**Requirements Specifications**

**P03 :MunasibMall.pk**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| **22100134** | **Abdur Rehman Masood** |
| **22100168** | **Daniyal Mumtaz** |
| **21100161** | **Muhammad Muzammil Khan** |
| **21100034** | **Waliullah Aitemad** |
| **22100199** | **Waqar Ul Haq Khatana** |

|  |  |  |
| --- | --- | --- |
| **Content** | **Totals** | **Obtained** |
| Introduction & system actors | 5 | 5 |
| Use case diagram | 10 | 10 |
| Use case descriptions | 20 | 10 |
| Class diagram | 20 | 7 |
| Sequence diagram | 20 | 0 |
| State diagram | 5 | 0 |
| Non-functional requirements | 5 | 5 |
| Who did what | 5 | 5 |
| Review checklist | 5 | 5 |
| Overall formatting/template | 5 | 3 |
| Late submission penalty | -20 |  |
| **Total** | **100** | **50** |
| Review | 20 |  |
| **Grand Total** |  |  |

**Table of Contents**

[1. Introduction 3](#_Toc82791699)

[2. System Actors 4](#_Toc82791700)

[3. Use Cases 5](#_Toc82791701)

[3.1 Use Case Diagrams 5](#_Toc82791702)

[3.2 Description of Use Cases 5](#_Toc82791703)

[3.2.1 Withdraw cash 5](#_Toc82791704)

[3.2.2 Transfer funds 6](#_Toc82791705)

[4. Class Diagram 7](#_Toc82791706)

[4.1 Diagram 7](#_Toc82791707)

[4.2 Description 7](#_Toc82791708)

[5. Sequence Diagrams 8](#_Toc82791709)

[5.1 Use case Name e.g., Withdraw cash 8](#_Toc82791710)

[5.2 Use case Name e.g., Transfer funds 8](#_Toc82791711)

[6. State Diagrams 9](#_Toc82791712)

[6.1 Diagram details 9](#_Toc82791713)

[6.2 Diagram 9](#_Toc82791714)

[7. Non-functional Requirements / Quality Attributes 10](#_Toc82791715)

[8. Who Did What? 11](#_Toc82791716)

[9. Review checklist 11](#_Toc82791717)

# Introduction

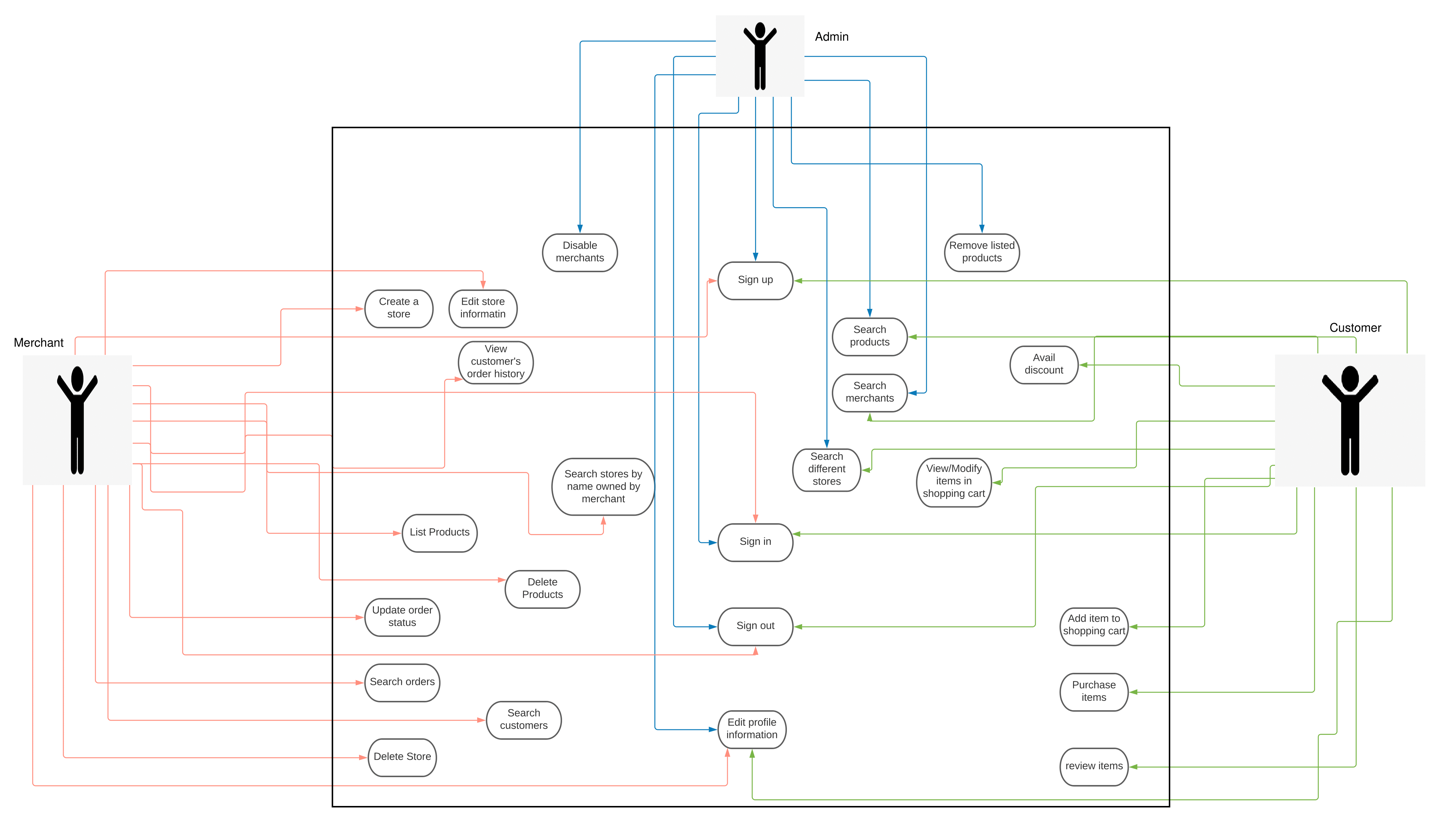
A mobile e-commerce centric application that will allow different merchants to have multiple stores of their own on the platform. Merchants will be given more autonomy than traditional services like Daraz.pk. They will also be given some customizability in making their page. Customers would be able to see specific pages or ‘shops’ as well as search for products in all of the shops on the platform and purchase the products that they like. Orders will be sent directly to the merchant who will then handle the delivery. Payment options like easy paisa or COD will be shown to the customer who can select their preferred method when purchasing. The system aims to provide a platform that gives more autonomy and the ability to handle orders to the merchants. Reviews and ratings will help the customer choose the right merchant.

# System Actors

|  |  |
| --- | --- |
| **Actor Name** | **Description** |
| Admin | Administrator that can monitor merchants and their products on the basis of different indexes. Most privileged actor. |
| Merchant | Merchant has the ability to create and manage multiple stores and make product listings on them. |
| Customer | A registered or unregistered user that can make purchases from different merchants. |

# Use Cases

## Use Case Diagrams



A better version of the use case diagram can be seen here:

https://lucid.app/lucidchart/9f8b3067-5b88-4bb3-b4ba-d1d66a503daf/edit?viewport\_loc=-1271%2C-407%2C4440%2C1848%2C0\_0&invitationId=inv\_9392ad2c-b51d-4200-a458-18347a97030e

## Description of Use Cases

[Write use case description in a way that the roles of actor and the system become clear at each step. Refer to the sample provided in the template.

In general, there are issues with overall content including pre and post conditions, format of description and alternative paths. See my comments in some selected use cases below for specific examples.]

Merchant: Sign Up:

|  |  |
| --- | --- |
| Identifier | 1 |
| Purpose | The user will be able to register to become a merchant |
| Pre-Condition | The user must have a valid email address |
| Post-Condition | The user is registered as merchant |
| **Step #** | **Typical Course of Action** |
| 1 | User clicks on sign up from the Home page and is redirected to the sign up screen |
| 2 | User enters his/her details |
| 3 | User clicks on Sign up button and is registered |
| **Step #** | **Alternate Course of Action** |
| 1 | User can also move to the sign up screen from the sign in screen |
| **Step #** | **Exception Path** |
| 1 | In step 2, if any of the input fields is empty or if the email does not contain @, user will be redirected to the sign up page |

Merchant: Sign In

|  |  |
| --- | --- |
| Identifier | 2 |
| Purpose | The user will be able to sign in using their credentials to be able to see their dashboard. |
| Pre-Condition | 1. The user must be registered. 2. The credentials entered by the user are valid, i.e., they are the same as those stored in the database. [This is not a pre-condition, it is part of the use case.] |
| Post-Condition | The user will be redirected to their dashboard. |
| **Step #** | **Typical Course of Action** |
| 1 | User clicks on Sign in from the Home page and is redirected to the sin in screen. |
| 2 | User enters his/her credentials |
| 3 | Upon verification of the credentials, user is redirected to their dashboard |
| **Step #** | **Alternate Course of Action** |
| 1 | User can also move to the sign in screen from the sign up screen |
| **Step #** | **Exception Path** |
| 1 | In step 2 and 3, if any of the input fields is empty or if the credentials are invalid, user is redirected to the sign in screen. |

Merchant: Edit Profile

|  |  |
| --- | --- |
| Identifier | 3 |
| Purpose | Merchants will be able to edit their profile information |
| Pre-Condition | User must be signed into the system. |
| Post-Condition | Changes that the user makes are saved into the database |
| **Step #** | **Typical Course of Action** |
| 1 | From the dashboard, user clicks on Edit Profile |
| 2 | User is redirected to the edit profile screen |
| 3 | User can change some of their details including password |
| 4 | User clicks on Save and the changes are saved |
| **Step #** | **Alternate Course of Action** |
|  | Nil |
| **Step #** | **Exception Path** |
| 1 | If the user clicks on Cancel or refreshes the page before Saving the changes, those changes will not be saved. |

Merchant: Log Out

|  |  |
| --- | --- |
| Identifier | 4 |
| Purpose | User will be able to log out of the system |
| Pre-Condition | User must be logged in |
| Post-Condition | User is logged out of the system and is redirected to the Home page. |
| **Step #** | **Typical Course of Action** |
| 1 | User clicks on Log Out from any screen |
| 2 | User’s Session is ended, is logged out of the system and is redirected to the Home page |
| **Step #** | **Alternate Course of Action** |
| 1 | A period of 30 minutes of inactivity will also result in user being logged out of the system |
| **Step #** | **Exception Path** |
|  | Nil |

Merchant: Create Store

|  |  |
| --- | --- |
| Identifier | 5 |
| Purpose | The user (Merchant) will be able to create a store |
| Pre-Condition | The user must be logged in |
| Post-Condition | The user will have a store hosted on our website |
| **Step #** | **Typical Course of Action** |
| 1 | User clicks on Create Store and is redirected to create store screen |
| 2 | User enters detailed personal information as well as the details of the store |
| 3 | User clicks on Create button |
| 4 | User is redirected to his/her store dashboard |
| **Step #** | **Alternate Course of Action** |
|  | Nil |
| **Step #** | **Exception Path** |
| 1 | In step 2, if any of the input fields is empty the user is redirected to the create store screen. If the user clicks on Cancel, he/she is redirected to the dashboard |

Merchant: Delete Store

|  |  |
| --- | --- |
| Identifier | 6 |
| Purpose | The user (Merchant) will be able to delete their store |
| Pre-Condition | 1. The user must be logged in. 2. The store to delete must exist and must belong to the user |
| Post-Condition | User’s store is deleted |
| **Step #** | **Typical Course of Action** |
| 1 | From the dashboard user(merchant) clicks on their store and is redirected to their store dashboard |
| 2 | From the store dashboard, user clicks on Delete button |
| 3 | A confirmation message is displayed to the user. |
| 4 | If the user clicks on Confirm, the user’s store is deleted |
| **Step #** | **Alternate Course of Action** |
|  | Nil |
| **Step #** | **Exception Path** |
| 1 | In step 3 if user clicks on Cancel, user’s store is retained |

Merchant: Edit Store Information

|  |  |
| --- | --- |
| Identifier | 7 |
| Purpose | The user (Merchant) will be able to edit their store information |
| Pre-Condition | 1. User must be logged in 2. Store to edit must exist and must belong to the user |
| Post-Condition | User’s store information is updated |
| **Step #** | **Typical Course of Action** |
| 1 | User clicks on Edit Store from the dashboard and is redirected to edit store screen |
| 2 | User can change some of the store information |
| 3 | User clicks on Save button upon which a confirmation message is displayed to the user |
| 4 | If the user clicks on Confirm, the store’s information is updated in the database |
| **Step #** | **Alternate Course of Action** |
|  | Nil |
| **Step #** | **Exception Path** |
| 1 | In step 4 if user clicks on Cancel, the store information is not changed and the user is redirected to the store dashboard. |
| 2 | If the user refreshes the page before Saving the changes, no change is made and the user is redirected to the edit store screen afresh |

Merchant: Search Stores

|  |  |
| --- | --- |
| Identifier | 8 |
| Purpose | The user (Merchant) will be able to search their stores by name |
| Pre-Condition | User must be logged in |
| Post-Condition | Stores belonging to the user with the relevant keywords are displayed on screen |
| **Step #** | **Typical Course of Action** |
| 1 | From the dashboard, user will enter keyword to search for their stores using the search bar |
| 2 | All the stores belonging to the user with the relevant keyword will be displayed |
| **Step #** | **Alternate Course of Action** |
|  | Nil |
| **Step #** | **Exception Path** |
| 1 | If none of the user’s stores match the keywords, A message saying “0 matches” will be displayed |

Merchant: List Products

|  |  |
| --- | --- |
| Identifier | 9 |
| Purpose | The user (Merchant) will be able to list multiple products on their store as well as upload photos of the products. |
| Pre-Condition | 1. User must be logged in 2. User must own a store |
| Post-Condition | Product(s) will be listed on user’s store |
| **Step #** | **Typical Course of Action** |
| 1 | From the store dashboard user will click on List Product and will be redirected to list product screen |
| 2 | User will add the details of the product including name, price, and photos of the product |
| 3 | User will click on Add and the product will be listed on the Store |
| **Step #** | **Alternate Course of Action** |
|  | Nil |
|  | **Exception Path** |
| 1 | In step 3, if user clicks on Cancel button, product information will not be processed and user will be redirected to the store dashboard. [This is not an error. It is user’s choice. It is alternative path.] |
| 2 | If the user refreshes the page before clicking on the Add button, product information will be wiped and user will be redirected to list product screen afresh. |

Merchant: Search Products

|  |  |
| --- | --- |
| Identifier | 10 |
| Purpose | The user (Merchant) will be able to search their products based on different filters. |
| Pre-Condition | 1. User must be logged in 2. User must own a store |
| Post-Condition | Products belonging to the user’s store with the relevant keywords will be displayed on screen. |
| **Step #** | **Typical Course of Action** |
| 1 | From the store dashboard user will enter the keywords in the search bar |
| 2 | Products with the relevant keywords will be displayed  User can filter results based on different filters such as price range, date listed, etc. |
| **Step #** | **Alternate Course of Action** |
|  | Nil |
| **Step #** | **Exception Path** |
|  | If none of the products satisfy the search result a message saying “0 matches” will be displayed. |

Merchant: Delete Products

|  |  |
| --- | --- |
| Identifier | 11 |
| Purpose | The user (Merchant) will be able to delete their products. |
| Pre-Condition | 1. User must be logged in 2. User must own a store 3. The product to be deleted must exist and must belong to the user |
| Post-Condition | Product is removed from the store. |
| **Step #** | **Typical Course of Action** |
| 1 | From the My Products panel, user will click on the product to be deleted |
| 2 | This will open the product details |
| 3 | User will click on the Delete button |
| 4 | A confirmation message will be displayed on screen  User will click on Confirm and the product will be deleted from the store |
| **Step #** | **Alternate Course of Action** |
| 1 | User can also quickly delete a product by clicking on the delete icon beside the product name in the My Products panel |
| 2 | A confirmation message will be displayed  User clicks on Confirm and the product is deleted from the store |
| **Step #** | **Exception Path** |
| 1 | In step 5, if the user clicks on Cancel, the product will not be deleted. |

Merchant: View Customer’s Order History

|  |  |
| --- | --- |
| Identifier | 12 |
| Purpose | The user (Merchant) will be able to view a particular customer’s order history |
| Pre-Condition | 1. User must be logged in 2. User must own a store |
| Post-Condition | Customer’s order history is displayed on screen. |
| **Step #** | **Typical Course of Action** |
| 1 | From the Customer panel, user will click on a particular customer |
| 2 | User will then click on Order History |
| 3 | The order history of that customer will be displayed on screen |
| **Step #** | **Alternate Course of Action** |
| 1 | User will click on Order panel |
| 2 | User will search for the orders of a particular customer using search filter |
| 3 | All the orders of that customer will be displayed on screen |
| **Step #** | **Exception Path** |
|  | Nil |

Merchant: Search Customer

|  |  |
| --- | --- |
| Identifier | 13 |
| Purpose | The user (Merchant) will be able to search customers based on different filters. |
| Pre-Condition | 1. User must be logged in 2. User must have a store |
| Post-Condition | Customers with the relevant keywords will be displayed on screen |
| **Step #** | **Typical Course of Action** |
| 1 | From the Customer panel, user will enter the keywords in the search bar |
| 2 | All the customers with the relevant keywords will be displayed on screen |
| 3 | User can filter search results based on different filters |
| **Step #** | **Alternate Course of Action** |
|  | Nil |
| **Step #** | **Exception Path** |
| 1 | If there is no customer with the relevant keyword, a message saying “0 matches” will be displayed on screen. |

Merchant: Update Order Status

|  |  |
| --- | --- |
| Identifier | 14 |
| Purpose | The user (Merchant) will be able to update order status such as in-transit, delivered, etc. |
| Pre-Condition | User must be logged in |
| Post-Condition | The order status is updated |
| **Step #** | **Typical Course of Action** |
| 1 | From the order panel, user will click on a particular order |
| 2 | This will open up the order details. User can change the status of the product from there |
| **Step #** | **Alternate Course of Action** |
|  | Nil |
| **Step #** | **Exception Path** |
|  | Nil |

Merchant: Search Orders

|  |  |
| --- | --- |
| Identifier | 15 |
| Purpose | The user (Merchant) will be able to search orders based on different filters. |
| Pre-Condition | 1. User must be logged in 2. User must have a store |
| Post-Condition | Orders with the relevant keywords will be displayed on screen |
| **Step #** | **Typical Course of Action** |
| 1 | From the Order panel, user will enter the keywords in the search bar |
| 2 | All the orders with the relevant keywords will be displayed on screen |
| 3 | User can filter search results based on different filters |
| **Step #** | **Alternate Course of Action** |
|  | Nil |
| **Step #** | **Exception Path** |
| 1 | If there is no order with the relevant keyword, a message saying “0 matches” will be displayed on screen. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | 16 | |
| **Purpose** | | The user can sign in as a customer to avail discounts/promotions on items. | |
| **Pre-conditions** | | Sign up use case must have been implemented once earlier i.e. the user must have a personal account already. | |
| **Post-conditions** | | The customer is signed in and can avail discounts. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The customer selects an item on discount. | |  |
|  | The portal asks the customer to sign in with a username and password incase it has an account (and is not currently signed in) or sign up if it is a guest customer. [The singup step should be taken to the alternative courses of action.] | |  |
|  | If its a successful sign in, customer can avail the discounted price on the desired item and add to shopping cart. | |  |
|  | It can view the original (old) price, the discount and the discounted price now and proceed to checkout. | |  |
|  | Use case ends. | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 2 if the customer does not already have an account/guest customer, it can first sign up and then resume from step 3. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 3, if there it was not a successful sign in or it was a banned user, then an error message is displayed accordingly and execution proceeds to step 5. | |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | 17 | |
| **Purpose** | | The customer will have the option to logout and if that customer is inactive for a given amount of time then that it should be logged out by the system automatically. | |
| **Pre-conditions** | | Customer must have completed the Log In use case | |
| **Post-conditions** | | Customer is signed out. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The customer selects the my profile icon on the top right corner. | |  |
|  | A dropdown menu appears. | |  |
|  | The customer selects the logout option. | |  |
|  | The customer is redirected to sign in / sign up page. | |  |
|  | The use case ends. | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In case a session expired/ timeout event occurs, the customer will resume from step 4. | |  |
| **Step #** | **Exception Paths** | |  |
|  | NA. | |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | 18 | |
| **Purpose** | | The customer can browse different stores and their respective inventories. | |
| **Pre-conditions** | | No conditions (it can be a registered or a guest customer.) | |
| **Post-conditions** | | The customer can view the searched store/brand and view it’s products. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The customer enters the name of the store in the search engine. | |  |
|  | All the relevant/similar name stores (that are registered) along with their unique logo will appear as a list. | |  |
|  | The user selects the desired store. | |  |
|  | All the products offered by that store and their information (discounts and availability) will be made visible to the customer. | |  |
| 6. | The use case ends. | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 1, if the user does not press enter button while typing but instead selects from the suggestions in the dropdown menu, it will be directly taken to step 4. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 1, if its not a registered name (not in our database), then an error message is displayed stating no results found and execution proceeds to step 6. | |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | 19 | |
| **Purpose** | | The customer can add an item to its shopping cart. | |
| **Pre-conditions** | | The customer must not be banned by either the merchant or the admin. The shopping cart must consist of items from one store only (not multiple stores). | |
| **Post-conditions** | | The customer can view the item along with the price and the quantity in the shopping cart. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The customer clicks the desired product. | |  |
|  | The system displays a button ‘add to cart’. | |  |
|  | The user clicks on the button. | |  |
|  | By default, one quantity of that item is added to the shopping cart. | |  |
|  | The user can view the name and price, and can proceed to checkout. | |  |
|  | The use case ends. | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In steps 1,2,3,4 the customer can cancel the transaction and go directly to step 6. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 3, if a product from a different store/brand is already present in the shopping cart, then an error message is displayed restricting to buying items from one store at a time and execution proceeds to step 6. | |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | 20 | |
| **Purpose** | | The customer can alter the quantity of the product before checkout. | |
| **Pre-conditions** | | The Add to cart use case has been initiated. | |
| **Post-conditions** | | The desired quantity, and the calculated new price is visible in your shopping cart. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The customer clicks the + button to add one more item and – to subtract one. | |  |
|  | The customer repeatedly presses + and – to get to the desired quantity. | |  |
| 3. | The system displayed the desired quantity. | |  |
| 4. | The user can press the add to cart button. [Why would the user press the “add” button now, + button is serving the same purpose?] | |  |
| 5. | The system displays the user the quantity and the collective/total price. | |  |
| 6. | The user is allowed to proceed to checkout. [This does not look like a step/action performed by the user. The description should be clear like *the user performed certain action*.] | |  |
| 7. | The use case ends. | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In step 1 if the customer is good with the default quantity (i.e. “1”) the customer can directly proceed to step 3. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 1, a quantity less than 0 and greater than 25 cannot be selected. [What will happen in this case?] | |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | 21 | |
| **Purpose** | | The customer can view its shopping cart at any time. | |
| **Pre-conditions** | | NA. | |
| **Post-conditions** | | The customer can view items added to shopping cart that have not yet been proceeded to checkout. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | The shopping cart icon indicates the number of items in the cart by displaying a number alongside it in the navigation bar. | |  |
|  | The customer selects the shopping cart icon from the navigation bar. | |  |
|  | The system shows all the items that are in the shopping cart (including previous incomplete sessions as well) and proceed to checkout option. | |  |
| 4. | The use case ends. | |  |
|  |  | |  |
|  | | | |
| **Step #** | **Alternate Courses of Action** | |  |
|  | In steps 1, 2, and 3, the customer can return to the home page by pressing the home button in the navigation bar. | |  |
| **Step #** | **Exception Paths** | |  |
|  | In step 1, if there are no items in the cart, no number will be displayed with the shopping cart icon. | |  |
|  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Identifier** | | | 22 | | | |
| **Purpose** | | | This takes the items in the customers shopping cart and processes them for a purchase i.e. checkout. | | | |
| **Pre-conditions** | | | The add to cart, use promotion/discount use cases (in case of registered users only) implemented. | | | |
| **Post-conditions** | | | The cash is dispatched, and the amount is deducted from the selected account. | | | |
|  | | | | | | |
| **Step #** | **Typical Course of Action** | | | | |  |
|  | The customer selects the review payment and address button. | | | | |  |
|  | The system asks the mode of payment (cash on delivery) | | | | |  |
|  | The user selects the feasible option. | | | | |  |
|  | The system confirms the customers bio data and address information (location). | | | | |  |
|  | The user reviews its provided address information. | | | | |  |
|  | If its correct, the user can press the order button now. | | | | |  |
| 7. | The use case ends. | | | | |  |
|  |  | | | | |  |
|  | | | | | | |
| **Step #** | **Alternate Courses of Action** | | | | |  |
|  | In steps 1-5 the customer can cancel the transaction/ delete from cart and go directly to step 7. | | | | |  |
| **Step #** | **Exception Paths** | | | | |  |
|  | In step 6, if the displayed address is no longer used, the user can change its location (it has shifted somewhere else), it can update its address location on profile and resume step 5. | | | | |  |
| **Identifier** | | | 23 | | | |
| **Purpose** | | | The customer purchases the item.. | | | |
| **Pre-conditions** | | | The checkout use case is completed successfully. Acted on when the customer presses the finalize order button in checkout | | | |
| **Post-conditions** | | | This decrements the inventory of all items within the order, email the user, create a log of the transaction, and check stock to see if a reorder needs to take place. | | | |
|  | | | | | | |
| **Step #** | **Typical Course of Action** | | | | |  |
|  | The customer presses the order button. | | | | |  |
| 2. | The system sends a notification email to the customer displaying its order details. | | | | |  |
| 3. | The use case ends. | | | | |  |
|  |  | | | | |  |
|  | | | | | | |
| **Step #** | **Alternate Courses of Action** | | | | |  |
|  | N.A | | | | |  |
| **Step #** | **Exception Paths** | | | | |  |
|  | In step 2, if there is some discrepancy in the order then the customer can reply to that email pointing out the error. | | | | |  |
| **Identifier** | | | 24 | | | |
| **Purpose** | | | The customer uses the promotion/discount code. If the user is a member they are presented with the option to enter in a promotion code that will take off a percentage from the total. | | | |
| **Pre-conditions** | | | The customer must be registered/ have an account/ completed the login use case. | | | |
| **Post-conditions** | | | The customer can add the discounted item and proceed to payment. | | | |
|  | | | | | | |
| **Step #** | **Typical Course of Action** | | | | |  |
|  | The customer selects the discounted item. | | | | |  |
|  | The system shows the percentage of discount. | | | | |  |
|  | If the user is a member, it adds the item to cart and proceed to checkout. | | | | |  |
|  | The user can view the discounted price after the discount is subtracted from the original price. | | | | |  |
| 5. | The use case ends. | | | | |  |
|  |  | | | | |  |
|  | | | | | | |
| **Step #** | **Alternate Courses of Action** | | | | |  |
|  | In steps 3, 5, and 7, the customer can cancel the transaction and go directly to step 9. | | | | |  |
| **Step #** | **Exception Paths** | | | | |  |
|  | In step 3, if it’s not a registered user, it will be directed to sign up page and step 5 will resume directly. | | | | |  |
| **Identifier** | | | 25 | | | |
| **Purpose** | | | The customer can give reviews when they receive the order. | | | |
| **Pre-conditions** | | | The checkout and purchase use cases have been implemented. | | | |
| **Post-conditions** | | | Anyone who searches the reviewed item can view this review/comment. | | | |
|  | | | | | | |
| **Step #** | **Typical Course of Action** | | | | |  |
|  | The customer goes to purchase history. | | | | |  |
|  | The customer writes its comments in the box. | | | | |  |
|  | The customer presses the submit review button. | | | | |  |
|  | The system confirms that a response has been submitted if the customer actually bought that item. | | | | |  |
| 5. | The use case ends. | | | | |  |
|  |  | | | | |  |
|  | | | | | | |
| **Step #** | **Alternate Courses of Action** | | | | |  |
|  | In steps 1,2,3,4 the customer can press the home button to go directly to step 5. | | | | |  |
| **Step #** | **Exception Paths** | | | | |  |
|  | In step 4, if customer di not buy, then an error message is displayed and execution proceeds to step 5. | | | | |  |
| **Identifier** | | | <unique id of the use case, e.g. UC-001 > | | | |
| **Purpose** | | | The customer requests cash and the ATM dispenses the cash. | | | |
| **Pre-conditions** | | | The *Validate PIN* use case is completed successfully. | | | |
| **Post-conditions** | | | The cash is dispatched, and the amount is deducted from the selected account. | | | |
|  | | | | | | |
| **Step #** | **Typical Course of Action** | | | | |  |
|  | The customer selects the withdrawal menu option. | | | | |  |
|  | The ATM asks the customer for the account from which to withdraw the cash. | | | | |  |
|  | The user selects the account. | | | | |  |
|  | The ATM asks the user for a cash amount. | | | | |  |
|  | The user enters the amount. | | | | |  |
|  | The ATM asks the user to confirm the amount. | | | | |  |
|  | The user confirms the amount. | | | | |  |
|  | If there are sufficient funds in the account, the money is dispensed, and the amount is withdrawn from the account. | | | | |  |
|  | The use case ends. | | | | |  |
|  |  | | | | |  |
|  | | | | | | |
| **Step #** | **Alternate Courses of Action** | | | | |  |
|  | In steps 3, 5, and 7, the customer can cancel the transaction and go directly to step 9. | | | | |  |
| **Step #** | **Exception Paths** | | | | |  |
|  | In step 8, if there are not sufficient funds, then an error message is displayed and execution proceeds to step 9. | | | | |  |
|  |  | |  | | |  |
| **Identifier** | | | | 26 | |
| **Purpose** | | | | The Admin can sign in to the admin portal for special privileges. | |
| **Pre-conditions** | | | | User should be on the admin login portal. | |
| **Post-conditions** | | | | The Admin is signed in and on sees the admin portal. | |
|  | | | | | |
| **Step #** | | **Typical Course of Action** | | |  |
|  | | Admin enters login credentials | | |  |
|  | | Taps on the login button | | |  |
| 3. | | System validates and verifies login credentials with database. | | |  |
| 4. | | If login credentials valid, system redirects to admin portal. | | |  |
| **Step #** | | **Alternate Courses of Action** | | |  |
| 4. | | If login credentials invalid, system refreshes page and prompts invalid credentials. | | |  |
|  | |  | | |  |
|  | |  | |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | 27 | |
| **Purpose** | | The Admin can view notable merchants on a dashboard. | |
| **Pre-conditions** | | User should be on the admin portal. | |
| **Post-conditions** | | The Admin is on the merchant dashboard | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
|  | Admin taps on the merchant dashboard button | |  |
|  | Admin is redirected to the merchant dashboard by the system. | |  |
| 3. | System sorts merchants by different indexes like best rated, highest grossing etc and displays top five of each index. | |  |
|  |  | |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | 28 | |
| **Purpose** | | Admin can temporarily disable a merchant | |
| **Pre-conditions** | | User should be on the admin portal. | |
| **Post-conditions** | | The specific merchant is disabled and customers will not see the merchant or their listings. | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | User searches for the merchant on merchant dashboard | |  |
| 2. | System shows relevant matches | |  |
| 3. | User taps on merchant name. | |  |
| 4. | System redirects to merchant profile and shows an option to disable merchant | |  |
| 5. | User taps on the disable merchant option. | |  |
| 6. | System changes the status of the merchant to disabled. | |  |
| 7. | System refreshes the view. | |  |
| **Step #** | **Alternate Courses of Action** | |  |
| 1 | User can tap merchant name from the admin dashboard directly or from any product listing | |  |
| **Step#** | **Exception Paths** | |  |
| 4. | If the merchant is already disabled, system will show an option to enable. | |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | 29 | |
| **Purpose** | | Admin can remove a listing | |
| **Pre-conditions** | | User should be logged in as admin and on a product listing page | |
| **Post-conditions** | | The product listing is removed | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | User taps on remove listing option | |  |
| 2. | System prompts “are you sure you want to remove?” message | |  |
| 3. | User taps on “yes” option | |  |
| 4. | System removes listing from database | |  |
| 5. | System redirects user to admin portal | |  |
| **Step #** | **Alternate Courses of Action** | |  |
| 3 | User taps “no” then the last page is displayed | |  |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Identifier** | | 30 | |
| **Purpose** | | Admin can ban a customer | |
| **Pre-conditions** | | User should logged in as admin and on the customer dashboard | |
| **Post-conditions** | | The customer ID is banned from logging in and put in a | |
|  | | | |
| **Step #** | **Typical Course of Action** | |  |
| 1. | User searches for the Customer by ID | |  |
| 2. | System shows relevant matches | |  |
| 3. | User taps on Customer name. | |  |
| 4. | System redirects to Customer profile and shows an option to Ban Customer | |  |
| 5. | User taps on the disable merchant option. | |  |
| 6. | System deletes the profile of the customer and puts their email ID in a blacklist | |  |
| 7. | System refreshes the view. | |  |
| **Step #** | **Alternate Courses of Action** | |  |
| 1 | User can tap Customer name from anywhere to see the profile. | |  |
|  |  |  |  |

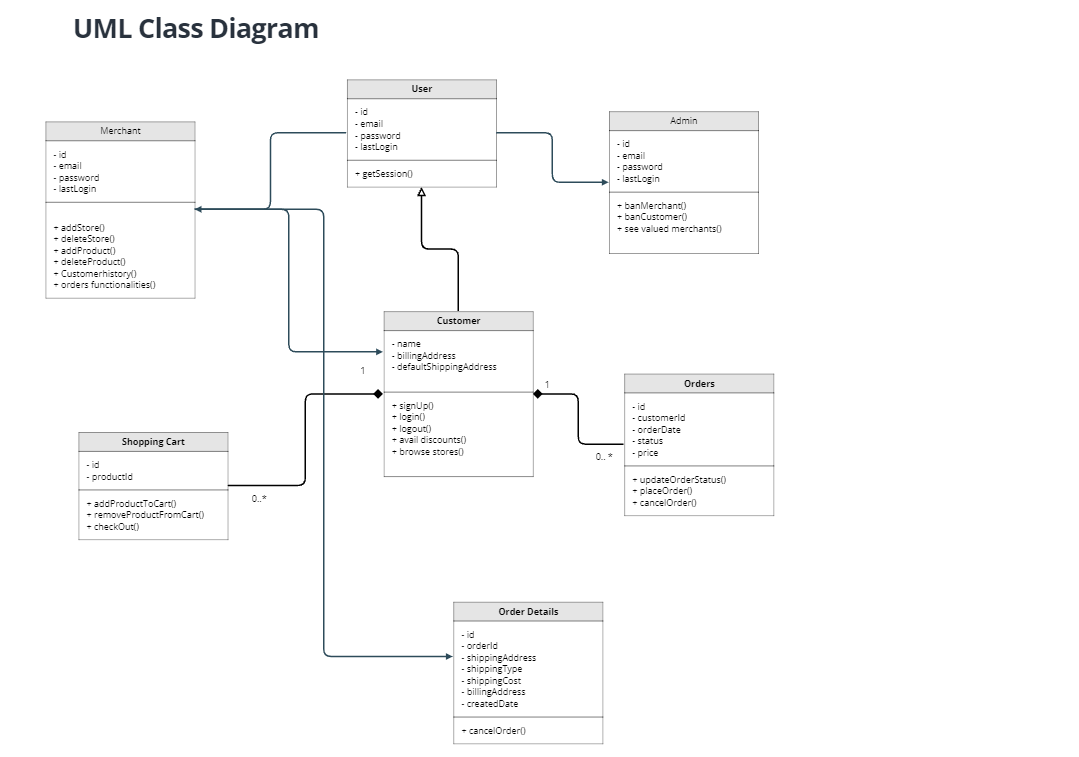
# Class Diagram

## Diagram

## A customer inherits from the user but not merchant and admin, why?

* What is the relationship between user and admin, user and merchant?
* Missing classes for stores, products, payments, items, uer roles.

<Use standard UML notation>



The diagram can be viewed clearly here: <https://app.moqups.com/I5TotF6tHt/view/page/aa458c046>

## Description

<Give brief description/purpose of each class in the class diagram. Give readable names to classes, attributes and operations.>

Classes include that of customer, admin and merchant with details of their roles.

# Sequence Diagrams

[Complete this section.]

<Create sequence diagram for each use case.>

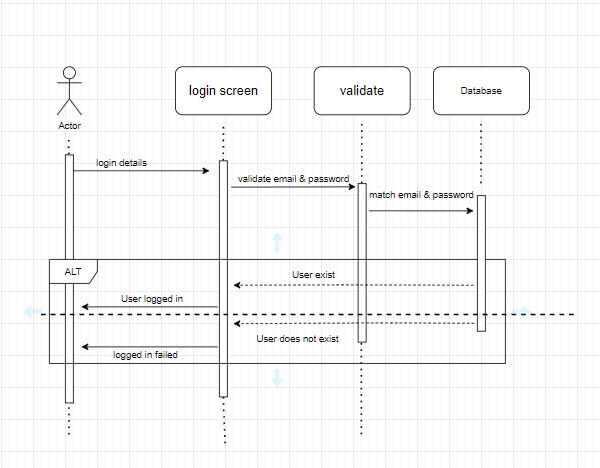
## Use case Name e.g., Withdraw cash

<Draw the sequence diagram of the use case using standard UML notation>

## Use case Name e.g., Transfer funds

<Draw the sequence diagram of the use case using standard UML notation>

Sequence Diagram for login of a valid user on the website.



# State Diagrams

<Repeat the following if you need to draw state diagrams of multiple objects>

## Diagram details

<Name of the object whose state is being shown along with details of states and messages.>

## Diagram

<Use standard UML notation>

…..

# Non-functional Requirements / Quality Attributes

|  |  |
| --- | --- |
| **Sr#** | **Requirements** |
| 1 | All transactions by any actor should not take more than 5s to be processed on a 2Mbps internet connection. |
| 2 | Maximum error rate in which an error might occur during a transaction will be low and on average not more than 1 error every 500 requests. |
| 3 | Ability to support 50,000 Customers and 200 merchants with the option to scale with time. |
| 4 | Throughput (transactions at a time) will be 200 users concurrently and the software will process 1500 requests on average in one day. |
| 5 | Scrolling on page will take at most 3s. |
| 6 | To secure user accounts, a Hash-based password login setup would be implemented. |
| 7 | Users can change password in case they forget it through an email sent to their registered email address. |
| 8 | Personal information of the users would be protected using encryption protocols. Passwords of users will be hashed before storing in the database. |
| 9 | Users will have the ability to stay logged in to the app. |
| 10 | In terms of usability, our interface will be user-friendly and easy to learn.  There will be a step-by-step registration process.  The icons used will be the ones most commonly used in some of the popular market places such as Daraz, Shopify, Alibaba.  Products display will be similar to Daraz. |
| 11 | The app should be available on google play store. |
| 12 | The app should run on android OS. |
| 13 | The uptime of the system will be at least 90%. |
| 14 | The app should support file/image uploads of up to 100 MB in size for a single seller for a particular product. |
| 15 | All functionalities will be supported by touch input methods. |
| 16 | The system should be hosted on a web service that can support a twofold increase in user interaction each year for the next five years without degradation in performance. The initial system will be able to support 50,000 customers and 200 merchants. |
| 17 | The hosting service must also support a twofold increase in the database volume each year for the next five years. Initial system would be able to store 5tb of data. |

# Who Did What?

|  |  |
| --- | --- |
| **Name of the Team Member** | **Tasks done** |
| Muzamil Khan | Actors, Admin Use Cases. |
| Walilullah Aitemad | Admin use cases, Introduction, Non-functional requirements |
| Daniyal Mumtaz | Actors, Customer Use Cases. |
| Abdur Rehman Masood | Actors, Customer Use Cases, and Diagram. |
| Waqar Ul Haq Khatana | Actors, Merchant Use Cases, and Use Case Diagram. |

# Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

|  |  |
| --- | --- |
| **Section** **Title** | **Reviewer Name(s)** |
| Use Cases Review | Abdur Rehman Masood |
| Use Cases Review | Daniyal Mumtaz |
| Diagrams and Use Cases | Waqar ul Haq Khatana |
| Diagrams, Use Cases, Non Functional Requirements. | Waliullah Aitemad |
| Use Cases. | Muhammad Muzammil Khan |