(item, requestInfo):
    # Item is the Item class, which holds item to be sold; requestInfo
    hold the information regarding the request
    saleRequest = itemSaleRequest(item, requestInfo)

    # List of reports in appended with this new report -> super user is
    alerted when there is a new report -> sorted using FIFO methodology
    saleRequests.append(saleRequest)

def offer(item, seller, buyer, offer) -> bool:
    if(balance >= item.price + tax(item.price) + item.shipping_fee):
        item.offer_count += 1
        seller.alert(offer)
        # If user has an active item to be sold, he will have his offers
        stored in a array
        # Using the alert function he will be able to accept or deny the
        offer
        return True

# Super User

def processApplication() -> bool:
    for application in applications:
        # iterate through the array, in FIFO fashion, and ensure that
        things such:

```

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

```

# username is available
# no inappropriate language
# email exist
if(): # SUs check the application and if they approve
    application.approve() # application being approved makes a
newUser = user(application.username, application.email,
application.password)
    # store everything in a database
    # remove the application from array
else:
    application.deny()
    # remove the element from the array and send message

def processSaleRequests():
    # Handled in FIFO fashion
    for saleRequest in saleRequest:
        # preview the request to the SUs
        if(): # if the request of the items is deemed fair by the SUs
            saleRequest.approve() # function approve will allow the item
to be posted
            # remove the element from the array
        else:
            saleRequest.deny()
            # If SUs deem the request to be false, they will be able to
send a message
            # remove the item from the array

def reviewUserComplaints():
    for complaint in userReports:
        if(): # if the complaint is deemed true
            comlpaint.user.giveWarning()
            # remove the element
        else:
            # remove the element
    # System will check warning count for user, each time a warning is
issued

```

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

```
# and if it is over the acceptable amount, user is removed

def accessStatistics(request):
    # based on the request, the information desired will be returned from
    the database
```

4.3. GUI and System Prototypes

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Homepage

TweakMart **Home** Market Contact Login/Register

Holiday DEAL!!!

Save Up to 50% off!

Score big on a deal while supplies last.

Today's Deals - All With Free Shipping [See More ->](#)

TweakMart **Home** Market Contact Login/Register

Today's Deals - All With Free Shipping [See More ->](#)

<i>Fluffy Bean Bag w/ Removable Cover</i>	<i>Microsoft Xbox Wireless Controller</i>	<i>Arlo Spotlight Wireless Camera</i>	<i>Robot Vacuum Cleaning System</i>
\$59.99	\$39.99	\$50.81	\$149.99
\$99.99 · 40% off	\$59.99 · 33% off	\$159.99 · 68% off	\$449.99 · 67% off

Market

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

TweakMart

Market

Contact

Login/Register

Find Items

Enter Title or Keywords

Enter Title or Keywords

Price Range

Show Item priced from \$ Min to \$ Max

Ratings

★★★★★

Search

Categories

Electronics

Collectibles & Art

Fashion

Toys & Hobbies

Sporting Goods

Health & Beauty

Books, Movies, & Music

Business & Industrial

Home & Garden

Other

Categories

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Nvidia RTX 3080 Founders Edition FE 10GB GDDR6X Graphics Card
\$630.00
 Free shipping or Best Offer
 36 sold

Lot of 5 HP NVIDIA Quadro K2000 2GB GDDR5 Graphic 713380-001
\$27.00
 \$13.57 shipping
 3 bids
 3d 1h

Contact

Got Questions? Need Help?
 We Will Get Back To You As Soon As Possible!

Your Name

Your Email

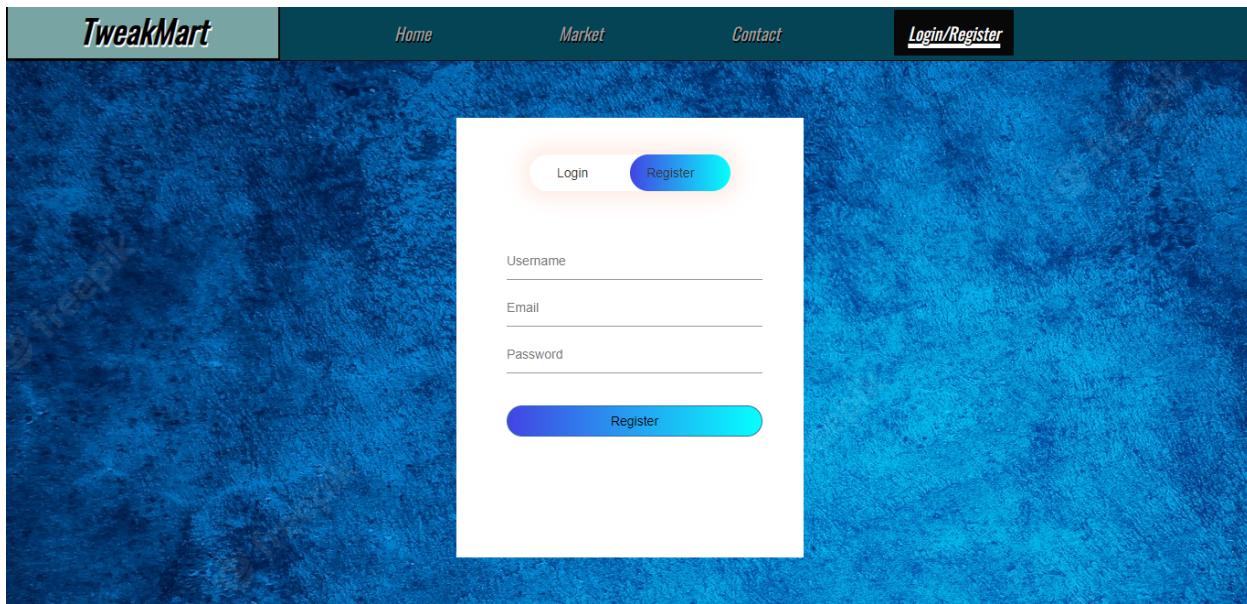
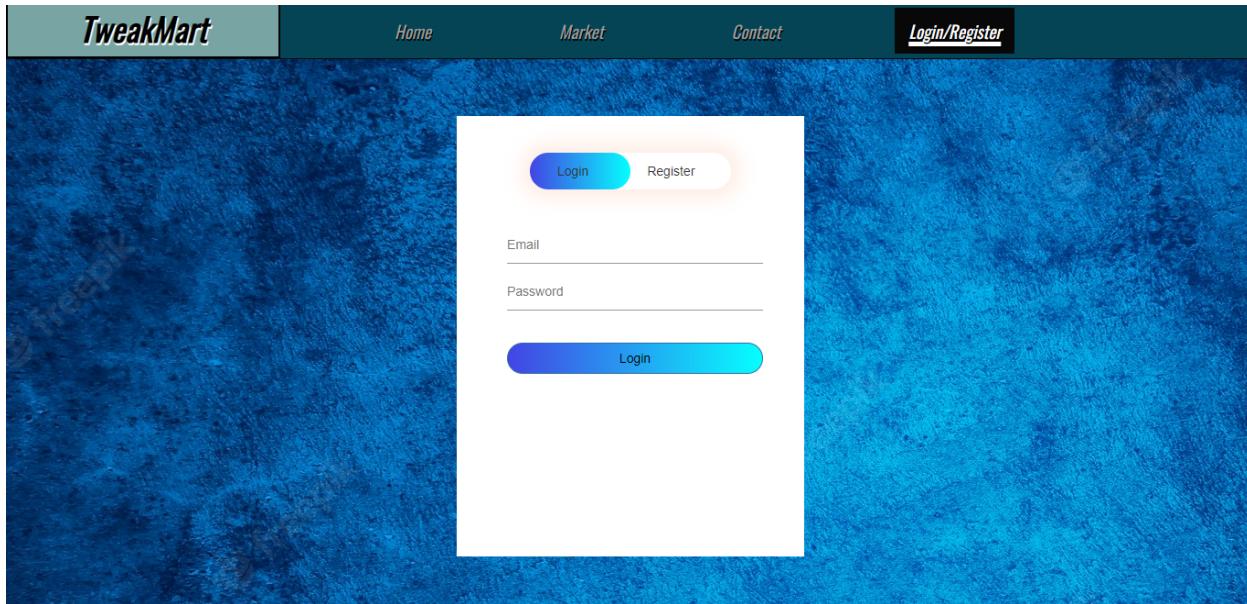
Message

Submit

Or you can email us directly at OfficialTweakMart@gmail.com

[Login/Register](#)

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	



After logging in → homepage

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Holiday DEAL!!!

Save Up to 50% off!

Score big on a deal while supplies last.

Today's Deals - All With Free Shipping [See More ->](#)

Personal Information

[Home](#)
[Personal Information](#)
[Payment & Wallet](#)
[Transaction](#)
[Logout](#)

Username <input type="text" value="New Username"/>	Old Password <input type="text" value="Old Password"/>
Phone Number <input type="text" value="New Phone Number"/>	New Password <input type="text" value="New Password"/>
Address <input type="text" value="New Address"/>	Retype Password <input type="text" value="Retype Password"/>
<input type="button" value="Save Changes"/>	

Payment & Wallet

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

Home
Personal Information
Payment & Wallet
Transaction
Logout

Add a credit card

Card Number:

Name On Card:

Expiration Date: MM YYYY

CVV: ###

[Add Your Card](#)

Balance: \$0

Deposit (\$): Amount Retype Amount

Withdraw(\$): Amount Retype Amount

Transaction

Home
Personal Information
Payment & Wallet
Transaction
Logout

Transaction History

There are no transaction made so far

SELL YOUR ITEM

Title
Use words people would search for when looking for your item
 Enter Title or Keywords

Time limit
 # of Day(s)

Price
 \$

Add a Photo
 No file chosen

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

4.4. Group meetings Memo

Date	Agenda Item	Updates	Participants
<20/10/2022>	<Decide on which programming languages to use, Share Ideas regarding the UI and functionalities, Review the requirements, Schedule the next meeting>	<Information Research, documentation, Set up a google doc to share and collaborate>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<27/10/2022>	<Roles are assigned to each member, front end role given to Johnson and Anova, back end given to Daphne, Konrad, and Dino, schedule the next available meeting, start working on phase 1 report>	<Chosen HTML, CSS, JS, and Django, set up the project environment, Github repo will be created later on>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<1/11/2022 - <2/11/2022>	<Outline of the website development, final revision of phase 1 report and submitted, schedule the next available meeting>	<Refresh and learn the programming languages that we decided on the project>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<7/11/2022>	<Share progress, Check up on each other, give advice or recommendation, help one another if needed, schedule the next available meeting>	<Website in development, brainstorm the basic structure of the UI, Github Repo has been created>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<12/11/2022> - <13/11/2022>	<Share progress, start working on phase 2 report, schedule the next available meeting>	<Website in development, figuring out how to merge code between frontend and backend>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>
<20/11/2022> - <24/11/2022>	<Share progress, Final revision on phase 2 report, schedule the next available meeting>	<Website in development, Pushing code to github repo>	<Johnson Chen, Dino Radoncic, Anova Sarkar, Daphne Tang, and Konrad Zielinski>

TweakMart	Version: <2.0>
Software Requirements Specification	Date: <24/11/2022>
<document identifier>	

4.5. Github Repo Address

- <https://github.com/ooDino/TweakMart/tree/master>