|  |
| --- |
| Nowadays, many organizations manage canteen systems still use hand writing to manage the list of items, food, bills... it really is a pretty heavy task. So we decided to create an app to better management. Application provides a user-friendly system using active facilitate rapid and effective for employees in the canteen. App provides support functions employees in add food at the dinner table and invoice management as well as the prices of the products of the canteen. Employees can add new items, add new products, the list of foods of product status, detailed statistics revenues in day/week/month, add new employees and  detailed statistics of imported products and export of the day/week/month so the warehouse management one convenient way. |

**Client/Project Undertaken:**

1. **List of input and output**

* **List of Employee**

**• Create Employees:**

**- Input:** Full name, Username, Password, Address, Birthday, Phone, Email, Type Employee, Area.

**- Output:** Notification and redirect to login form.

**• Login:**

**- Input:** Username, Password.

**- Output:** Notification and redirect to home.

**• Update Employees:**

**- Input:** Full name, Username, Password, Address, Birthday, Phone, Email, Type Employee, Area.

**- Output:** Notification and redirect to home.

* **List of Sell**

**• Create tables:**

**- Input:** Quantity.

**- Output:** Show list tables and status.

**• Orders:**

**- Input:** Food code, Food name, Quantity food, Employee ID, Table code.

**- Output:** Notification and redirect to PayBill form.

**• PayBill:**

**- Input:** Orders information.

**- Output:** Redirect to home.

**• Search Orders:**

**- Input:** Input date.

**- OutPut:** Show Orders information.

**• Total:**

**- Input:** Input date.

**- OutPut:** Show total money, total order .

* **List of Product:**

**• Distributor:**

**- Input:** Name, Address, Phone, Email.

**- Output:** Send message.

**• Product:**

**- Input:** Name, Unit, Price.

**- Output:** Send message.

**• Import product:**

**- Input:** Quantity, Product name, Price.

**- Output:** Send message.

* **List of Food:**

**• Category:**

**- Input:** Category food.

**- Output:** Show list category food.

**• Food:**

**- Input :** Name, Quantity, Price.

**- Output:** Show notification.

**• Promotion:**

**- Input:** Name, Begin time, End time.

**- Output:** Show notification, show detail promotion.

1. **Overview of processes involved in the system**

* **Employee manage:**

• **Create employee :** Input information and create account.

• **Login:** Once having an account registed, users can be able to sign in for using all member’s functions.

**• Changing account:** After signed in, user can change their personal information.

* **Sales Manager:**

• **Create Orders:** After login, user can create Orders.

• **Pay bill:** After create Orders, user can pau bills.

• **Search Orders:** Search and show detail Orders.

**• Total.**

* Food manage:

• **Add Food :** After login, user can add food.

• **Food manage:** Food management, quantity of food and distribution of food.

• **Promottion**: Create promotion, Manage start time and end of promotion.

* **Product mange:**

• **Distributor:** Distributor management.

• **Product mange:** Product management, product quantity and inventory.

1. **Hardware and software required for implementing the project**

• **Hardware:** Intel - compatible processor with a minimum speed of 1 GHz or a faster proccessor, minimum 1 GB ram , 100 MB of disk space.

**• Software:** Java Runtime Environment 1.8 or over.

|  |  |
| --- | --- |
| **Sr.No** | **Customer’s Acceptance Criteria** |
| 1. | Admin can manage user accounts, backup and recovery data. |
| 2. | Program can execute without problems. |
| 3. | Interface is user-friendly. |
| 4. | Manager can set permission for members. |
| 5. | Search data easily. |
| 6. | Ensure data accuracy’s. |
| 7. | Generate the bills. |

**Project Plan**

* **Project Details**

• **Name of Client:** Canteen Manage.

• **Date of Project Plan:** 27/02/2017

• **Project Vision/Objectives:** CanteenManage directed to a software with a user-friendly interface is fully functional in service demand manage.Convenient and fast operation.

* **Scope:**

- CanteenManage provides convenient and fast management.

- The software provides functionality for creating and managing employee accounts. And update, edit, delete list employee.

- Log in and log out of software.

- Show list table and set status for table.

- Provides functionality for customers to order and pay invoices.

- Software can update, edit, delete, search orders.

- Show detail orders.

- Calculate total amount and total order quantity during the day.

- Create food and category. Show list food.

- Create promotion, notify start time and end time.

- Update, edit, delete food and promotion.

- Distributor management. - Product management.

- Show list product, update, edit, delete product and distributor.

* **Our understanding of the Client organization:** ABC Company is a fast food company that needs software to manage its products.
* **Project Organization with Responsibilities and Authorities:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Member** | **Role** | **Responsibilities** |
| 1 | Nguyen Thanh Nhan | Leader | Assign jobs to members. Design database. |
| 2 | Huynh Van Man | Design, programer | Program interface. |
| 3 | Tran Phuoc Vinh | Design, programer | Analyze customer requirement. |
| 4 | Tran Thanh Lac | Design, programer | Program interface. |

* **Project Initiation/Requirement Documents:**
  + Information Employee.
  + Information Orders.
  + Information Distributor.
  + Information Product.
  + Information Promotion.
  + Information Food.
* **Deliverables:**
* Design document.
* Installation manual.
* User manual.
* **Project Dependencies:**

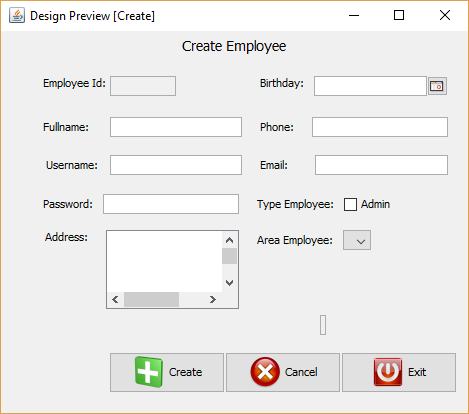
**-** Limite of knowledge and number of members.

* **Major Project Milestones:**
* Define Problem: 27/02/2017 – 1/3/2017
* Analyze customer requirement: 05/03/2017 – 08/03/2017
* Design phase: 10/03/2017 – 15/03/2017
* Development phase: 15/03/2017 – 20/03/2017
* Testing: 20/03/2017 – 22/03/2017
* **Quality Plan:**
* Review Activities
* Claim client information.
* Update information and request from client.
* Exchange, get idea from team members.
* Coding.
* Testing Activities
* Check database.
* Check syntax error.
* Check exception and forms.
* Backup and recovery strategies (in case of disk crash, network failures)
* Daily check database.
* Backup database weekly.

**Document Design**

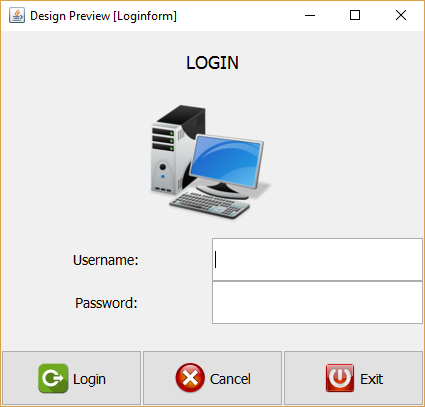
|  |  |
| --- | --- |
| **Prototype** | **Value** |
| Background color | [0,102,51],While, [73,37,1] |
| Title-Font Size | 13px |
| Title-Font Color | #003555 |
| Title-Font Style | 13px/150% Times New Roman ,Verdana, Arial, Helvetica, sans-serif |
| Title-Alignment | Left |
| Background color of the Controls on the form | #FFFFFF |
| Foreground color of the Controls on the form | #FFFFFF |
| Control Caption Font Size | 13px |
| Control Caption Font Color | 13px |
| Control Caption Font Style | 12px/150% Times New Roman ,Verdana, Arial, Helvetica, sans-serif |
| Control Caption and controls-Alignment | Left |
| Command button Alignment | Left |
| Size | 1280x720 |

**List of forms to be created**  
  
**1. Create Employee.**



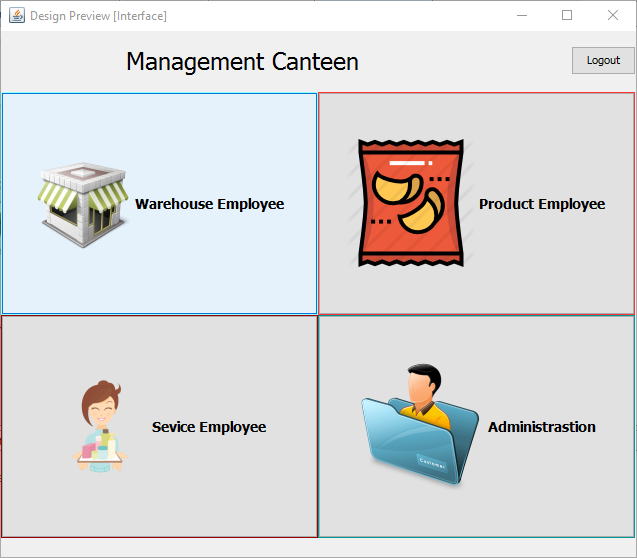
* **Controls :** 7 textfield, 1 textarea, 1 radiobutton, 1 combobox, 3 button. All controls must be enter

**2. Login.**



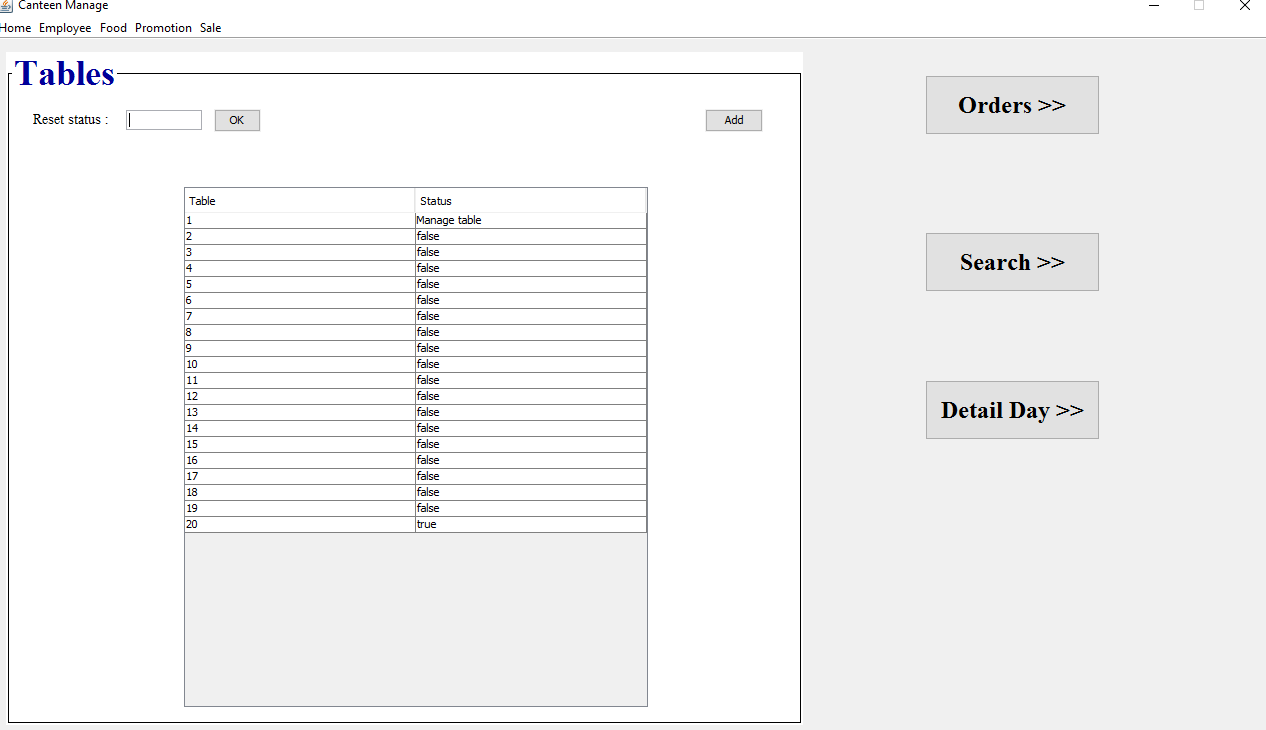
* Controls: 2 textfield for Username and Password. All controls must be enter.

**3. Home.**

****

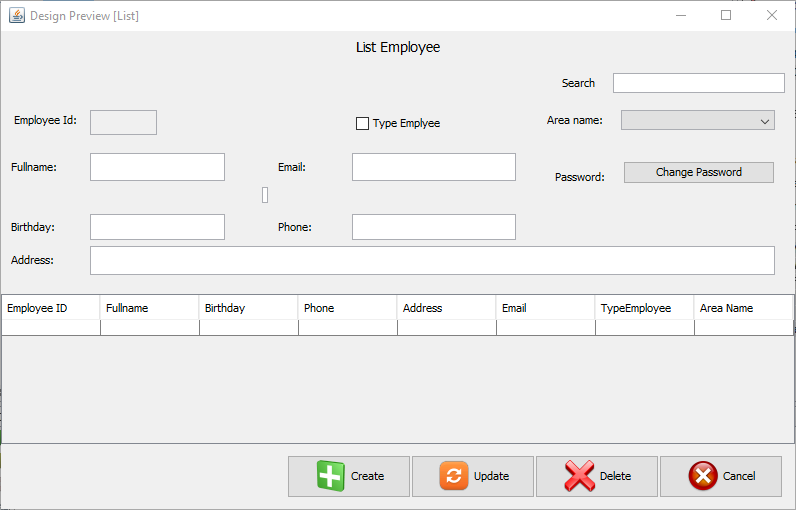
* Controls: 4 button. All controls must be enter.

**4. List Table.**



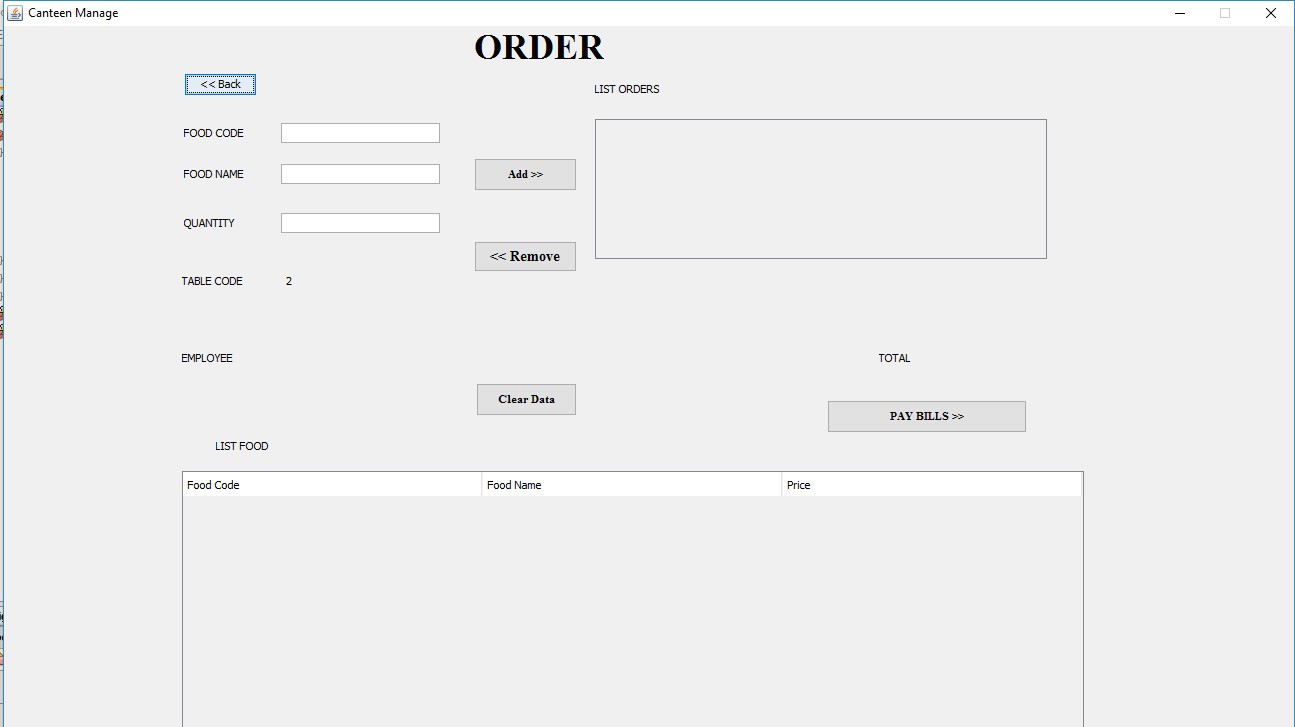
* Controls : 1 jtable, 1 texfield, 2 button. View list table. All controls must be enter.

**5. View Employee.**



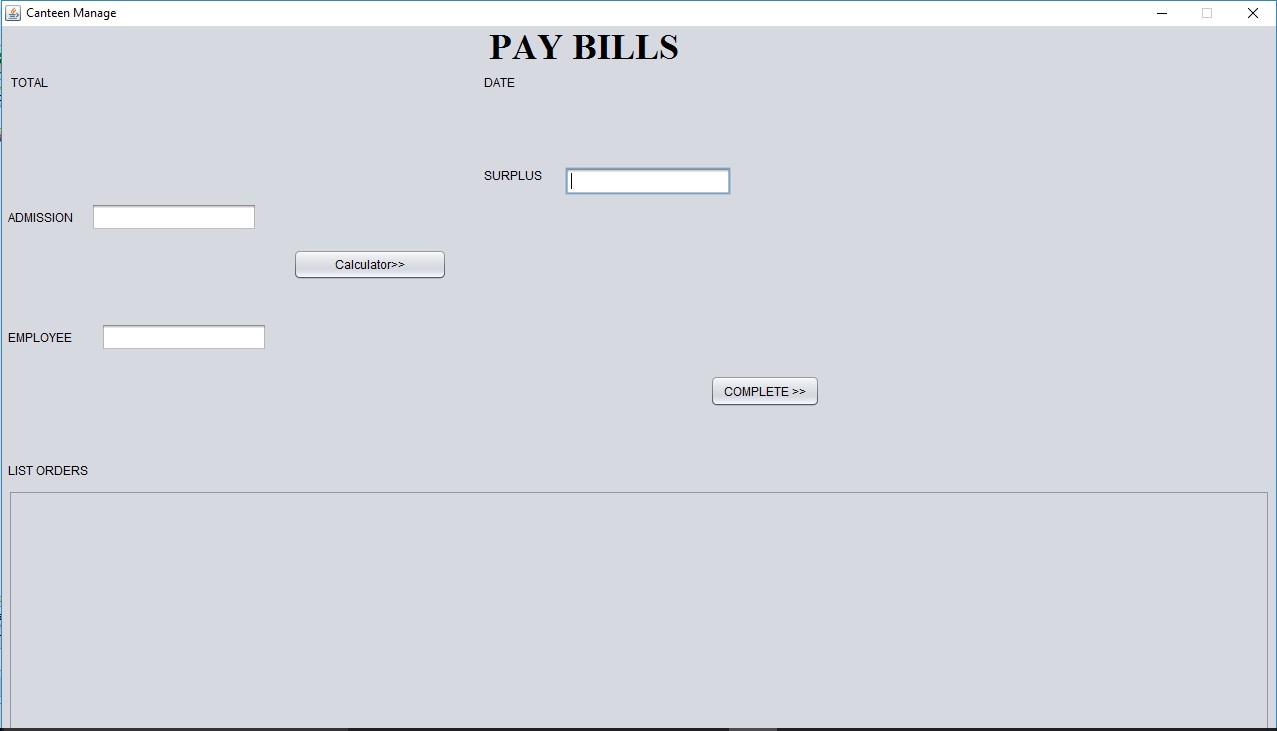
* Controls: 6 textfield, 1 jtable, 1 button, 1 radiobutton, 1 combobox. Show list and detail Employee.

**6. Create Orders.**



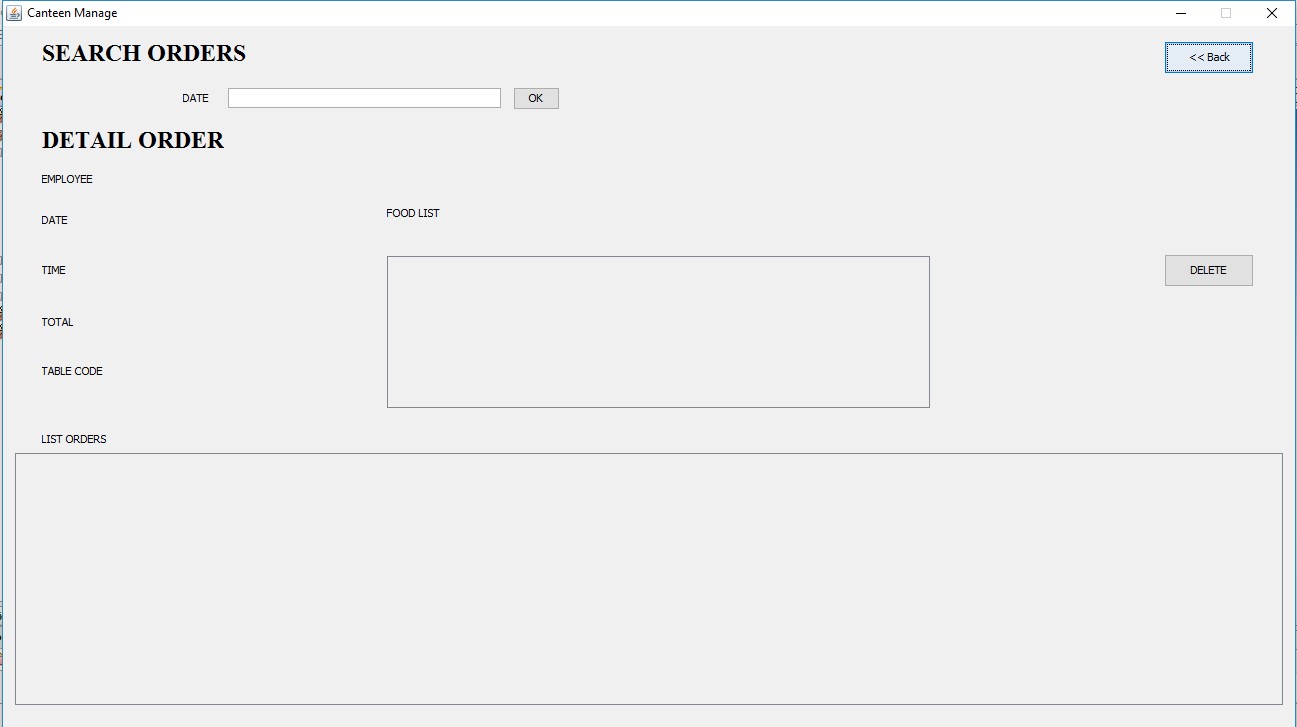
* Control: 3 textfieid, 5 label, 1 textarea, 1 jtable, 4 button. All controls must be enter.

**7. Pay Bills.**



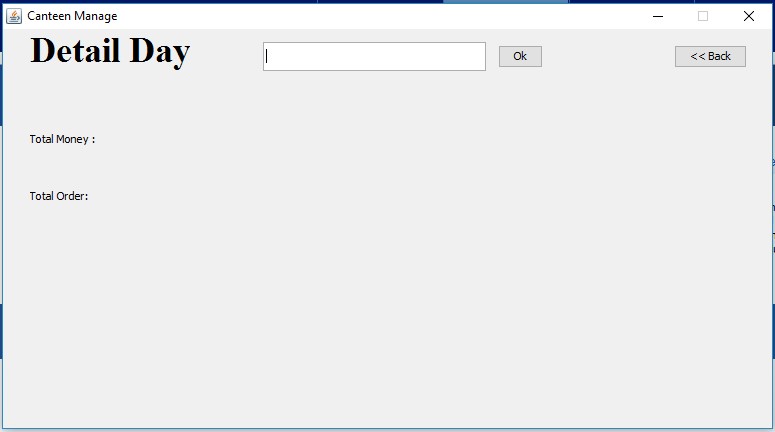
* Controls: 3 textfield, 4 label, 1 jtable, 2 button. All controls must be enter.

**8. Search Orders.**



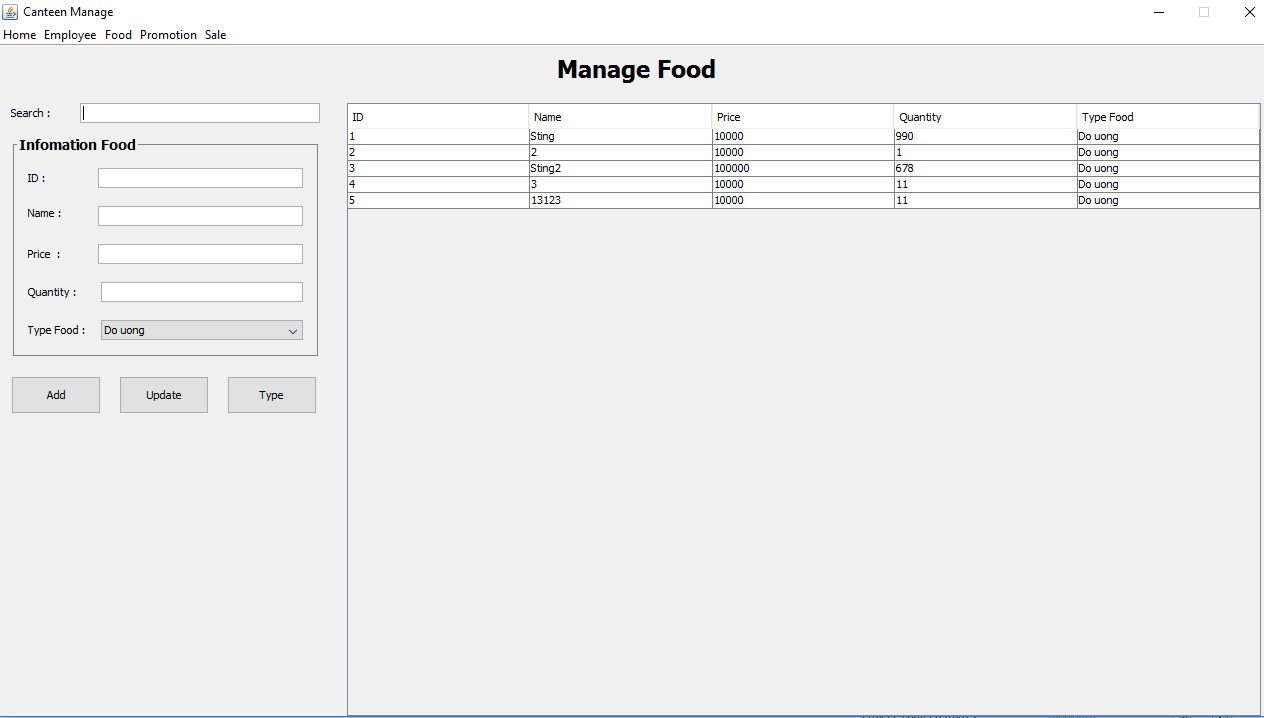
* Controls: Textfield, label, jtable, button. All controls must be enter. Show detail Orders.

**9. Detail Day.**



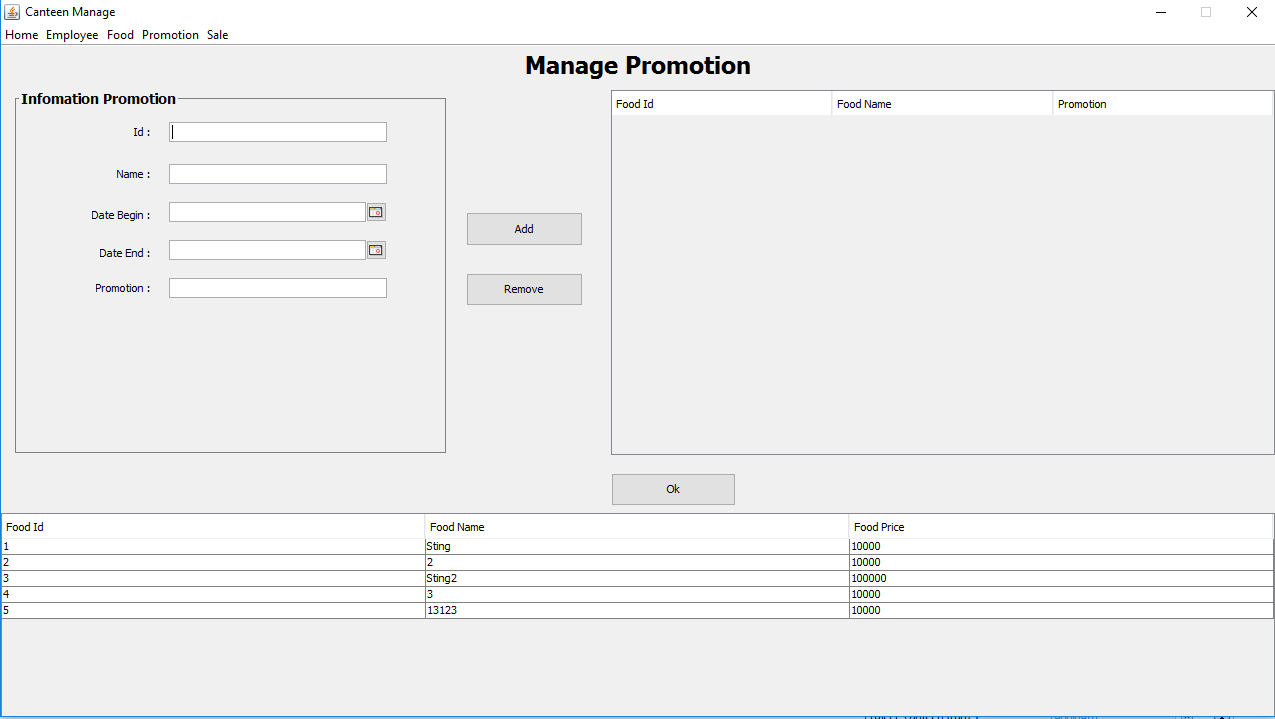
* Controls: 1 textfield, 2 button, 2 label. Show detail day.

**10.Manage Food.**



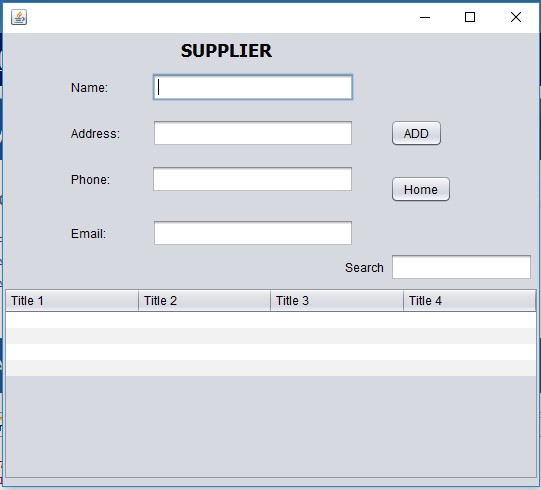
* Controls: 3 button, 5 textfield, 1 combobox,6 label, 1 jtable.

**11.Manage Promotion.**



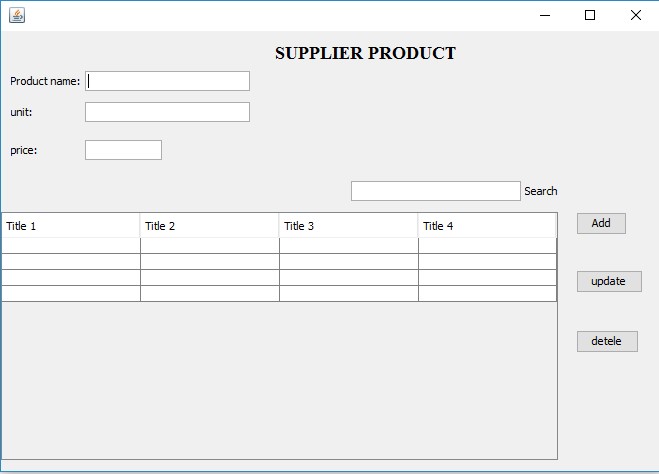
* Controls: 2 jtable, 3 button, 3 textfield, 2 jDateChoose

**12.Supplier.**



* Controls: 5 textfield, 1 jtable, 2 button. Add and search Supplier.

**13.Manage Product.**



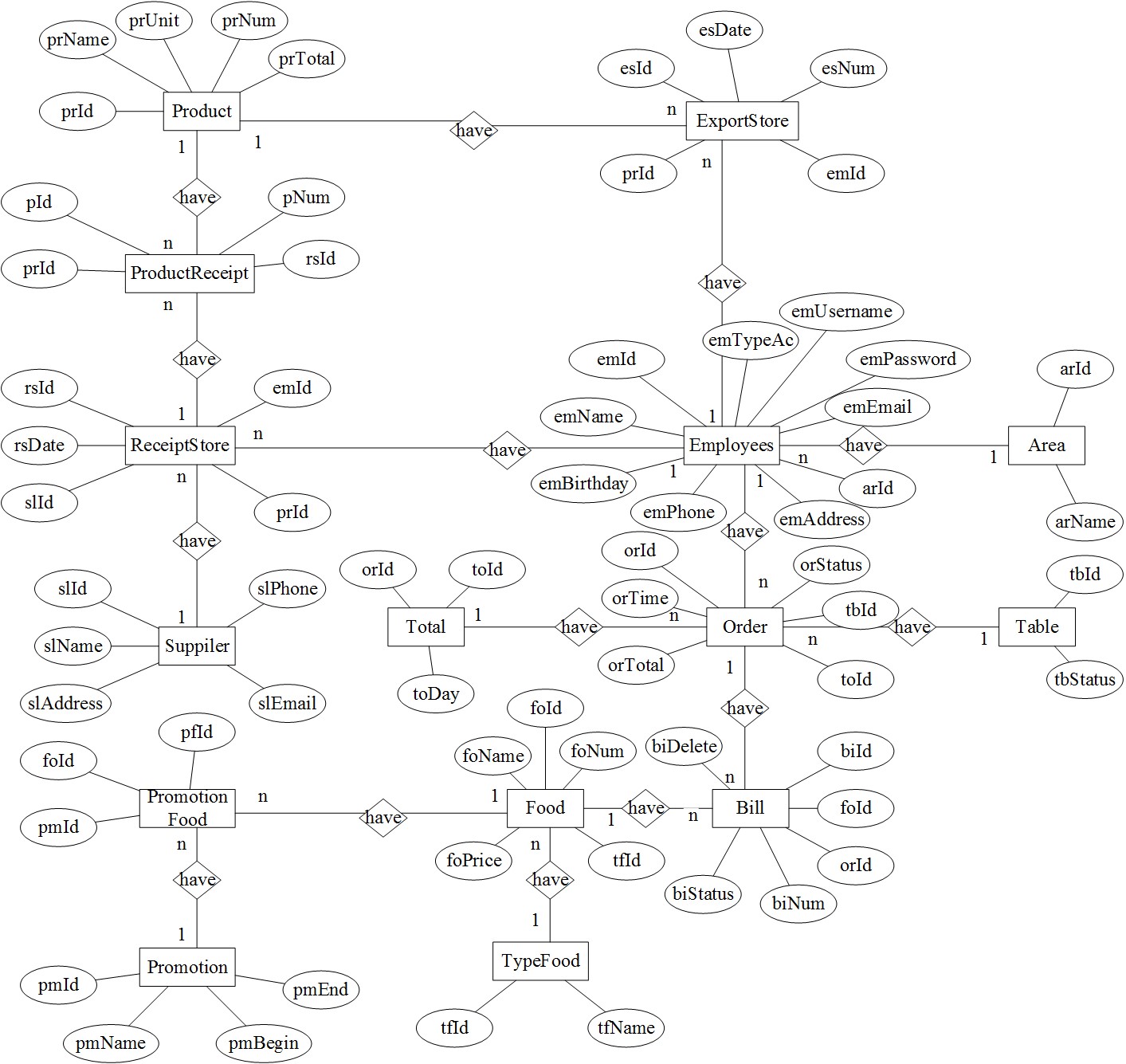
* Controls: 4 textfield, 3 button, 1 jtable. Create, edit, delete product.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name of the Table** | **Table Description** | **Number of**  **Fields** | **Primary Key** | **Related Tables** | **Foreign Key** |
| Area | Employee area | 2 | arId |  |  |
| Employee | Information Employee | 10 | emId | Area | arId |
| Orders | Information Orders | 6 | orId | Table | tbId |
| Bill | Detail Orders | 3 | biId | Order, Food | orId,foId |
| Tables | Quantity and status table | 2 | tbId |  |  |
| Total | Sum money | 2 | toId |  |  |
| Food | Information Food | 5 | foId | TypeFood | tfId |
| TypeFood | Category Food | 2 | tfId |  |  |
| Promotion | Information Promotion | 4 | prId |  |  |
| PromotionFood | Detail Promotion | 3 | pfId | Food, Promotion | foId, pmId |
| Product | Information Product | 5 | prId |  |  |
| ExportStore |  | 5 | esId | Product | prId |
| ReceiptStore |  | 5 | rsId | Product,  Suppiler | prId,  slId |
| Suppiler | Information Distributor | 5 | slId |  |  |

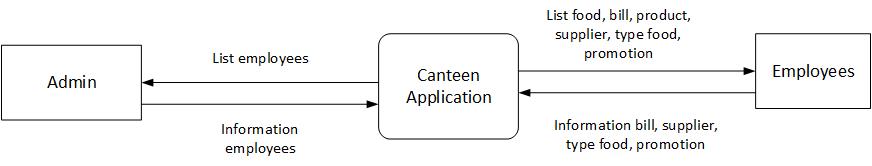
|  |  |  |
| --- | --- | --- |
| **Table Name** | **Colum Name** | **Data Type** |
| Area | arId | Int |
| arName | Nvarchar(200) |
| Employee | emId | Int |
| emName | Nvarchar(200) |
| emBirthday | Date |
| emPhone | Varchar(15) |
| emAddress | Nvarchar(250) |
| emEmail | Nvarchar(200) |
| emUsername | Varchar(200) |
| emPassword | Varchar(200) |
| emTypeAc | Bit |
| arId | Int |
| Supplier | slId | Int |
| slName | Nvarchar(200) |
| slAddress | Nvarchar(200) |
| slPhone | Varchar(15) |
| slEmail | Varchar(200) |
| Product | prId | Int |
| prName | Nvarchar(200) |
| prUnit | Varchar(20) |
| prNum | Varchar(200) |
| prTotal | Varchar(200) |
| ReceiptStore | rsId | Int |
| rsDate | Date |
| slId | Int |
| emId | Int |
| prId | Int |
| ExportStore | esId | Int |

|  |  |  |
| --- | --- | --- |
|  | esDate | Date |
| esNum | Varchar(200) |
| emId | Int |
| prId | Int |
| Type | tfId | Int |
| tfName | Nvarchar(200) |
| Food | foId | Int |
| foName | Nvarchar(200) |
| foPrice | Varchar(200) |
| foNum | Varchar(200) |
| tfId | Int |
| Promotion | prId | Int |
| prName | Nvarchar(200) |
| prBegin | Date |
| prEnd | Date |
| PromotionFood | pfId | Int |
| pfPromotion | Varchar(20) |
| foId | Int |
| prId | Int |
| Tables | tbId | Int |
| tbStatus | Bit |
| Orders | orId | Int |
| orTime | Time |
| orDate | Date |
| orStatus | Bit |
| emId | Int |
| orTotal | Varchar(200) |
| tbId | Int |
| Bill | biId | Int |
| biNum | Int |
| biStatus | Bit |
| biDelete | Bit |
| foId | Int |
| orId | Int |
| Total | toId | Int |
| today | Varchar(200) |
| orId | Int |

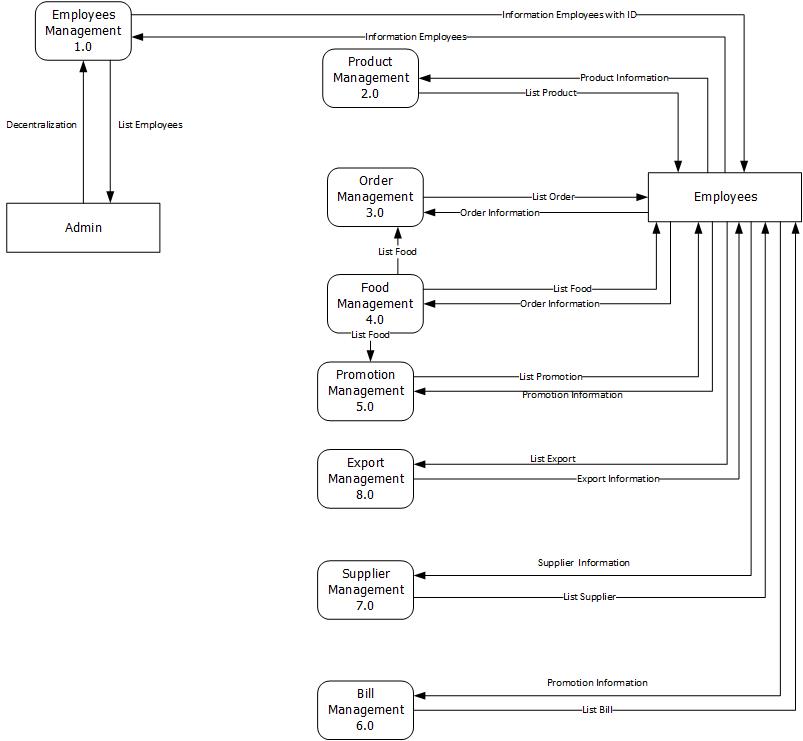
* **E-R :**



* **Data flow diagram :**

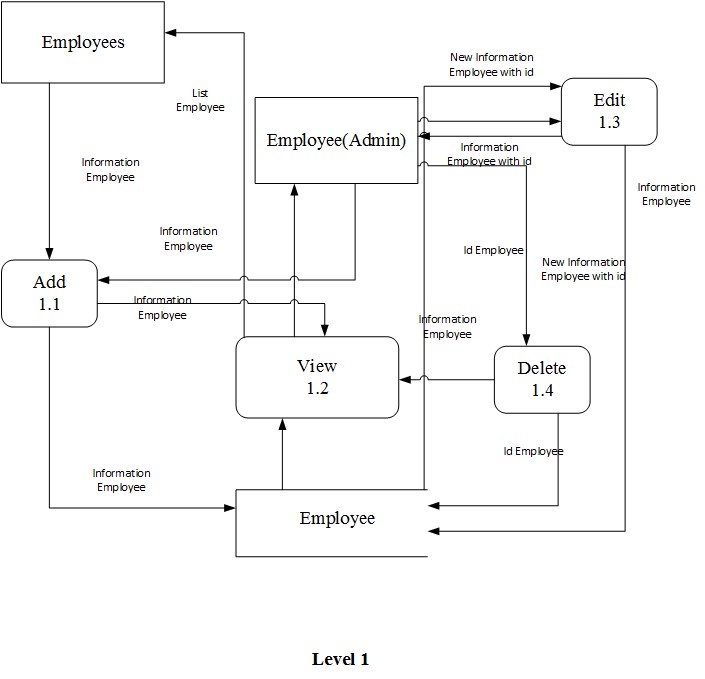


**Level Content**

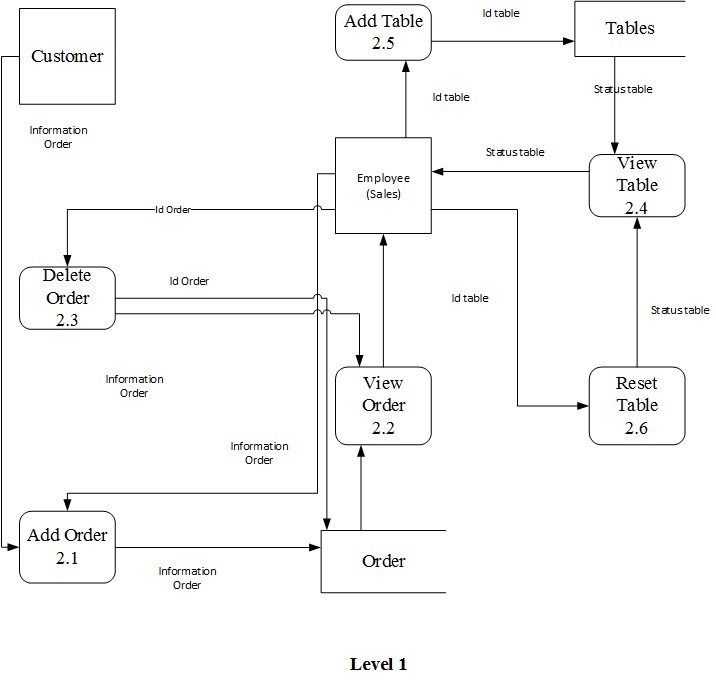


**Level 0**

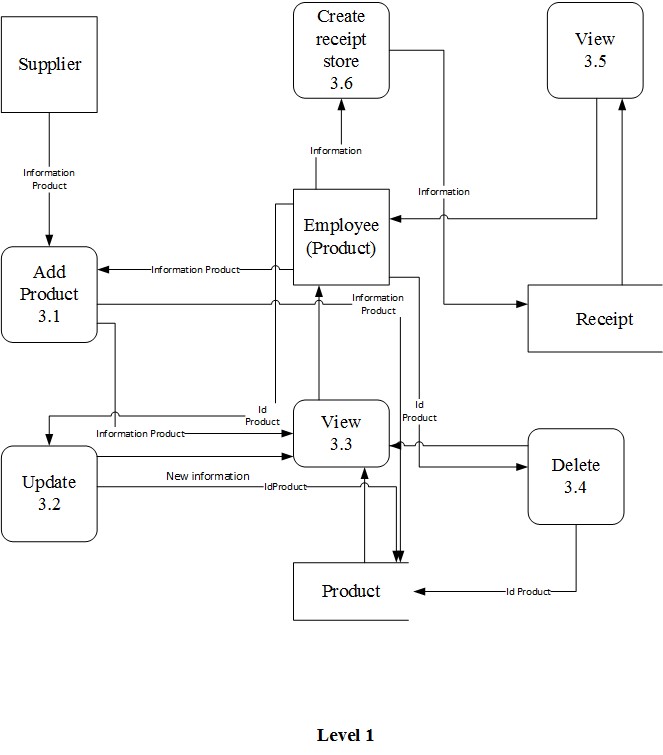
* **Data flow diagram employee:**



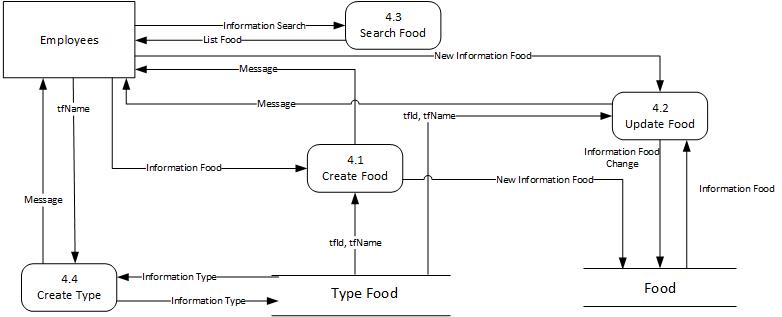
* **Data flow diagram sales:**



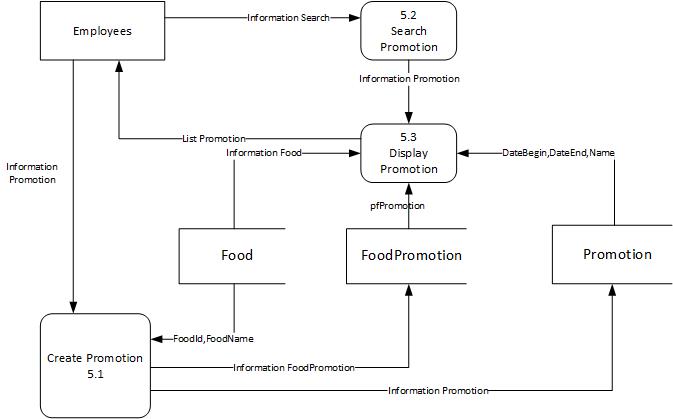
* **Data flow diagram product:**



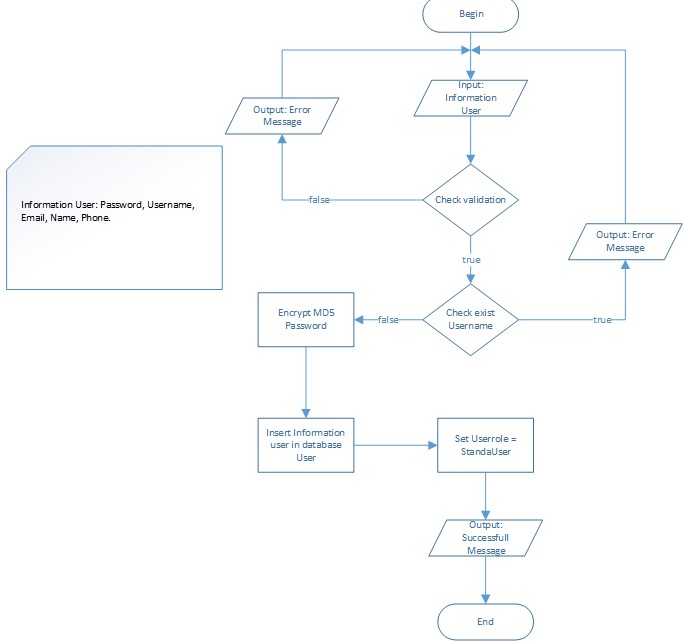
* **Data flow diagram food:**



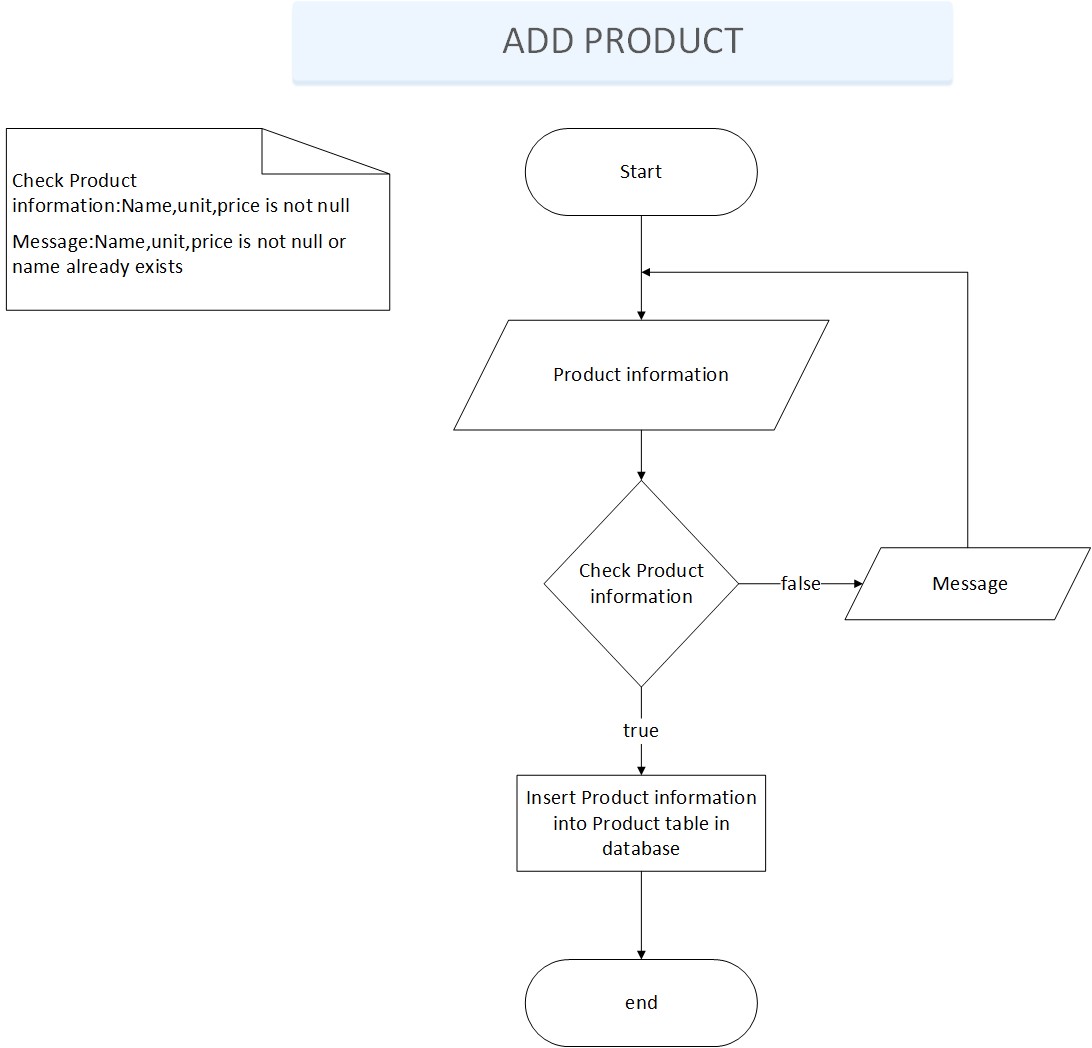
* **Data flow diagram promotion:**



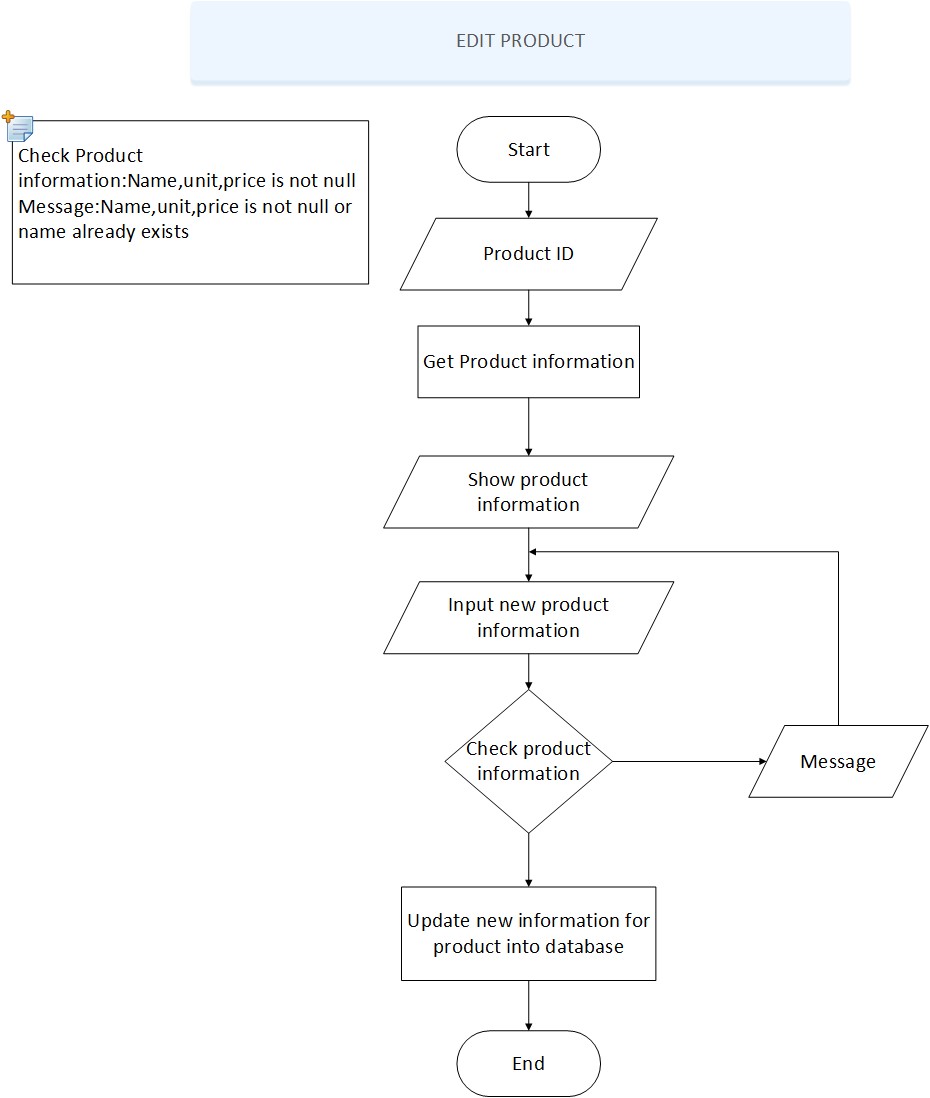
1. **Create Acount Employee.**



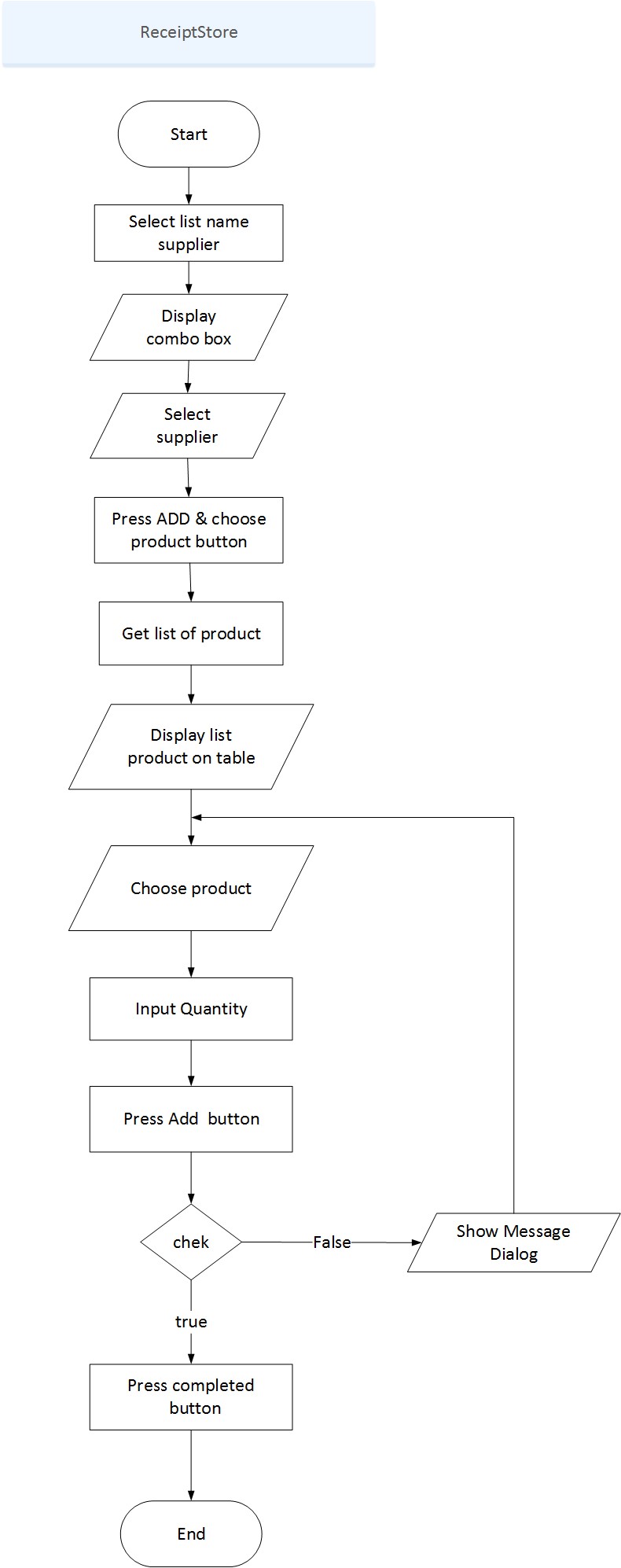
1. **Add Product.**



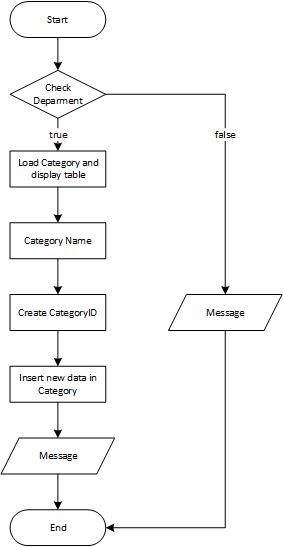
1. **Edit Product.**



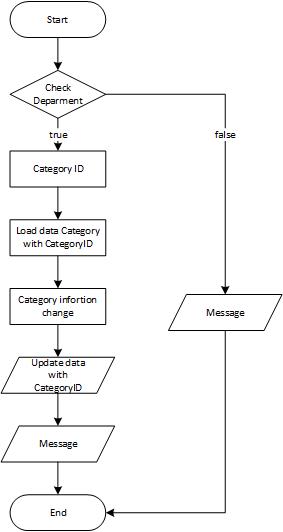
1. **ReceiptStore.**



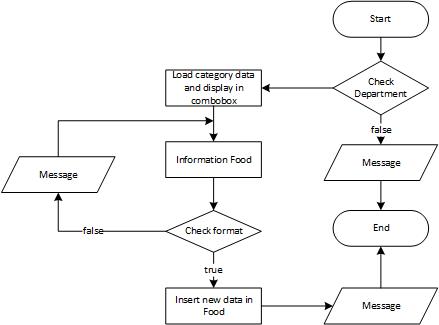
**5. Create Category.**



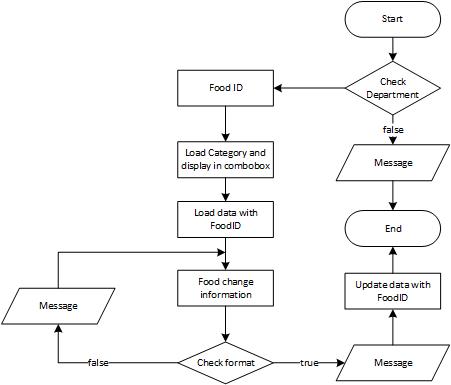
**6.Edit Category.**



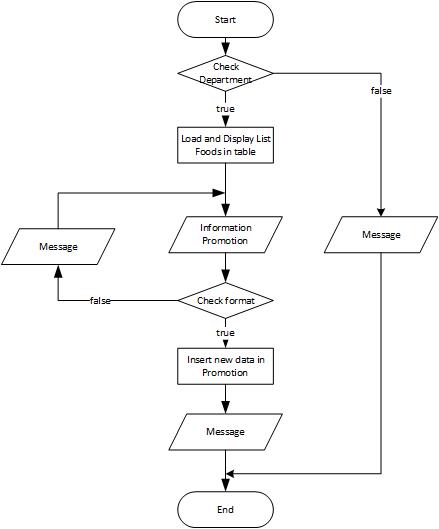
**7. Create Food.**



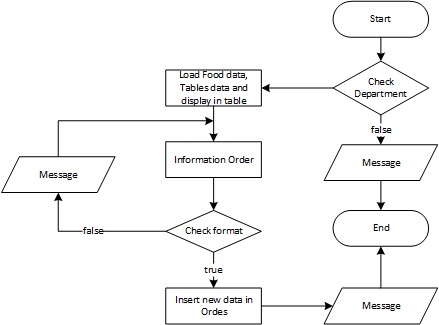
1. **Edit Food.**



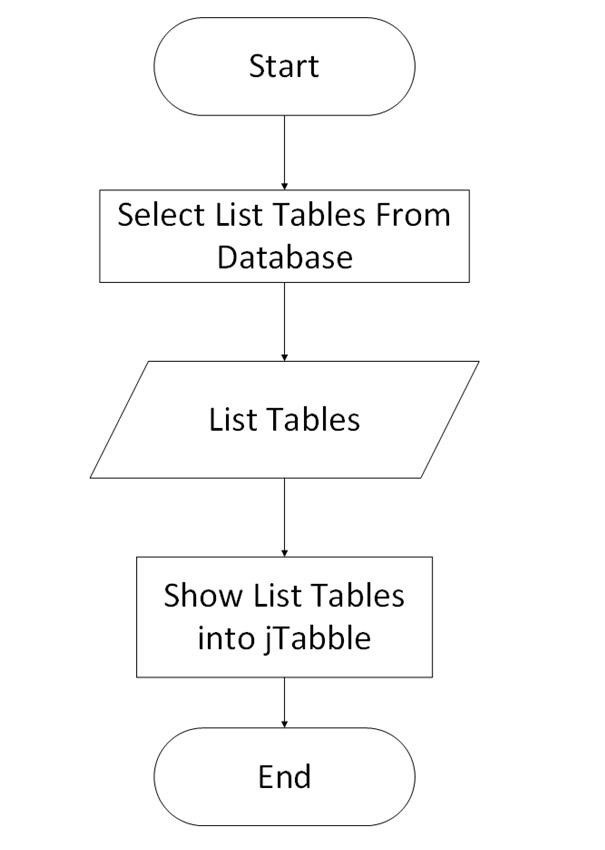
1. **Create Promotion.**



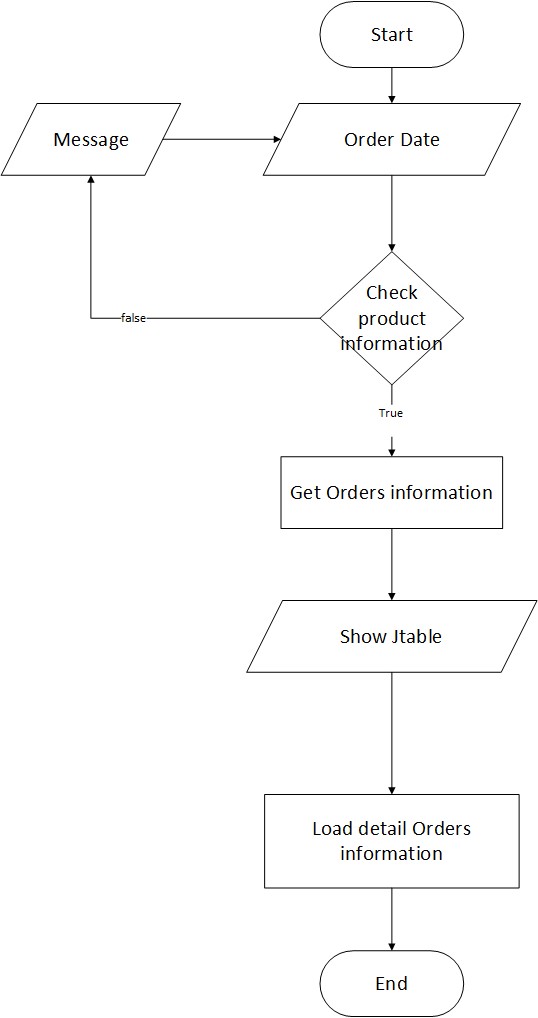
1. **Add Order.**



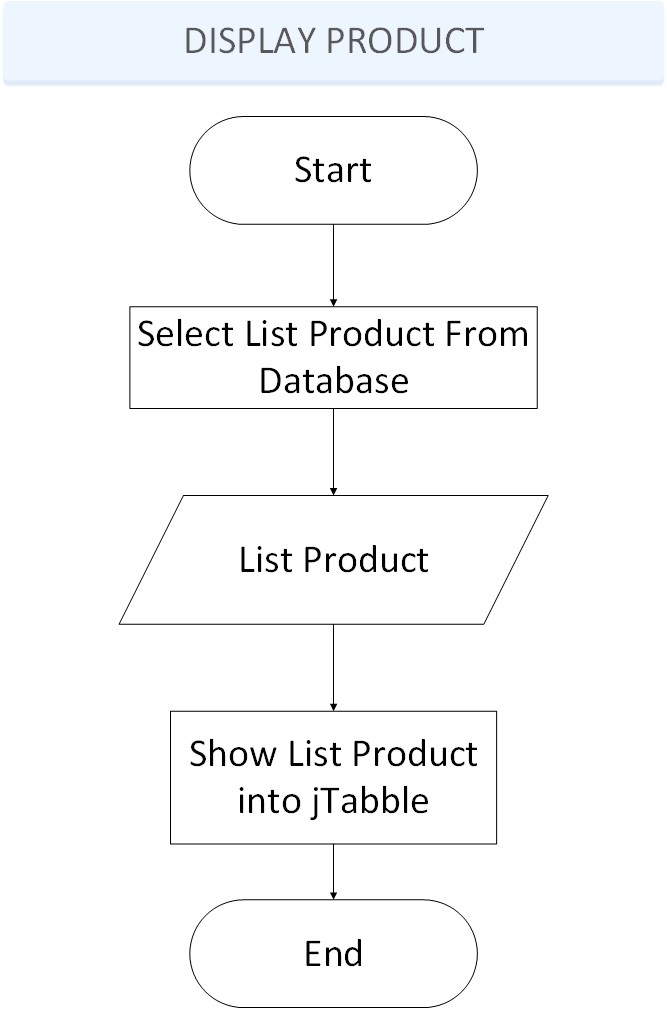
1. **View Table.**



1. **Search.**



**13.View Product.**



14.Login Employee



15.Add Emoloyee



16.Update Employee



17.Delete Employee



18.Change Password Employee



19.Display list Employee



1. **Programing Standars**

- Coding standards, when implemented correctly, can provide huge advantages to the produced code and the companies implementing them. They can make a great difference when maintaining software. Maintenance is arguably the most important step in the Software Development Life Cycle (SDLC) and is definitely the most recurring. When the developers who are responsible for maintaining the code are different from the ones who produced it, it relies on them understanding code developed by another developer.

1. **Standars for code writing style**

- Code must be clear, concise.

1. **Standars for declaring Variables**

- Package name should use only lower case letters and digits and no underscores.

- Names should start with uppercase letter, and capitalize the first letter of subsequent words as well any letters that are part of acronym. All other characters in the name have to be lower case. Avoid using underscores for separating words.

- Same as Class.

- Same as Method. However variable names should be nouns or noun phrases.

1. **Standars for Function Declarations**

- Names should start with a lowercase letter. Capitalize the first letter of subsequent words as well as any letters that are part of an acronym. Method names should be verbs or verb phrases.

1. **Orther Standars**

|  |  |
| --- | --- |
| **Prefix** | **Control Description** |
| txt | textField |
| are | textArea |
| lb | Label |
| c | checkbox |
| r | radioButton |
| btn | button |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project Ref. No.:** | **Project Title:**  **MangaGalax**  **y** | **Activity Plan Prepared By:** | | **Date of Preparation of Activity Plan:** | | |
| **Task**  **Sub**  **divition** | **Description** | **Planned**  **Start**  **Date** | **Actual**  **Start**  **Date** | **Actua**  **l Days** | **Team**  **Membe**  **r**  **Names** | **Status** |
| Analyze and  create database |  | 28/02/201 7 | 02/03/201 7 | 3 | Nguyen  Thanh  Nhan | Complet  e |
| Design interface of program |  | 02/03/201 7 | 05/03/201 7 | 3 | Huynh  Van  Man | Complet  e |
| Design flowchart |  | 05/03/201 7 | 07/03/201 7 | 2 | Tran  Phuoc  Vinh | Complet  e |
| Form  Manage  Employe  e | Create,  Login, Edit, Delete | 07/03/201 7 | 12/03/201 7 | 5 | Tran  Thanh Lac | Complet  e |
| Form  Sale  Manage | Add order, Edit, Delete | 08/03/201 7 | 12/03/201 7 | 4 | Nguyen  Thanh  Nhan | Complet  e |
| Form  Food  Manage | Add food, Edit | 10/03/201 7 | 15/03/201 7 | 5 | Huynh  Van  Man | Complet  e |
| Form  Product  Manage |  | 10/03/201 7 | 16/03/201 7 | 6 | Tran  Phuoc  Vinh | Complet  e |

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Features Tested** | **Remarks** |
| 1 | Test user interface | Good |
| 2 | Test database | Good |
| 3 | Test functions of software | Good |
| 4 | Test activities | Good |
| 5 | Test final | Good |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Project Plan**  **Activity/Milest one** | **Work**  **Spectificatio n** | **Status of the**  **Activit y** | **Remark**  **s** | **Responsib**  **ility** |
| 27/02/2017 | Get user information | Exchange with customers | Finish | Good |  |
| 02/03/2017 | Design | Group meeting | Finish | Good |  |
| 05/03/2017 | Coding | Coding | Finish | Good |  |
| 20/03/2017 | Test | Group meeting | Finish | Good |  |

|  |  |  |
| --- | --- | --- |
| **Sr.No.** | **Aspect tested** | **Suggestion/Remarks** |
| 1. | Have all the modules been properly intergrated and are they completely functional? |  |
| 2. | Does each unit meet its objective and purpose?  Are all the validation happening as specified in Process Design? |  |
| 3. | Have all Design and Coding standards been followed and implemented? |  |
| 4. | Is the GIU design consistent all over? |  |
| 5. | Are the codes working as per the specification? |  |
| 6. | Does the application’s functionality resolve the client problem, and satisfy his needs completely? |  |
| 7. | Have the hardware and software been correclly chosen? |  |
| 8. | Additional feature and utilities that give value addition to the entire project. |  |