****

**Project**

**Agricultural  
Product Management**

**Subject: Java Desktop Application**

**The Drifter**

**Group 6:**

**Phan Tan Phat – CE130319 (\*)**

**Ha Van Ngoan – CE130192**

**Dang Buu Hoa – CE130427**

**Teacher: Vo Hong Khanh**

**TABLE OF CONTENTS**

Contents

**I. Problem definition1**

**II. User requirement2**

1. Function requirement2

a. List input to the application2

b. List output expected from the application2

c. Overview of processes involved in the application2

2. Hardware requirement2

a. Client Computer2

b. Server Computer2

3. Software requirement3

a. Client Computer3

b. Server Computer3

**III. Schedule and role 4**

1. Schedule details4

2. Role details4

**IV. Design pattern5**

1. Database diagram5

2. Database details5

a. Customers table6

b. Employees table6

c. Invoices table6

d. Orders table6

e. Order items table6

f. Price table7

g. Products table7

3. Class diagram8

4. Interface10

a. Interface displays information of customers10

b. Interface displays information of employees 11

c. Interface to add an order12

d. Interface to add the quantity of products13

**V. Maintenance14**

**VI. Checklist15**

# **PROBLEM DEFINITION**

There is a customer named 'A', he is very headache when facing with lots of bills and books every day. While other stores all use software to manage the store, Mr. A still manages the store in a traditional fashion every day. Mr. A wanted to build a store management software for his shop, so he went to meet the programmers to order. Programmers introduced Mr. A to the benefits of using management programs compared to traditional-style management.

* Advantages when using management program:

- Save time and cost

- Effectively manage resources.

- Suitable for all businesses.

- Coordinate different departments

- Improve management ability.

- Handling a large amount of data.

- Improve business efficiency.

# **USER REQUIREMENT**

1. **Functions requirement**
2. **List inputs to the application:**

* Information of products.
* Information of invoices.
* Information of customers.
* Information of employees.
* Information of orders, ordered items
* Information of prices.

1. **List outputs expected from the application**

* Information of sold products.
* Quantity of remaining products.
* The bills.

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

* Get prices, customers information, ordered items, employees’ information.
* Calculate the bills.
* Update available items in the store and warehouse.
* Export the bills.

1. **Hardware requirement:**
2. **Client Computer:**

|  |  |
| --- | --- |
| CPU | Intel Core i5 4th Gen or higher |
| Ram | 4GB |
| Storage | HDD: 10GB |

1. **Server Computer:**

|  |  |
| --- | --- |
| CPU | Intel Quad-core Q6600 or higher |
| Ram | 8GB |
| Storage | HDD: 10GB |

1. **Software requirement:**
2. **Client Computer:**

|  |  |
| --- | --- |
| Operating system | Window 7 or higher |
| JRE | 1.8 or higher |

1. **Server Computer:**

|  |  |
| --- | --- |
| Operating system | Window Server 2008 or higher |
| Database system | Microsoft SQL Server 2017 |

# **SCHEDULE AND ROLE**

1. **Role details:**

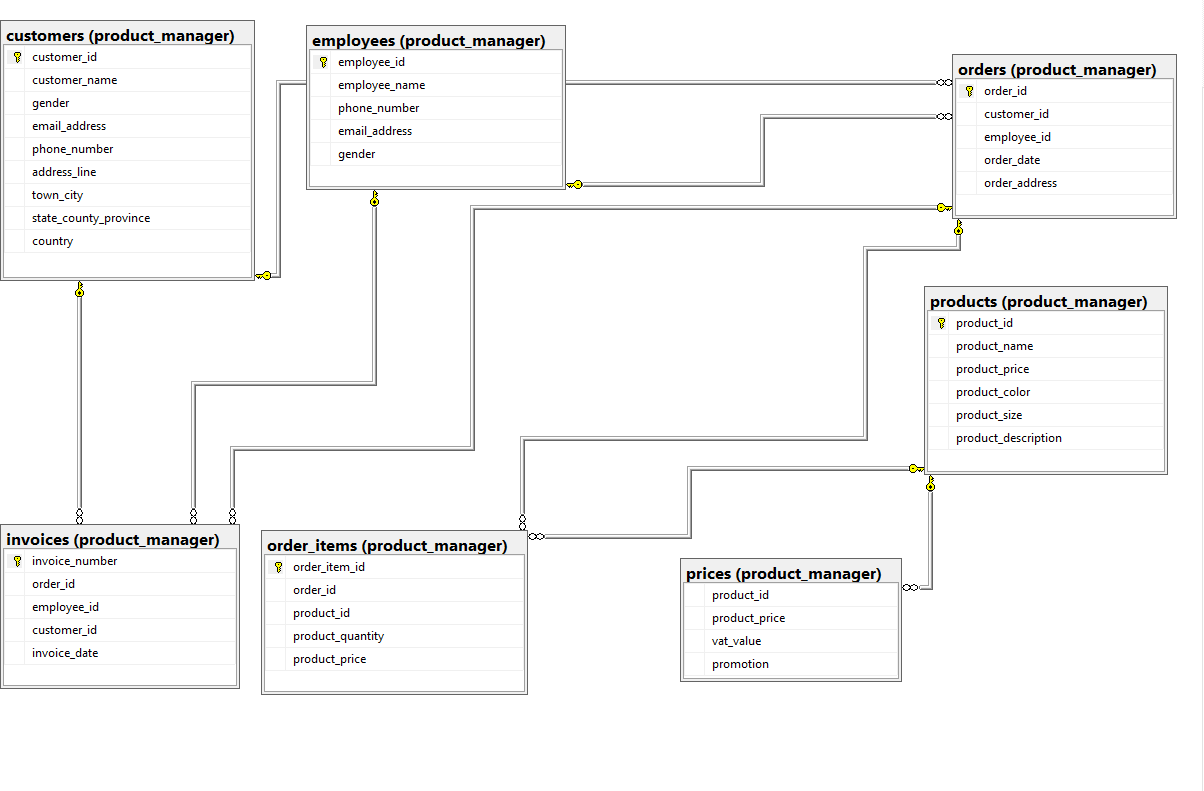
|  |  |  |  |
| --- | --- | --- | --- |
| Order | Member | Role | Responsibilities |
| 1 | Phat | Group leader | Manage the project |
| 2 | Hoa, Phat, Ngoan | Analyst | Define problem, input, output, process.  Determinate how to see problem together. |
| 3 | Ngoan, Hoa | Designer | Design interface |
| 4 | Phat, Ngoan, Hoa | Programmer | Program all functions for applications. |
| 5 | Hoa, Phat, Ngoan | Tester | Testing all functions which have been programming |

1. **Schedule details:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Order | Project title | Planned Start Date | Hour | Members’ names |
| 1 | Cover | 05/03/2019 | 0.5 | Hoa |
| 2 | Problem definition | 20/01/2019 | 0.5 | Ngoan |
| 4 | User requirement | 05/03/2019 | 0.5 | Hoa, Ngoan |
| 5 | Code – interface | 21/03/2019 | 2 | Ngoan, Hoa |
| 6 | Code – backend | 21/03/2019 | 5 | Phat, Ngoan, Hoa |
| 7 | Schedule and role | 05/03/2019 | 0.5 | Hoa |
| 8 | Database diagram | 20/01/2019 | 1 | Hoa |
| 9 | Class diagram | 05/03/2019 | 1 | Ngoan, Phat |
| 10 | Test | 23/03/2019 | 1 | Ngoan, Hoa |
| 11 | Debug | 23/03/2019 | 2 | Ngoan, Phat, Hoa |
| 12 | Maintenance | 23/03/2019 | 1.5 | Ngoan, Hoa |
| 13 | Checklist | 23/03/2019 | 0.5 | Ngoan |

# **DESIGN PATTERN**

1. **Database diagram**



1. **Database detail:**
2. **Customers table**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraints | Description |
| customer\_id | Int | primary key | Customer’s ID |
| customer\_name | nvarchar(50) | not null | Customer’s name |
| gender | nvarchar(50) | not null | Customer’s gender |
| email\_address | nvarchar(50) | not null | Customer’s email address |
| phone\_number | nvarchar(50) | not null | Customer’s phone number |
| address\_line | nvarchar(50) | not null | Customer’s address line |
| town\_city | nvarchar(50) | not null | Customer’s town or city |
| state\_county\_province | nvarchar(50) | not null | Customer’s state or province |
| country | nvarchar(50) | not null | Customer’s country |

1. **Employees table**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraints | Description |
| employee\_id | int | primary key | Employee’s ID |
| employee\_name | nvarchar(50) | not null | Employee’s name |
| gender | nvarchar(50) | not null | Employee’s gender |
| email\_address | nvarchar(50) | not null | Employee’s email address |
| phone\_number | nvarchar(50) | not null | Employee’s phone number |

1. **Invoices table**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraints | Description |
| invoice\_number | int | primary key | Invoice’s number |
| order\_id | int | foreign key | Order’s ID |
| employee\_id | int | foreign key | Employee’s ID |
| customer\_id | int | foreign key | Customer’s ID |
| invoice\_date | date | not null | Invoice’s date |

1. **Orders table**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraints | Description |
| order\_id | int | primary key | Order’s ID |
| employee\_id | int | foreign key | Employee’s ID |
| customer\_id | int | foreign key | Customer’s ID |
| date\_order | date | not null | Order’s date |
| address\_order | nvarchar(50) | not null | Order’s address |

1. **Order items table**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraints | Description |
| order\_item\_id | int | primary key | Order item’s ID |
| order\_id | int | foreign key | Order’s ID |
| product\_id | int | foreign key | Product’s ID |
| product\_price | float | not null | Product’s price |
| product\_quantity | nvarchar(50) | not null | Product’s quantity |

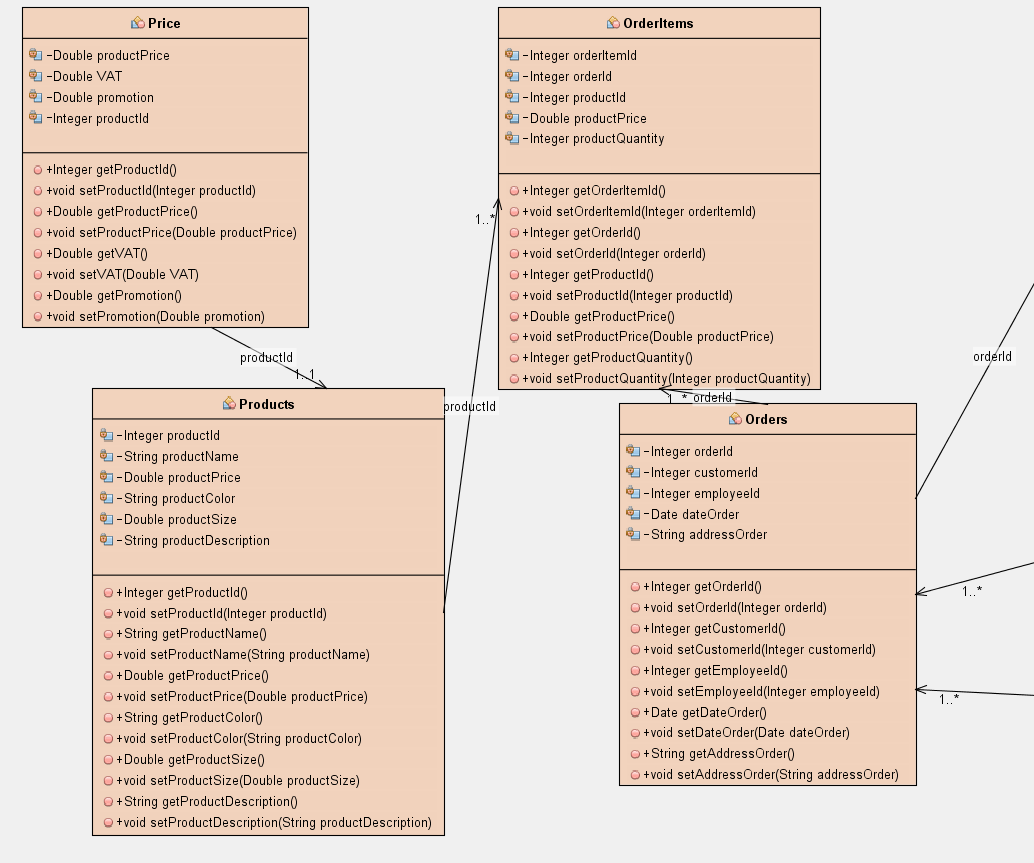
1. **Price table**

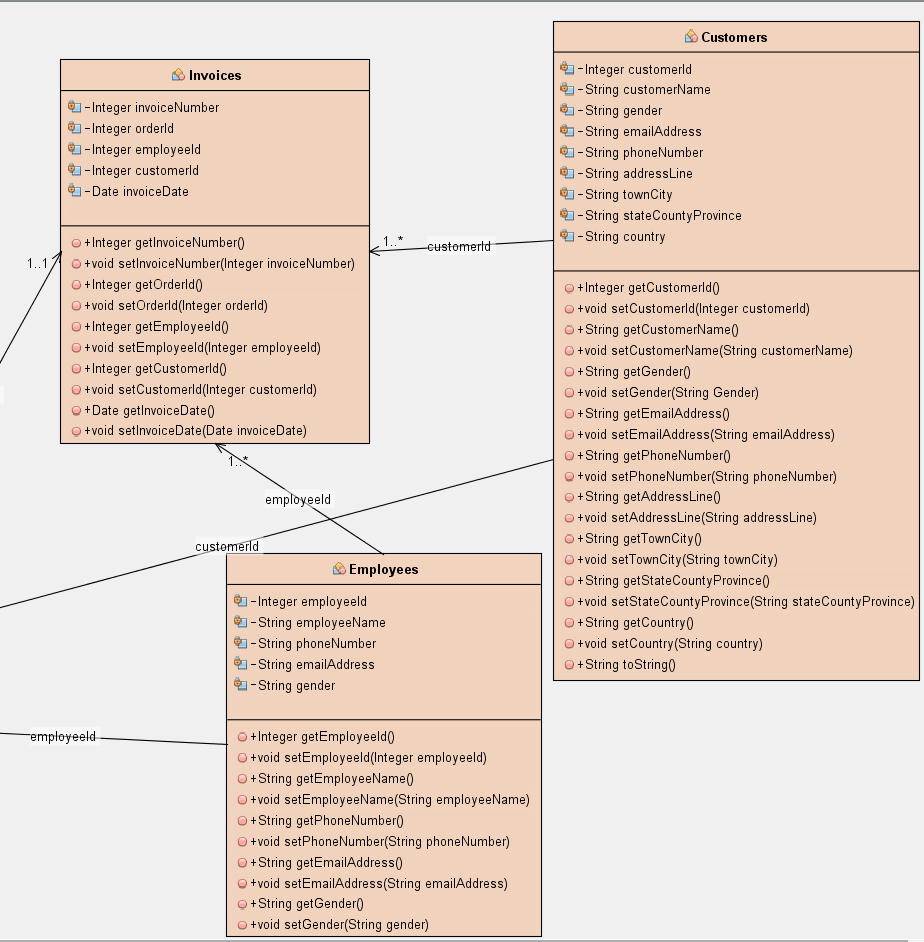
|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraints | Description |
| order\_item\_id | int | primary key | Order item’s ID |
| order\_id | int | foreign key | Order’s ID |
| product\_id | int | foreign key | Product’s ID |
| product\_price | float | not null | Product’s price |
| product\_quantity | int | not null | Product’s quantity |

1. **Products table**

|  |  |  |  |
| --- | --- | --- | --- |
| Field name | Data type | Constraints | Description |
| product\_id | int | foreign key | Product’s ID |
| product\_name | nvarchar(50) | not null | Product’s name |
| product\_price | float | not null | Product’s price |
| product\_color | nvarchar(50) | not null | Product’s color |
| product\_size | float | not null | Product’s size |
| product\_description | nvarchar(500) | not null | Product’s description |

1. **Class diagram**

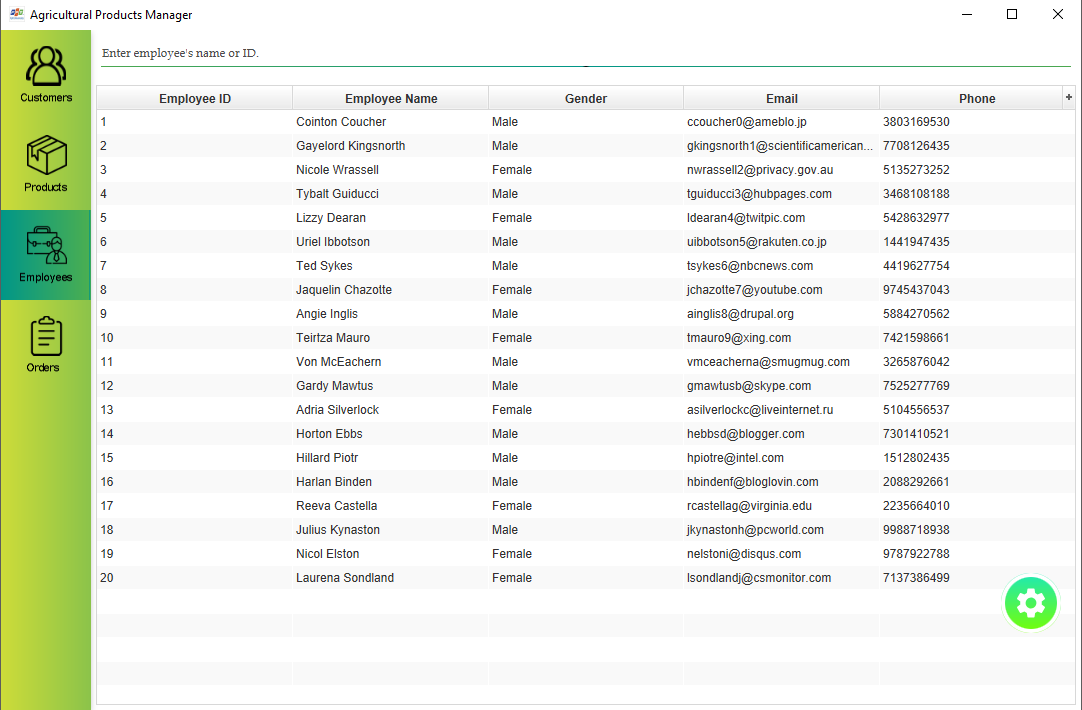




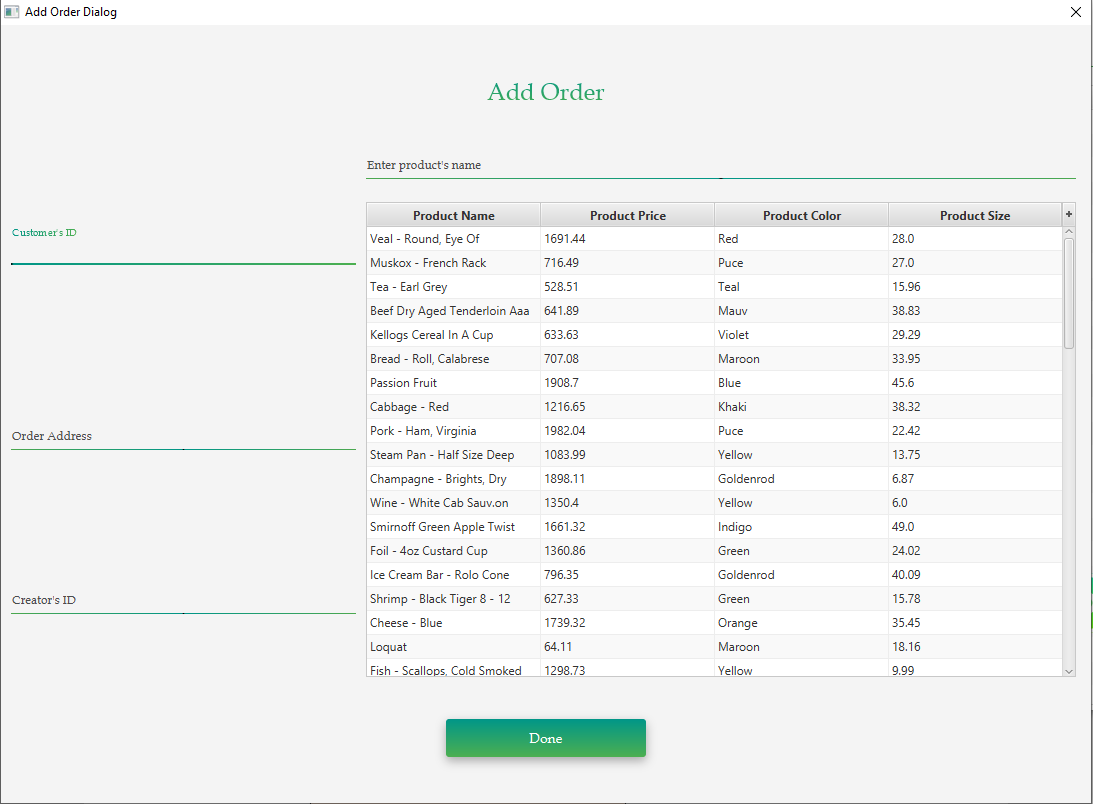
1. **Interface**
2. **Interface displays information of customers**



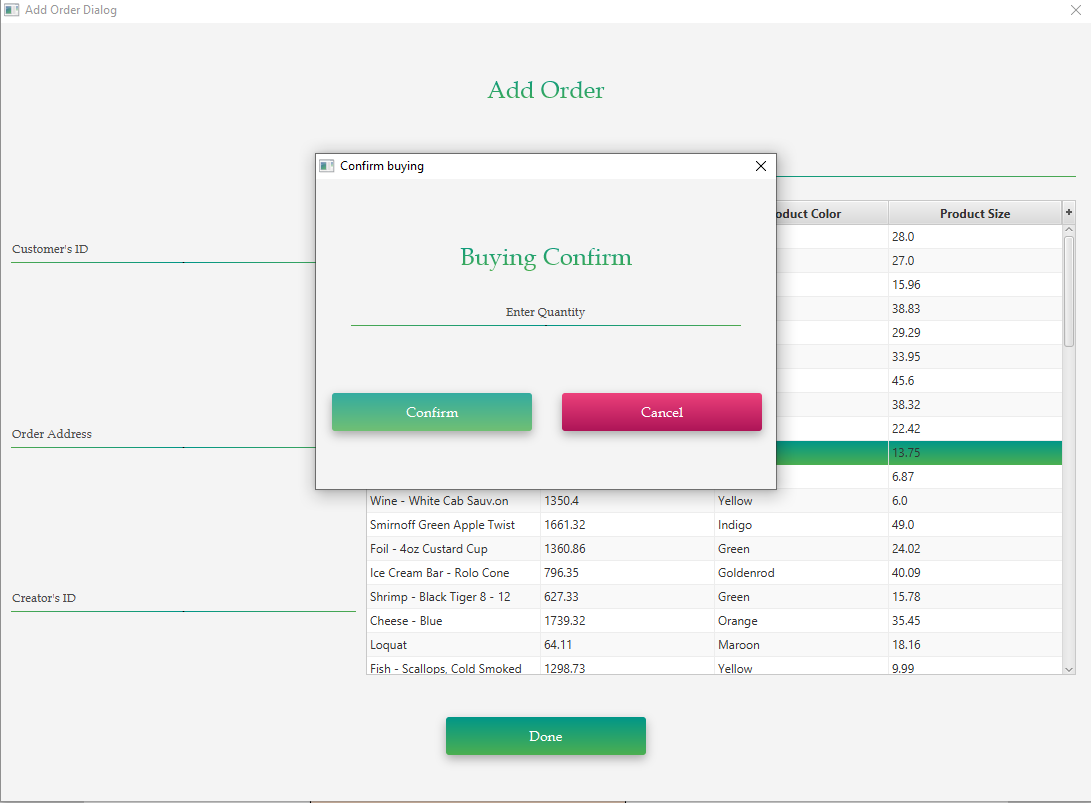
1. **Interface displays information of employees**

****

1. **Interface to add an order**

****

**d. Interface to add the quantity of products**

****

# **MAINTENANCE**

- Backup document on Google drive and three laptops of Ngoan, Phat and Hoa.

- Backup code on Github and Google drive.

- Backup daily.

- Backup the database weekly.

# **CHECKLIST**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Order | Project title | Planned Start Date | Expected  Hour | Actual Date | Actual  Hour | Team mate names | Status |
| 1 | Cover | 05/03/2019 | 0.5 | 05/03/2019 | 0.5 | Hoa | Complete |
| 2 | Problem definition | 20/01/2019 | 0.5 | 20/01/2019 | 0.5 | Ngoan | Complete |
| 3 | User requirement | 05/03/2019 | 0.5 | 05/03/2019 | 0.5 | Hoa, Ngoan | Complete |
| 4 | Code – interface | 21/03/2019 | 2 | 21/03/2019 | 2 | Ngoan, Hoa | Complete |
| 5 | Code – backend | 21/03/2019 | 5 | 21/03/2019 | 6 | Ngoan, Phat, Hoa | Complete |
| 6 | Schedule and role | 05/03/2019 | 0.5 | 05/03/2019 | 0.5 | Hoa | Complete |
| 7 | Database diagram | 20/01/2019 | 1 | 20/01/2019 | 2 | Hoa | Complete |
| 8 | Class diagram | 05/03/2019 | 1 | 05/03/2019 | 2 | Ngoan, Phat | Complete |
| 9 | Test | 23/03/2019 | 1 | 23/03/2019 | 1 | Ngoan, Hoa | Complete |
| 10 | Debug | 23/03/2019 | 2 | 23/03/2019 | 2 | Ngoan, Phat, Hoa | Complete |
| 11 | Maintenance | 23/03/2019 | 1.5 | 23/03/2019 | 1 | Ngoan, Hoa | Complete |
| 12 | Checklist | 23/03/2019 | 0.5 | 23/03/2019 | 0.5 | Ngoan | Complete |