**DB Final Project Proposal**

**General Store Management System**

A logo of a computer science

Description automatically generated

**Session**: 2022 – 2026

**Submitted by:**

Tabish Akhtar 2022-CS-78

Saqlain Mansab 2022-CS-80

Zulqarnain Akram 2022-CS-84

**Supervised by:**

Mr. Nazeef Ul Haq

Department of Computer Science

**University of Engineering and Technology**

**Lahore Pakistan**

Table of Contents

[Project Description: 3](#_Toc163073731)

[Project Features: 3](#_Toc163073732)

[Technology Stack: 4](#_Toc163073733)

[Project Actors: 4](#_Toc163073734)

[Admin: 4](#_Toc163073735)

[Employee: 4](#_Toc163073736)

[Use Cases: 5](#_Toc163073737)

[Use Case 1(Add Employees): 5](#_Toc163073738)

[Use Case 2(Edit Employees): 6](#_Toc163073739)

[Use Case 3(Add Customers): 7](#_Toc163073740)

[Use Case 4(Edit Customers): 8](#_Toc163073741)

[Use Case 5(Add Supplier): 8](#_Toc163073742)

[Use Case 6(Edit Supplier): 9](#_Toc163073743)

[Use Case 7(Add Categories): 10](#_Toc163073744)

[Use Case 8(Update Categories): 11](#_Toc163073745)

[Use Case 9(Delete Categories): 12](#_Toc163073746)

[Use Case 10( Employees Attendance): 13](#_Toc163073747)

[Use Case 11(Discounts): 14](#_Toc163073748)

[Use Case 12(Generate Reports): 15](#_Toc163073749)

[Use Case 13(Place Order): 16](#_Toc163073750)

[Use Case 14(Purchase Products): 17](#_Toc163073751)

[User Interface Details 18](#_Toc163073752)

[Classes: 19](#_Toc163073753)

[ER Design: 20](#_Toc163073754)

[One to One Relationship 20](#_Toc163073755)

[Many to Many Relationship: 20](#_Toc163073756)

[One to Many Relationship 20](#_Toc163073757)

[Multi-Level Inheritance: 20](#_Toc163073758)

[Weak Entities: 20](#_Toc163073759)

[Strong Relationship: 20](#_Toc163073760)

[Multivalued Attributes: 20](#_Toc163073761)

[Computed Attributes: 20](#_Toc163073762)

[Complex Attributes: 20](#_Toc163073763)

[Database Design 21](#_Toc163073764)

[Transactions 21](#_Toc163073765)

[Views 21](#_Toc163073766)

[Stored Procedures 21](#_Toc163073767)

[Triggers 21](#_Toc163073768)

[Indexes 21](#_Toc163073769)

[Exceptions: 22](#_Toc163073770)

[Project Plan 22](#_Toc163073771)

# Project Description:

Our project aims to revolutionize the management of general shops, simplifying tasks such as inventory tracking, sales monitoring, and customer, supplier, and employee management. The main authority is given to the admin, who utilizes the management system to oversee and manage employees, products, and suppliers for their shop. The admin has the capability to define product categories and sub-categories tailored to the shop's offerings. Moreover, the admin can implement discounts through the issuance of coupon codes for specific products, enhancing promotional activities and more sale.

Employees also play a crucial role in the system, with the authority to add customers and suppliers to the database. When a customer visits the shop, employees promptly enter their information into the system and record the products purchased. Similarly, when a supplier comes to the shop, employees add their data and any products purchased from them. The system accommodates both cash payments and product borrowing options for customers, providing flexibility in transaction methods.

Additionally, the admin has the capability to manage employee attendance efficiently. Through the system, the admin can mark employee attendance, ensuring accurate records of staff presence. Furthermore, the system automatically logs the time of employee logins, maintaining a comprehensive record of system usage for record-keeping purposes.

Furthermore, our project incorporates advanced reporting and analytics features to provide valuable insights into shop operations. Through customizable reports and dashboards, shop owners can gain a deeper understanding of sales trends, customer behavior, and product performance. This data-driven approach empowers shop owners to make informed decisions and optimize business strategies for growth and success.

Moreover, our system prioritizes user-friendly design and intuitive navigation, ensuring ease of use for both admin and employees. With a simple and intuitive interface, users can quickly access key features and functionalities, enhancing efficiency and productivity in day-to-day operations.

In conclusion, our project represents a comprehensive solution for general shop management, integrating essential features and functionalities to streamline operations, enhance productivity, and drive business growth.

# Project Features:

Here are project features available to the user of application.

**Admin:**

**Manage Employees:**

Add, update, and remove employee details, including its name, profession, salary and contact information.

**Manage Customers:**

Add, update, and remove employee details, including its name, address and contact information.

**Manage Categories:**

Add, update, and remove products categories and sub categories of products.

**Manage Products:**

Add, update and remove products for shop by selecting their sub category and other information.

**Manage Suppliers:**

Add, update and remove supplier details including its name, address, description and contact information.

**Discount Coupons:**

Set up and manage discount coupons for specific products to attract customers.

**Attendance Tracking:**

Monitor employee attendance and manage work hours for payroll purposes.

**Reporting and Analytics:**

Access comprehensive reports and analytics to track sales, inventory, and employee performance.

**Employees:**

**Manage Customers:**

Add, update, and remove employee details, including its name, address and contact information.

**Manage Suppliers:**

Add, update and remove supplier details including its name, address, description and contact information.

**Place Order:**

Employees can place order by entering product id, customer id , quantity and coupon code(optional) for customers.

**Purchase Products:**

Employees can purchase products from suppliers by entering their details and product details for shop.

**Payment Options:**

Accept cash payments and manage product borrowing for customers.

# Technology Stack:

|  |  |
| --- | --- |
| Language (C#/Java) | C# |
| Platform (Web/Desktop) | Desktop |
| Frontend Technology | C# Window Forms |
| IDEs | Visual Studio Community 2022, SSMS 2022 |

# Project Actors:

The project actors who will use this system are:

## Admin:

The admin is main authority of the shop management system. Admin oversee all the customer, employees, products, products categories discount coupon. Admin can also see specific reports like stock available, most wanted products, less selling products, employee attendance, customers preferences. This comprehensive oversight empowers the admin to make informed decisions, optimize business strategies, and drive success within the shop environment.

## Employee:

The employee plays a vital role in the shop management system, being responsible for various tasks. They handle customer details, including adding, updating, or removing information such as name, address, and contact details. Similarly, employees manage supplier details, ensuring accuracy in name, address, description, and contact information. When processing customer orders, employees input essential details like product ID, customer ID, quantity, and, if applicable, a coupon code. Additionally, they handle the purchasing process by entering supplier and product details to replenish shop inventory. Lastly, employees facilitate transactions by accepting cash payments and managing product borrowing for customers.

# 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(Add Employees):

|  |  |
| --- | --- |
| Use Case ID | U01 |
| Name | Add Employees |
| Actor | Admin |
| Description | The "Add employees" feature allows admin to add new workers to the shop system. They fill in details like names, emails, and salaries. The system checks if all needed info is entered correctly. It's easy to use, with clear instructions and boxes for different info types. Once everything is filled out right, the system creates a special ID for the new worker and adds them to the system. This feature makes it easy for businesses to keep track of their employees and helps them get started quickly. |
| Layout in pencil tool |  |
| Validators | **Name and FName box:**  The Name box accepts strings of up to 16 characters in length and cannot be null (empty).  **Email box:**  Email box only accept string of email address format (e.g., [example@gmail.com](mailto:example@gmail.com) ).  **Salary, Account# and Contact box:**  These boxes only accept an integer of up to 15 characters.  **Address box:**  The address box accept string up to 32 characters.  **Credentials check box:**  user can select or un select this check box.  **Add button:**  The Add button will only add a new employee if all required fields, such as Name, Profession, etc., are filled out. |

## Use Case 2(Edit Employees):

|  |  |
| --- | --- |
| Use Case ID | U02 |
| Name | Edit Employees |
| Actor | Admin |
| Description | The "Edit Employees" feature lets admins modify details of employees already in the system. They can choose an employee from the list and change things like name, email, and salary. Before saving any updates, the system checks to ensure the new info is correct. Admins can also delete an employee by selecting them and confirming the deletion. The system double-checks to make sure the deletion is intentional. If confirmed, the employee's record, including all associated info, is permanently removed from the database. After changes are confirmed, the employee's record is updated accordingly. |
| Layout in pencil tool |  |
| Validators | **Data Grid View:**  Admin can click on any row, and the data from that row will appear in the designated boxes. They can then modify this information.  **Name and FName box:**  The Name box accepts strings of up to 16 characters in length and cannot be null (empty).  **Email Box:**  Email Box only Accepts string of email address format (e.g., [example@gmail.com](mailto:example@gmail.com) ).  **Salary, Account# and Contact box:**  These Boxes accepts an integer of up to 15 characters.  **Address Box:**  This box accepts string up to 32 characters.  **Credentials radio buttons:**  Admin can select one of two radio buttons to assign or remove credentials for further actions.  **Update button:**  The UPDATE button will update the information of an employee if their current ID exists in the database.  **Delete button:**  The Delete button will update the information of an employee if their current ID exists in the database.  **Dependency Check:**  Before deleting an employee, the system should check for any dependencies or associated data, such as assigned tasks or transactions. If dependencies exist, the system should prompt the user to resolve them before proceeding with the deletion.  **Logging:**  When Admin removes an employee then it’s all data like timestamp and information remains on database but will not display on list of employees.  **Confirmation Prompt:**  When user selects a row from data grid view and click on delete button then a confirmation prompt appears on screen with OK button. This helps prevent accidental deletion. |

## Use Case 3(Add Customers):

|  |  |
| --- | --- |
| Use Case ID | U03 |
| Name | Add Customers |
| Actor | Admin, Employees |
| Description | The "Add Customers" feature enables user to add new customers to the shop management system. User access the designated interface for adding customers and input relevant details such as name, email, contact information, profession, CNIC and address. The new customer is added to the system with the specified details and a unique customer ID.  The customer's information is stored securely in the database, allowing for easy access and management. If the user decides to cancel the customer addition process at any point, they can navigate back or close the form without saving changes. |
| Layout in pencil tool |  |
| Validators | **Name, Profession and FName box:**  These boxes accept strings of up to 16 characters in length and cannot be null (empty).  **Email Box:**  Email Box only Accepts string of email address format (e.g., [example@email.com](mailto:example@email.com) ).  **Contact box:**  These boxes only accept an integer of up to 15 characters.  **Address Box:**  This box Accepts string up to 32 characters.  **CINC Box:**  CNIC box accepts an integer of length 13 digits.  **Add button:**  The Add button will only add a new customer if all required fields, such as Name, Profession, etc., are filled out. |

## Use Case 4(Edit Customers):

|  |  |
| --- | --- |
| Use Case ID | U04 |
| Name | Edit Customers |
| Actor | Admin, Employees |
| Description | The "Edit Customers" feature allows user to modify details of customers already in the system. They can select a customer from the list and change information like name, email, contact information, profession, CNIC and address details. Before saving any changes, the system checks to ensure the new information is accurate. User can also delete a customer by selecting them and confirming the deletion. The system double-checks to ensure the deletion is intentional. If confirmed, the customer's record, including all associated details, is permanently removed from the database. After changes are confirmed, the customer's record is updated accordingly. |
| Layout in pencil tool |  |
| Validators | **Name, Profession and FName box:**  These boxes accept strings of up to 16 characters in length and cannot be null (empty).  **Email Box:**  Email Box only Accepts string of email address format (e.g., [example@email.com](mailto:example@email.com) ).  **Contact box:**  These boxes only accept an integer of up to 15 characters.  **Address Box:**  This box Accepts string up to 32 characters.  **CINC Box:**  CNIC box accepts an integer of length 13 digits.  **Delete Button:**  When delete button clicked then it checks the selected row of customer from data grid and only deletes if current ID exists in database.  **Update button:**  The Update button will update the information of an employee if their current ID exists in the database. |

## Use Case 5(Add Supplier):

|  |  |
| --- | --- |
| Use Case ID | U05 |
| Name | Add Supplier |
| Actor | Admin, Employees |
| Description | The "Add Supplier" feature allows users to add new supplier data to the shop management system. User access the designated interface for adding supplier and input relevant details such as name, email, contact information, description and address. The new supplier is added to the system with the specified details and a unique supplier ID.  The supplier's information is stored securely in the database, allowing for easy access and management. Next time whenever that supplier come to shop to sell products to us then its ID will be used for products purchase. |
| Layout in pencil tool |  |
| Validators | **Name box:**  The Name box accept string of up to 16 characters in length and cannot be null (empty).  **Email Box:**  Email Box only Accepts string of email address format (e.g., [example@email.com](mailto:example@email.com) ).  **Contact box:**  These boxes only accept an integer of up to 15 characters.  **Address Box:**  This box Accepts string up to 32 characters.  **Description Box:**  Here user can write any string up to 64 characters.  **Add button:**  The Add button will only add a new supplier if all required fields, such as Name, email, contact, etc., are filled out. |

## Use Case 6(Edit Supplier):

|  |  |
| --- | --- |
| Use Case ID | U06 |
| Name | Edit Supplier |
| Actor | Admin, Employees |
| Description | The "Edit Supplier" feature allows user to modify details of supplier already in the system. They can select a supplier from the list and change information like name, email, contact information, description and address details. Before saving any changes, the system checks to ensure the new information is accurate. User can also delete a supplier by selecting them and confirming the deletion. The system double-checks to ensure the deletion is intentional. If confirmed, the supplier's record, including all associated details, is permanently removed from the database. After changes are confirmed, the customer's record is updated accordingly. |
| Layout in pencil tool |  |
| Validators | **Name box:**  The Name box accept string of up to 16 characters in length and cannot be null (empty).  **Email Box:**  Email Box only Accepts string of email address format (e.g., [example@gmail.com](mailto:example@gmail.com) ).  **Contact box:**  These boxes only accept an integer of up to 15 characters.  **Address Box:**  This box Accepts string up to 32 characters.  **Description Box:**  Here user can write any string up to 64 characters.  **Status Radio buttons:**  User can select one of these two radio buttons either active or in-active.  **Delete Button:**  When delete button clicked then it checks the selected row of customer from data grid and only deletes if current ID exists in database.  **Update button:**  The Update button will update the information of a supplier if their current ID exists in the database. |

## Use Case 7(Add Categories):

|  |  |
| --- | --- |
| Use Case ID | U07 |
| Name | Add Categories |
| Actor | Admin |
| Description | The feature “Add Categories” empowers admin to enhance their shop's product organization within the management system. Admins navigate to the category management section, where they can view existing categories in a data grid. Upon selecting a category, admins gain the ability to add subcategories directly on the same screen. These categories and subcategories serve as essential organizational tools for managing products effectively. By assigning products to appropriate categories and subcategories, admins ensure a user-friendly browsing experience for simplify general management processes. |
| Layout in pencil tool |  |
| Validators | **Name box:**  The Name box accept string of up to 16 characters in length and cannot be null (empty).  **Description box:**  Here user can write any string up to 64 characters.  **Category combo box:**  Only added categories will be displayed in category combo box.  **Add button:**  This button when clicked checks all the required boxes and if these boxes fulfill requirement then the categories or subcategories will be added to database. |

## Use Case 8(Update Categories):

|  |  |
| --- | --- |
| Use Case ID | U08 |
| Name | Update Categories |
| Actor | Admin |
| Description | The "Update Categories" feature allows admins to modify existing categories in the shop management system. Admins access the category management section, where they can view a list of current categories. Upon selecting a category to update, admins can make changes to its name, description, or any other relevant details. The system verifies the updates to ensure accuracy and completeness before applying them. This feature provides admins with the flexibility to adjust category information as needed, ensuring that the product organization remains up-to-date and relevant. |
| Layout in pencil tool |  |
| Validators | **Name box:**  The Name box accept string of up to 16 characters in length and cannot be null (empty).  **Description box:**  Here user can write any string up to 64 characters.  **Category combo box:**  Only added categories will be displayed in category combo box.  **Update button:**  This button when clicked checks all the required boxes and if these boxes fulfill requirement then the categories or subcategories will be updated. |

## Use Case 9(Delete Categories):

|  |  |
| --- | --- |
| Use Case ID | U09 |
| Name | Delete Categories |
| Actor | Admin |
| Description | The "Delete Categories" feature enables admins to remove existing categories from the shop management system. Admin navigate to the category section, where there is list of current categories. Admin can select one category then click the "Delete" button. When a category is deleted then all the products associated with that category and its subcategories would be deleted. When click on delete button a prompt message appears for confirmation of deletion. After confirmation then categories or subcategories would be deleted. |
| Layout in pencil tool |  |
| Validators | **Data Grid:**  To delete a category, the user must click on the specified row in the data grid and then click on the delete button. If the user wants to delete a subcategory, they must select the subcategory from the subcategory data grid.  **Delete button:**  When clicked, this button checks if a row is selected. If a row is selected, it proceeds to delete the corresponding category or subcategory. |

## Use Case 10( Employees Attendance):

|  |  |
| --- | --- |
| Use Case ID | U010 |
| Name | Employee Attendance |
| Actor | Admin |
| Description | The " Employee Attendance" feature enables admins to monitor and manage employee attendance within the shop management system. Admins access the attendance management section, where they can view a list of employees and their attendance records. Employees are marked as present, absent, or on leave based on their attendance status for a specific date. Admins have the ability to manually mark attendance and edit existing records. |
| Layout in pencil tool |  |
| Validators | **Select previous attendance button:**  When this button is clicked, a drop-down box and an "Ok" button appear, allowing the admin to choose a date to display previous attendance records.  **Drop Down Box:**  Admin can only select the date from within the drop-down box.  **Ok Buttons:**  The "Ok" button, located near the drop-down, displays the attendance for the selected date when a date is chosen from the drop-down.  **Choose date for attendance button:**  When this button is clicked, a date-time picker and an "Ok" button appear, enabling admins to mark attendance.  **Date Time Picker:**  Admin can only select current and previous dates to mark attendance.  **OK button:**  When the Ok button located near date time picker is clicked, employee data is displayed on the grid, and admins can mark the attendance of these employees.  **Data Grid:**  To delete an entry, users must click on the specified row in the data grid and then click on the delete button. If users want to delete a subcategory, they must select the subcategory from the subcategory data grid.  **Delete button:**  When clicked, this button checks if a row is selected. If a row is selected, it proceeds to delete the corresponding category or subcategory. |

## Use Case 11(Discounts):

|  |  |
| --- | --- |
| Use Case ID | U011 |
| Name | Discounts |
| Actor | Admin |
| Description | Discount Management allows admin to create and manage discounts within the shop management system. Admins access the discount section, where they can view existing discounts and create new ones. Each discount can be customized with parameters such as discount type, amount, validity period, and applicable products or categories. Admins can also edit or delete existing discounts as needed. The system ensures that discounts are applied correctly during checkout by inputting a discount code. |
| Layout in pencil tool |  |
| Validators | **Name, Description Box:**  These input fields accept strings of up to 64 characters in length.  **Coupon Code Box:**  Coupon code box accept any type of string for special code purpose that not already present.  **Percentage box:**  This box accepts an integer for percentage discount.  **Data Grid:**  To delete an entry, users must click on the specified row in the data grid and then click on the delete button. If users want to delete a subcategory, they must select the subcategory from the subcategory data grid.  **Update button:**  This button when clicked checks all the required boxes and if these boxes fulfill requirement then the it updates discounts if its ID matches with ID of discount in database.  **Add button:**  This button when clicked checks all the required boxes and if these boxes fulfill requirement then the it adds the discount to database if it not already exists.  **Delete button:**  When clicked, this button checks if a row is selected. If a row is selected, it proceeds to delete the corresponding discount. |

## Use Case 12(Generate Reports):

|  |  |
| --- | --- |
| Use Case ID | U012 |
| Name | Generate reports |
| Actor | Admin |
| Description | Generating reports with the Crystal Report extension in C# WinForms involves several steps to provide users with useful information. First, data is fetched from databases or other sources. Then, using the Crystal Reports designer tool in Visual Studio, developers design the report layout. After designing, the fetched data is connected to this layout. The Crystal Reports engine is then used to generate the report based on the connected data. Users can view or export the report within the WinForms application. Additionally, customization options allow developers to tailor the report layout to meet specific needs. |
| Layout in pencil tool |  |
| Validators | **Category drop-down:**  The admin can only select predefined categories from the dropdown menu.  **Select Report drop-down:**  The "Select Report" dropdown menu contains various report topics, from which the admin can choose one.  **Generate button:**  This button is activated only when both dropdown menus have selections, and upon clicking, it generates a report in PDF format based on the chosen category and report topic. |

## Use Case 13(Place Order):

|  |  |
| --- | --- |
| Use Case ID | U013 |
| Name | Place Order |
| Actor | Employee |
| Description | When a customer visits the shop, employees add their details to the customer list. When customers want to buy something, employees enter their purchases into the system and place their orders. To attract customers and promote sales, the shop displays coupon codes upfront. When customers use these codes, they get discounts on their purchases. This makes the shop a great place to buy things. |
| Layout in pencil tool |  |
| Validators | **Product ID, Customer ID boxes:**  These boxes contain authentic information of product Id and customer id.  **Quantity drop-down:**  This drop-down accepts only integers.  **Coupon code:**  Coupon code box accept any type of string for special code purpose that not already present. Its optional part.  **Transaction type radio buttons:**  User can only select one radio button from Transaction type either payment or borrowing.  **Price details label:**  This is not editable by user, it only updates when employee enter the details of products and it adjusts its discount and total price values accordingly.  **Order button:**  When all the required boxes are filled with accurate information, then this button adds the order to orders table. |

## Use Case 14(Purchase Products):

|  |  |
| --- | --- |
| Use Case ID | U014 |
| Name | Purchase Products |
| Actor | Employee |
| Description | When a supplier comes to shop, then employees register its information. Employees purchase products from that supplier by getting its ID, product details and quantity. The total price becomes automatically and employee purchases products for the shop. The price and details of products is already stored. If a new product comes to shop then admin must approve and add it first to products list then employee will be able to purchase it. |
| Layout in pencil tool |  |
| Validators | **Product ID, Supplier ID boxes:**  These boxes contain authentic information of product Id and supplier id.  **Quantity drop-down:**  This drop-down accepts only integers.  **Price details label:**  This is not editable by user, it only updates when employee enter the details of products and it adjusts its total price values accordingly.  **Purchase button:**  When all the required boxes are filled with accurate information, then products are purchased from supplier. |

# User Interface Details

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Use Case Id | TextBox | DropDown | Password Box | Table | Date Field | Buttons | AutoComplete | Radio Button | CheckBox | Menu | Text Area | ProgressBar |
| U01 | 7 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| U02 | 7 | 0 | 0 | 1 | 0 | 2 | 0 | 2 | 0 | 1 | 0 | 0 |
| U03 | 6 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| U04 | 6 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 |
| U05 | 5 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| U06 | 5 | 0 | 0 | 1 | 0 | 2 | 0 | 2 | 0 | 1 | 0 | 0 |
| U07 | 4 | 1 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 |
| U08 | 4 | 1 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 |
| U09 | 4 | 1 | 0 | 2 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 |
| U10 | 0 | 1 | 0 | 1 | 1 | 6 | 0 | 0 | 0 | 1 | 0 | 0 |
| U11 | 4 | 0 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 1 | 0 | 0 |
| U12 | 0 | 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |
| U13 | 3 | 1 | 0 | 0 | 0 | 1 | 0 | 2 | 0 | 1 | 0 | 0 |
| U14 | 2 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 |

# 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)** |
| Order | Domain | No | No | Yes |
| Customer |
| Supplier |
| Employee |
| Shipment |
| Category |
| SubCategory |
| Product |
| Discount |
| Transaction | Software | Yes |
| Person |
| Item |
| Credentials | No |
| Invoice |
| PDFGenerator |
| ExceptionGenerator |
| Admin | Yes | No |
| Shop Details | Yes |

# ER Design:

## One to One Relationship

Add UML diagram of each example(at least three).

## Many to Many Relationship:

At least 2 examples with UML diagrams.

## One to Many Relationship

10 examples with UML diagrams

## Multi-Level Inheritance:

Two examples with UML diagrams

## Weak Entities:

At least 3 examples with UML diagram

## Strong Relationship:

At least 2 examples with UML diagram

## Multivalued Attributes:

At least 3 examples with UML diagram

## Computed Attributes:

At least 5 examples with UML diagram

## Complex Attributes:

At least 5 examples with UML diagram

# Database Design

# 

# Transactions

# Views

# Stored Procedures

# Triggers

# Indexes

# 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 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Project Plan

This section should include the implementation plan and work division among the members. All the estimated dates should be before May 05, 2024

|  |  |  |  |
| --- | --- | --- | --- |
| **Use Case Id** | **Use Case Name** | **Member Name** | **Estimated Completion Date** |
| U01 | Add Employees | Tabish Akhtar | 02-05-2024 |
| U02 | Edit Employees |
| U03 | Add Customers |
| U04 | Edit Customers |
| U05 | Add Supplier |
| U06 | Edit Supplier | Saqlain Mansab | 03-05-2024 |
| U07 | Add Categories |
| U08 | Update Categories |
| U09 | Delete Categories |
| U10 | Employees Attendance |
| U11 | Discounts | Zulqarnain Akram | 04-05-2024 |
| U12 | Generate Reports |
| U13 | Place Order |
| U14 | Purchase Product |