Project Title Here

CS262- Design Document



Project Supervisor

Mr. Samyan Qayyum Wahla

Project Members(Group Id Here)

|  |  |
| --- | --- |
| Member 1 Name | Member 1 Registration No. |
| Member 2 Name | Member 2 Registration No. |

Department of Computer Science

University of Engineering and Technology, Lahore

Pakistan

Table of Contents

[Project Description: 3](#_Toc119621420)

[Project Features: 3](#_Toc119621421)

[Technology Stack: 3](#_Toc119621422)

[Project Actors: 3](#_Toc119621423)

[Use Cases: 3](#_Toc119621424)

[Use Case 1(Write the name of the use case): 3](#_Toc119621425)

[User Interface Details 4](#_Toc119621426)

[Classes: 5](#_Toc119621427)

[Object Oriented Features: 5](#_Toc119621428)

[Composition: 5](#_Toc119621429)

[Inheritance: 5](#_Toc119621430)

[Multiple Inheritance: 5](#_Toc119621431)

[Multi-Level Inheritance: 5](#_Toc119621432)

[Polymorphism: 5](#_Toc119621433)

[Detailed Object Oriented Design: 5](#_Toc119621434)

[Data Structure: 6](#_Toc119621435)

[Exceptions: 6](#_Toc119621436)

[Data Storage: 6](#_Toc119621437)

[Email Sending: 6](#_Toc119621438)

[Project Plan 6](#_Toc119621439)

# Project Description:

The system is designed for a company that provides:

Logistics (delivery of products to its client), product management (crud operations), and effective communication with their employee and clients.

The company has its office, warehouse, and rider.

It has a different contract with multiple firms to take the shipment and store it in dedicated warehouses. The rider will take orders from the shopkeeper. Their order is received at the office, and the office will create the feasibility report according to their client's needs and instructions generated for their warehouse manager to fulfill their order. The available rider will receive an email about their order. The office will send a confirmation email to their client.

There are a total of four actors in the system and one stakeholder. Their name and role are:

* CEO: The owner of the company could manage all the operations.
* Employee: They directly report to CEO and help in company operations.
* Warehouse Manager: Ready the shipment for the rider and managed other expenses.
* Rider: Received the order detail and delivered the product to pre-subscribed routes.

The stakeholders are:

* Shopkeepers: The rider will take the order from the shop owner and deliver it. The rider will receive all orders and payments.

All the actors will be able to create their accounts, and the system will give specific security codes to them. It helps to protect the system from security breaches.

There first dedicated dashboard for the owner where they monitor all operations. The operations manage their employees, products, and expenses and send emails. The CEO is the only person in the system with access to all operations. CEOs could analyze company operations, including the performance of their workers. The system will generate the company expenditure report.

The second dashboard is for the office employees directly contacting the CEO. They have access to manage emails, clients' orders, vendors' orders, and company expenses. The company's expenses are the CEO, rider, and warehouse salaries. The payment of the vendor and clients. An employee will enter all the shipments that the company receives their record. They add the product name, SKU number, weight, volume, cost price, manufacturer, and many more to identify confirmed products.

The third dashboard is for the warehouse manager, who receives feasibility reports of office employees and readies the order for the rider. The warehouse manager must record the labor used in preparing the order. It could provide the miscellaneous expenses of the warehouse, like electricity costs, etc. They can view the product and make suitable changes according to the requirements.

The fourth dashboard is for a rider who takes orders from the warehouse and delivery them to the company client. The system will provide the routes for the destination with the order detail. The rider received a specific amount of fuel to perform the operations. The prescribed fuel is calculated according to the formula. They can see all the products. The product will be sorted in order, like assessing and descending. Search for a specific product from a wide range of available products. The system will deploy different sharp algorithms to access the desired date quickly. Able to place the order and view the detail of the order as well. The order is placed according to the stakeholder shopkeeper's needs.

The system will provide the report to the CEO according to the performance of their worker, expenditure, and profit.

Like how many products are received in the warehouse, how many products are left, how many products are delivered to company clients, how many riders have done shipments, which rider performs most shipments, and which rider needs to perform better. It also includes how many orders a shopkeeper placed and whether the company received the payment.

The email notification mechanism is embedded in the system, which helps the company communicate within and outside with other vendors and clients. The internal communication will send the order details to the warehouse manager to prepare the shipment for the rider. The rider also received the email for the delivery of the order. The employee emails the CEO for any need of assistance with an issue. The warehouse manager and rider also mail to the company office for any assistance. In external communication, the client will receive a confirmation email from the system about their order. They also take assistance from the company with any issue.

The flow of new users will be like this. A potential user provides the required detail for account creation, and an employee or CEO selects their role in the company. A unique username is allowed to login into the system. Account successfully creation the, they create the security code to use the system according to their role.

All the data is stored in an effective data structure to extract the data according to the need of the system actor and stakeholder.

# Project Features:

1. CEO are able to manage employee, warehouse manager, rider and shopkeeper.
2. CEO and Employee manage product related operations.
3. CEO will be able to analyze company operations.
4. Warehouse manager ready the shipment for rider.
5. Rider delivered the shipment to their shopkeeper.
6. Rider are able to selected the shortest route to reach the destination.
7. One user is able to notify other user through email.
8. Shopkeeper are able to view products and place order.
9. Dedicated security password for each user.
10. Company expenditure report will be generated.

In this section, write down the project features clearly. These are the features that will be available to the end user of your project directly. You should be able to describe, how to write the features of the project.

# Technology Stack:

|  |  |
| --- | --- |
| Language | C# (.net framework 4.8) |
| IDEs | Microsoft Visual Studio 2022 |

# Project Actors:

* **CEO:** The owner of company could manage all the operations
* **Employee:** They directly reported to CEO and help in company operations.
* **Warehouse Manager:** Ready the shipment for rider and manage other expenses.
* **Rider:** Received the detail of order and delivered the product to presubscribed routes.

**Stakeholders:**

* **Shopkeeper:** Are able to outlook products, place order and check detail.

# 

# Use Cases:

All the use cases should be written in the following format. The name of use should be start with a verb e.g. Add Student. It should not be as Student/ Student Add. Add Student is separate use case and Edit Student is separate use case: (Reference Document for use cases: <https://www.craiglarman.com/wiki/downloads/applying_uml/larman-ch6-applying-evolutionary-use-cases.pdf> )

## Use Case 1(Log In):

|  |  |
| --- | --- |
| Use Case ID | U02 |
| Name | Login Screen |
| Actor | CEO, Employee, Rider, Warehouse Manager |
| Description | Respected user will login to this system by providing valid role, username and password. After clicking on Login Button, System will check for its validity if this user is already exist on database. After successful validation, respected panel will show them. |
| UI Interface in JUSTINMIND | C:\Users\AA\AppData\Local\Microsoft\Windows\INetCache\Content.Word\UI-002 Login@1x.png |
| Pre-Condition | Respective User Will initiate this system. |
| Flow | Main Scenario:   1. By Default Login Screen will appear first 2. Select Your Role from combo box then. 3. Provide your username you chose while signing up. 4. Then enter password for your account 5. Then Click on Login Button   Alternate Flow:   * If forgot password button is clicked  1. Interface I02 will get open  * If user provides invalid username and password  1. Message box with a message of invalid input will be displayed 2. System remains on the same page  * If user provides unregistered information  1. Message Box with a message of “User Not Found” will get displayed  * If user don’t provide all required information  1. Alert Message will be displayed.  * If Exit Button is Clicked |
| Post-Condition | User will successfully login to this system. |

## Use Case 2(Forgot Password):

|  |  |
| --- | --- |
| Use Case ID | U02 |
| Name | Forgot Password |
| Actor | CEO, Employee, Rider, Warehouse Manager |
| Description | Already registered user can change his/her password. Provide Role, username and your new password. Also confirm your password then after completion your password will be changed. |
| UI Interface in JUSTINMIND | C:\Users\AA\AppData\Local\Microsoft\Windows\INetCache\Content.Word\UI-003 Forgot password@1x.png |
| Pre-Condition | User Must be registered in the system. |
| Flow | Main Scenario:   1. Click on Forgot password Button on Log In Screen. i.e., U02 2. Select Your Role from combo box then. 3. Provide your username you chose while signing up. 4. Then enter password for your account 5. Again type for confirmation 6. Then Click on Update Button   Alternate Flow:   * If user provides his previous password  1. He/she will get an error message 2. System return to same page  * If user don’t provide all required information  1. Alert Message will be displayed. 2. System remain to same page  * If password is not of length equal to or greater than 8  1. Password will not be changed  * If password text and confirm password text not matched  1. Again error message will be displayed |
| Post-Condition | Password will successfully changed |

## Use Case 3(CEO Dashboard):

|  |  |
| --- | --- |
| Use Case ID | U04 |
| Name | CEO Dashboard |
| Actor | CEO |
| Description | After Successful Logging in to this system by CEO, CEO dashboard will be shown. CEO is the main handler of our system. This interface provides CEO to perform multiple CRUD operations. He has all the authority to add, search, update and delete any information related to all actors and other entities. He is able to view and generate reports. He can send emails as well. |
| UI Interface in JUSTINMIND | E:\Final Term Project\IMG\UI-004 CEO Dashboard@1x.png |
| Pre-Condition | CEO must have an account and logged in to this system |
| Flow | After Successful logging by a role of CEO, CEO Dashboard will appear  He can now has all the authorities and can do multiple operations |

## Use Case 4(Account Details):

|  |  |
| --- | --- |
| Use Case ID | U05 |
| Name | Accounts Detail |
| Actor | CEO, Employee |
| Description | Respective Actors can view all the details of each user who have registered themselves on this system. They can filter out users with certain designation, certain attribute and can apply multiple filters to search out specific users. This provides ability to edit and delete (only CEO) selected users. |
| UI Interface in JUSTINMIND | D:\Downloads\UI-006 View%2FDelete Account@1x.png |
| Pre-Condition | Select Account Details from Option Menu Strip on Top Left  All users detail will automatically be displayed |
| Flow | Main Scenario:   1. Select Your desired Row for required operations 2. Click Delete Button at the bottom of the screen if you want to delete selected row respectively. 3. Clicking Edit Button will open a new interface I05 with selected row data.   Alternate Flow:   * If text filters are selected and word text area is empty  1. An Alert message of Empty input is entered  * If delete button is pressed without selecting any row data  1. Exception will be handled and an exception error will be displayed to the user 2. User will remain on the same page  * If Close Button is Clicked  1. CEO or Employee Dashboard will again open |
| Post-Condition | Data from selected row will be deleted or updated |

## Use Case 5(Update Account):

|  |  |
| --- | --- |
| Use Case ID | U06 |
| Name | Accounts Detail |
| Actor | CEO, Employee |
| Description | Respective Actors can view all the details of each user who have registered themselves on this system in Account Details Screen. You can edit your desired row data by providing modified information related to that user. |
| UI Interface in JUSTINMIND | D:\Downloads\UI-005 Update Account@1x.png |
| Pre-Condition | Desired Row Must be selected  Click Edit Button |
| Flow | Main Scenario:   1. Designation is already selected and non-editable 2. Already Available Data will be shown in their respective areas 3. Modify data entity of your choice 4. Click Update Account Button   Alternate Flow:   * If user don’t provide all required information  1. Alert Message will be displayed. 2. System remain to same page  * If password is not of length equal to or greater than 8  1. Password will not be changed  * If password text and confirm password text not matched  1. Again error message will be displayed |
| Post-Condition | Selected Data will get modified |

## Use Case 6(Add Products):

|  |  |
| --- | --- |
| Use Case ID | U07 |
| Name | Add Products |
| Actor | CEO, Employee |
| Description | These two concerned actors will add products in for their warehouses. They have to provide all details of the bought products and all the products will get entered in the system records. |
| UI Interface in JUSTINMIND | D:\Downloads\UI-008 Add Product@1x.png |
| Pre-Condition | Click on Add product Button on CEO or Employee Dashboard |
| Flow | Main Scenario:   1. Add Product Form will be displayed 2. It will ask the users for all details of products 3. Click on Add Button to add product details   Alternate Flow:   * If Reset is Clicked  1. All text and input areas will get clear 2. Control will remain on the same page  * If Close is Clicked  1. Form will be closed 2. CEO and Employee Dashboard will again get displayed  * If not necessary conditions are provided  1. Error Message will be displayed with a message |
| Post-Condition | Product will be added to system and will be sent to the selected Warehouse |

## Use Case 7(Products Details):

|  |  |
| --- | --- |
| Use Case ID | U08 |
| Name | Products Details |
| Actor | CEO, Employee |
| Description | This form allows the respective users to view products and perform certain operations on data. Certain operations include search with filters and edit and delete options (only for CEO) as well. |
| UI Interface in JUSTINMIND | D:\Downloads\UI-010 View%2FDelete Products@1x.png |
| Pre-Condition | Click on Option on Top Left Corner  Select products details |
| Flow | Main Scenario:   1. On Clicking product details, product forms will be displayed 2. Select Row from the data and click delete to delete and edit to edit selected item   Alternate Flow:   * If text filters are selected and word text area is empty  1. An Alert message of Empty input is entered  * If delete button is pressed without selecting any row data  1. Exception will be handled and an exception error will be displayed to the user 2. User will remain on the same page  * If Close Button is Clicked  1. CEO or Employee Dashboard will again open  * If Clear Button is Clicked  1. All filters will get cleared |
| Post-Condition | Selected Row Data will get updated or deleted |

## Use Case 8(Update Products):

|  |  |
| --- | --- |
| Use Case ID | U09 |
| Name | Update Products |
| Actor | CEO, Employee |
| Description | This form allows the respective users to edit products. Select the product from the data. Click Update in the I09. This form will get open. All previous product details will also be there in its respective fields. Modify Details according to your need. |
| UI Interface in JUSTINMIND | D:\Downloads\UI-009 Update Product@1x.png |
| Pre-Condition | Click on Update Button in I09 |
| Flow | Main Scenario:   1. Clicking on Update Button will open this screen 2. Modify Details and Fill all areas 3. Click Update Button   Alternate Flow:   * If user don’t provide all required information  1. Alert Message will be displayed. 2. System remain to same page  * Clicking Reset  1. Clear the fields  * Clicking Close Button  1. Close the Form 2. Return to Products Details UI |
| Post-Condition | Selected Data (Product) will get modified |

## Use Case 9(Add Rider):

|  |  |
| --- | --- |
| Use Case ID | U10 |
| Name | Add Rider |
| Actor | CEO, Employee |
| Description | Irrespective of self-signup by rider himself, CEO and Employee can add rider details into their system. They have to provide his personal information and the information related to his vehicle. Rider himself will tell his details to Employee or his details will be provided to the CEO. |
| UI Interface in JUSTINMIND | E:\Final Term Project\IMG\AddRider@1x.png |
| Pre-Condition | Click on Add Rider from CEO or Employee Dashboard |
| Flow | Main Scenario:   1. Designation will be selected automatically 2. Provide All Necessary personal information 3. Also Add Information Related to vehicle 4. Click on Add Button |
| Post-Condition | Rider Information will be Added to your system |

## Use Case 10(Update Rider):

|  |  |
| --- | --- |
| Use Case ID | U11 |
| Name | Update Rider |
| Actor | CEO, Employee, Rider |
| Description | Selecting Riders Details and then Clicking Update button after selecting row, will allow you to modify selected Riders Details. Change your required fields and Save the information |
| UI Interface in JUSTINMIND | E:\Final Term Project\IMG\EditRider@1x.png |
| Pre-Condition | Click on Update in Rider Details after selecting rider |
| Flow | Main Scenario:   1. Designation will be selected automatically 2. Provide All Necessary personal information 3. Also Add Information Related to vehicle 4. Click on Add Button   Alternate Flow:   * Clicking Close Button  1. Return to Riders Details |
| Post-Condition | Rider Information will be modified and will get save to your system |

## Use Case 11(Order Product):

|  |  |
| --- | --- |
| Use Case ID | U12 |
| Name | Order Product |
| Actor | Rider |
| Description | Rider will approach the shopkeepers, take orders from them, recommend them some products and handed over the order to Employee in the company. Shopkeeper can be shown the summary of products he asked to buy |
| UI Interface in JUSTINMIND | D:\Downloads\UI-011 Order Product@1x.pngD:\Downloads\UI-012 Order Summery@1x.png |
| Pre-Condition | Click on Order Product in the Rider Dashboard |
| Flow | Main Scenario:   1. At first all available products will be shown in the grid list 2. Rider will ask shopkeeper for the order 3. Rider will add the products by searching them and providing quantity 4. Select the product and click Add to Cart 5. After adding all the required products to cart, click Buy Now. 6. Order Summary Screen will open 7. Provide Necessary Details and Click Buy   Alternate Flow:   * Clicking Close Button  1. Return to Riders Dashboard  * Clicking the Reset will reset the Cart |
| Post-Condition | Employee will get notified and order will be added to the system |

## Use Case 12(Send Email):

|  |  |
| --- | --- |
| Use Case ID | U13 |
| Name | Send Email |
| Actor | Rider, CEO, Warehouse Manager, Employee |
| Description | To communicate with each other and notify about each activity, each user can contact with one another through email. |
| UI Interface in JUSTINMIND | D:\Downloads\UI-013 SendEmail@1x.png |
| Pre-Condition | You must be a user of this system before |
| Flow | Main Scenario:   1. Click on Send Email from your dashboard 2. Type To: 3. Type From: 4. Type Subject: 5. Type Email 6. Click Send   Alternate Flow:   * Clicking Close Button  1. Return to Respective Dashboard  * Clicking the Reset will reset the Email |
| Post-Condition | Email will be send to the person specified in To Section |

## Use Case 13(New Warehouse):

|  |  |
| --- | --- |
| Use Case ID | U14 |
| Name | New Warehouse |
| Actor | CEO, Employee |
| Description | CEO and Employee Dashboard have an option to add a new warehouse for a company. Provide Necessary details like warehouse volume, capacity, etc and manager name as well. |
| UI Interface in JUSTINMIND | D:\Downloads\UI-014 AddWarehouse@1x.png |
| Pre-Condition | Click on Add Warehouse in Employee and CEO Dashboard |
| Flow | Main Scenario:   1. New Warehouse UI screen will be displayed 2. Provide details of a new warehouse 3. Assign a New manager 4. Click on Add   Alternate Flow:   * Clicking Close Button  1. Return to Respective Dashboard  * Clicking the Clear will reset the fields |
| Post-Condition | A new Warehouse will be added to the office |

## Use Case 14(View/Delete Warehouse):

|  |  |
| --- | --- |
| Use Case ID | U15 |
| Name | View/Delete Warehouse |
| Actor | CEO, Employee |
| Description | This form allows you to view all the warehouse with their details. Only CEO has the authority to delete warehouses. The concerned user can also filter out the warehouse details. |
| UI Interface in JUSTINMIND | D:\Downloads\UI-016 View%2FDelete Warehouse@1x.png |
| Pre-Condition | Click on Option in CEO or Employee Dashboard Click Warehouse details |
| Flow | Main Scenario:   1. Warehouse Details UI will appear 2. View Details 3. For deletion, select data row and click delete from below.   Alternate Flow:   * Clicking Edit Button  1. Will open a Edit Warehouse Form  * Clicking Clear Button  1. Will reset the filters |
| Post-Condition | Detailed Warehouse data will be displayed or deletion occurs if any |

## Use Case 15(Edit Warehouse):

|  |  |
| --- | --- |
| Use Case ID | U16 |
| Name | Edit Warehouse |
| Actor | CEO, Employee |
| Description | Selecting Warehouse Details and then Clicking Update button after selecting row, will allow you to modify selected Warehouse Details. Change your required fields and Save the information |
| UI Interface in JUSTINMIND | D:\Downloads\UI-015 EditWarehouse@1x.png |
| Pre-Condition | Click on Update in Warehouse details after selecting data |
| Flow | Main Scenario:   1. Update Warehouse UI will show 2. Modify already filled fields with your requirements 3. Click Update   Alternate Flow:   * Clicking Clear Button  1. Clear All fields  * Clicking Back Button  1. Back to Previous Warehouse Details |
| Post-Condition | Selected Warehouse details will be modified |

# Use Interfaces:

|  |  |
| --- | --- |
| Interface ID | I01 |
| Name | Intro |
| Linked Use Case | NULL |
| UI Screen (JustInMind) | D:\Downloads\UI-001 Intro@1x.png |
| Validators | Add here the name of validators that will be applied on this page |

|  |  |
| --- | --- |
| Interface ID | I02 |
| Name | Login |
| Linked Use Case | U01 |
| UI Screen (JustInMind) | D:\Downloads\UI-002 Login@1x.png |
| Validators | Add here the name of validators that will be applied on this page |

|  |  |
| --- | --- |
| Interface ID | I03 |
| Name | Forgot Password |
| Linked Use Case | U02 |
| UI Screen (JustInMind) | D:\Downloads\UI-003 Forgot password@1x.png |
| Validators | Add here the name of validators that will be applied on this page |

|  |  |
| --- | --- |
| Interface ID | I04 |
| Name | Create Account |
| Linked Use Case | U03 |
| UI Screen (JustInMind) | D:\Downloads\UI-004 Create Account@1x.png |
| Validators | Add here the name of validators that will be applied on this page |

|  |  |
| --- | --- |
| Interface ID | I05 |
| Name | Update Information |
| Linked Use Case | U04 |
| UI Screen (JustInMind) | D:\Downloads\UI-005 Update Account@1x (1).png |
| Validators | Add here the name of validators that will be applied on this page |

|  |  |
| --- | --- |
| Interface ID | I06 |
| Name | Account Detail |
| Linked Use Case | U05 |
| UI Screen (JustInMind) | D:\Downloads\UI-006 View%2FDelete Account@1x (1).png |
| Validators | Add here the name of validators that will be applied on this page |

|  |  |
| --- | --- |
| Interface ID | I07 |
| Name | CEO Dashboard |
| Linked Use Case | U06 |
| UI Screen (JustInMind) | D:\Downloads\UI-007 CEO Dashboard@1x.png |
| Validators | Add here the name of validators that will be applied on this page |

|  |  |
| --- | --- |
| Interface ID | I08 |
| Name | Add Product |
| Linked Use Case | U07 |
| UI Screen (JustInMind) | D:\Downloads\UI-008 Add Product@1x (2).png |
| Validators | Add here the name of validators that will be applied on this page |

# User Interface Details

In this section, fill the table for summary that which use case will have the required component. Inside each box, write the counts for each component. If component is not used, write zero.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Interface Id | TextBox | DropDown | Password Box | Table | Date Field | Buttons | AutoComplete | Radio Button | CheckBox | Menu | Text Area | ProgressBar |
| I01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| I02 |  |  |  |  |  |  |  |  |  |  |  |  |

# Classes:

In this section, we do not require detailed design diagram. But identify the tentative classes with the requirement Fill the following table for details. Note that class name should follow naming conventions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Class Name** | **Software/ Domain** | **Is Abstract (Yes/No)** | **Is Singleton (Yes/No)** | **Is the class will has parametrized constructor(Yes/No)** |
| CEO | Domain | No | Yes | Yes |
| Company | Domain | No | Yes | Yes |
| Office | Domain | No | Yes | Yes |
| WareHouse | Domain | No | No | Yes |
| User | Domain | No | No | Yes |
| Rider | Domain | No | No | Yes |
| Employee | Domain | No | No | Yes |
| WareHouse Manager | Domain | No | No | Yes |
| ShopOwner | Domain | No | No | Yes |
| Shop | Domain | No | No | Yes |
| Ledger | Domain | No | Yes | Yes |
| Order | Domain | No | No | Yes |
| Product | Domain | No | No | Yes |
| Vehicle | Domain | No | No | Yes |
| Stack | Software | No | No | Yes |
| Queue | Software | No | No | Yes |
| Linked List | Software | No | No | Yes |
| Binary Search Tree | Software | No | No | Yes |

# Object Oriented Features:

## Composition:

In our Project there are 8 places where we use Composition:

* Company Class has composition of Ledger Class
* Company Class has composition of Office Class
* Company Class has composition of Warehouse Class
* Company Class has composition of CEO Class
* Warehouse Class has composition of Warehouse Manager Class
* Rider Class has composition of Vehicle Class
* Office Class has composition of User Class ( Employee, Rider)
* Shop Owner Class has composition of Shop Class

## Inheritance:

In our project inheritance is used in following places:

* User inherits the class of CEO
* User inherits the class of Rider
* User inherits the class of Shopkeeper
* User inherits the class of Warehouse Manager

## Multi-Level Inheritance:

In our project Multilevel inheritance is used as:

* User class inherits the CEO class and CEO class inherits the Employee Class

## Aggregation

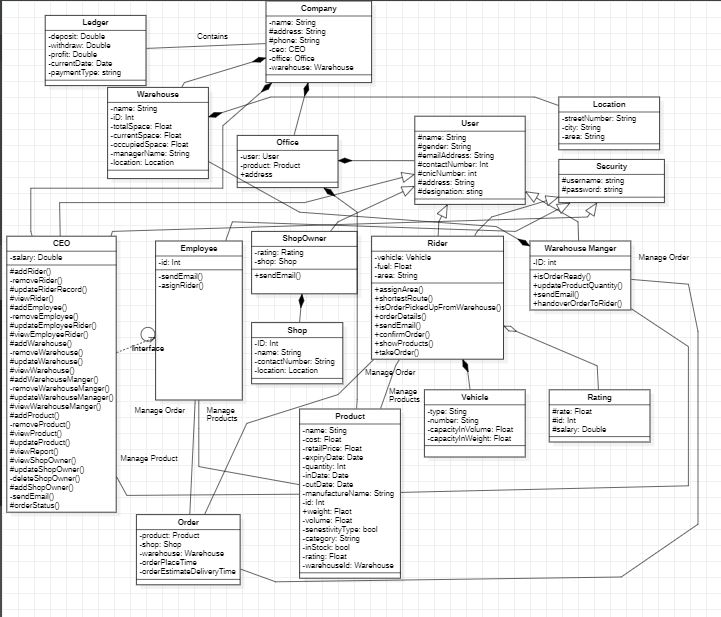
* Rider Aggregate the Rating Class in our project.

## Association:

* Warehouse Manager manages the order.
* CEO manages the products
* Rider take the order
* Rider adds the order
* Employee adds the products
* Employee manages the order

## UML Diagram:

## 



# Data Structure:

|  |  |  |
| --- | --- | --- |
| **Use Case ID** | **Data Structure Used** | **Justification for the use of data structure** |
| UI-002 | Linked List | In the UI-002 (LOGIN), search and compare the user from the list so when the user data is found it returns the action. |
| UI-004 | Linked List | In the UI-004(Create Account), Linked list is used to add user. |
| UI-005 | Linked List | In the UI-005 (Create Account), Update data of the user present in the Linked List |
| UI-006 | Linked List | In the UI-006 (Account Details), Grid of the added users shown lists where all the users are stored(added) |
| UI-008 | Linked List | In the UI-008(Add Product), Add the product data in the List . |
| UI-009 | Linked List | In the UI-009(Update Product), Update the product data in the list where the data of the products are added. |
| UI-0010 | Linked List | In the UI-0010(View Product), View the product data in the Grid that are stored in the list at the backend. |
| UI-0011 | Queue | In the UI-0011(Order Product), To place the order we use the mechanism of First in and First Out (first order item will be placed first) |
| UI-0012 | Queue | In the UI-0012(Order Summary), To place the order we use the mechanism of First in and First Out (first order item will be placed first)  So the show of data of order products in the grid also use queue. |
| UI-0013 | Stack | In the UI-0013(Email Product), To send the mail and view the mail (first send mail is shown in the last and the most recent one in the first) |

# Exceptions:

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Exception | Why this exception will occur | Use Case Id in which exception could be occurred | How you will handle the exception |
|  |  |  |  |
| Incorrect Format | By default system, take all input in string and the deploy system need to convert into desire format. If the input data is not converted into other datatype like int and float the future task not performed e.g. string 2 and int 2 behave different in CPU. | 008,026,021, | Restrict the user to enter the required data in correct format. |
| File not Exit | The required file not in the correct path and CPU not recognize it. | 016,006,012, | Restrict the user first select the file then perfrom future action |
| Incorrect URL | The wrong URL of the website broken the link with the DNS and required data not fetch from server. | 024 | Apply stick constrain to avoid it. |

# Data Storage:

1. **Mails (CSV):**

Columns data names are:

* 1. Employee and Rider
  2. Rider and Shopkeeper
  3. Warehouse Manager and Employee
  4. CEO and Employee

1. **Products (CSV):**

Columns data names are:

* 1. Name
  2. Cost
  3. Retail price
  4. Expiry Date
  5. Quantity
  6. In date
  7. Out date
  8. Manufacturer
  9. ID
  10. Weight
  11. Volume
  12. Category
  13. Sensitivity
  14. In stock
  15. Rating
  16. Warehouse Hold ID

1. **Users (CSV):**

Columns data names are:

1. Name
2. Gender
3. Email address
4. Contact Number
5. CNIC number
6. Address
7. Desigination

# Email Sending:

At first,

* Rider will email the employee about the order of shopkeeper
* Employee email warehouse manager to ready the shipment for rider
* Rider will email the Employee and Shopkeeper that he has picked the order and cc to CEO.

# Project Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case Id** | **Use Case Name** | **Member Name** | **Estimated Completion Date** |  |
| 001-008 |  | Syed Hashir | 10/12/2021 |  |
| 009-0018 |  | M.Hammad Hassan | 14/12/2021 |  |
| 019-0026 |  | M. Kabir Ahmed | 18/12/2021 |  |

# Analytical Reports

In our project, we can use the analytical reports:

* Salary Report
* Rider Capture Order Report
* Profit Report
* Saled Product