

**PROJECT REPORT**

# Milk Tea Mini Shop Management

|  |  |
| --- | --- |
| **Semester:** | Programming Fundamentals |
| **Class:** | PF08 |
| **Group:** | Group 2 |
| **Instructor** | Nguyễn Xuân Sinh |
| **Team members:** | NDE 19055 – Nông Hồng Cương  NDE 19024 – Nguyễn Long Nhật |

# Index

Milk Tea Mini Shop Management 1

Index 2

I. Project introduction 3

II. Analyze System Requirements 3

III. Design Details 17

IV. Test 24

V. Assign work to each team member 24

VI. Installation Instructions 25

Appendix 26

Document format 27

# Project introduction

User login to system and performs the functions such as: add, update drink and order. Finally, the user prints the invoice for the customer

1. Proposed System

Help user manage our milk tea shop more conveniently.

1. The scope of the project to be applied

Apply in many milk tea mini shops.

1. System Name

Milk Tea Mini Shop Management

1. Deployment Environmet

* Hardware: Computers: PCs, Laptops; Servers.
* Software:

+ Computers: Window, MacOS, Linux,...

+Servers: MySQL Server,...

1. Development Tools

- Visual Studio Code

- IntelliJ IDEA

- MySQL Workbench 8.0 CE

1. Customer Requirements

(System features)

-UC01 Login

-UC02 Create order

-UC03 Update order

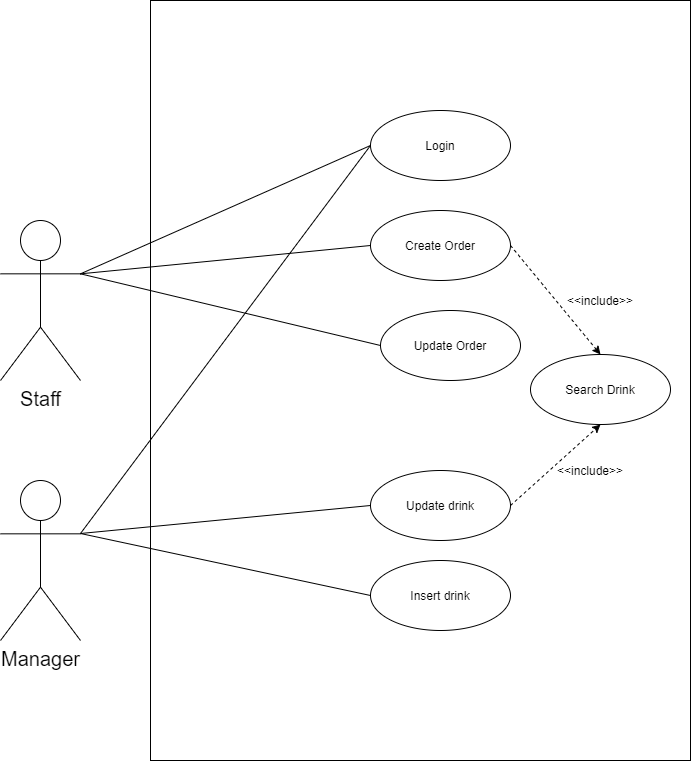
-UC04 Insert drink

-UC05 Update drink

# Analyze System Requirements

< Specify the requirements of the system required to meet customer requirements. This is the content of the discussion presented in more detail >

1. Use Case



* 1. Use case 1: Login

|  |  |
| --- | --- |
| Use Case Name | Login |
| Use Case ID | UC01\_Login |
| Description | Staff/Manager must log in to the system. |
| Actor | -Staff  -Manager |
| Organizational Benefits | Authenticate user |
| Frequency of Use | Often |
| Triggers | Staff/Manager want to use the system. |
| Preconditions | Staff/manager must have account and password to log in. |
| Postconditions | Show features of the staff/manager. |
| Main Course | 1. User input account and password. 2. Valid account/password.(see AC1) 3. Show features of user:    1. If user is Manager, display Manager’Menu .   3.2 If user in Staff, display Staff’s Menu. |
| Alternate Courses | AC3.1: Manager’Menu   * 1. Show all drink order by sold.   2. Insert drink   3. Update drink   AC3.2: Staff’s Menu   * 1. Create Order   2.2 Update Order |
| Exceptions | EX1. Valid Username:   * 1. Maximum 20 characters.   2. Minimun 8 characters.   3. Must contain letter, at least 1 number and 1 upper letter.   EX2. Valid Password  2.1. Minimum password length 8 characters.  2.2. Must contain letter, at least 1 number and 1 upper letter.  EX3. Incorrect password  3.1. Return user to Main Course Step 1  EX4. Account is not existed  4.1.Return user to Main Course Step1 |

1.2. Use case 2: Insert Drink

|  |  |
| --- | --- |
| Use Case Name | Insert Drink |
| Use Case ID | UC02\_Insert\_Drink |
| Description | Add new drink to database. |
| Actor | Manager |
| Organizational Benefits | Add new drink to database. |
| Frequency of Use | Often |
| Triggers | Select “Insert Drink” |
| Preconditions | User is the manager logged in to the system. |
| Postconditions | Return tre previous menu. |
| Main Course | 1. Input drink information. 2. Save drink to database. 3. System prompt that drink is save successfully. 4. Return the Manager’s Menu. |
| Alternate Courses | AC1.1: Input drink information   * 1. Code   2. Name   3. Unit price |
| Exceptions | EX1: Product existed(code existed)   * 1. Return user to Main courses step 1.1   EX2: Valid unit price  2.1.Unit price > 0  2.2. Return to Main courses step 1.4  EX3: Insert fails   * 1. System prompt insert fails   3.2 Return to Main Course step 1. |
|  |  |

1.3. Use case 3: Update drink

|  |  |
| --- | --- |
| Use Case Name | Update Drink |
| Use Case ID | UC03\_Update\_Drink |
| Description | Update drink’s information existed in database. |
| Actor | Manager |
| Organizational Benefits | Update change of drink information. |
| Frequency of Use | Sometimes |
| Triggers | Select “Update Drink” |
| Preconditions | User is the manager logged in to the system. |
| Postconditions | Return the previous menu. |
| Main Course | 1. Input drink Code 2. Show drink’s information 3. Show update menu. 4. Input new unit price 5. Update drink’s information. 6. System prompt that drink is update successfully. 7. Return Manager’s menu. |
| Alternate Courses |  |
| Exceptions | EX1.1: Product has Existed   1. Return user to Main courses step 1   EX4.1: Valid unit price   1. Unit price > 0. 2. If unit price <= 0, prompt and return Main courses step 4   EX3.1: Update fails   1. System prompt insert fails 2. Return to Main Course step 1. |

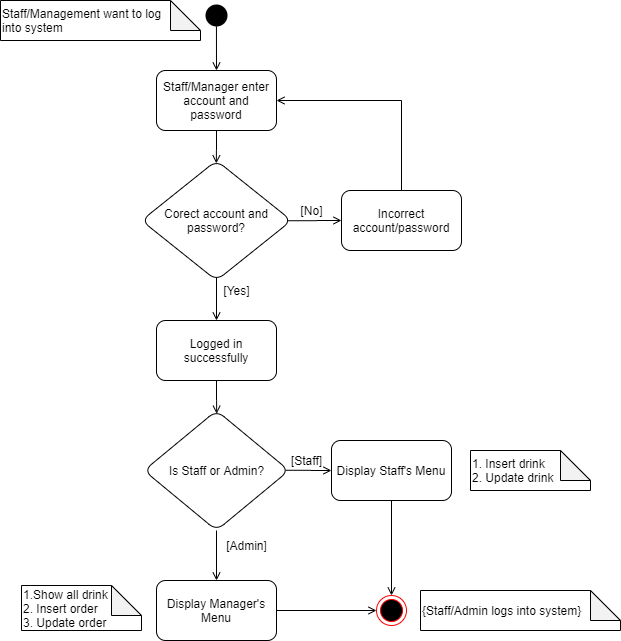
1.4 Use case 4: Create Order

|  |  |
| --- | --- |
| Use Case Name | Create Order |
| Use Case ID | UC04\_Create\_Order |
| Description | Create new order. |
| Actor | Staff |
| Organizational Benefits | Easy to manage sales and invoices. |
| Frequency of Use | Often |
| Triggers | Staff selects “Create order” in Staff’s menu. |
| Preconditions | Staff logged in to the system. |
| Postconditions | Print invoice & return the previous menu. |
| Main Course | * 1. Input drink code   2. Input quantity   3. Save order to database.   4. System prompt that order is saved successfully.   5. Show order information.   6. Print invoice and return the previous menu. |
| Alternate Courses | AC1.1. Check drink code   1. Code is existed in order.  * Quantity = quantity + new quantity.  1. Code is not existed in order.  * Add drink into order. |
| Exceptions | EX1.1. Drink code is not existed   1. System prompt that drink code is not existed. 2. Return Main Course Step 1.   EX2.1 Valid quantity   1. Quantity >0. 2. If quantity <= 0, prompt and return Main courses step 2.   EX3.1 Save order fails   1. System prompt that order is not saved successfully. 2. Return to Main Course Step 1. |

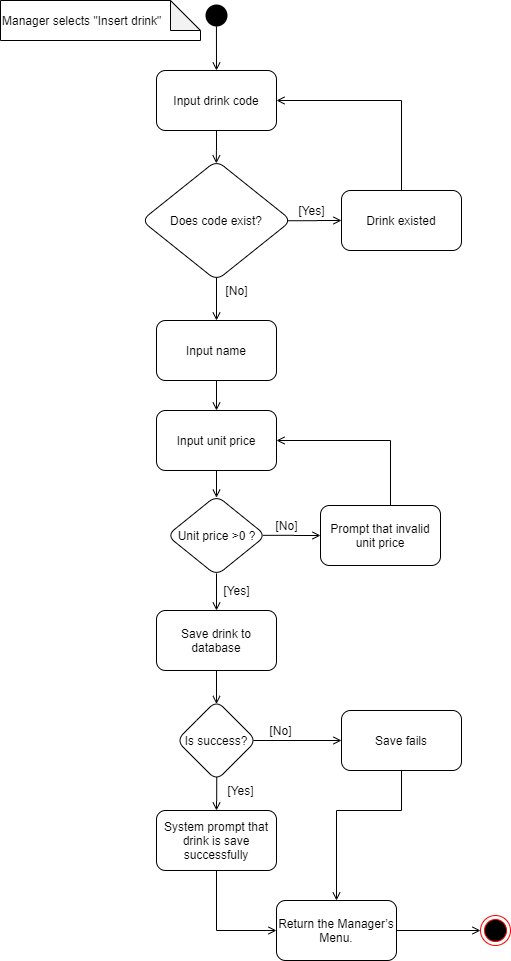
1.5 Use case 5: Update Order

|  |  |
| --- | --- |
| Use Case Name | Update Order |
| Use Case ID | UC05\_Update\_Order |
| Description | Update drink’s information in order. |
| Actor | Staff |
| Organizational Benefits | Update drink’s information in order. |
| Frequency of Use | Sometimes |
| Triggers | Staff selects “Update order” in Staff’s menu. |
| Preconditions | Staff logged in to the system. |
| Postconditions | Print updated invoice & return the previous menu. |
| Main Course | 1. Input order ID. 2. Show order details. 3. Input drink code to update. 4. Input new quantity. 5. Save change to database. 6. System prompt that order is saved successfully. 7. Show order information. 8. Print invoice and return the previous menu. |
| Alternate Courses |  |
| Exceptions | EX1.1 Order ID doesn’t exist.   1. Return Main Course Step 1.   EX3.1 Drink code doesn’t exist   1. System prompt that drink code is not existed. 2. Return to Main Course Step 3.   EX4.1 Valid quantity   1. Quantity >0. 2. If quantity <= 0, prompt and return Main courses step 2.   EX5.1 Save order fails   1. System prompt that order is not saved successfully. 2. Return to Main Course Step 1. |

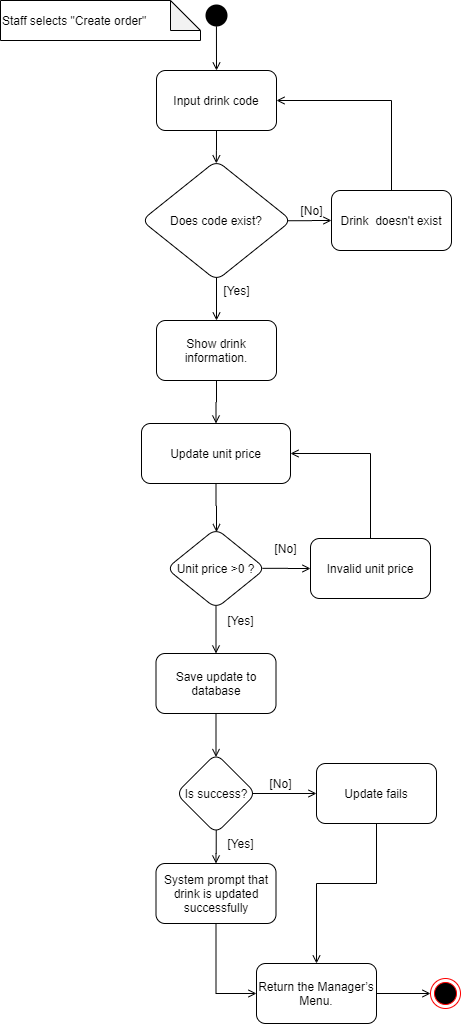
1. Activity Diagram:
   1. Login



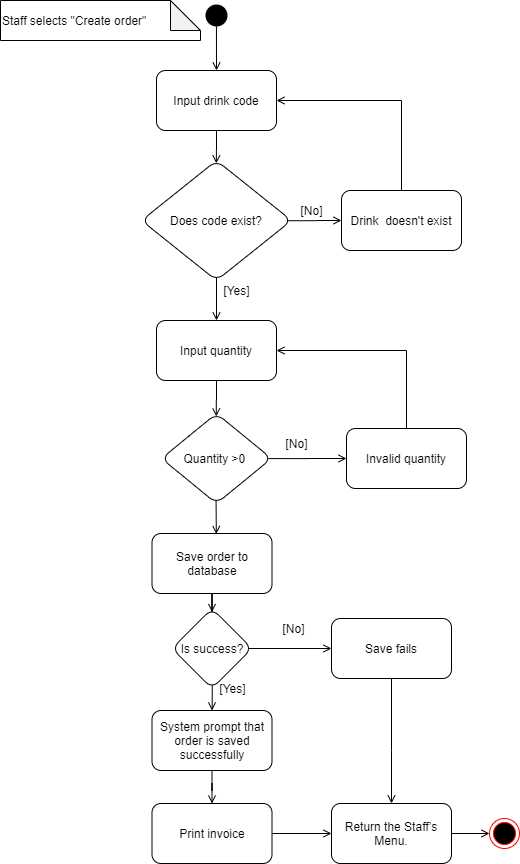
* 1. Insert drink



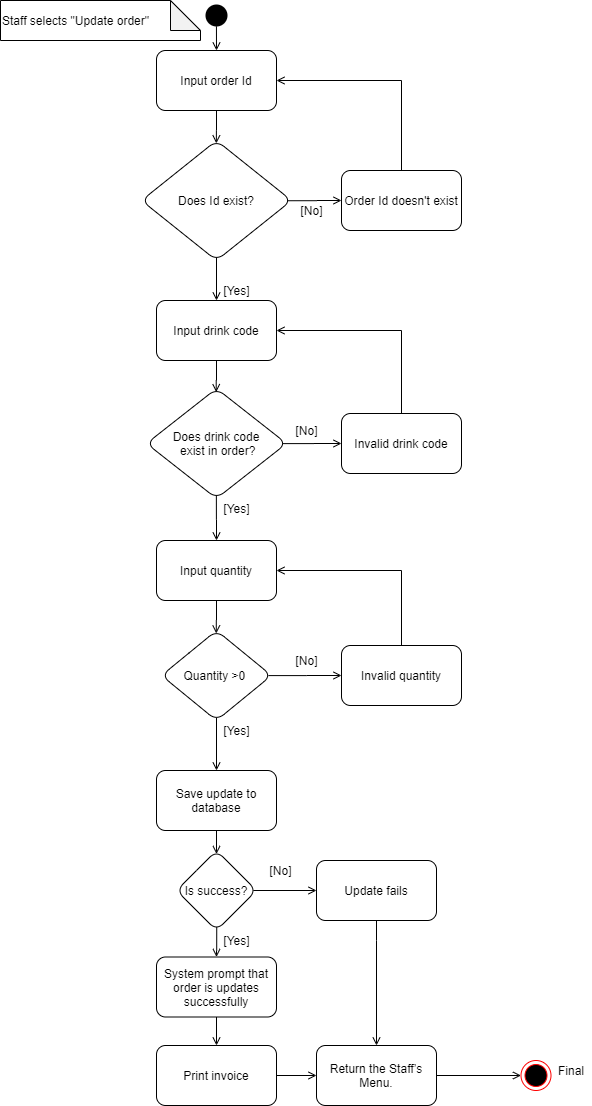
* 1. Update drink



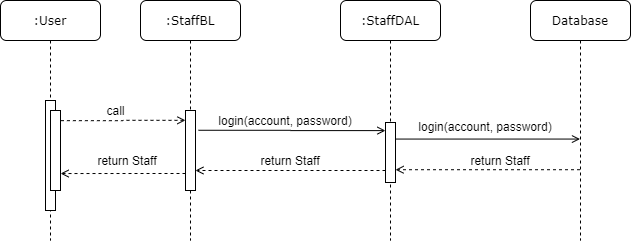
* 1. Creat Order



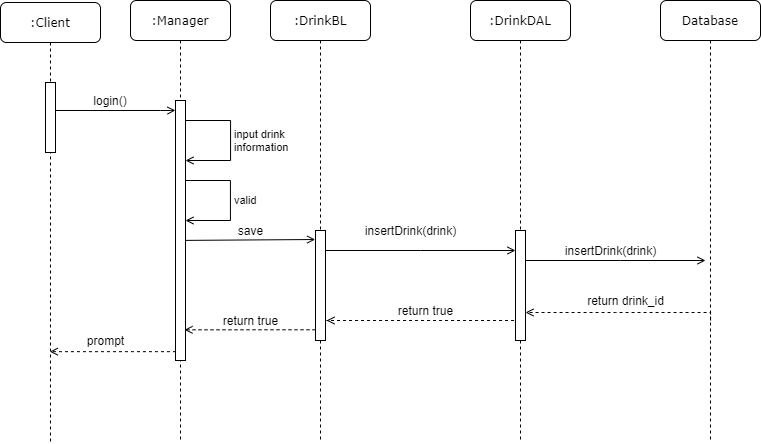
* 1. Update order



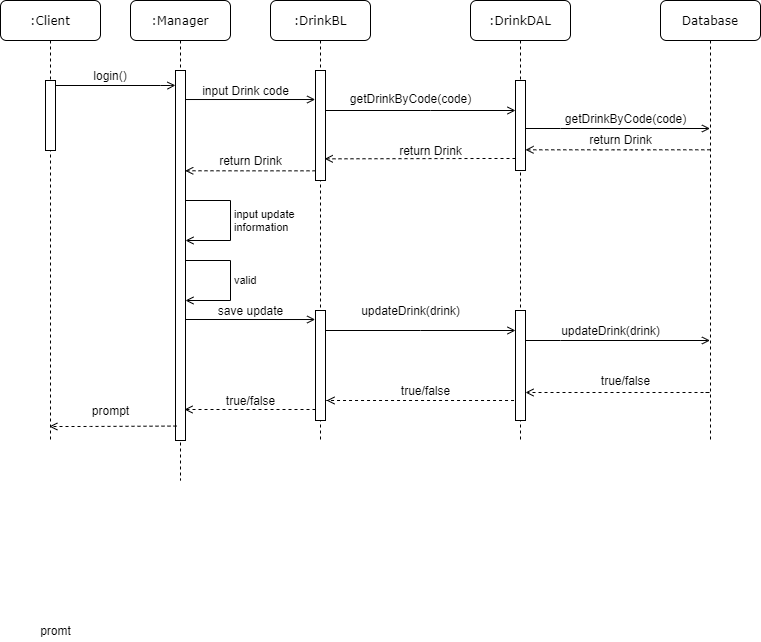
1. Sequence Diagram

3.1 Login

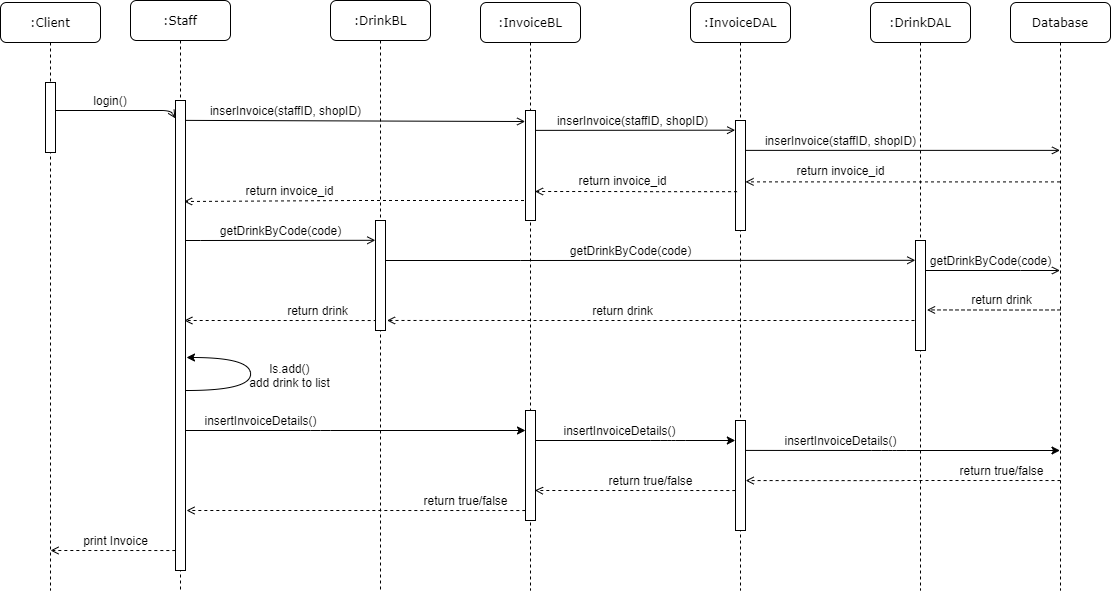
3.2 Insert drink



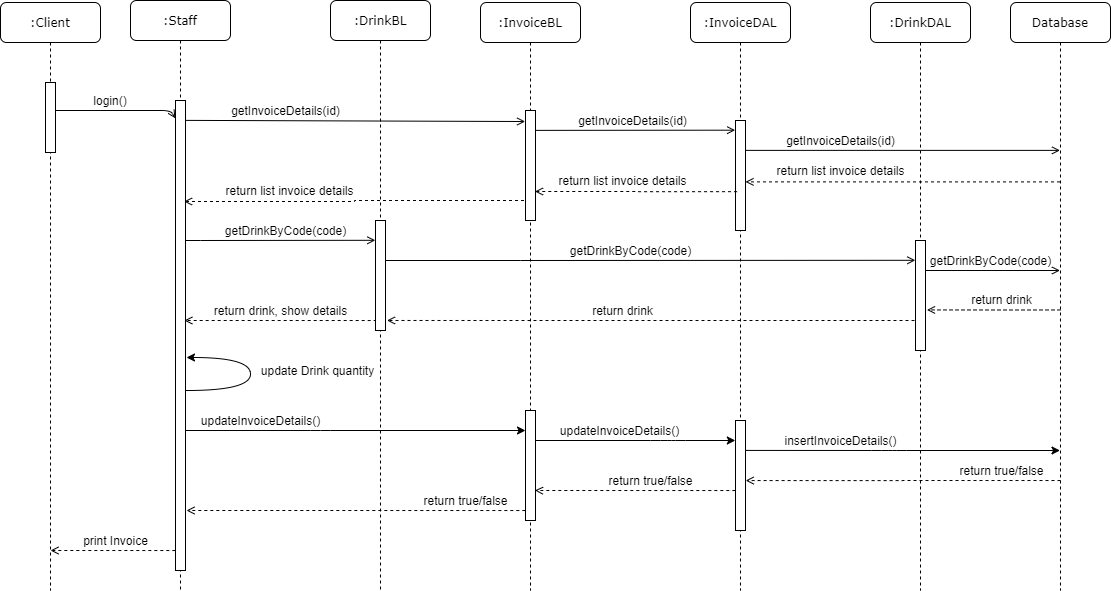
3.3 Update Drink



3.4 Insert Invoice

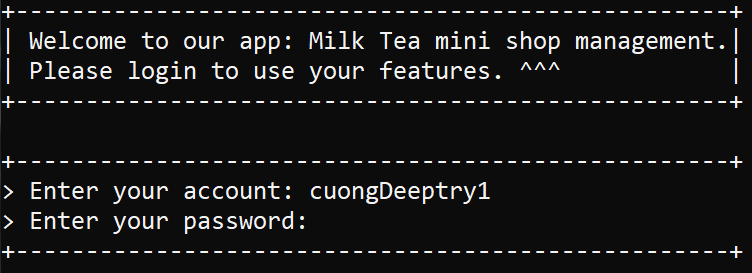


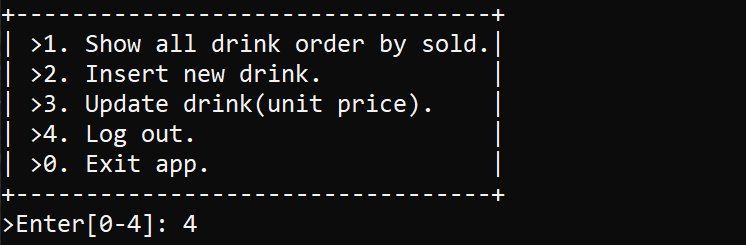
3.5 Update Invoice

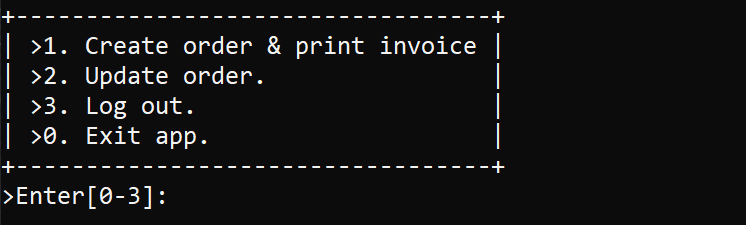


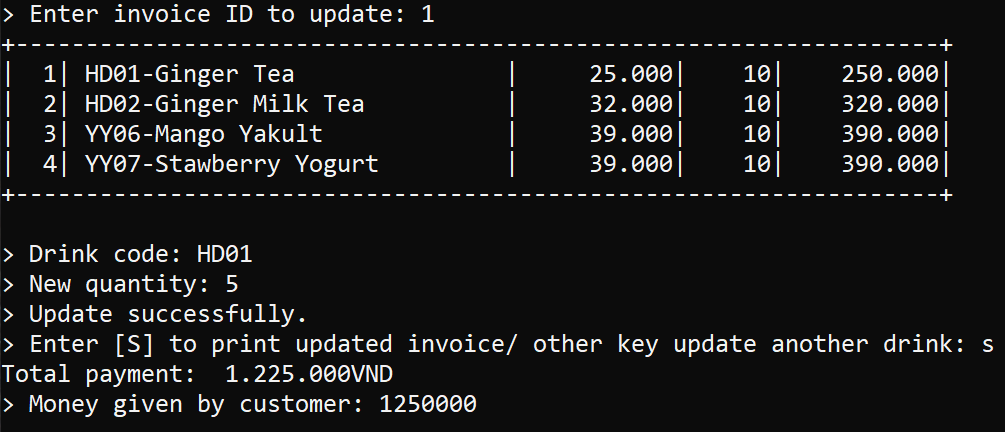
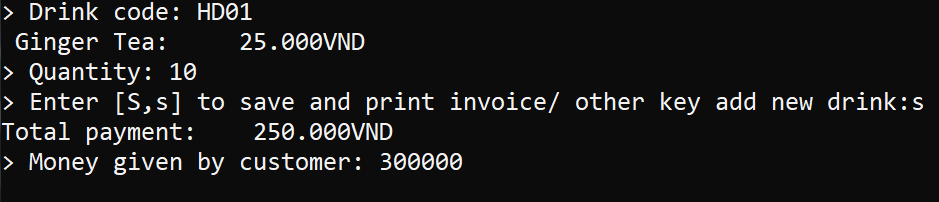
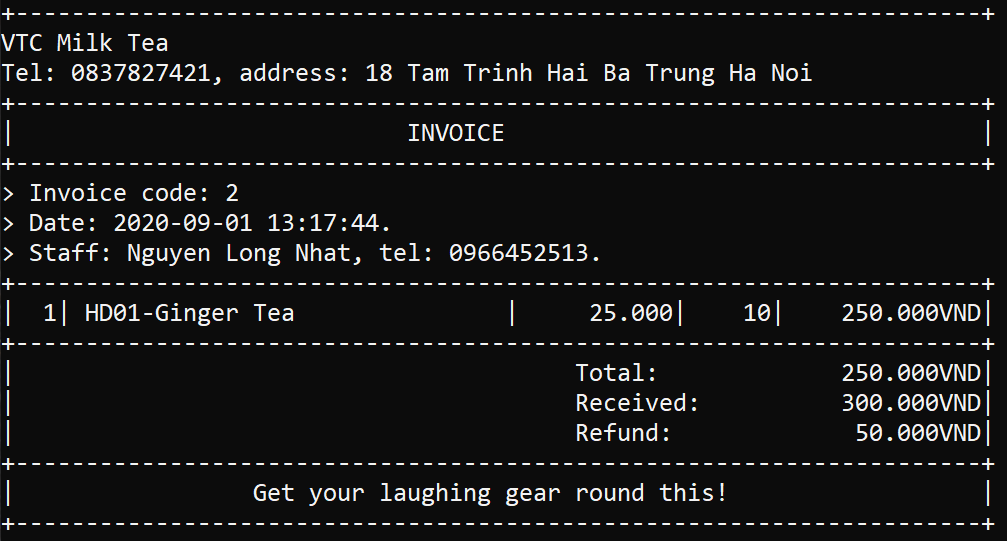
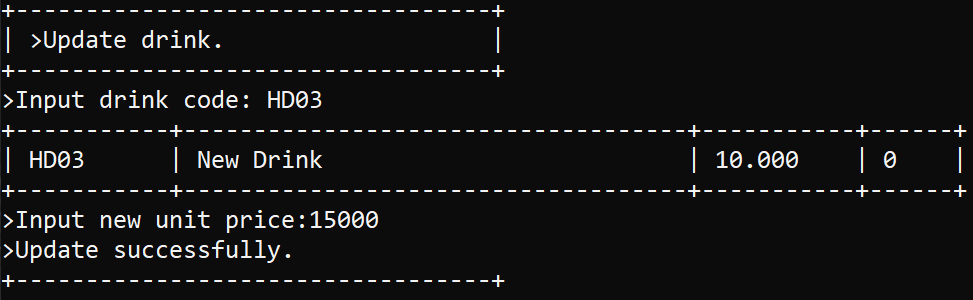
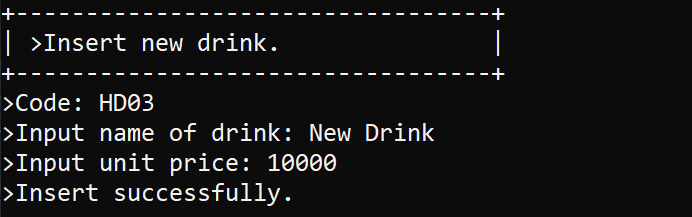
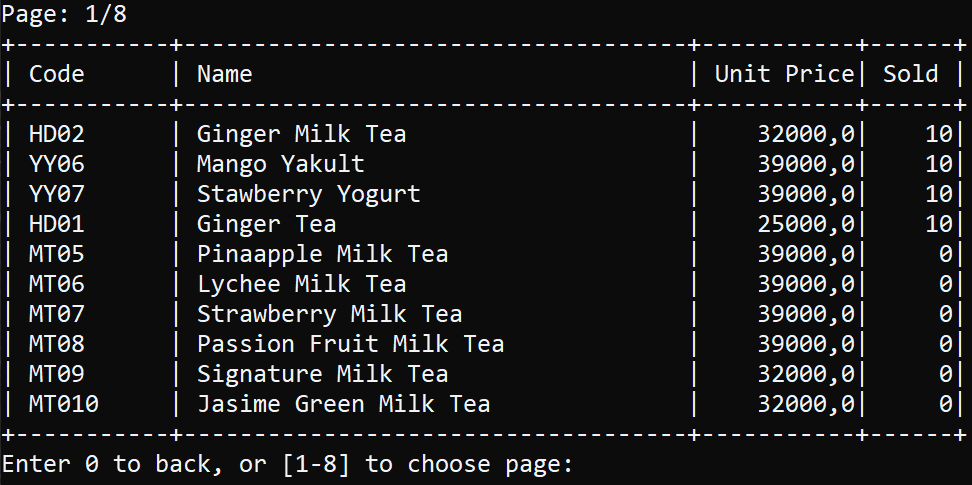
# Design Details

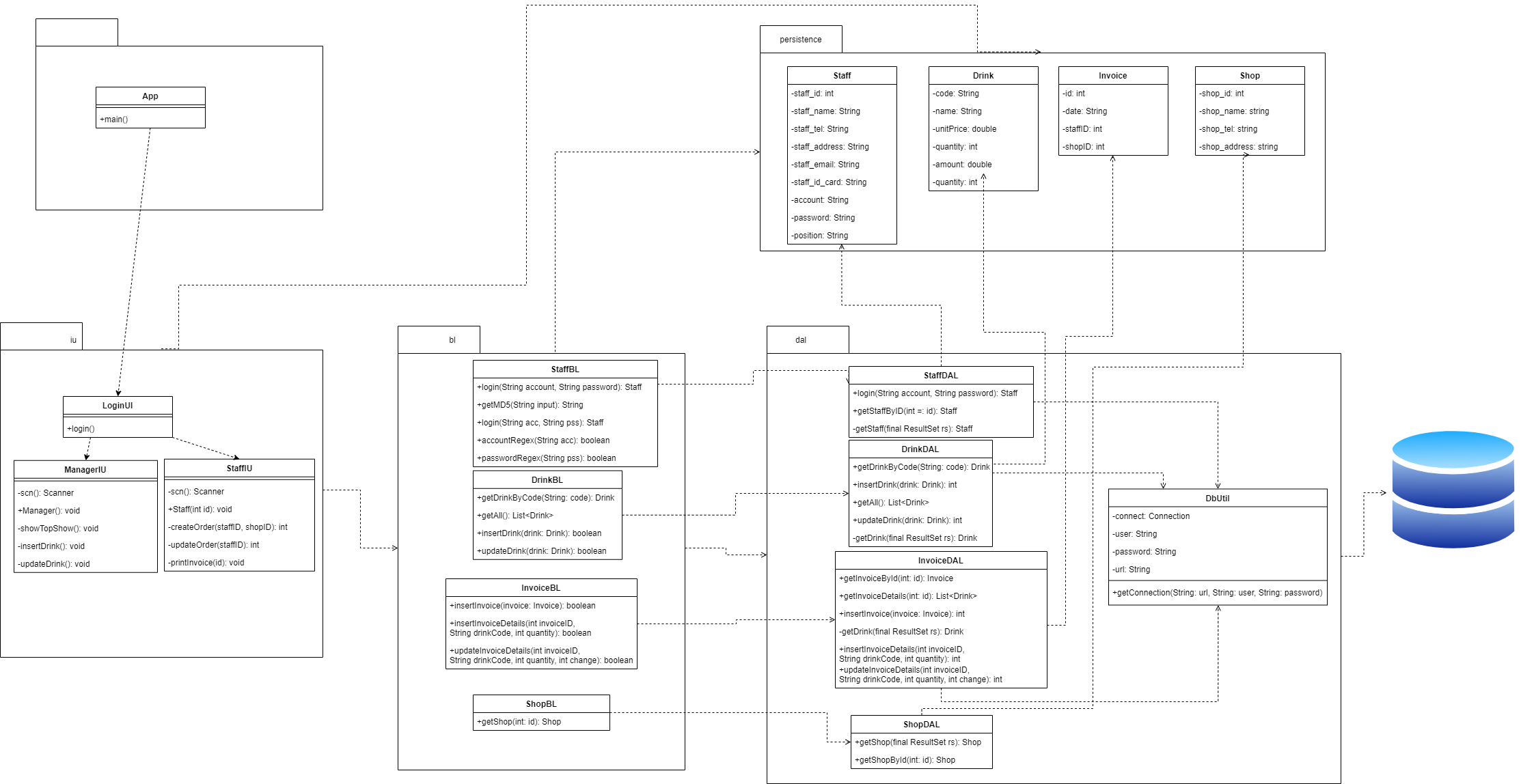
1. UI Design< Design the main user interface and for each features of the software >

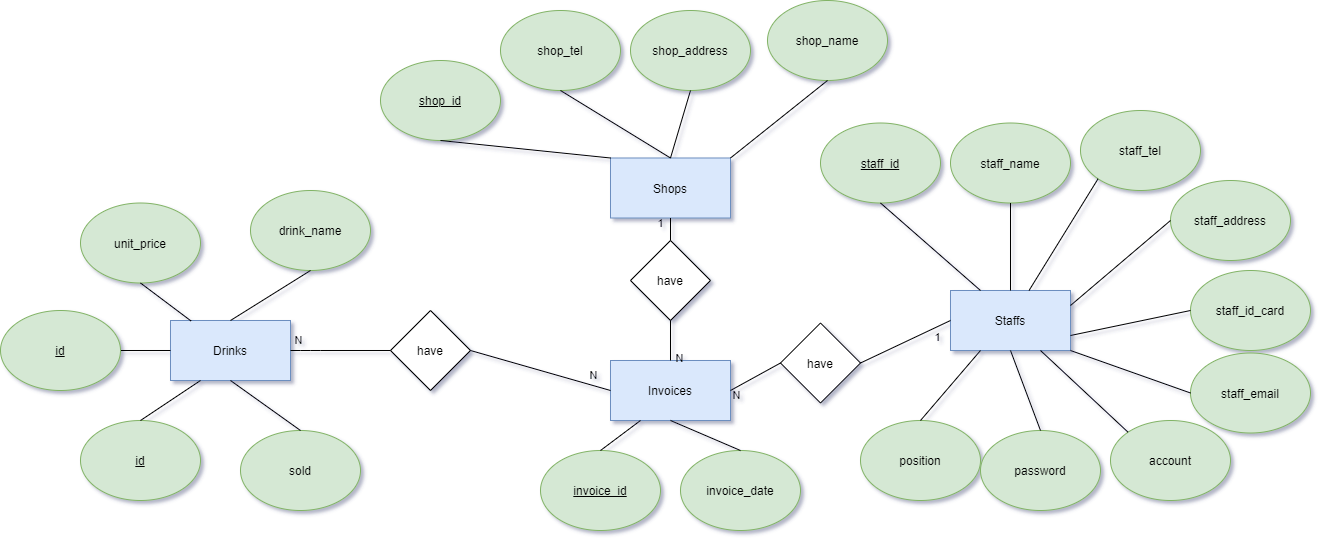




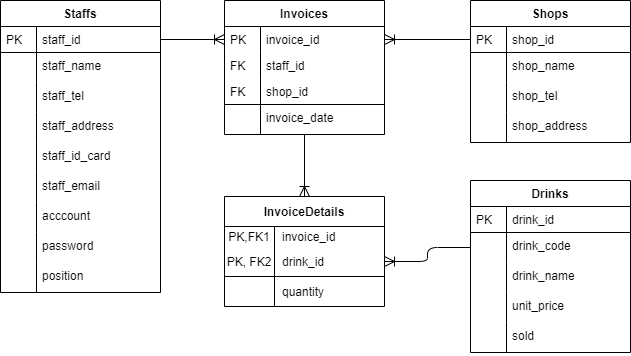




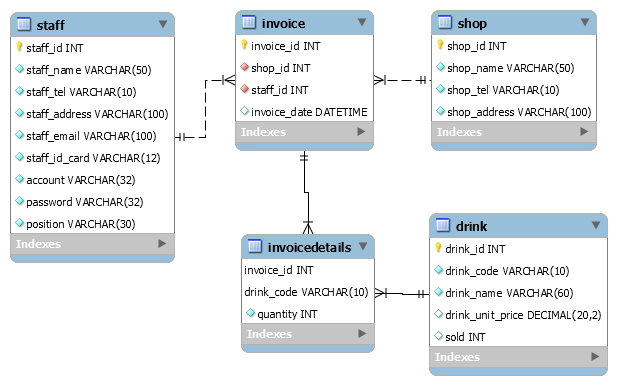
1. Code Design (Class Diagram)
2. Database Design
   1. Entity Relationship Diagram



Or



* 1. Database Design Details



|  |  |  |  |
| --- | --- | --- | --- |
| Staff | | | |
| Column Name | Data Type | Constraints | Description |
| staff\_id | int | Primary key, Auto increment, not null |  |
| staff\_name | varchar(50) | Not null | User’s full name |
| staff\_tel | varchar(10) | Not null | User’s phone number |
| staff\_address | varchar(100) | Not null | User’s address |
| staff\_email | varchar(100) | Not null | User’s email |
| staff\_id\_card | varchar(12) | Not null | User’s id card number |
| account | varchar(45) | Not null | User’s account |
| password | varchar(45) | Not null | User’s password |
| position | varchar(100) | Not null | User’s position |

|  |  |  |  |
| --- | --- | --- | --- |
| Drink | | | |
| Column Name | Data Type | Constraints | Description |
| drink\_id | int | Primary key, Auto increment,not null | Drink’s id |
| drink\_code | varchar(10) | Unique, not null | Drink’s code |
| drink\_category | varchar(45) | Not null | Drink’s category |
| drink\_name | varchar(100) | Not null | Drink’s name |
| unit\_price | decimal(20,2) | Not null, default 0 | Drink’s unit price |
| sold | int | Not null, default 0 | Drink’s sold |

|  |  |  |  |
| --- | --- | --- | --- |
| Shop | | | |
| Column Name | Data Type | Constraints | Description |
| shop\_id | int | Primary Key, Auto increment, Not null |  |
| shop\_name | varchar(45) | Not null | Shop’s name |
| shop\_tel | varchar(10) | Not null | Shop’s phone number |
| shop\_name | varchar(100) | Not null | Shop’s address |

|  |  |  |  |
| --- | --- | --- | --- |
| Invoice | | | |
| Column Name | Data Type | Constraints | Description |
| invoice\_id | int | Primary key, auto increment |  |
| staff\_id | int | Foreign Key, not null | Staff’s Id |
| shop\_id | int | Foreign Key, not null | Shop’s Id |
| invoice\_date | date | Not null | Invoice date |
| total\_due | decimal(20,2) | Not null, default 0 | Total due to pay |

|  |  |  |  |
| --- | --- | --- | --- |
| InvoiceDetails | | | |
| Column Name | Data Type | Constraints | Description |
| invoice\_id | int | Foreign key, not null | Invoice’s Id |
| drink\_id | int | Foreign key, not null | Drink’s Id |
| quantity | int | Not null, default 0 | Quantity of drink |

# Test

|  |  |
| --- | --- |
| Test Case Number |  |
| Test Case Name |  |
| Test Case Description |  |
| Preconditions |  |
| Test Case Input |  |
| Test Case Expected Output |  |
| Test Case Steps |  |
| Default Value Preverving |  |

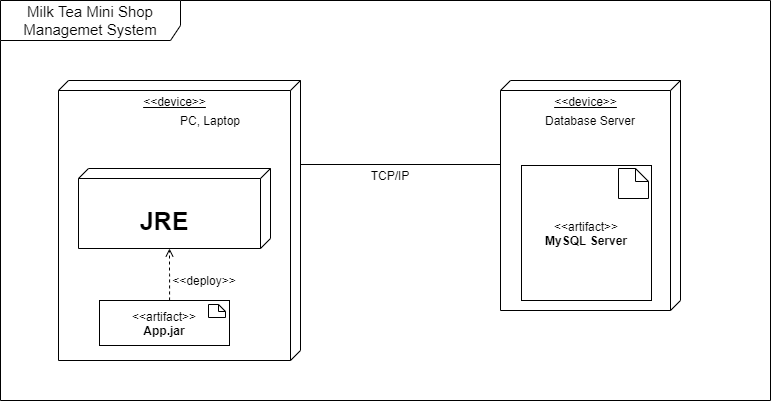
# Assign work to each team member

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Group Name | Project Name | | | | | |
| No | Task name | Description | Start Date | End Date | Member | Self assessment |
| 1 |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |

# Installation Instructions

< Consistently listing conventions when installing software (Detailed instructions with pictures and notes)>

1. Deployment Diagram



* 1. Install and set path JRE

[https://docs.oracle.com/goldengate/1212/gg-winux/GDRAD/java.htm#BGBFJHAB](https://docs.oracle.com/goldengate/1212/gg-winux/GDRAD/java.htm%23BGBFJHAB)

* 1. Install MySQL Server

<https://dev.mysql.com/doc/refman/8.0/en/installing.html>

1. Installation steps
   * Database Install
   * Server install
   * Application install

# Appendix

Terms and abbreviations <if available>

<listing terms and abbreviations here>

References <if available >

<List of references here>

Some other issues <if available >

<Results, limitations, experiences, techniques, and other considerations when implementing a project>

# Document format

**Report corver:**

*The cover is printed in blue.*

*With the format as the first page of this document.*

**Paper Size:**

*The report is presented on A4 size paper (210 mm x 297 mm)*

**Top header**

*Left: Logo của VTC Academy*

*Right: Project Name*

*Font: Helvetica Neue (Light)*

*Font size: 12pt*

**Bottom header**

*Left: Class\_Name –Project\_Name*

*Right: Page\_Number*

*Font: Helvetica Neue (Light)*

*Font size: 12pt*

**Report Content:**

*Font: Helvetica Neue (Light)*

*Font size: 12pt*

*Minimum 20 pages*

**Page margin (for A4 size paper)**

*Top: 20 - 25 mm;*

*Bottom: 20 - 25 mm;*

*Left: 30 - 35 mm;*

*Right: 15 - 20 mm;*