|  |
| --- |
| HÃ¬nh áº£nh cÃ³ liÃªn quan |
| **GROUP 03**  **MILK TEA SHOP** |
|  |
| Can Tho, November 2019 |

|  |  |  |
| --- | --- | --- |
| **Design Plan:** | **Document name: Problem Definition Document** | **SWP/Form No.1** |
| **Effective Date:** | **Version: 1** | **Page No** |

**Problem definition of Milk Tea Shop**

Milk Tea Shop is a milk tea shop specializing in selling milk tea and fast food. The main customers that it serves is students. When customers come to the shop, they order food and choose their own seat that they like then pay for bill at cashier.

But recently, the number of customers is increasing. The shop manager feels difficult to record and manage not only the invoice but also revenue of the shop. In addition, employees can not write lots of order bill every day. So, they want to have a system can help them to manage their shop that customers can order bill simpler and employees don’t work too much.

The system will provide functions:

* Customers can order food, choose their table that they like then pay at cashier
* Customers can register and log into system or enter discount code to receive the offer from shop
* Employees and manager must log in to use system.
* Employees can manage order bills in food tables, status of food tables, and food
* Manager can manage bills, employees, accounts and income of shop.
* Manager’s actions include view, insert, update, drop. But employee cannot drop or update anything. The order bill can be canceled when in the order process but not drop when customer has paid.

Beside that:

* The system must run smoothly, without errors from the system side.
* Fast execution speed, serving the crowded customer flow

|  |  |  |
| --- | --- | --- |
|  | **Prepare by (Student)** | **Approve by (Teacher)** |
| **Date** |  |  |

|  |  |  |
| --- | --- | --- |
| **Design Plan:** | **Document name: CRS** | **SWP/Form No.2A** |
| **Effective Date:** | **Version: 1** | **Page No** |

# **List of input to the system**

Account : User account in the system.

* Id: id account.
* Account: account name.
* Password: password (default “1” when create account).
* Roll: roll of account (1: customer, 2: employee, 3: manager).
* Name: name of account owner.
* Gender: gender of account owner.
* Birthday: the date of birth of account owner.
* Address: address of account owner.
* Phone: phone number of account owner.
* Email: email of account owner.

Table Food: Information of table.

* Id: id table.
* Status: status of table (Disable or Enable).
* X: coordinates X of table.
* Y: coordinates Y of table.

Category: Category of food.

* Id: id category.
* Name: type name.

Food: Information of food.

* Id: id food.
* Name: name of food.
* Id Category: classify on food (foreign key of Category).

Discount: Information of discount

* Id: id discount.
* Name: code value.
* Sale: discount values (example: 15%).

Bill: Information of bill.

* Id: id of bill.
* Id Table Food: id number of table (foreign key of Table Food).
* Id Customer: id number of Customer (foreign key of Account).
* Id Employee: id number of Employee (foreign key of Account).
* Id Discount: id number of Discount (foreign key of Discount).
* Check In: the time that customer paid bill.
* Check Out: the time that customer left the shop.

Bill Detail: Information of bill detail.

* Id: id bill detail.
* Id Bill: id number of table Bill (Foreign Key of bill).
* Id Food: id number of table Food (Foreign Key of Food).
* Quantity: quantity of food.

# **List of output expected from the system**

For Customers:

* Display food list.
* Display food category.
* Display bill details and total price when ordering.
* Display information of account when logged in.
* Display table list when paid bill.

For Employees:

* Display table list.
* Display bill list of each table in that day.
* Display bill details of each bill.
* Display bill details that customers are ordering.
* Display information of account when logged in.
* Display food list.
* Display food category.

For Managers:

* Display table list.
* Display employee list and account of each employee.
* Display customer list and account of each customer.
* Display discount list.
* Display bill list and bill details of each bill.
* Display food list.
* Display food category.
* Display income.

# **Overview of processes involved in the system**

For Customers

* Watch category food and food list, choose food
* Watch bill details that ordered
* Customers can cancel order if they have not paid
* Pay bill and watch table list, choose table
* Login account to discount bill
* Enter discount code
* Watch information of account
* Edit information of account if wrong
* Register account

For Employees

* Watch category food and food list
* Watch bill details that customer is ordering
* Watch table list
* Watch bill list and bill details of each table in that day
* Watch information of account
* Edit information of account if wrong
* Check out the bill of table

For Managers

* Manage category food: watch, insert, update, drop categories food
* Manage food: watch, insert, update, drop foods
* Manage bill: watch bills and bill details
* Manage employee: watch, insert, update, drop information and account of employees
* Manage customer: watch, insert, update, drop information and account of customers
* Manage discount code: watch, insert, update, drop discount code
* Manage income of shop, export report

# **Hardware and software required for implementing the project**

|  |  |
| --- | --- |
| **Hardware** | |
| CPU | 1.8 GHz or faster processor. Quad-core or better recommended. |
| Ram | 2 GB of RAM. 8 GB of RAM recommended. |
| Hard disk space | Minimum of 800MB up to 256 GB or more of available space depending on increasing database. |
| Hard disk speed | To improve performance, install Operating System and Project on a solid state drive (SSD). |
| Video card | Supports a minimum display resolution of 720p (1280 by 720). Project work best at a resolution of WXGA (1366 by 768) or higher. |
| **Languages** | |
| Supported Languages | Project is available in English. |
| **Operating System, Environment run time and Software** | |
| Operating System | Project will install and run on the following operating systems (Windows 10 64 bit recommended) |
| Environment run time | Java 11.0.4 |
| Database Server | Microsoft SQL Server (SQL Server 2016 recommended) |
| Database Server Management tool | Microsoft SQL Server Management Studio 18 |

|  |  |  |
| --- | --- | --- |
|  | **Prepare by (Student)** | **Approve by (Teacher)** |
| **Date** |  |  |

|  |  |  |
| --- | --- | --- |
| **Design Plan:** | **Document name: CRS/ Acceptance Criteria** | **SWP/Form No.2B** |
| **Effective Date:** | **Version: 1** | **Page No** |

# **Acceptance Criteria**

|  |  |
| --- | --- |
| S.No | Acceptance Criteria |
| For Customers: | |
| 1 | Customers can watch food and price in food list. |
| 2 | Customers can order food and pay bill. |
| 3 | Customers can cancel ordering when they had not pay. |
| 4 | Customers can choose table food after paying bill. |
| 5 | Customers can login or enter discount code in order to discount bill. |
| 6 | Customers can register account, account will create automatically with information that customer registered. |
| 7 | Customers can watch information of their account and edit it. |
| For Employees: | |
| 8 | Employees must login to use system. |
| 9 | Employees can watch bill and table that customer is ordering. |
| 10 | Employees can watch list of food. |
| 11 | Employees can watch status of tables and bills in that day of each table. |
| 12 | Employees can watch information of their account and edit it. |
| For Managers: | |
| 13 | Managers can watch, edit information of customer account. |
| 14 | Managers can watch, insert, update, drop employee and their account. |
| 15 | Managers can watch, insert, update, drop information of food. |
| 16 | Managers can watch, insert, update, drop food table. |
| 17 | Managers can watch, insert, update, drop discount code. |
| 18 | Managers can watch bill details. |
| 19 | Managers can watch income of shop and export file report. |

|  |  |  |
| --- | --- | --- |
|  | **Prepare by (Student)** | **Approve by (Teacher)** |
| **Date** |  |  |

|  |  |  |
| --- | --- | --- |
| **Design Plan:** | **Document name: Project Plan** | **SWP/Form No.3** |
| **Effective Date:** | **Version: 1** | **Page No** |

1. **Project Details:**
2. Name Project:

Milk Tea Shop

1. Date of Project Plan:

15-10-2019

1. Project Vision/Objectives:

Develop a management system application which stores information of bills, categories food, foods, tables food, employees, customers and manages them, also has enough necessary functions for the shop's owner to work effectively, such as statistic bill, calculate revenue, export report. Beside that, the application supports customer can order food, pay bill and choose table at cashier easily. It helps employees can manage tables and bills of each table in that day effectively.

1. Scope:

Milk tea shop, statistic bill, calculate revenue, export report, order food, pay bill.

1. Our understanding about the Client organization

Milk Tea Shop is a milk tea shop specializing in selling milk tea and fast food. The main customers that it serves is students. When customers come to the shop, they order food and pay bill. The shop's owner need application which supports managing bill, calculate revenue, export report.

1. Project with Organization with Responsibilities and Authorities:

* Dao Minh Trung Thuan – Leader: System, database analysis and design, user interface design, coding, testing.
* Ha Thi Hong Tham – Member: User interface design, coding.
* Tan Tan Lap – Member: User interface design, coding.
* Tran Quoc Cuong – Member: coding, testing.

1. **Project initiation/Requirement Documents**

* Java SE JDK 11: <https://docs.oracle.com/en/java/javase/11/>
* IntelliJ IDEA: <https://www.jetbrains.com/idea/documentation/>
* JavaFX 2: <https://docs.oracle.com/javafx/2/>
* SQL Server:

<https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation?view=sql-server-ver15>

* JDBC Driver for SQL Server:

<https://docs.microsoft.com/en-us/sql/connect/jdbc/microsoft-jdbc-driver-for-sql-server?view=sql-server-ver15>

1. **Deliverable:**
2. Application.
3. User guide.
4. Project document.
5. **Project dependencies:**
6. Lacking of time.
7. Illness and family busy.
8. Members slow process or out of the project.
9. **Major project prototype**
10. **Quality plan**
11. Review activities:

At least 3 times a week, each meeting there is a check for each member in group to review processing, help to solve hard trouble,….

1. Testing activities:

Ask for some help from a real shop and test the project with their customers, employees and shop’s owner.

1. Backup and recovery strategies

One copy store in Git-hub.

Each member store a copy of project in their own computer and one in their Google drive.

|  |  |  |
| --- | --- | --- |
|  | **Prepare by (Student)** | **Approve by (Teacher)** |
| **Date** |  |  |

|  |  |  |
| --- | --- | --- |
| **Design Plan:** | **Document name: GUI Standards Document** | **SWP/Form No.4** |
| **Effective Date:** | **Version: 1** | **Page No** |

|  |  |  |
| --- | --- | --- |
| **Name** | **Property** | **Value** |
| **Documentation** | | |
|  | Document theme | Default |
| Color scheme | Default |
| Form-background color | White |
| Title-Font size | 13 |
| Title-Font color | Black |
| Title-Font style | Time New Roman |
| Title-Alignment | Justify |
| **Form** | | |
|  | Theme | Modena (FX8) |
| Background color | Default background |
|  | Use AnchorPane for Root and set padding 0 | |
| **Containers** | | |
|  | Padding | 10 - 10 - 10 - 10 |
| Spacing | 10 |
| Min Width | USE\_COMPUTED\_SIZE |
| Min Height | USE\_COMPUTED\_SIZE |
| Max Width | USE\_COMPUTED\_SIZE |
| Max Height | USE\_COMPUTED\_SIZE |
| Alignment | Center |
| **Grid Pane** | Valignment | Center |
| Fill Height | TRUE |
| **Title** | | |
|  | Font | System 24px Bold |
| Text Fill | #2e6681 |
| Text Alignment | Left |
| Alignment | Center |
| **Controls** | | |
|  | Font | System 14px |
| Text Fill | Black |
| Text Alignment | Left |
| **Button** | Set default button if form has only one button | |
| Font | System 14px |
| Text Fill | Black |
| Text Alignment | Center |

|  |  |  |
| --- | --- | --- |
|  | **Prepare by (Student)** | **Approve by (Teacher)** |
| **Date** |  |  |

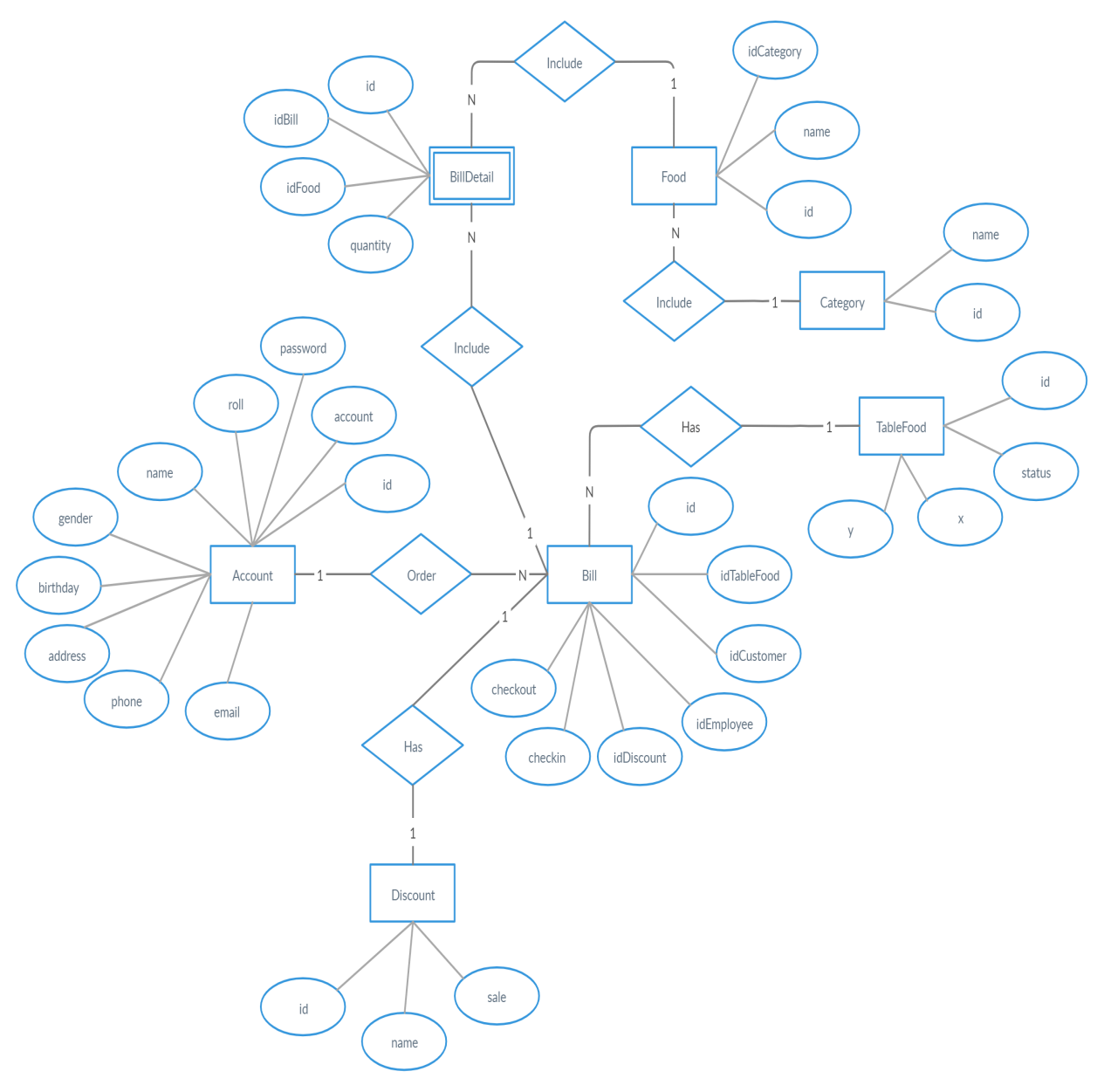
|  |  |  |
| --- | --- | --- |
| **Design Plan:** | **Document name: Document Design** | **SWP/Form No.6** |
| **Effective Date:** | **Version: 1** | **Page No 1/2** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name of the Table** | **Table Description** | **Number of Fields** | **Primary Key** | **Related Tables** | **Foreign Key** |
| Account | User account in the system. | 10 | id |  |  |
| TableFood | Information of table. | 4 | id |  |  |
| Category | Category of food. | 2 | id |  |  |
| Food | Information of food. | 3 | id | 1.Category | idCategory |
| Discount | Information of discount | 3 | id |  |  |
| Bill | Information of bill. | 7 | id | 1.TableFood  2.Account  3.Discount | idTableFood  idCustomer  idEmployee  idDiscount |
| BillDetail | Information of bill detail. | 4 | id | 1. Food 2. Bill | idBill  idFood |

|  |  |  |
| --- | --- | --- |
|  | **Prepare by (Student)** | **Approve by (Teacher)** |
| **Date** |  |  |

|  |  |  |
| --- | --- | --- |
| **Design Plan:** | **Document name: Document Design** | **SWP/Form No.6** |
| **Effective Date:** | **Version: 1** | **Page No 2/2** |

|  |  |  |
| --- | --- | --- |
|  | **Prepare by (Student)** | **Approve by (Teacher)** |
| **Date** |  |  |

****

|  |  |  |
| --- | --- | --- |
| **Design Plan:** | **Document name: Coding Standard Document** | **SWP/Form No.8** |
| **Effective Date:** | **Version: 1** | **Page No** |

1. **Programming Standards.**

* Every button should have an Icon for expressing, image icons have to store in the folder at source “<ProjectName>/src/asset” and have extension is “.png”.
* Must have declaration part for classes and interfaces using for the project.

1. **Standards for code writing style**

* The package name should be in all lowercase letters. Example:

controller, view, etc.

* Names for data types must be nouns, and be spelled out by words that begin with an uppercase character. Example:

ViewLogin, ViewChangePassword, etc.

* Variable names must start with a lowercase letter, the next words begin with a capital letter. Example:

name, controllerLogin, etc.

* The constant name must be all capital letters, words separated by underscore "\_". Example:

CONNECT\_STRING, etc.

* The method names must be verbs that start with a lowercase letter, and subsequent words are clearly spelled out by words starting with a capital letter. Example:

getName(), setName(), login(), update(), etc.

* Abbreviations should not include all capital letters (unless it is in a constant name).
* All names should be written in English.
* Wide range (global) variables should be long-named, narrow-range (local) variables should be short-named.
* The keyword "set / get" must be placed in methods that directly access the property. Example:

For local variables:

for (int i = 0; i < 10; i++) {

// Do somethings.

}

For global variables:

public class ControllerAdminLogin {   
 private TextField name;   
 private TextField password;  
}

* The keyword "find" can be used in the search methods.
* The name class will be placed after the package name. (unless it is class in package Model). Example:

Class in Controller package:

ControllerAdminLogin, ControllerChangePassword…

Class in Model package:

Account, Bill, Food, etc.

* The array, list (set of multiple objects) name should be named in the plural. Example:

foods, bills, accounts, etc.

* Additional parts should be used. Examble:

get/set, add/remove, create/destroy, start/stop, insert/delete, increment/decrement, old/new, begin/end, first/last, up/down, min/max, next/previous, old/new, open/close, show/hide, suspend/resume, etc.

|  |  |  |
| --- | --- | --- |
|  | **Prepare by (Student)** | **Approve by (Teacher)** |
| **Date** |  |  |

|  |  |  |
| --- | --- | --- |
| **Design Plan:** | **Document name:** | **SWP/Form No.9** |
| **Effective Date:** | **Version: 1** | **Page No** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Project**  **Ref.No:** | **Project**  **Type** | **Plan prepared By:** | | **Date of Preparation of Activity plan:** | | |
| **Task sub division** | **Description** | **Planned Start Date** | **Actual**  **Start Date** | **Actual Days** | **Team Member Names** | **Status** |
| Form No 1 | Problem Definition  Document | 9/10/2019 | 10/10/2019 | 2 | Ha Thi Hong Tham | 100% |
| Form No 2A | CRS | 9/10/2019 | 10/10/2019 | 2 | Tan Tan Lap | 100% |
| Form No 2B | CRS/Customer Acceptance Criteria | 9/10/2019 | 10/10/2019 | 2 | Dao Minh Trung Thuan | 100% |
| Form No 3 | Project Plan | 15/10/2019 | 16/10/2019 | 2 | Dao Minh Trung Thuan | 100% |
| Form No 4 | GUI Standards Document | 15/10/2019 | 16/10/2019 | 2 | Dao Minh Trung Thuan | 100% |
| Form No 5 | Interface Design Document | 15/10/2019 | 16/10/2019 | 2 | Dao Minh Trung Thuan | 100% |
| Form No 6 | Table Design  Document |  |  |  |  |  |
| Form No 7 | Process Design Document |  |  |  |  |  |
| Form No 8 | Coding Standards  Document |  |  |  |  |  |
| Form No 9 | Task Sheet | 9/10/2019 | 10/10/2019 |  |  |  |
| Form No 10 | Testing Document |  |  |  |  |  |
| Form No 11 | Project Review |  |  |  |  |  |
| Form No 12 | Final Check List |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **Prepare by (Student)** | **Approve by (Teacher)** |
| **Date** |  |  |