

Carrefour Multinational Corporation

Project Supervisor

Mr. Samyan Qayyum Wahla

Project Members(G-02)

|  |  |
| --- | --- |
| Nabeel Yousaf | 2020-CS-103 |
| Haider Ali | 2020-CS-137 |

Department of Computer Science

University of Engineering and Technology, Lahore

Pakistan

Table of Contents

[Project Description: 3](#_Toc71624246)

[Project Features: 4](#_Toc71624247)

[Technology Stack: 5](#_Toc71624248)

[Project Actors: 6](#_Toc71624249)

[Use Cases: 7](#_Toc71624250)

[Use Case 1(Write the name of the use case): 7](#_Toc71624251)

[User Interface Details 8](#_Toc71624252)

[Classes: 9](#_Toc71624253)

[Object Oriented Features: 10](#_Toc71624254)

[Composition: 10](#_Toc71624255)

[Inheritance: 10](#_Toc71624256)

[Multiple Inheritance: 10](#_Toc71624257)

[Multi-Level Inheritance: 10](#_Toc71624258)

[Polymorphism: 10](#_Toc71624259)

[Collections: 11](#_Toc71624260)

[Exceptions: 12](#_Toc71624261)

[Data Storage: 13](#_Toc71624262)

[Email Sending: 14](#_Toc71624263)

[Project Plan 15](#_Toc71624264)

# Project Description:

"**Carrefour**" has been a part of the English language since the 15th century. It once referred to an intersection of four roads at a single point, but later came to refer to any public square or plaza. The primary Carrefour shop (not a hypermarket) was opened in 1960, inside rural Annecy, near a crossroad. The gathering was made in 1958 by Marcel Fournier, Denis Defforey and Jacques Defforey, who attended and were influenced by several seminars in the United States led by "the Pope of retail" Bernardo Trujillo. The Carrefour bunch was the first in Europe to open a hypermarket, an enormous general store, and a retail chain under a similar rooftop. Finally the supermarket corporation is formed called **“Carrefour –French Multinational Corporation”**

The system is under control of a **CEO** which is acting as a head of the corporation. The CEO is basically the

Owner of the whole system and he is responsible of adding the corporation managers and review the decision made by managers. A corporation manager is people who is responsible of all the functionalities of the corporation like registering clients allotting them the space and apply sales decide GST and land tax and rent of shop. So in other words he is the main active member of the system the next member is counter holder a counter holder is haired by the manager and his biggest responsibility is to calculate bill of the customer. The next actor of the system is client which invests and buy a shop in corporation and sale products he may belongs to a organization or be a standalone buyer he is allotted with an account and he may pays rent for his place. The last actor of the system staff these are also haired by manager and are responsible of sale products and do office tasks the salesmen are also another entity belongs to staff the client may take one or many salesmen according to the needs of his system but not free he needs to pay the salary and flounces to that salesman.

Now lets discuss the cycle of the system the system is leaded by CEO first the CEO adds corporation manager then the corporation manager has all the authority to register the clients the manager also responsible of allotment of space to the client. First of all the client send a proposal to take place on rent he needs to provide some basic information and the manager then on the bases of the information and interview selects or rejects the requests the list of all the selected and rejected requests are then sent to CEO the CEO the review them if some of them is un appropriate then he have the authority to forcedly reject it and he send mail to manager about the rejection cause. Only when the request is approved from both the manager and CEO the clients gets the confirmation email if the client receives the mail he also have the ability to cancel his proposal but he needs to send mail to manager about the cause and basically it happens when place or the rent is not appropriate so once the requests are accepted from both ends the manager needs to provide the client an id and password of his account just like the CEO assigns the manager so through this account the manager and client are linked they can send mails the client can print his challans and receipts more over the client is capable of keep the record of the goods and sales he may add products and view the amount of sale yet the client is restricted to pay rent gst and lands tax all the amount is uploaded on his account he may pay online or print it and submit to bank the client is capable to increase the amount and apply sales on products the manager also supposed to apply sales on specific days the more the client sale products the more he supposed to pay gst the gst is calculated on per item sale a specific amount of tax is paid to organization the bill is calculated on sales point where an employee opens up his account and start calculating the bill the bill is paid through cash or debit card each shop is assigned a unique shop id and the product belongs to that id then the price is added to that particular shop owner and the sales tax is added to company account. The client have option to withdraw his sales amount. The client can cancel his membership anytime but he needs to send an email to the manager before cancelling and manager also have ability to communicate with client before approving the cancellation request. The cancellation request is rejected if the tie is less than the contracted time.

# Project Features:

In this section, write down the project features clearly. These are the features that will be available to the customer of your project directly.

# Technology Stack:

|  |  |
| --- | --- |
| Language(C#/Java) |  |
| Platform(Web/Desktop) |  |
| Frontend Technology(Simple HTML/Bootstrap/any other library) |  |
| IDEs | List the IDEs here that you will use for development |

# Project Actors:

In this sections, write the name of actors and brief description, who will be using the system.

# Use Cases:

All the use cases should be written in the following format. The name of use should be start with a word 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:

## Use Case 1(Write the name of the use case):

|  |  |
| --- | --- |
| Use Case ID | U01 |
| Name |  |
| Actor |  |
| Description | Brief description of the use case. With at least 50 words. |
| Layout in pencil tool |  |
| 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.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case Id | TextBox | DropDown | Password Box | Table | Date Field | Buttons | AutoComplete | Radio Button | CheckBox | Menu | Text Area | ProgressBar |
| U01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| U02 |  |  |  |  |  |  |  |  |  |  |  |  |

# Classes:

In this section, we do not require detailed design diagram. But identify the tentative classes with the requirement that you should have at least 10 domain classes and 5 software classes, 3 abstract classes, 2 singelton classes. 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)** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Object Oriented Features:

## Composition:

In this section, Identify the at least three solid examples where composition can be perform. Add UML diagram of each example.

## Inheritance:

At least 2 examples where inheritance will take place with UML diagrams.

## Multiple Inheritance:

Two example with UML diagrams of multiple inheritance.

## Multi-Level Inheritance:

Two examples with UML diagrams of multi-level inheritance.

## Polymorphism:

At least 3 examples with UML diagram for polymorphism.

# Collections:

In this section, describe how and where you will use the following collections and why you are forced to use these collections.

ArrayList, LinkedList, Queue, Stack, HashSet and TreeSet

# Exceptions:

In this sections, identify at high level which type of exceptions you can face in your code and what are the solutions. Add more rows in the table as per requirements.

|  |  |  |
| --- | --- | --- |
| Type of Exception | Why this exception will occur | How you will handle the exception |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Data Storage:

In this sections, describe the five files with their format from where you will read or store data. In case of database table, write down the names of columns for tables. At least 5 files/ tables are required.

# Email Sending:

In this section, describe the points where you will be required to send the email from the code. Additionally, write down the sample subject and email content.

# Project Plan

This section should include the implementation plan and work division among the members. All the estimated dates should be before June 15, 2021

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Id** | **Use Case Name** | **Member Name** | **Estimated Completion Date** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |