|  |  |  |
| --- | --- | --- |
|  | **CAN THO UNIVERSITY SOFTWARE CENTER** |  |
| **MEKONG DELTA - APTECH** |

**Aptech Certified Computer Professional (ACCP)**

**SEMESTER II**

**FEKER Store**

|  |  |
| --- | --- |
| **Instructor:** | **Class: CP2396G01** |
| MSr: Nguyễn Võ Thông Thái | **Team:** |
| **Students:**  1. A21157- Bùi Trung Nguyên |

Can Tho, 07/2019

**PROBLEM DEFINITION OF PROJECT**

Feker Store specializes in providing mechanical keyboard products. With a passion for mechanical keyboards such as custom keyboards, metal keyboards, plastic keyboards, keyboards with LEDs, keyboards without LEDs, ... a variety of keyboards, we create Feker products to serve everyone, all ages, all professions with a common passion for keyboards.

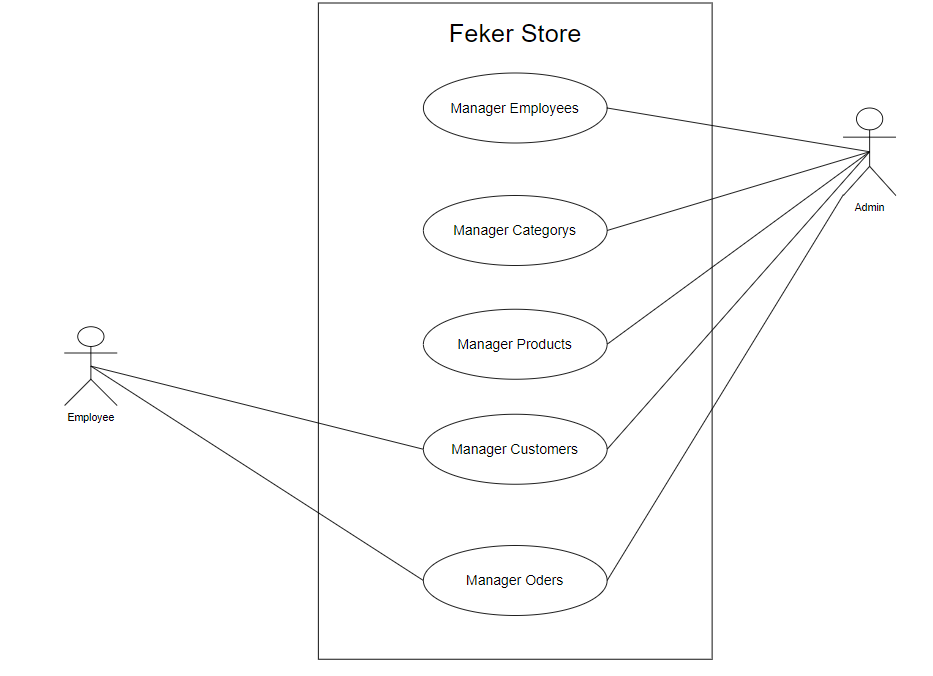
Feker stores are growing and are being trusted and chosen by more and more people to shop, more work must be done on paper such as product quantity management, invoice management, customer management, ... this causes many difficulties for the management of the store, paper management makes it difficult to find products, invoices when needed, documents are easily lost and procedures are quick when having to make invoices by hand. Therefore, the store needs to change to a newer technology management system and change the management work on paper to management on the management system software and create a management application for the store, helping the store's operations become easier to manage, clearer and more transparent.

The appearance of the Feker Store application is the answer to the difficulties that the store is facing. With simple features such as Logging into the system, having a customer management page, product management, product type management, invoice management, store staff management, changing the way of operating and creating a store app is inevitable to help store management become easier and help customers'

**CUSTOMER REQUIREMENTS SPECIFICATION (CRS)**

**Name of Project**:

**Modeling requirements of customer (Use case Diagram)**

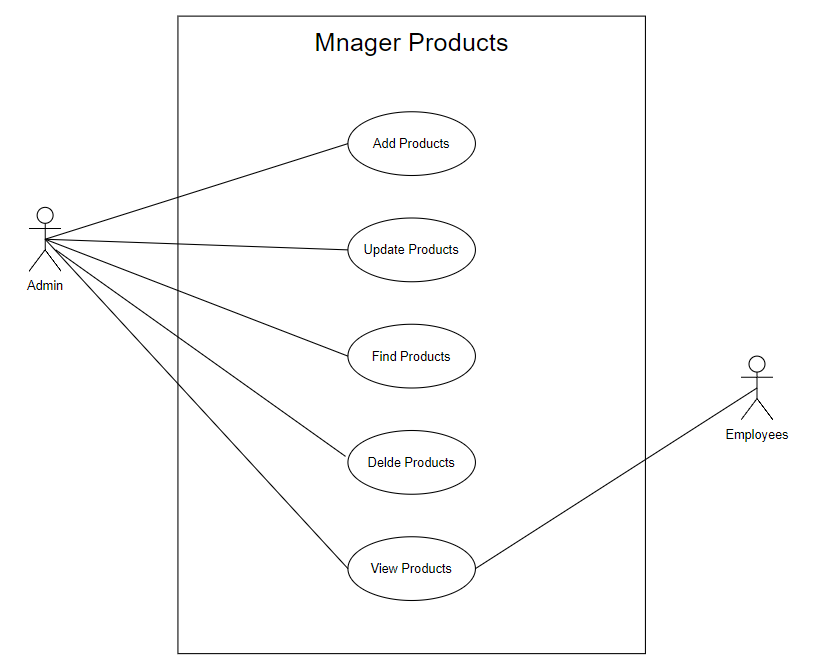


**Actor Description**

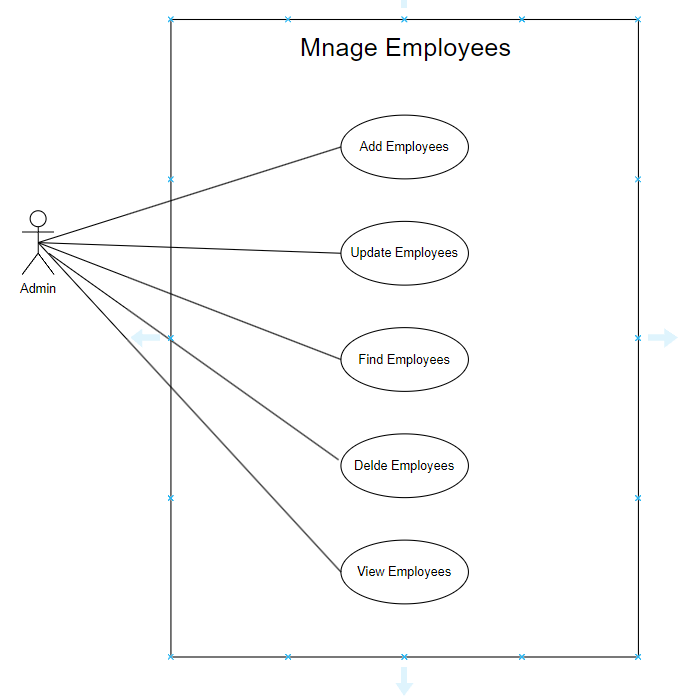
|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Actor** | **Use Case** | **Description** |
|  | Admin | Manage Customer | This subsystem allows the administration of adding and editing customers of the store |
| Manage Product | This subsystem allows Administrators to manage product information including adding new products, updating product information and deleting products. |
| Login | If you want to access the store's management system, you need to be granted an account to be able to log into the system and continue working. |
| Manager Employees | Managers can add, edit and delete at will with the employee management function |
| Manager Orders | When customers come to buy products, they can add, edit, and delete customer orders in the order management section of the software |
| Manager Categories | For product categories, product classification is required in the product type management function. Managers can add, edit, and delete product classifications. |
|  | Employees | View Products | For employees in the product management function, they can only see the product table. |
| Manager Customers | Store employees can register to add customers to the customer list in the customer management function and have the right to edit customer information as required. |
| Login | For employees, they also need an account administrator to be able to log in to the system and work. |
| Manager Orders | After the employee registers a customer for membership, each customer will make a purchase and the employee will add an invoice and select which customer belongs to that order and can edit the customer's order as required. |

**Use case Description**

* + **Manage Products:** This subsystem allows administrators to manage product information including adding new products, updating product information and deleting products.

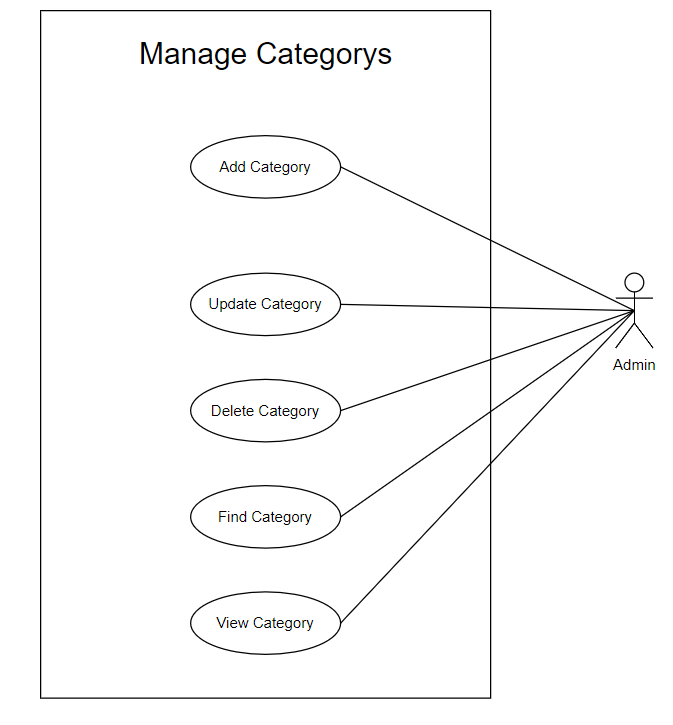


* + - **Add New Product**
* Purpose: this function allows Adminto add new information about the product.
* Inputs:
  + Select the product's Name
  + Select the product's CategoryName
  + Select the product's Img
  + Select the product's price
  + Select the product's Description
* Outputs: New information about the product is added to the table of the database.
  + - **Update Product**
* Purpose:Purpose: This function allows Admin to edit existing product information.
* Inputs:   
   Edit Product Name  
   Edit Product CategoryName  
   Edit Product Img  
   Edit Product Price  
   Edit Product Description
* Outputs: New information about the product is added to the table of the database.
  + - **Delete Product**
* Purpose: Allows managers to select and delete a product from the product table
* Inputs:
* Outputs:Successfully deleted a product from the database

**3.1.4. FInd Product** Purpose: Allows users to enter product names in the search bar and the data will be searched and filtered based on the name entered in the search bar.  
 Inputs: Enter the name in the search bar  
 Outputs: Filter the search list by name and display if there is no name, the table will be blank.  
**3.2. Manage Employees :** This subsystem allows administrators to manage Employee information including adding new Employees, updating Employee Information, and deleting Employees.

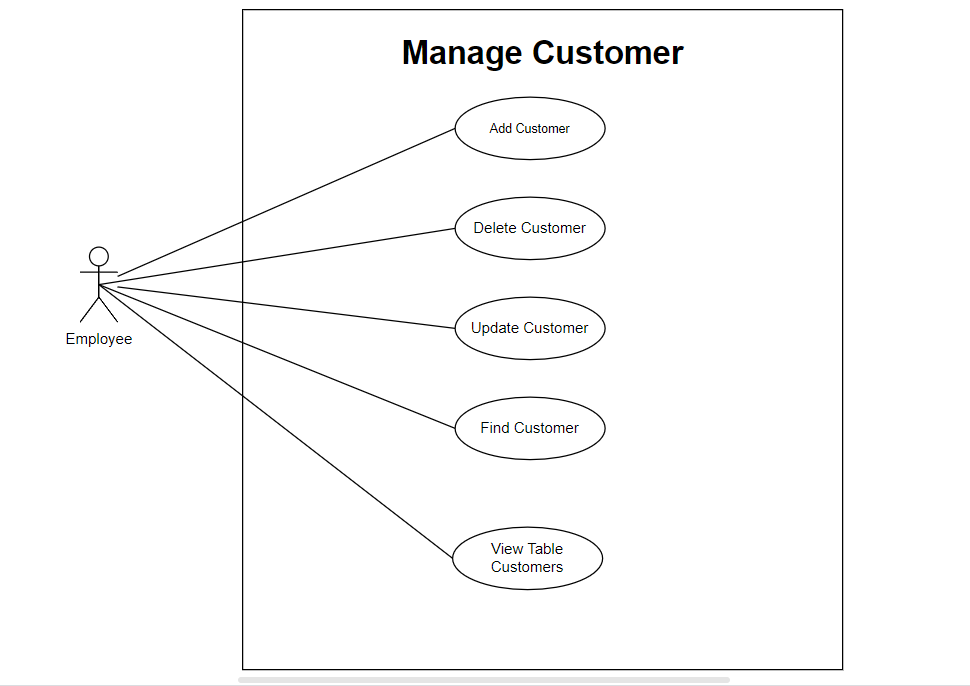
* **Add Employees**
  + Purpose:this function allows Adminto add new information about the Employees.
  + Inputs:
    - MSNV
    - Name
    - Phone
    - Address
    - Img
    - Password
  + Outputs:New information of employees is added to table of database.
* **Update Employees** 
  + Purpose: This function allows Admin to edit existing Employees information.
  + Inputs:
    - MSNV
    - Name
    - Phone
    - Address
    - Img
    - Password
  + Outputs::New information of employees is added to the table of the database.
* **Delete Employees** 
  + Purpose:Allows managers to select and delete a employees from the employees table
  + Inputs:
  + Outputs:Successfully deleted a employees from the database
* **Find Employees**
  + Purpose: Allows users to enter employees names in the search bar and the data will be searched and filtered based on the name entered in the search bar.
  + Inputs: Enter the name in the search bar.
  + Outputs:Filter the search list by name and display if there is no name, the table will be blank.

**3.3. Manage Category:** This subsystem allows administrators to manage product type information including adding new product types, updating Product Type Information, and deleting product types.



* **Add Categories:**
  + Purpose:this function allows Adminto add new information about the Categories
  + Input:
    - Name
    - Description
  + Outputs:New information of Category added to the table of the database.
* **Update Categories:**
  + Purpose: This function allows Admin to edit existing Category information.
  + Input:
    - Name
    - Description
  + Outputs: New information of Category added to the table of the database.
* **Delete Categories:**
  + Purpose:Allows managers to select and delete a Category From the Category table
  + Input:
  + Outputs:Successfully deleted a employees from the database
* **Find Categories:** 
  + Purpose: Allows administrators to easily search product categories
  + Input: input Name Category in JtextFIeld Search Name
  + Outputs: show rows with the name of the category or without and not showing

**3.4 Manager Customer :This subsystem allows administrators to manage customers including adding new customers, updating Customer Information and deleting Customers.**

****

* **Add Customer** 
  + Purpose: This function allows Admin to add new customers
  + Input
    - Name
    - Phone
    - Address
    - MSKH ( Random)
  + Output: New information of Customer added to the table of the database.
* **Edit Customer**
  + Purpose: Allows editing customer information if there are errors
  + Input
    - Name
    - Phone
    - Address
    - MSKH ( Random)
  + Output: New information of Customer added to the table of the database.
* **Delete Customer**
  + Purpose: Allows deleting customers if needed
  + Input
  + Output:Successfully deleted from database
* **Find Customer** 
  + Purpose: Allows searching for customers
  + Input
    - Name
  + Output: Successful search format displays the customer column

**3.5 Manager Order:Allows users to add products to order or delete products if they choose incorrectly. and saved to the Order and OrderDetail tables.**

* **Add Order**
* **Delete Order**

**Hardware and Software required for implementing the project**

|  |  |  |
| --- | --- | --- |
|  | **Server** | **Client** |
| **Hardware** |  |  |
| **Software** |  |  |
| **Additional**  **Technology** |  | |

**CUSTOMER ACCEPTANCE CRITERIA**

|  |  |
| --- | --- |
| **No.** | **Customer’s Acceptance Criteria** |
|  | Function: Add, edit, delete and view employee list. |
|  | Add, edit, delete and view product lists. |
|  | Add, edit, delete and view product categories list. |
|  | Add, edit, delete and view customer lists. |
|  | : Create and manage orders for each customer. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**PROJECT PLAN**

1. **Project Details**
2. **Name of the Client:** CUSC APTECH Cần Thơ
3. **Date of the Project Plan:**2/5/2024
4. **Project Vision/Objectives: (Define the project vision/objectives as stated by the client)**

Project Vision:

Build a keyboard store management application, providing effective employee, product, product type, customer and order management functions.

Objectives:

Create a system to help manage and track employee information.

Manage product information.

Manage product categories.

Customer Information Management.

Create and manage orders for customers.

1. **Scope: (Mention the scope of the project giving the location that will be covered, processed, range of services, and so on)**

**Location**: The application will be deployed and used at all branches of the keyboard store.

**Handled**: All processes related to employee, product, product type, customer and order management.

**Service range**:

Staff manager

Product Management

Manage product types

Customer management

Create and manage orders

1. **Our understanding of the client organization: (Give the range of services, functions, overview of processes, and so on)**

**Services**: Supply and retail of keyboards and related accessories.

**Function**: Manage employees, products, product categories, customers and orders.

**Procedure**:

Manage and track employee information.

Manage product information.

Manage product catalog.

Customer Information Management.

Create and manage orders for customers.

1. **Project Organization with Responsibilities and Authorities: (Give the name of Project team members their roles and responsibilities)**

|  |  |  |  |
| --- | --- | --- | --- |
| No | Member | Roles | Responsibilities |
| 1 | Bui Trung Nguyen | |  | | --- | | Project Manager |  |  | | --- | |  | | Project management, planning and monitoring project progress |
| 2 | Bui Trung Nguyen | |  | | --- | | Business Analyst |  |  | | --- | |  | | Collect and analyze requirements from customers |
| 3 | Bui Trung Nguyen | Developer | Application development and testing |
| 4 | Bui Trung Nguyen | |  | | --- | | UI/UX Designer |  |  | | --- | |  | | User interface and user experience design |
| 6 | Bui Trung Nguyen | |  | | --- | | QA Tester |  |  | | --- | |  | | Testing and ensuring product quality |
|  | Bui Trung Nguyen | |  | | --- | | Database Admin |  |  | | --- | |  | | Database management and data security |

1. **Project Initiation/Requirement Documents**

(Information required from the client as inputs regarding his/her system; could be the information about the services, processes, and so on)

* Information required from the client:
  + Detailed information about current management processes.
  + Ask in detail about the application's functions.
  + List of available products, product categories and customers.

1. **Deliverables**

(The documents to be handed over to the client - such as CRS, Design Document, Installation Manual, User Manual, Maintenance Manual, and Code of documents)

* Documents that need to be handed over to customers:
  + Customer Requirement Specification (CRS)
  + Design Document
  + Installation Manual
  + User Manual
  + Maintenance Manual
  + Code Documentation

1. **Project Dependencies**

(Any event or task outside the scope of the project which will affect the success of the project)

* Dependencies
  + Ability to provide information and feedback from customers.
  + Customer's technology infrastructure (hardware, network, etc.).
  + Cooperation of project stakeholders.

1. **Major Project Milestones**

(Generating CRS, Building a Prototype, Gartt - describe the main milestones of the project)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Week Task | 1 | | | | | 2 | | | | | 3 | | | | | 4 | | | | |
|  | M | T | W | T | F | M | T | W | T | F | M | T | W | T | F | M | T | W | T | F |
| Analyze customer requirements |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Design system |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Coding |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **Quality Plan**
2. Review Activities (Review meeting participants, frequency, and so on)

* Participants
  + Bùi Trung Nguyên
* Frequency
  + every day.

1. Testing Activities (Final Test)

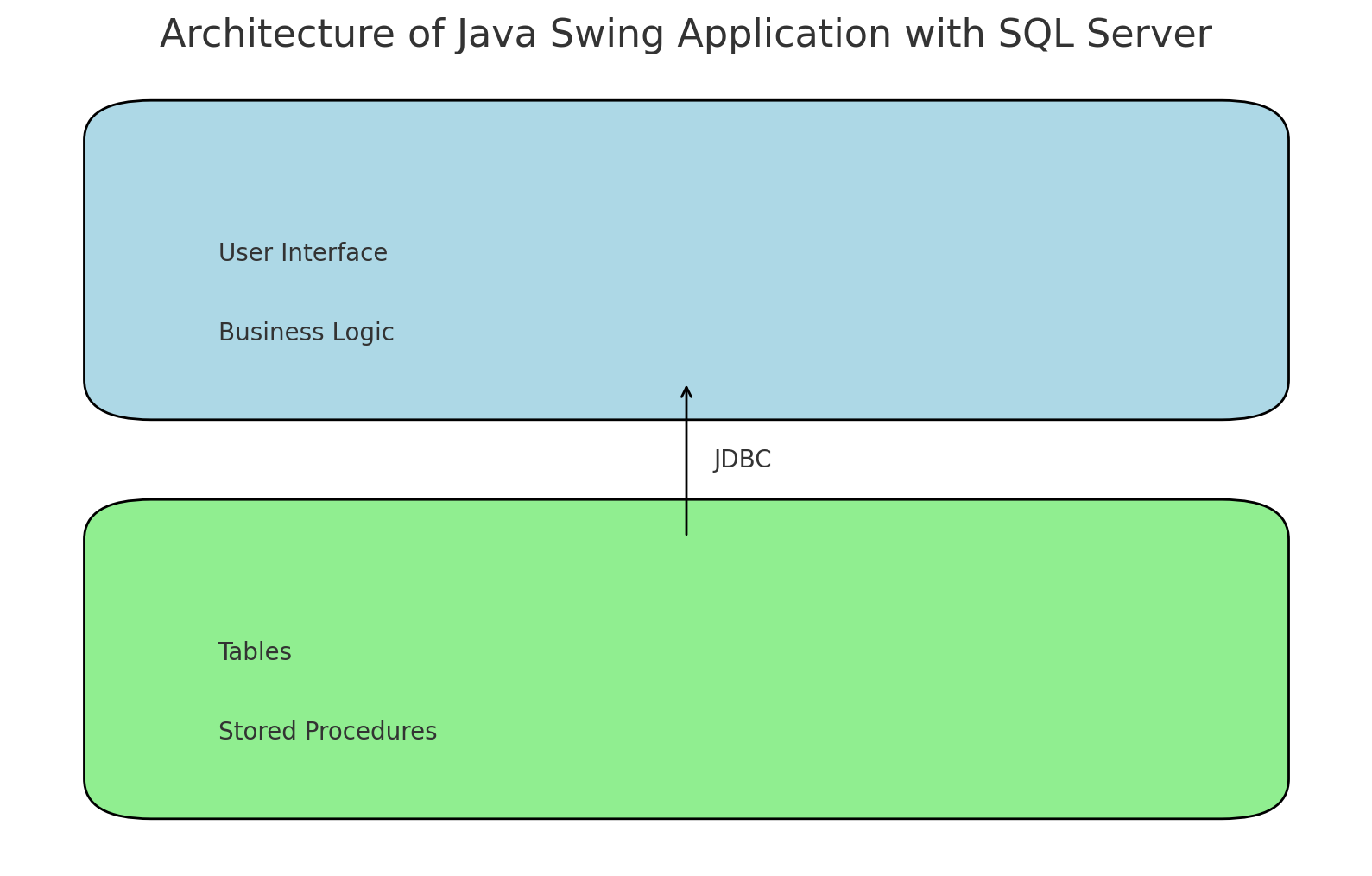
Take the final test 5 days before completing the project.

1. Backup and Recovery Strategies (In case of disk crash, network failures, and so on)

Use Github to save: https://github.com/trungnguyen1812/ProjectSem2 .

**ARCHITECTURE & GUI STANDARD OF APPLICATION**

1. **Architecture of Application**



1. **GUI Standard Application**

|  |  |
| --- | --- |
| **Property** | **Value** |
| Document Theme and Color Scheme | The main colors are blue and white, with a darker blue for the left navigation bar and function buttons. |
| Form - Background Color | White (#FFFFFF). |
| Title - Font Size | Title - Font Size: Large font size, estimated about 24pt. |
| Title - Font Color | Title - Font Color: Black (#000000). |
| Title - Font Style | Title - Font Style: Bold font style. |
| Title - Alignment | Title - Alignment: Center alignment. |
| Background color of Controls on the form | Add button: Light green.  Edit button: Light blue.  Delete button: Light red.  Search box and search function buttons: White. |
| Foreground color of Controls on the form | Add button: White (plus icon).  Edit button: White (pencil icon).  Delete button: White (trash can icon).  Search box and search function buttons: Black for text. |
| Control Caption - Font Size | Control Caption - Font Size: About 12pt. |
| Control Caption - Font Color | Control Caption - Font Color: Black (#000000). |
| Control Caption - Font Style | Control Caption - Font Style: Regular font style. |
| Control caption and controls - Alignment | Left alignment for table items.  Center the function buttons. |
| Command button - Alienment | Command Button Alignment: Function buttons (Add, Edit, Delete) are centered horizontally above the customer data table. |

**INTERFACE DESIGN DOCUMENT**

List of forms to be created

1. **Document Name: Manager Employees**
   1. **Description:**

The employee management function is an important part of any organization's management system. It helps manage employee information effectively and conveniently. Below is a detailed description of the main functions including adding, editing, deleting and searching employees.

* 1. **Controls on the Document**

**Button:  click to display an employee creation form.**

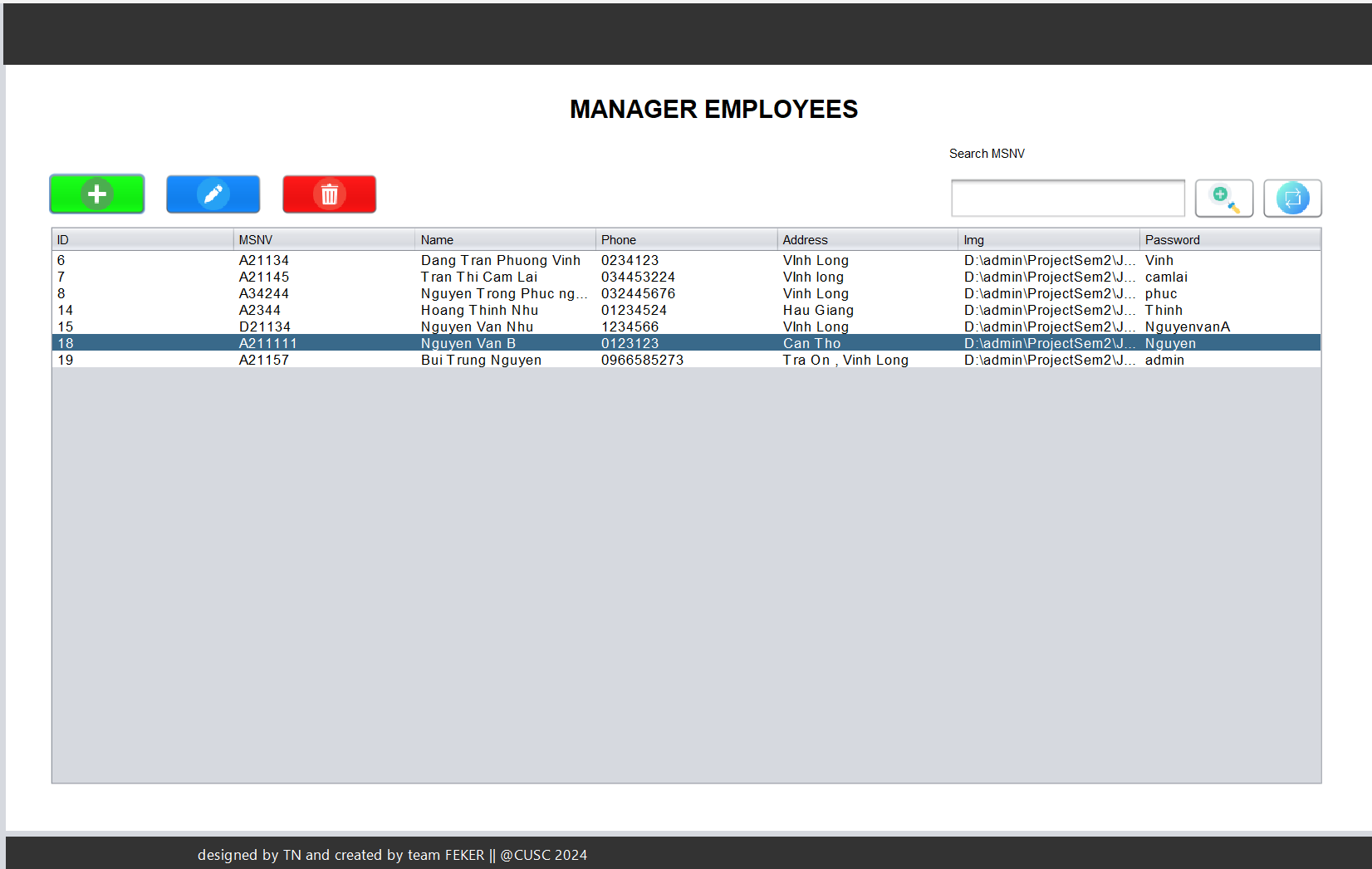
**Button:  click to display an employee edition form.**

**Button:  click to delete employee information.**

**Button:  click to search employee information.**

**Button:  click to reset the employee table.**

* 1. **UI design**

****

1. **Document Name: Manager Category**
   1. **Description**

The product category management function in a sales management or warehouse management system is an important part that helps administrators easily organize and operate product categories. Below are the main functions including adding, editing, deleting and searching for product types.

* 1. **Controls on the Document**

**Button:  click to display a Category creation form.**

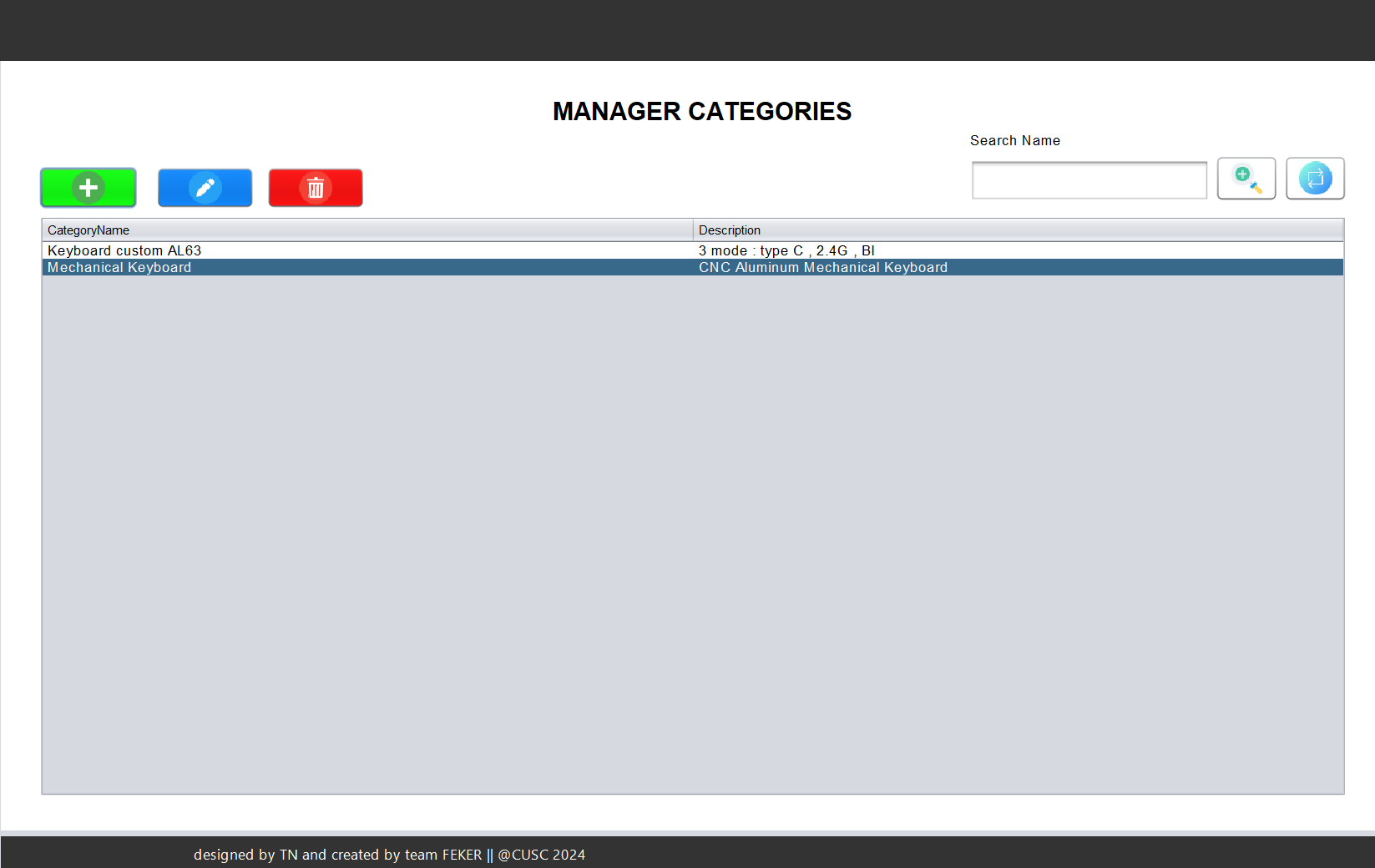
**Button:  click to display a Category edition form.**

**Button:  click to delete Category information.**

**Button:  click to search Category information.**

**Button:  click to reset the Category table.**

* 1. **UI design**



1. **Document Name: Manager Product**
   1. **Description**

Product Management functionality is an important feature of our keyboard sales application. With this feature, users can easily add, edit, delete and search keyboard products quickly and effectively.

* 1. **Controls on the Document**

**Button:  click to display a Product creation form.**

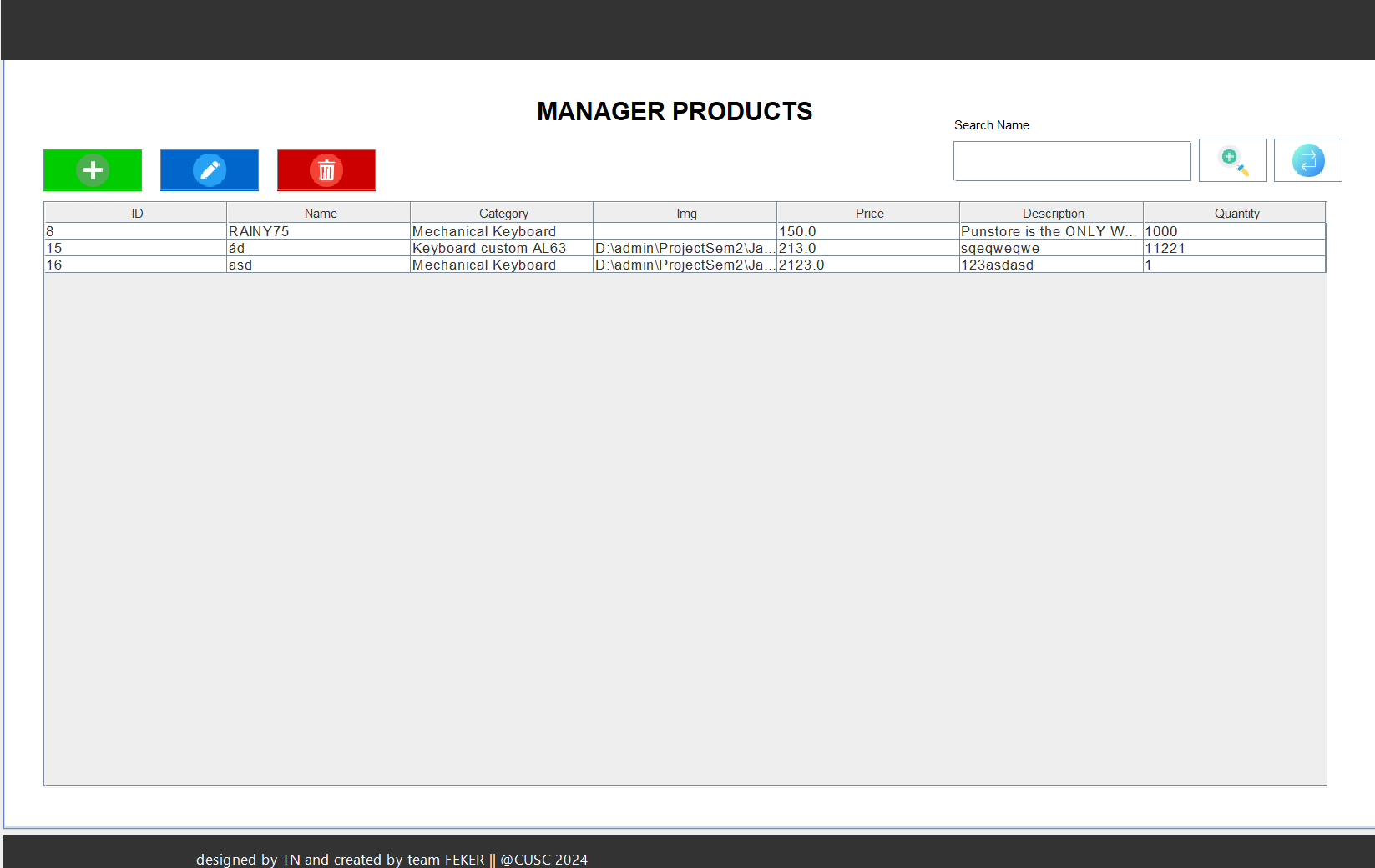
**Button:  click to display a Product edition form.**

**Button:  click to delete Product information.**

**Button:  click to search Product information.**

**Button:  click to reset the Product table.**

* 1. **UI design**



1. **Document Name: Manager Customer**
   1. **Description**

Customer Management function is an important feature in our keyboard sales application. With this feature, users can easily add, edit, delete and search customer information, and manage their orders.

* 1. **Controls on the Document**

**Button:  click to display a Customer creation form.**

**Button:  click to display a Customer edition form.**

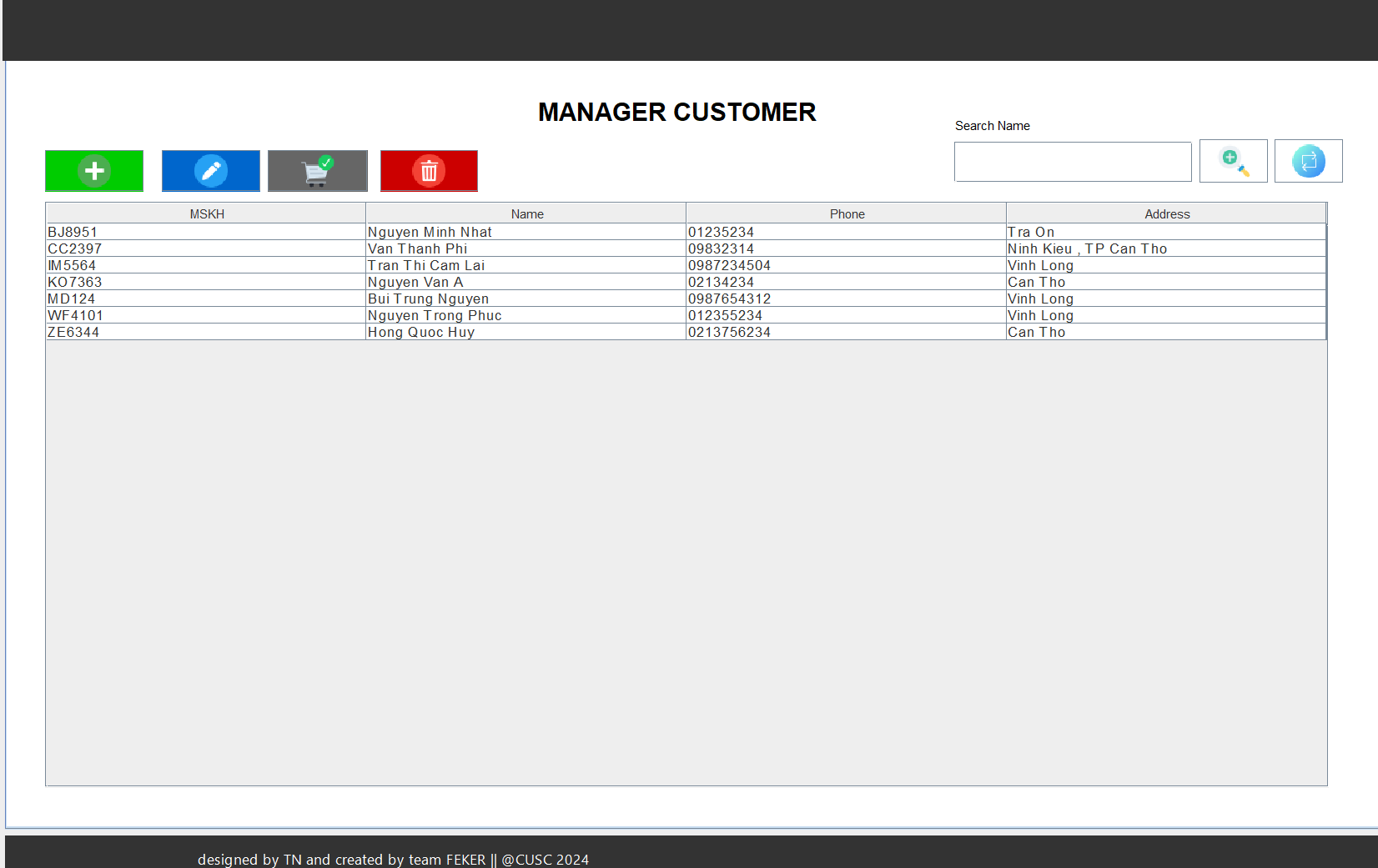
**Button:  click to delete Customer information.**

**Button:  click to search Customer information.**

**Button:  click to reset the Customer table.**

**Button:  Click to display the product order form.**

* 1. **UI design**



1. **Document Name: Manager Order**
   1. **Description**

The Order function in the keyboard store manager application provides a simple and intuitive interface for creating and managing orders. With a few clicks, employees can easily select keyboard models, quantities and shipping information to create a new order. The application will automatically update order status and notify staff when there are any changes.

* 1. **Controls on the Document**

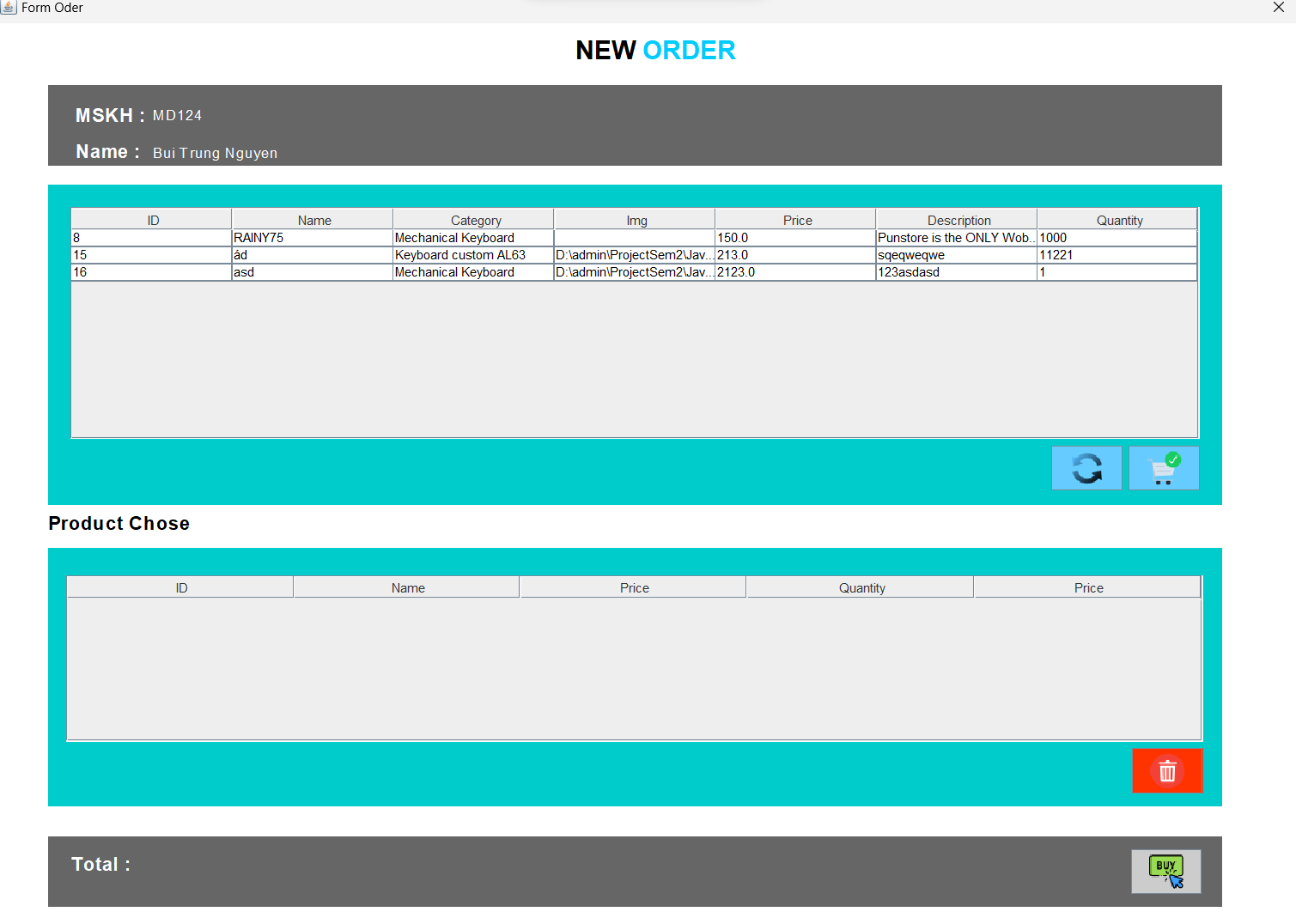
**Button:  Click to select the chosen product table.**

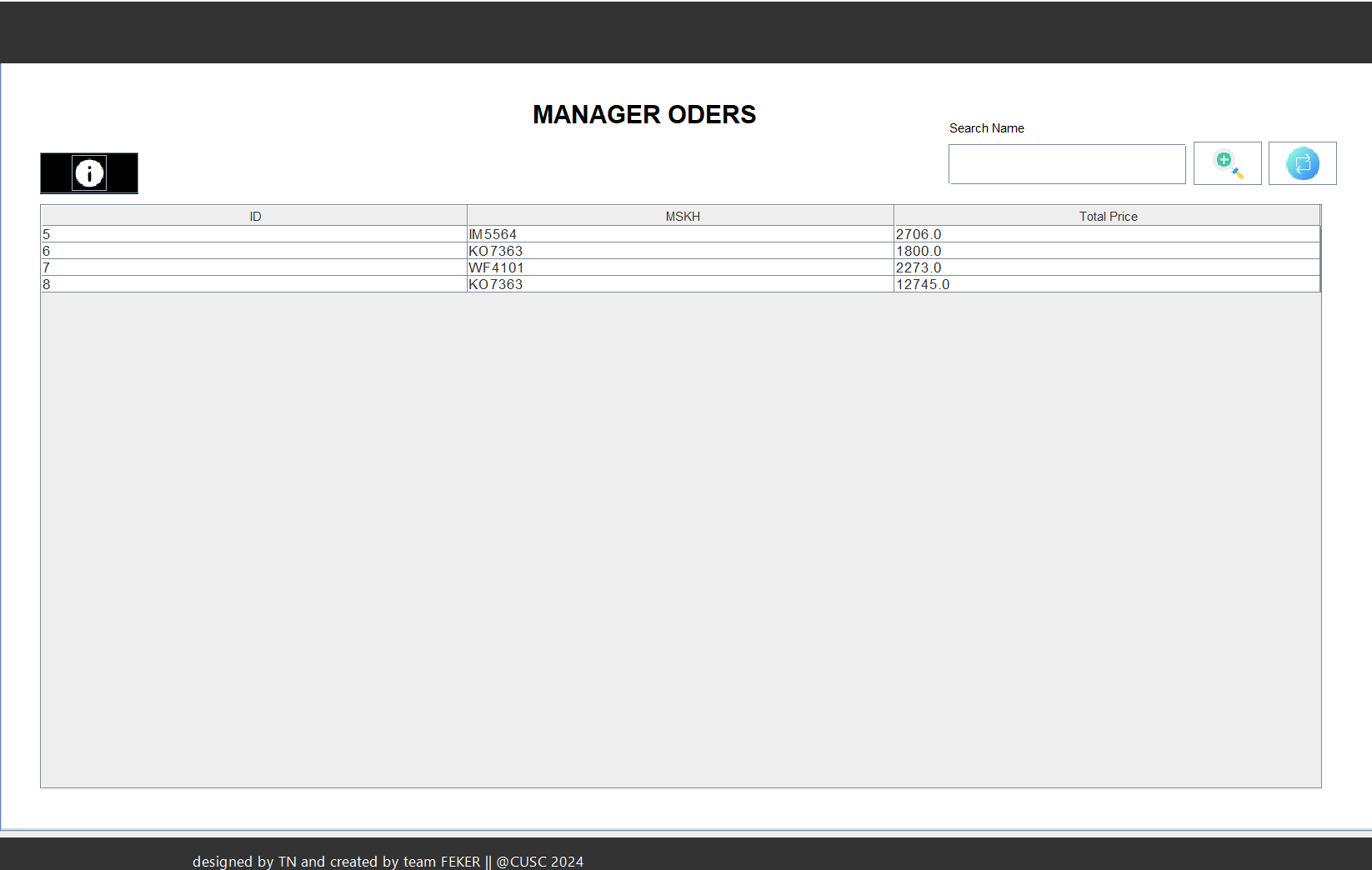
**Button:  Click to create orders for customers**

**Button: Click to remove the product from the choose product table.**

**Button:  click to reset the Product table in form new order.**

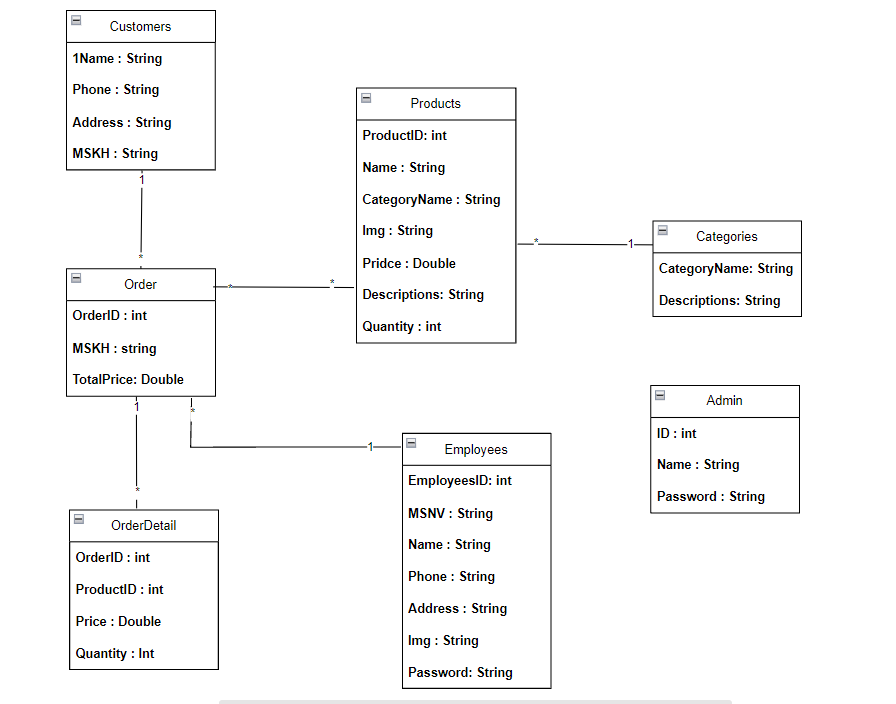
* 1. **UI design**





**DATABASE DESIGN DOCUMENT**

**I. ER/Class Diagram**



**II. Table Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Employees** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| EmployeesID | int | Primary Key, Identity | Employee code |
| MSNV | nchar(10) | Not null | customer code |
| Name | nvarchar(100) | Not null | Employee Name |
| Phone | nchar(10) | Not null | Employee Phone |
| Address | nvarchar(500) | Not null | Employee Address |
| Img | nvarchar(500) | Not null | image of staff |
| Password | nvarchar(50) | Not null | employee password |

|  |  |  |  |
| --- | --- | --- | --- |
| **Products** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| ProductID | Int | Primary Key, Identity | Product code |
| CategoryName | nvarchar(50) | Foreign Key, Not Null | Category name |
| Name | nvarchar(50) | Not Null | Product name |
| Img | nvarchar(500) | Not Null | image product |
| Price | float | Not Null | price of the product |
| Quantity | Int | Not Null | product quantity |
| Description | text | Not null | Product introduction |

|  |  |  |  |
| --- | --- | --- | --- |
| **Categories** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| CategoryName | nvarchar(50) | Primary Key | Category name |
| Description | nvarchar(500) | Not Null | Category description |

|  |  |  |  |
| --- | --- | --- | --- |
| **Admin** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| ID | int | Primary Key, Identity | Admin code |
| Name | nvarchar(50) | Not Null | Admin Name |
| Password | nvarchar(50) | Not null | Admin Password |

|  |  |  |  |
| --- | --- | --- | --- |
| **Customers** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| Name | nvarchar(50) | Primary Key | Customer Name |
| Phone | nchar(10) | Not null | Customer Phone |
| Address | nvarchar(50) | Not null | Customer Address |
| MSKH | nvarchar(50) | Not null | customer code |

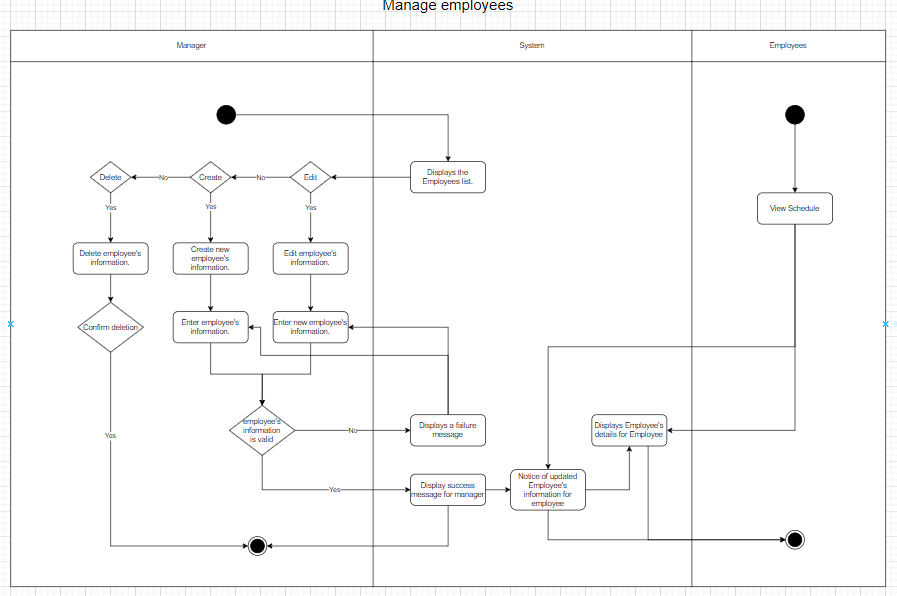
|  |  |  |  |
| --- | --- | --- | --- |
| **Order** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| OderID | int | Primary Key, Identity | Order code |
| MSKH | nvarchar(50) | Foreign Key, Not Null | Customer code |
| TotalPrice | float | Not null | push out the bill |

|  |  |  |  |
| --- | --- | --- | --- |
| **Order Detail** | | | |
| **Field Name** | **Data Type** | **Constraint  (Primary Key, Foreign Key, Identity, Not Null,...)** | **Description** |
| OderID | int | Primary Key,Foreign Key, Identity | Order code |
| ProductID | int | Primary Key,Foreign Key, Identity | Product code |
| Price | float | Not null | price of the product |
| Quantity | int | Not null | the number of products |

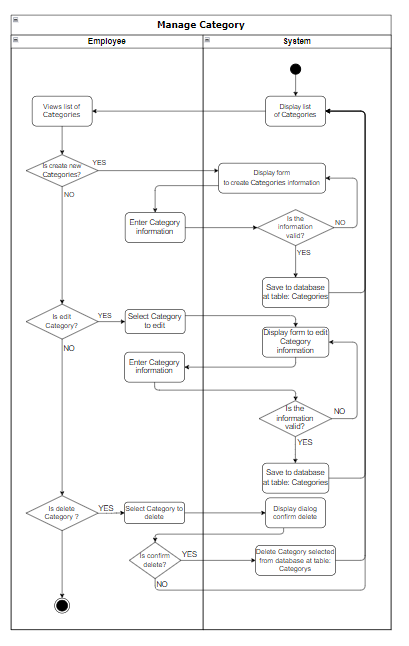
**PROCESS DESIGN DOCUMENT**

(Activity diagram, sequence diagram)

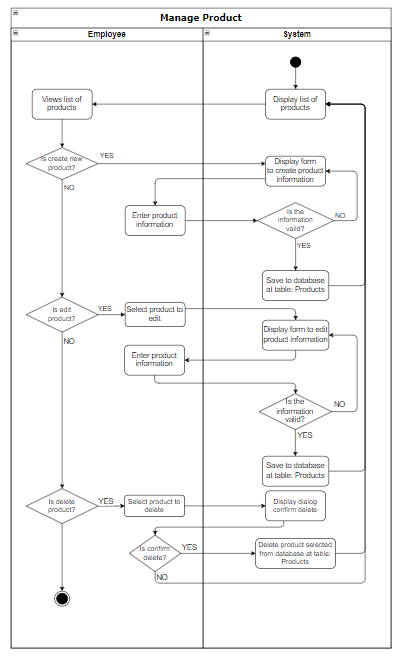
1. Process Name
   1. Employee manager.
   2. Category manager.
   3. Product manager.
   4. Customer manager.
   5. Manage import orders
2. Process Description and Details
   1. Employee manager.



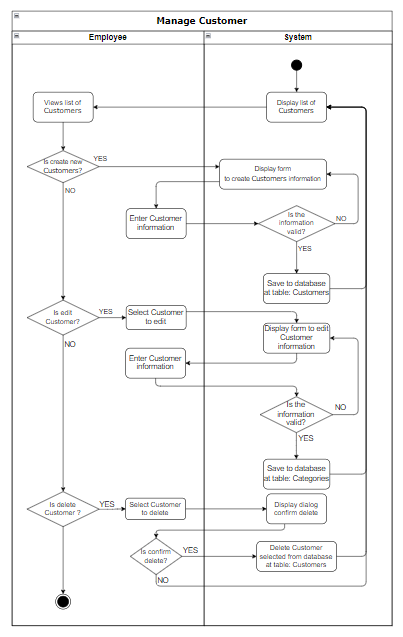
* 1. Category manager.



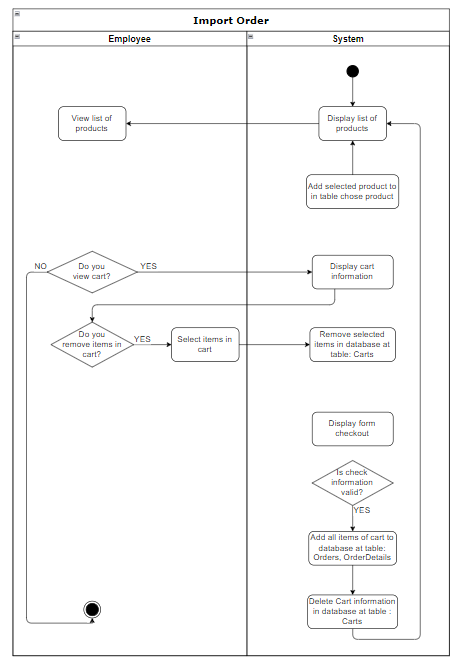
* 1. Product manager.



* 1. Customer manager.



* 1. import orders



**CODING STANDARDS DOCUMENT**

1. Programming Standards

1. General convention:

File Names: Do not use spaces in file names. Use only alphabetical characters (no numbers). File names should not exceed 40 characters.

2. File Suffixes:

File Type, Suffix, Java source, Java byte code,.java, .class

3. Naming Java source files (.java)

General conventions: Combine the management function, main content of the file, and the programming function in the name.

<Management function> + <Main content of the file> + <Programming function>

Standards for code writing style

1. Standards for code writing style

A file consists of multiple sections, which should be separated by blank lines and optional comments to identify each section.

Avoid writing files longer than 2000 lines.

1. Standards for declaring Variables

1. Line length:

Avoid writing lines longer than 80 characters.

Note: Code snippets used in reference materials should have shorter lengths, typically less than 70 characters.

2. Curly braces:

The opening curly brace "{" should be placed at the end of the line., the closing curly brace "}" should be placed on a new line.

3. Indentation:

Lines at the same "level" should have consistent indentation, indentation levels should be evenly spaced.

4. Line breaks:

when a single expression exceeds a single line, break it according to the following rules: break after a comma, break before an operator., higher-priority break conventions take precedence., the new line should be indented by 2 tabs.

### 5. General Rules

* Case Sensitivity: Java variable names are case-sensitive.
* Start with a Letter: Variable names should start with a letter, a currency character ($), or an underscore (\_). They should not start with a number.
* No Special Characters: Avoid special characters such as @, #, -, etc., except for $ and \_.
* Keywords: Do not use Java reserved keywords (e.g., class, int, public) as variable names.

1. Standards for Function Declarations

Annotations should be used to provide general information about code and provide additional information that may not be apparent just by reading the code. Annotations should only contain information relevant to understanding the program. For example, information about how a package is created and which directory it is located in should not be included in annotations.

1. Other Standards

**TASK SHEET**

1. Project Title:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No.** | **Task** | **Actual Start Date** | **Actual Days** | **Teammate Name** | **Status** |
|  | Manager employees | 22/4/2024 | 7 | Bui Trung Nguyen | Finished |
| Manager categories | 29/4/2024 | 7 | Bui Trung Nguyen | Finished |
| Manager products | 6/5/2024 | 14 | Bui Trung Nguyen | Finished |
| Manager customer | 20/5/2024 | 3 | Bui Trung Nguyen | Finished |
| New order | 24/5/2024 | 3 | Bui Trung Nguyen | Finished |
| Manager orders | 27/5/2024 | 13 | Bui Trung Nguyen | Finished |

*Note: You can use list or Gartt*

**TESTING DOCUMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Test Case Name** | **Task** | **Result** | |
| **Expected** | **Actual Result**  Pass/Fail-Comments |
|  | Add employee details | Enter employee details  MSNV :A21176  Fullname: Nam Viet  Phone:0234234534  Address: Vinh Long  IMG : icon.png  Password: admin@123 | Record is added to the database | Pass |
|  | Add New Category | Enter category detail:  Name : Mechanical Keyboard  Descriptions:CNC Aluminum Mechanical Keyboard | Record is added to the database | Pass |
|  | Add Product | Enter Product detail  Name: Feker ik 75 pro  Price: 29,51  Quantity:100  Select Category: Mechanical Keyboard  Img:product1.png  Description:test | Record is added to the database | Pass |
| 4 | Add Customer | Enter Customer detail  Name:Bui Trung Nguyen  Phone 0366599761  MSkH:MK2134  Address:Can Tho | Record is added to the database | Pass |
| 5 | Add Order | Enter order detail  Select customer to place order:MD124 Select the product to order:RAINY75  Select the buy button to complete the order | Record is added to the database | Pass |

**PROJECT REVIEW**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Project Plan Activity/Milestone** | **Work Specification** | **Status of the Activity** | **Remarks** | **Responsibility** |
| 22/4/2024 | Analyze requirements and assign work | All Form | Done | Good | All Teammate |
| 2/5/2024 | Discuss and resolve the app's functionality | App functionality | Done | Good | All Teammate |
| 9/6/2024 | Complete the work | Code | Done | Good | All Teammate |

**FINAL CHECK LIST**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Aspect tested** | **Suggestion/Remarks** |
|  | Have all the modules been property integrated and are they completely functional? | Yes |
|  | Are all the validations happening as specified in Process Design? | Yes |
|  | Have all Design and Coding standards been followed and implemented? | Yes |
|  | Have all the modules been property integrated and are they completely functional? | Yes |
|  | Are all the validations happening as specified in Process Design? | Yes |
|  | Have all Design and Coding standards been followed and implemented? | Yes |
|  | Have all the modules been property integrated and are they completely functional? | Yes |
|  | Are all the validations happening as specified in Process Design? | Yes |

shopping procedures become faster, helping to increase productivity when selling store products.