

PROJECT REPORT

# **Trip Planner**

Semester: Programming Fundamentals

Class: K22P.

Group: BCS - Ba Con Sâu

Instructor : Võ Công Đình.

Team members: TRƯƠNG CHIÊU TÀI.

ĐỖ VĂN HÒA

NGUYỄN THUẬN

1. **Project introduction**

The app provides two main features: following the available tours and users arranging their own itinerary. The app provides functions such as transportation, discount codes, amusement parks, renting cars, tour guides, food and beverage options, check-in locations, combo deals, tour details, and order details.

1. **The suggested system includes:**

- Hot travel tours.

- Destinations.

- Vouchers.

1. **The target users are:**

- Food enthusiasts.

- Locals.

- Business travelers.

- People planning their trips.

- Travel enthusiasts.

- The system name is Trip Planner.

1. **Deployment environment:**

Windows, Linux, Mac.

1. **Development tools:**

Visual Studio Code

Draw.io

MySQL Workbench 8.0 CE

1. **Customer requirements:**

- Login/Register.

- Display service list.

- Create itinerary, register for tours.

- Display order list.

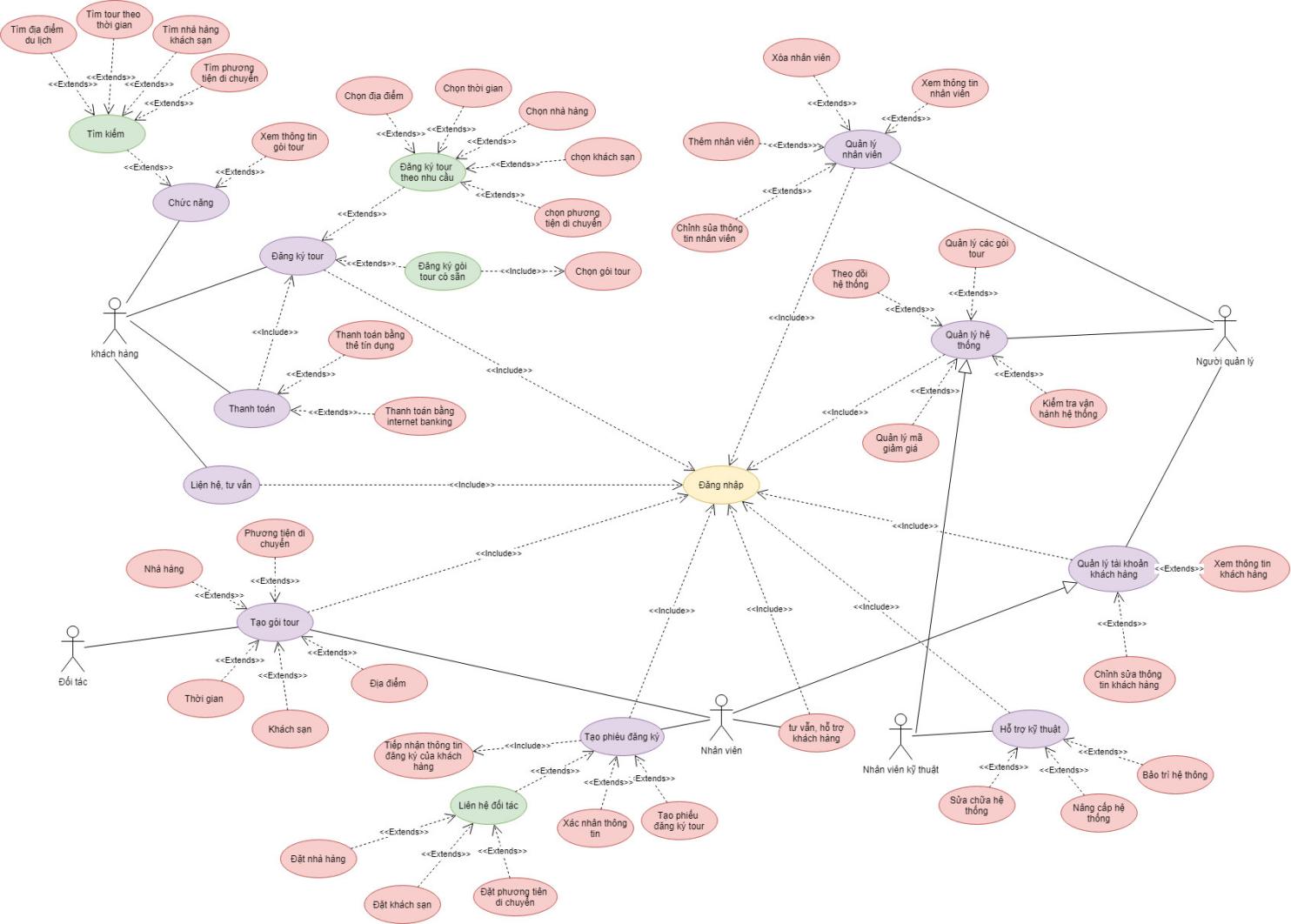
- View order details.

- Confirm payment or cancel orders.

1. Analyze System Requirements

This app was created to help travelers book and manage their trips more easily. It can help users find more information about the destinations they want to visit, such as prices, attractions, activities, culture, etc. so that they can plan their trips according to their preferences and needs. The main function of this app for users is that it allows them to create their own itinerary by choosing from a variety of options and packages. For travel agents, this app allows them to view the information of the bookings and confirm the payment if they have received enough money from the users, cancel the booking if the users change their mind or pause the payment in case of necessity.

* 1. Use Case
  2. Use Case



* 1. Xác định các tác nhân và ca sử dụng .

1.1.1 .Xác định các Tác nhân

- Tác nhân : Người quản lý .

- Tác nhân : Đối tác.

- Tác nhân : Khách hàng.

- Tác nhân : Nhân viên .

- Tác nhân : Nhân viên kỹ thuật .

1.1.2 .Xác định các ca sử dụng.

- Dựa trên việc mô tả bài toán và việc phân tích để tìm ra tác nhân, ta xác định

Được các ca sử dụng sau .

+ Chức năng :.

+ Đăng ký tour : .

+ Thanh toán .

+ Liên hệ tư vấn

+ Đăng nhập

+ Tạo gói tour

+ Tạo phiếu đăng ký .

+ Tư vấn hỗ trợ khách hàng .

+ Quản lý tài khoản khách hàng .

+ Hỗ trợ kỹ thuật.

+ Quản lý hệ thống .

+ Quản lý nhân viên .

|  |  |
| --- | --- |
| Tác nhân | Ca sử dụng |
| Khách hàng | 1. Chức năng . 2. Đăng ký tour . 3. Thanh toán . 4. Liên hệ tư vấn . 5. Đăng nhập . |
| Đối tác | 1. Đăng nhập . 2. Tạo gói tour . |
| Nhân viên | 1. Tạo gói tour . 2. Tạo phiếu đăng ký . 3. Tư vấn hỗ trợ khách hàng . |
| Nhân viên kỹ thuật | 1. Đăng nhập . 2. Hỗ trợ kỹ thuật . 3. Quản lý hệ thống . |
| Người quản lý | 1. Đăng nhập . 2. Quản lý thông tin nhân viên . 3. Quản lý thông tin khách hàng . 4. Quản lý hệ thống . |

* 1. Đặc tả User case.
     1. Đặc tả UC “Chức Năng”.

- Tên ca sử dụng : Chức năng

- Mục đích : Cho phép khách hàng xem thông tin tour và có thể tìm kiếm thông tin về các tour như :

+ Tìm địa điểm du lịch,

+ Tìm tour theo thời gian,

+ Tìm nhà hàng, khách sạn,

+ Tìm phương tiện di chuyển .

- Nội dụng : Khách hàng tìm kiếm thông tin về tour du lịch hoặc xem thông tin tour .

- Đối tác : khách hàng .

|  |  |
| --- | --- |
| Hành động của tác nhân | Phản ứng của hệ thống |
| 1. .Chọn chức năng tìm kiếm | 1. . hiện thanh tìm kiếm để khách hàng có thể nhập từ để tìm kiếm các từ khóa. |
| 1. .Điền từ khóa tìm kiếm | 1. . Hiển thị thông tin về từ khóa mà khách hàng đã nhập |

* + 1. .Đặc tả UC “Đăng ký tour” .

- Tên ca sử dụng : Đăng ký tour .

- Mục đích : cho phép khách hàng đăng ký tour theo gói tour có sẵn hoặc có thể đặt tour theo nhu cầu riêng như :

+ Chọn địa điểm .

+ Chọn thời gian .

+ Chọn nhà hàng .

+ Chọn khách sạn .

+ Chọn phương tiện di chuyển .

- Nội dung : khách hàng chọn một gói tour có sẵn hoặc tự đăng ký một gói tour theo nhu cầu rồi ấn đăng ký và kết thúc .

- Đối tác sử dụng : khách hàng.

- Điều kiện đầu vào : ca sử dụng bắt đầu khi nhân viên đăng nhập thành công vào hệ thống.

- Diễn biến :

|  |  |
| --- | --- |
| Hành động tác nhân | Phản ứng hệ thống |
| 1. Khách hàng xem tour, tìm kiếm rồi chọn để đăng ký | 1. Yêu cầu khách hàng đăng nhập |
| 1. Khách hàng nhập thông tin cá nhân | 1. Kiểm tra và xác thực thông tin khách hàng |
| 1. Đăng ký tour theo nhu cầu hoặc đăng ký gói tour có sẵn và thanh toán | 1. Gửi thông tin đăng ký cho nhân viên và xác thực thông tin |

* + 1. . Đặc tả UC “thanh toán”.

- Tên ca sử dụng : Thanh toán,

- Mục đích : cho phép khách hàng thanh toán gói tour mà mình đăng ký.

- Nội dụng : Khách hàng sau khi đăng ký tour thì phải thanh toán gói tour trước.

- Đối tác sử dụng : Khách hàng.

- Điều kiện đầu vào : ca sử dụng bắt đầu khi khách hàng Đăng ký gói tour thành công.

- Diễn biến :

|  |  |
| --- | --- |
| Hành động tác nhân | Phản ứng hệ thống |
| 1. Thanh toán sau khi đăng ký tour | 1. Yêu cầu khách hàng chọn phương thức thanh toán. |
| 1. Chọn phương thức và thanh toán | 1. Xác nhân thông tin thanh toán thành công. |

1.2.4 .Đặc tả UC “liên hệ tư vấn”

- Tên ca sử dụng: liên hệ tư vấn,

- Mục đích: giúp đỡ hỗ trợ khách hàng giải đáp thắc mắc của khách hàng.

- Tóm lược : ca sử dụng cho phép nhân khách hàng liên hệ với nhân viên để giải đáp thắc mấc.

1.2.5 .Đặc tả UC “Tạo gói tour”.

- Tên ca sử dụng : Tạo gói tour.

- Mục đích : Cho phép nhân viên và đối tác tạo những gói tour theo có sẵn của đối tác và đăng

Tải lên hệ thống.

- Đối tượng : đối tác , nhân viên.

- Điều kiện đầu vào : ca sử dụng bắt đầu khi đối tác ,nhân viên đăng nhập thành công vào hệ thống.

-Diễn biến

|  |  |
| --- | --- |
| Hành động tác nhân | Phản ứng hệ thống |
| 1. Chọn tạo phiếu tour | 1. Hiển thị mục để điền thông tin gói tour |
| 1. Điền thông tin gói tour | 1. xác nhận thông tin |
| 1. Đăng tải gói tour lên trang wed | 6.hiện thị gói tour lên trang wed |

1.2.6 .Đặc tả UC “Tạo phiếu Đăng ký”

- Tên ca sử dụng ; Tạo phiếu đăng ký tour

- Mục đích : Cho phép nhân viên tạo phiếu đăng ký tour cho khách hàng bằng hình

thức qua trang wed của công ty.

- Tóm lược : Sau khi khách hàng dăng ký tour theo nhu cầu của khách hàng thì nhân viên dựa

Theo những nhu cầu của khách hàng để liên hệ đối tác đặt lịch.

- Trên phiếu đăng ký tour có thông tin của khách hàng và lựa chọn điểm đi , nhà hàng, khách sạn, phương tiện

Di chuyển của khách hàng.

- Đối tác : nhân viên.

- Điều kiện đầu vào : ca sử dụng bắt đầu khi nhân viên đăng nhập thành công vào hệ thống.

1.2.7 .Đặc tả UC “Hỗ trợ khách hàng”.

- Tên ca sử dụng : Tư vẫn hỗ trợ khách hàng.

- Mục đích : hỗ trợ giải đáp thắc mắc khách hàng và tư vấn về gói tour cho khách hàng hiểu rõ hơn khi có yêu cầu

Của khách hàng.

- Tóm lược : Ca sử dụng cho phép nhân viên dùng chức năng để tư vấn giải đáp thắc mắc cho khách hàng.

- Đối tác : nhân viên.

- Điều kiện đầu vào : ca sử dụng bắt đầu khi nhân viên đăng nhập thành công vào hệ thống.

1.2.8 . Đặc tả UC “Hỗ trợ kỹ thuật”.

- Tên ca sử dụng : Hỗ trợ kỹ thuật.

- Mục đích : Cho phép nhân viên kỹ thuật sửa chữa hệ thống, nâng cấp hệ thống và bảo trì hệ thống.

- Tóm lược : ca sử dụng trên cho phép nhân viên hỗ trợ chỉnh sửa hệ thống .

- Đối tác : Nhân viên kỹ thuật.

1.2.9 .Đặc tả UC “Quản lýtài khoản khách hàng”.

- Tên ca sử dụng : Quảng lý tài khoản khách hàng.

- Mục đích : cho phép nhân viên chỉnh sửa thông tin của khách hàng và cho phép người quản lý

xem thông tin của khách hàng như :

+ Xem tên của khách hàng.

+ Xem Số đth phương thức liên lạc

+ Xem thông tin gói tour mà khách hàng đã đăng ký.

- Tóm lược : Cho phép nhân viên và người quản lý xem thông tin và chỉnh sửa thông tin khách hàng.

- Điều kiện đầu vào : ca sử dụng bắt đầu khi nhân viên hoặc người quản lý đăng nhập thành công vào hệ thống.

1.2.10 .Đặc tả UC “Quản lý nhân viên”.

- Tên ca sử dụng : Quản lý nhân viên

- Mục đích : Giúp nhà quản lý luôn nắm chắc thông tin về nhân viên của mình.

- Tóm lược : Ca sử dụng cho phép người quản lý duy trì thông tin của nhân viên trong hệ thống

Bao gồm các thao tác:

+ Xem thông tin nhân viên

+ Chỉnh Sửa thông tin nhân viên.

+ Thêm nhân viên.

+ Xóa nhân viên.

- Đối tác: người quản lý

- Điều kiện đầu vào : ca sử dụng bắt đầu khi người quản lý đăng nhập thành công vào hệ thống.

1.2.11 .Đặc tả UC “Quản lý hệ thống”

-Tên ca sử dụng : Quản lý hệ thống.

- Mục đích : giúp người quản lý điều hành hệ thống và quản lý hệ thống bao gồm các thao tác sau:

+ Kiểm tra vẫn hành hệ thống.

+ Theo dõi hệ thống

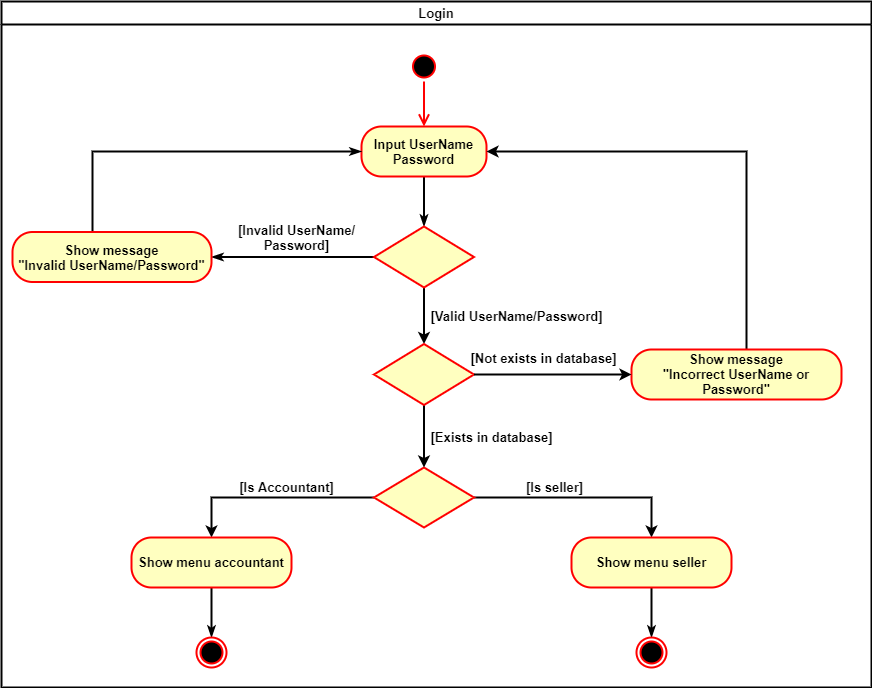
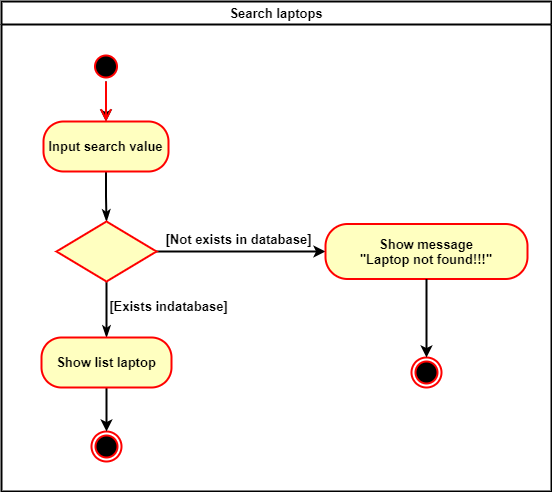
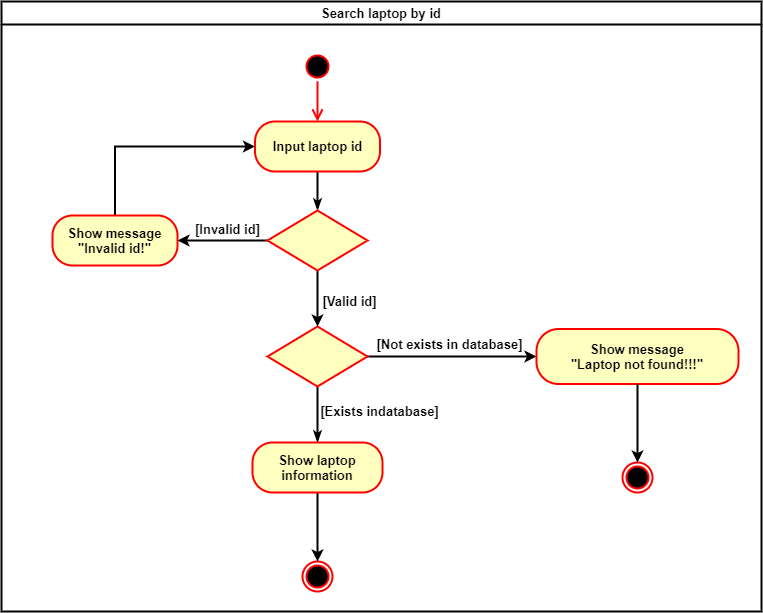
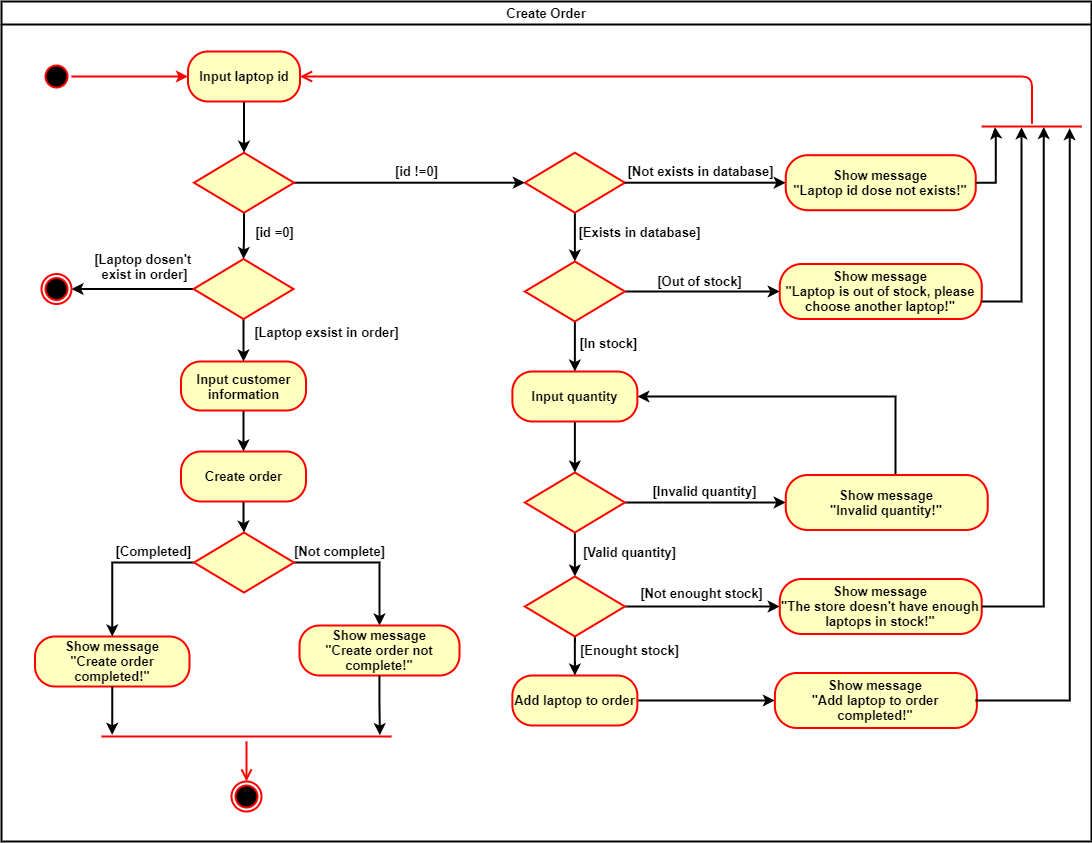
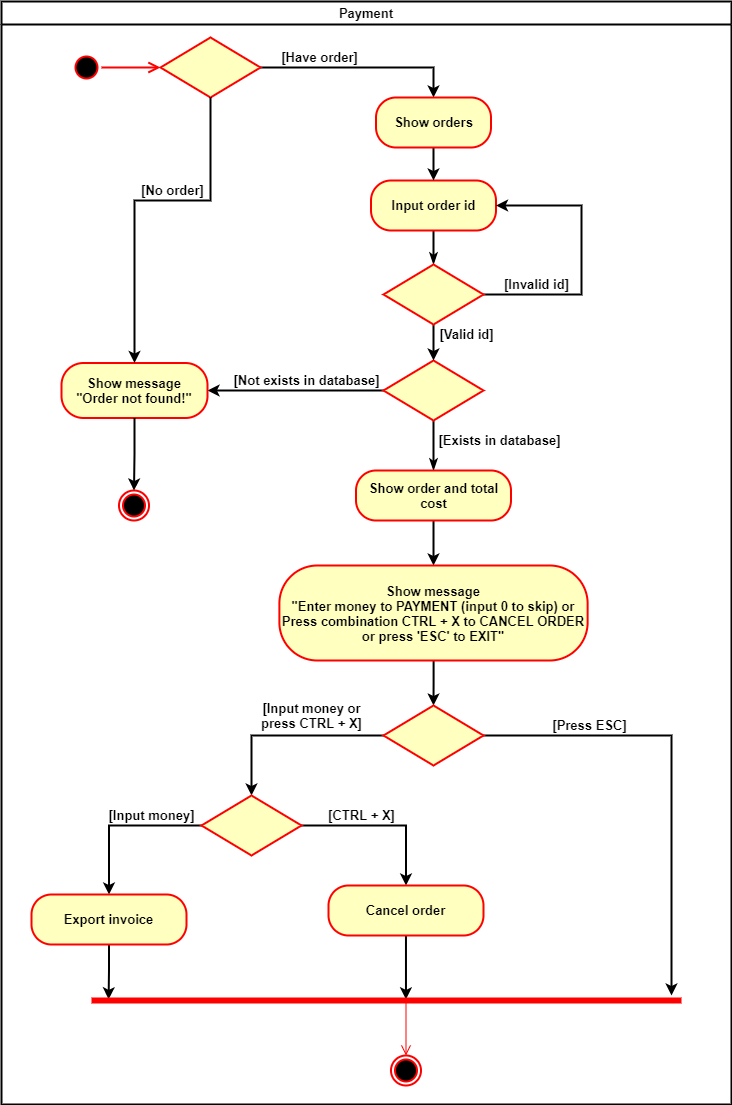
+ Quản lý các gói tour.

+ Quản lý mã giả giá.

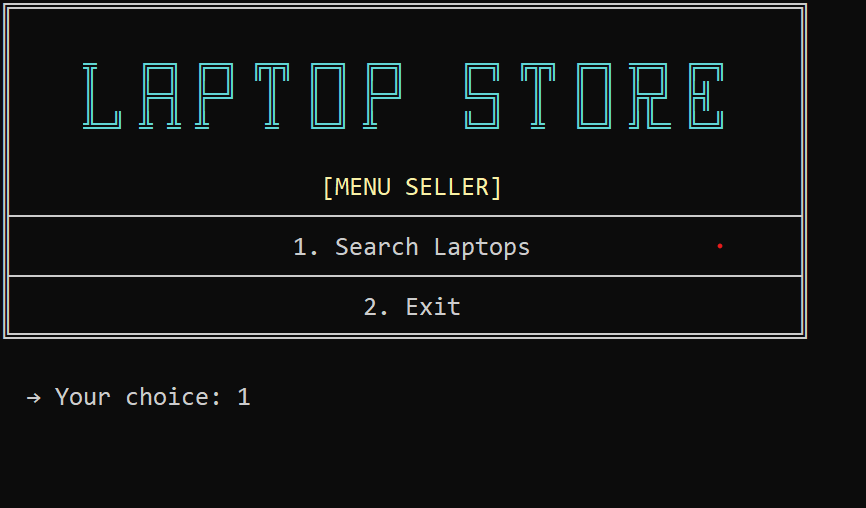
- Tóm lược: Ca sử dụng giúp người quản lý có thể điều hành và quản lý hệ thống .

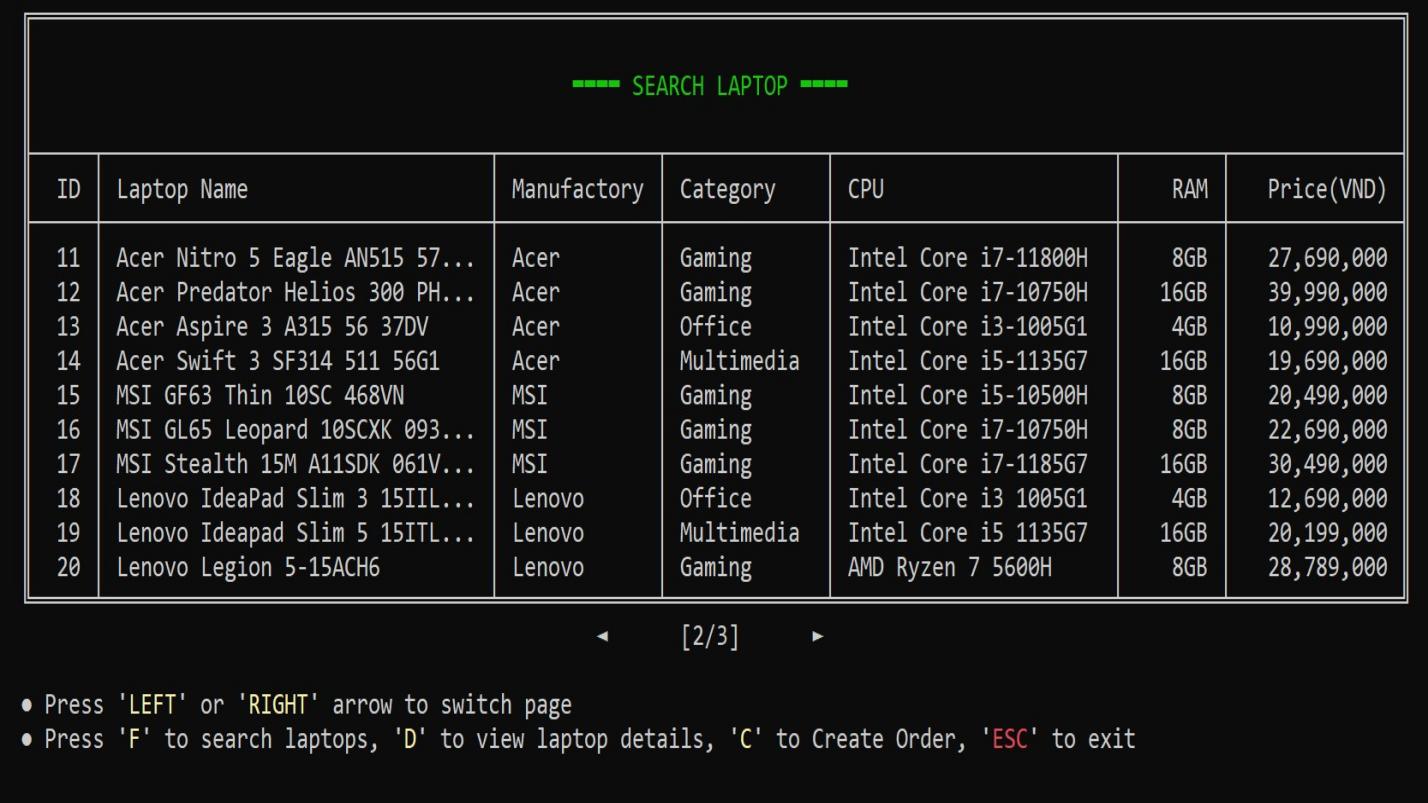
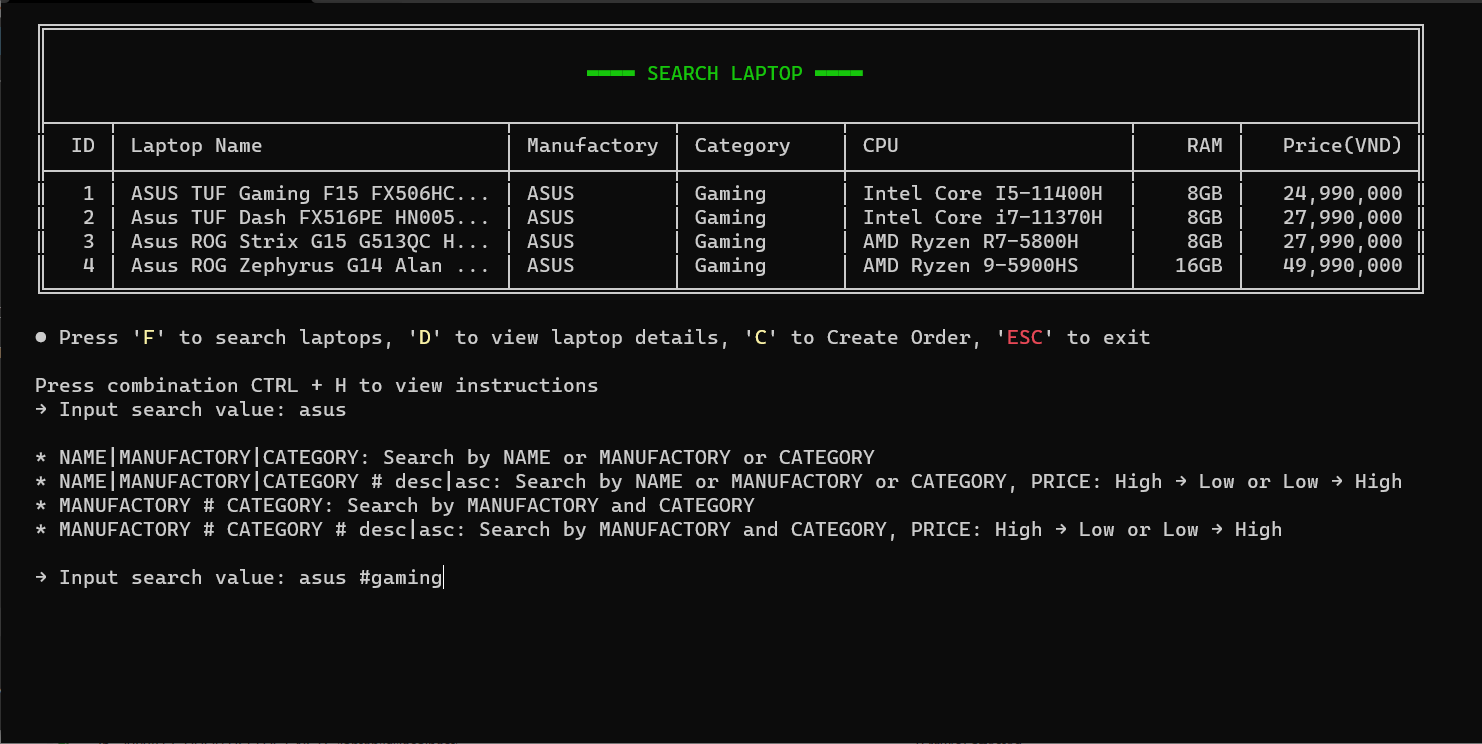
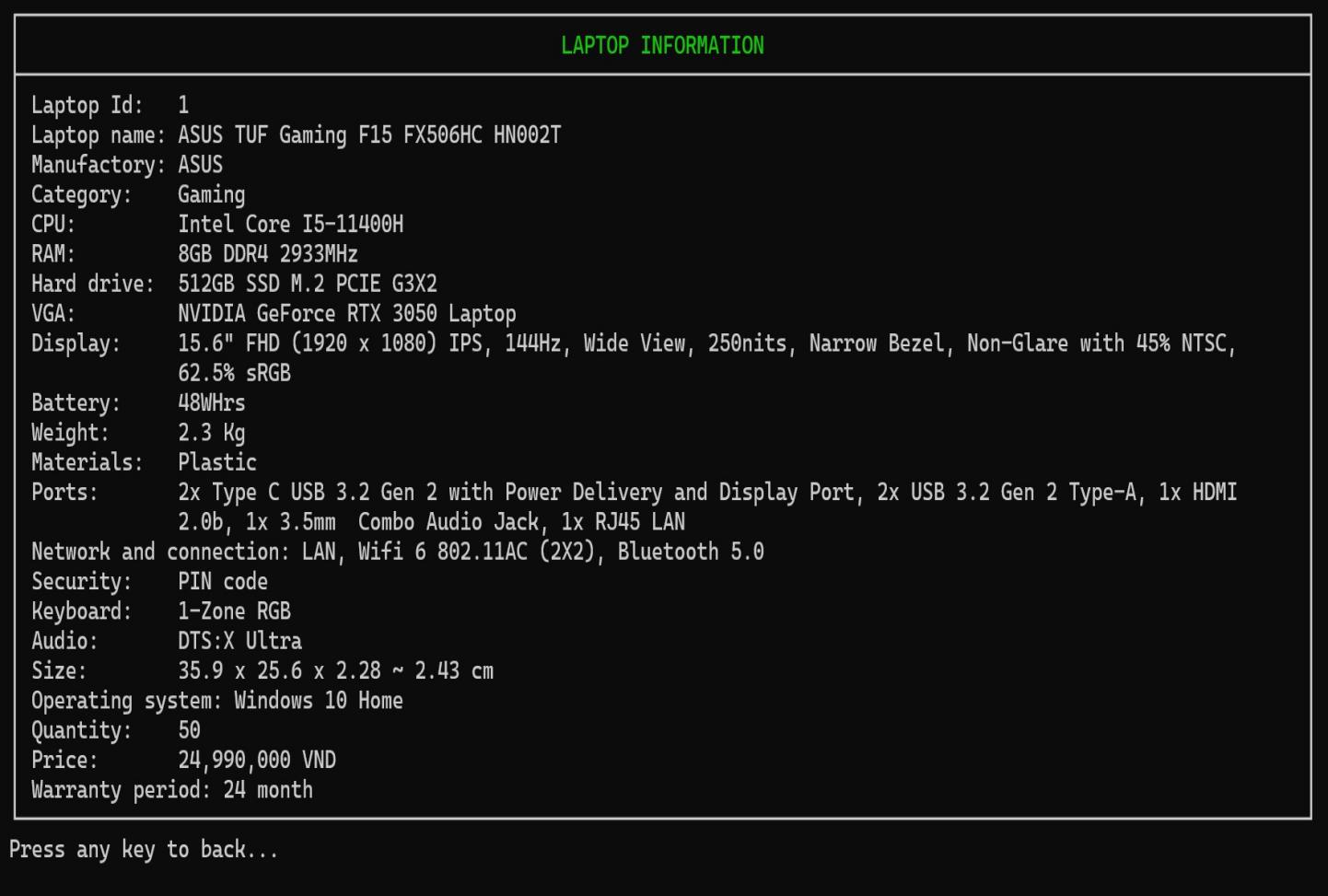
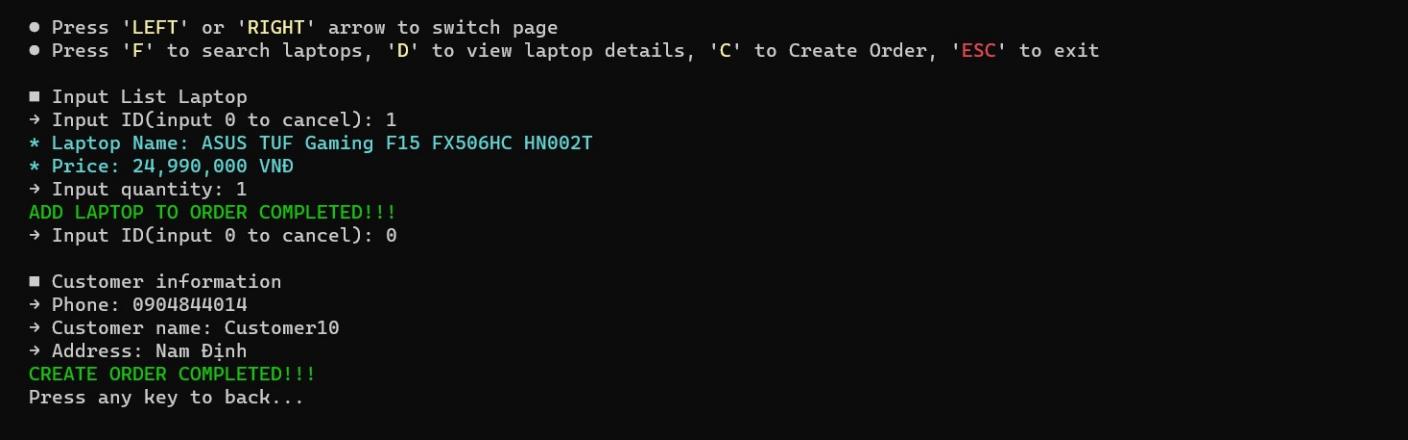
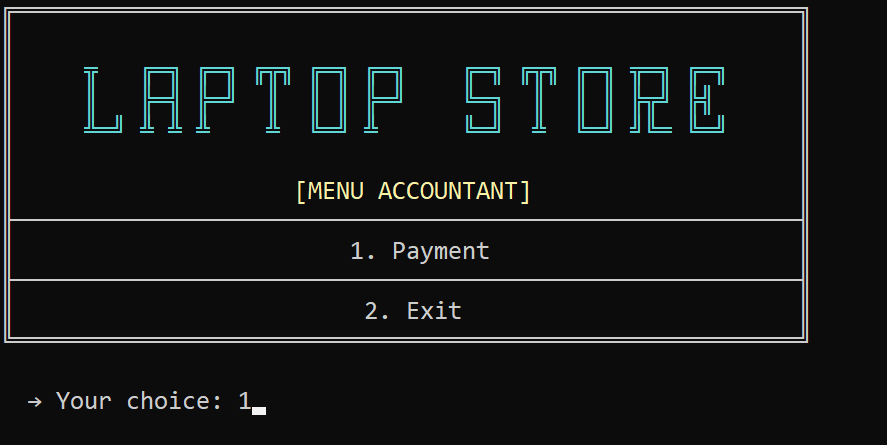
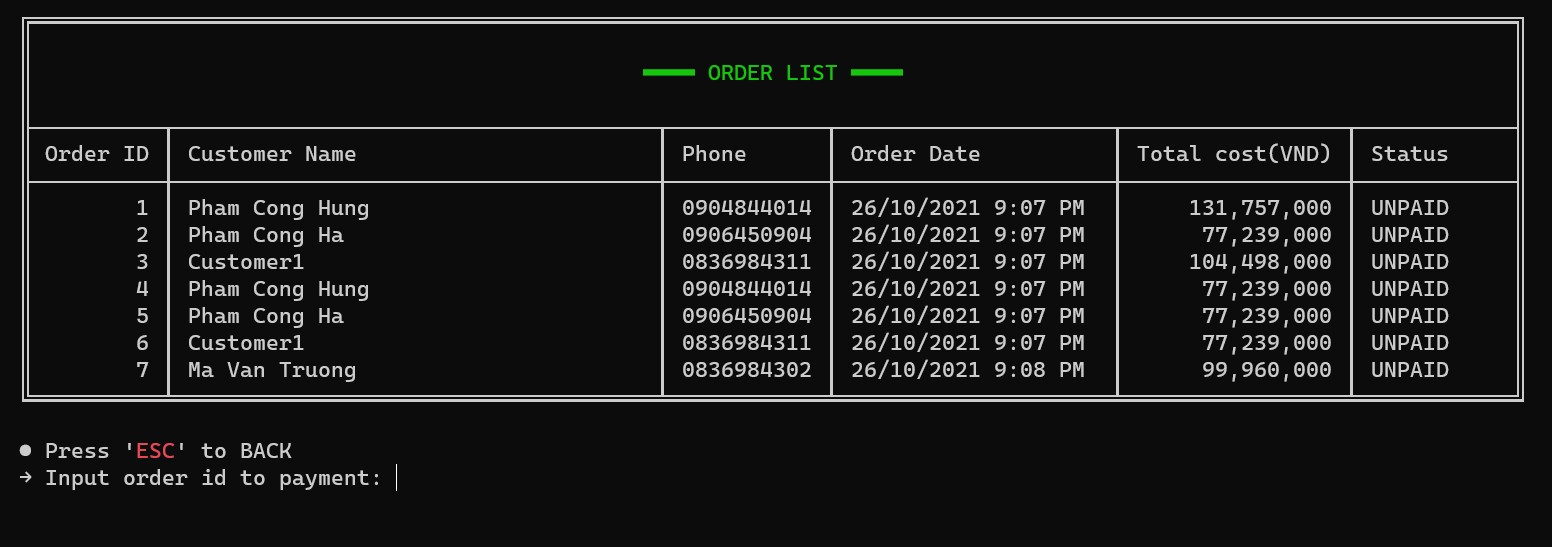
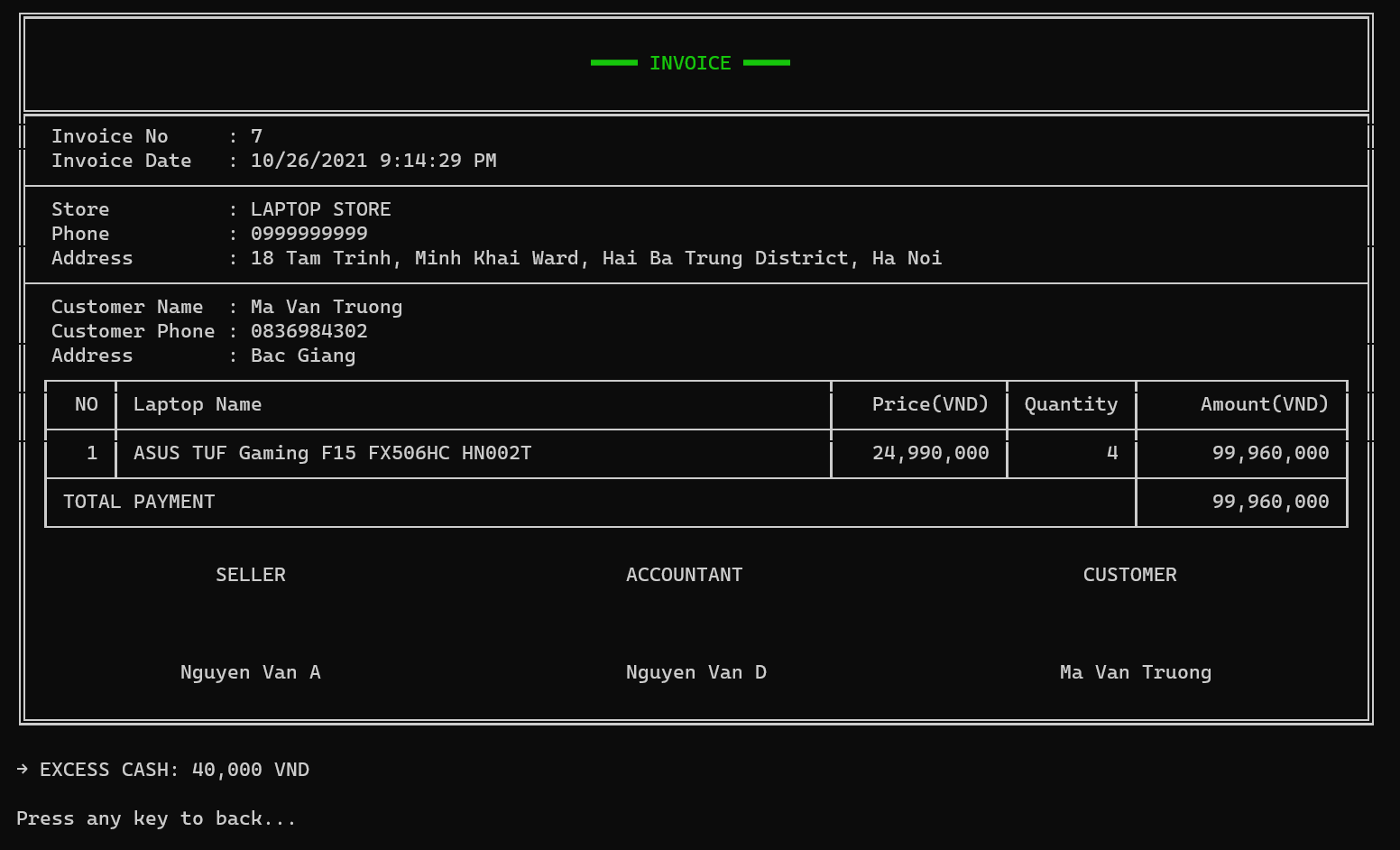
- Đối tác : Người quản lý

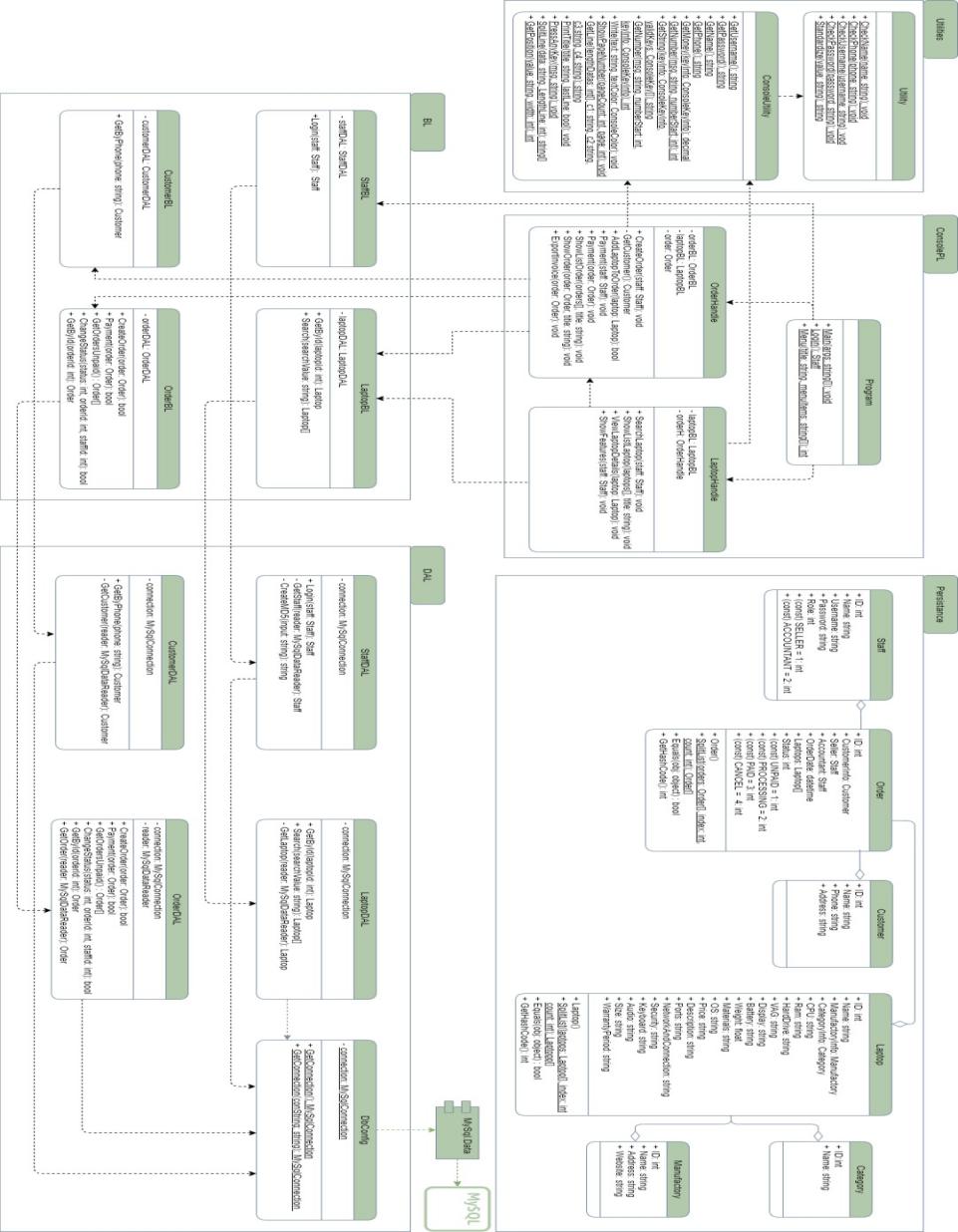
- Điều kiện đầu vào : ca sử dụng bắt đầu khi người quản lý đăng nhập thành công vào hệ thống.

* 1. Activity Diagram
  2. Login
  3. Search laptop
  4. Search laptop by id
  5. Create order
  6. Payment

1. Design Details
   1. UI Design
   2. Login
   3. Menu seller

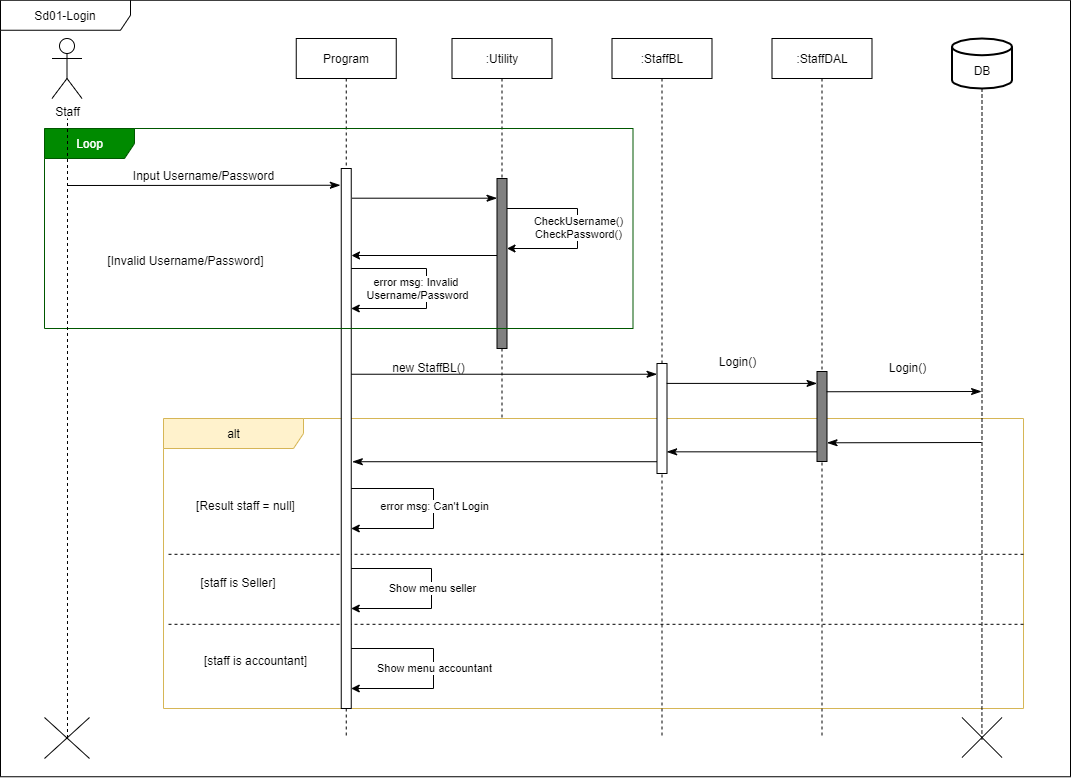


* 1. List of laptop
  2. Search laptop
  3. Show laptop information
  4. Create order
  5. Menu Accountant
  6. Show order list
  7. Payment
  8. Invoice

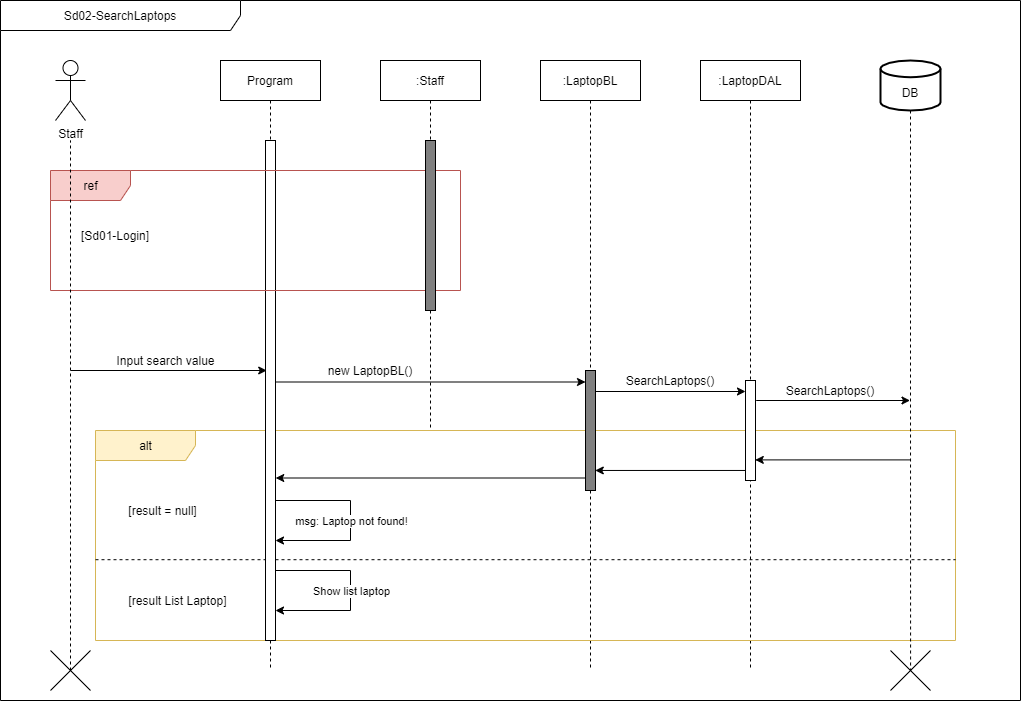


* 1. Code Design (Class Diagram)

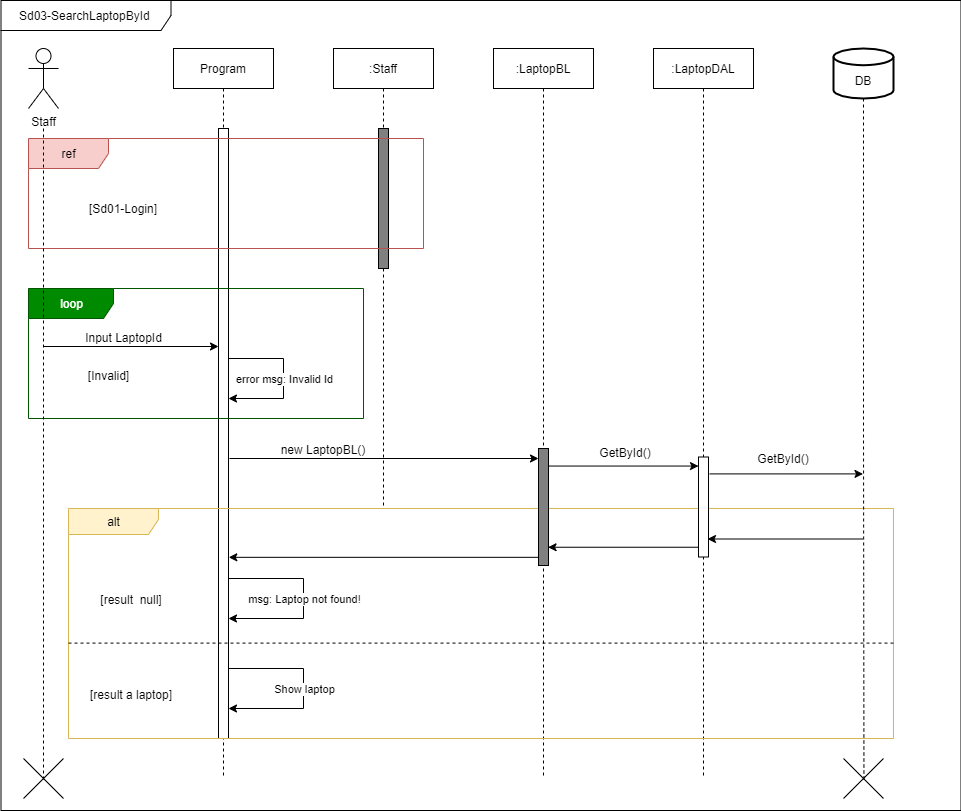
* 1. Sequence Diagram
  2. Login

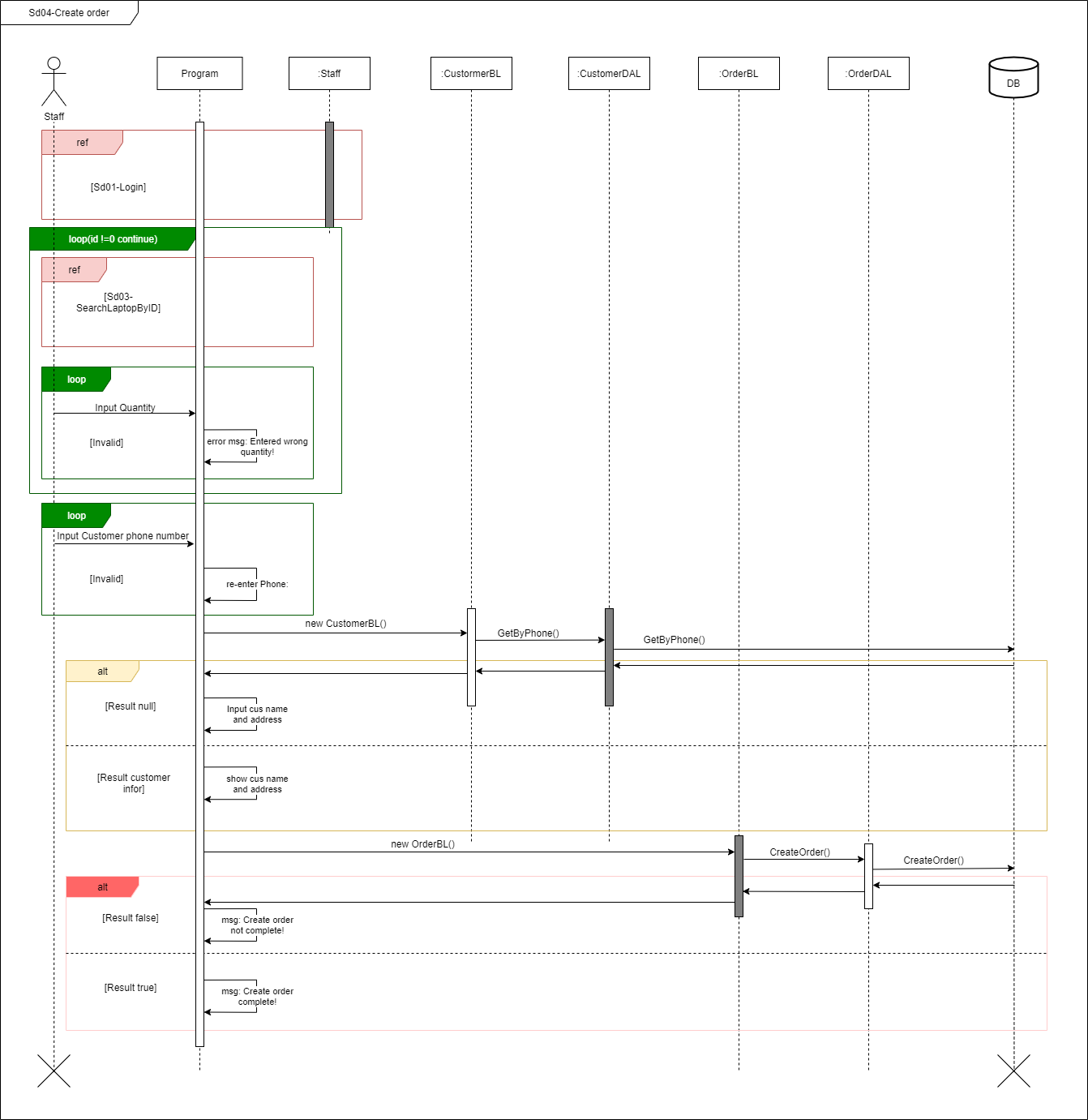
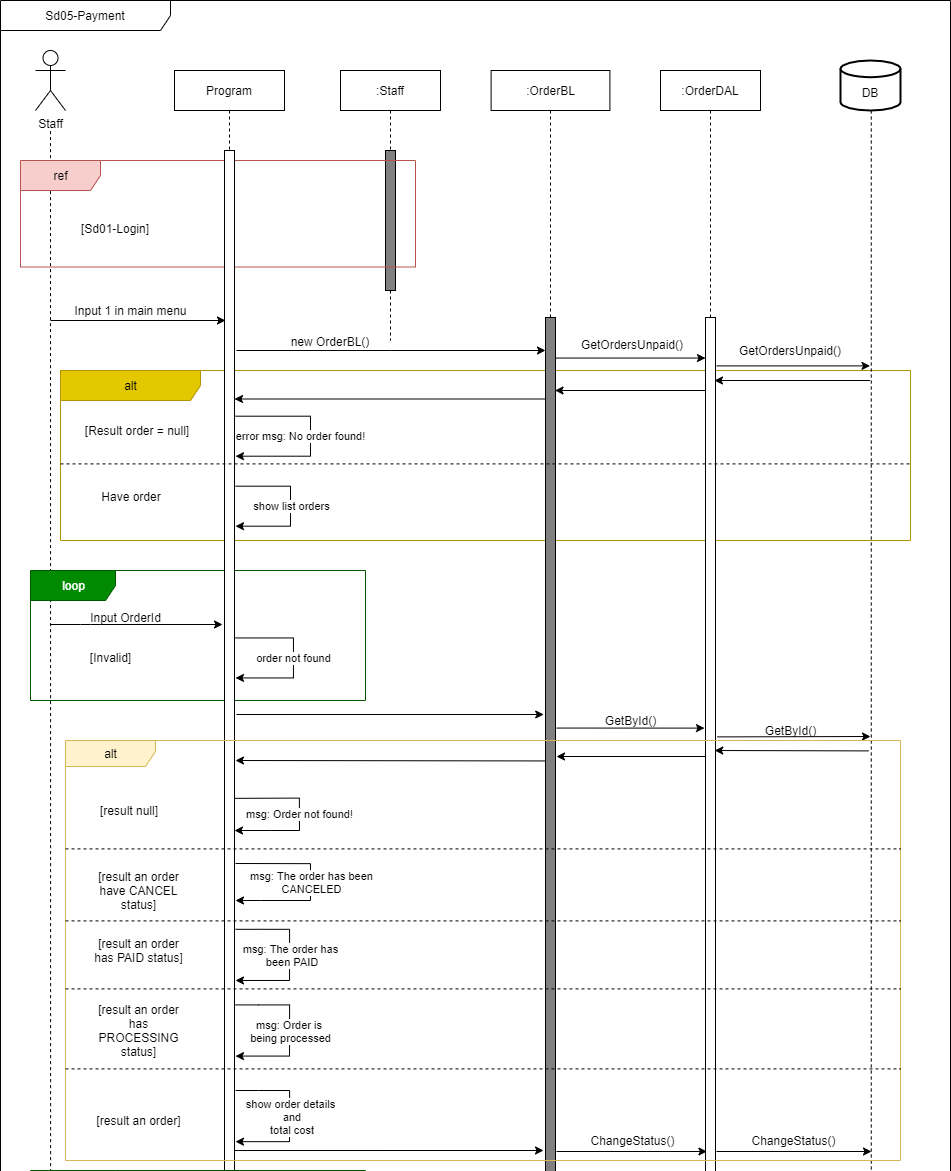


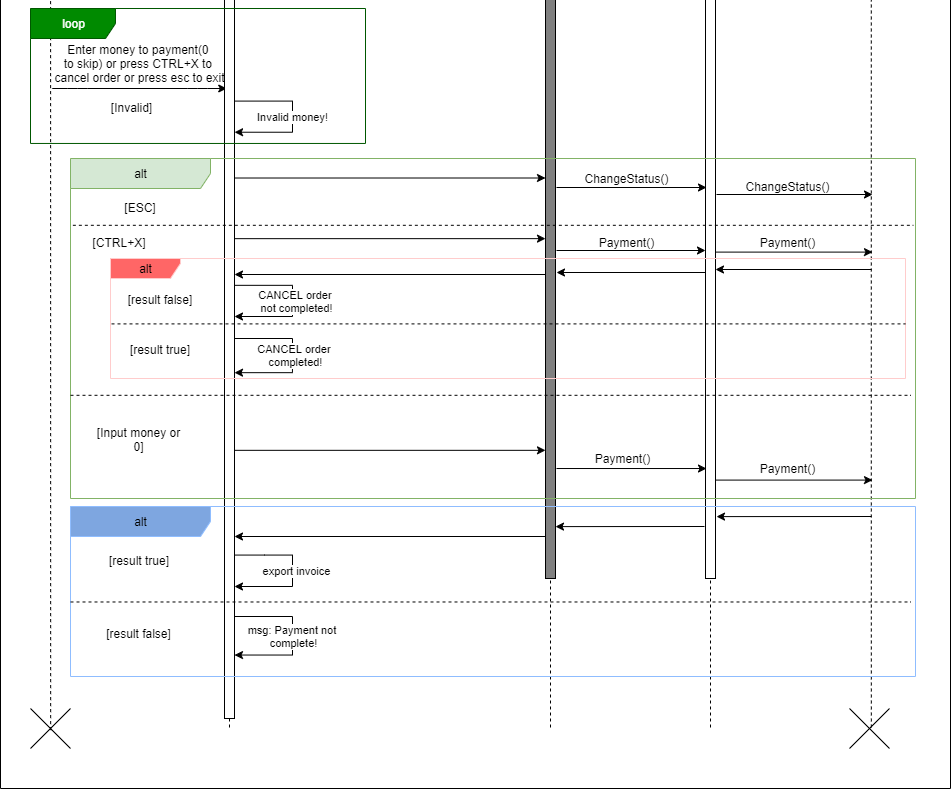
* 1. Search laptops



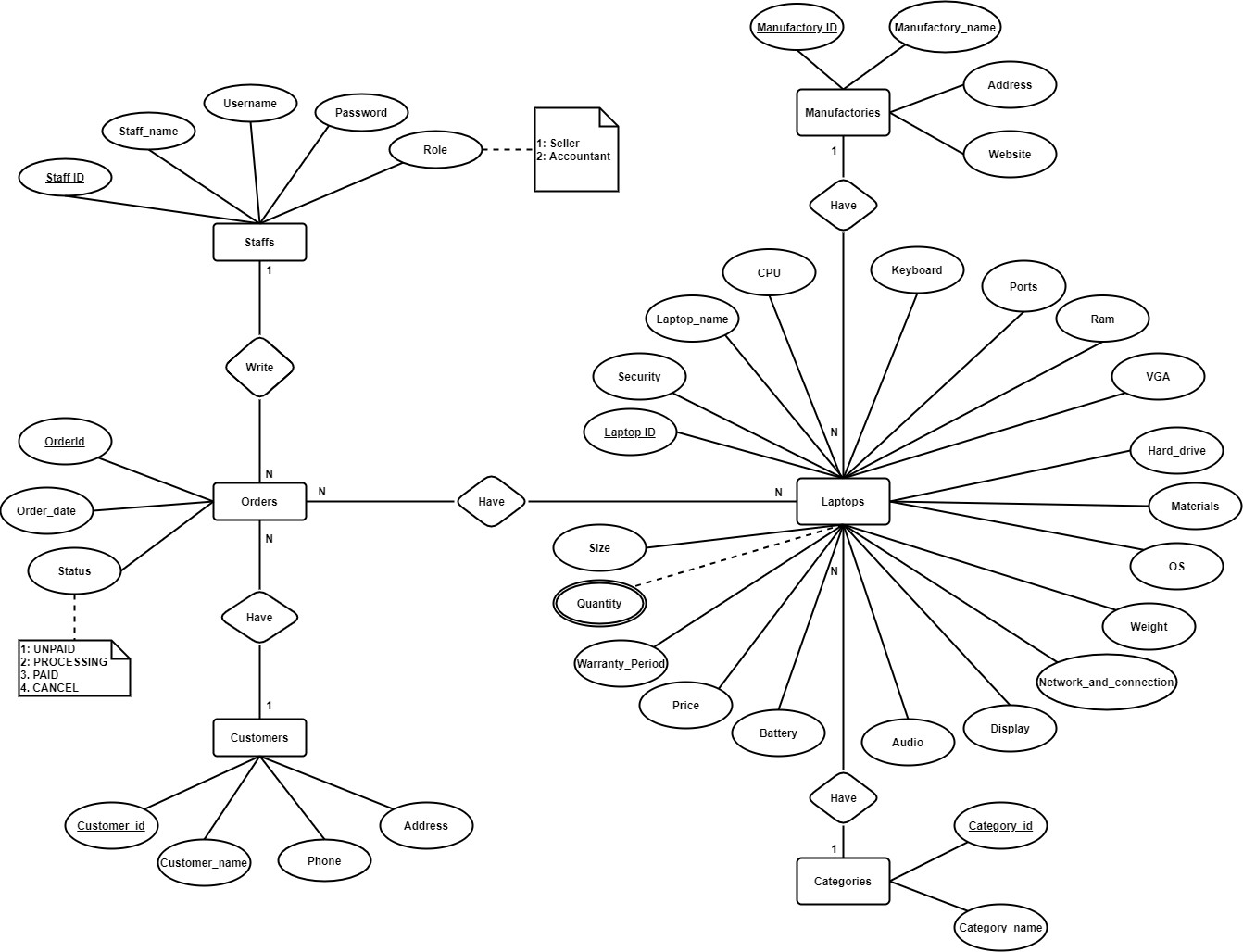
* 1. Search laptop by id



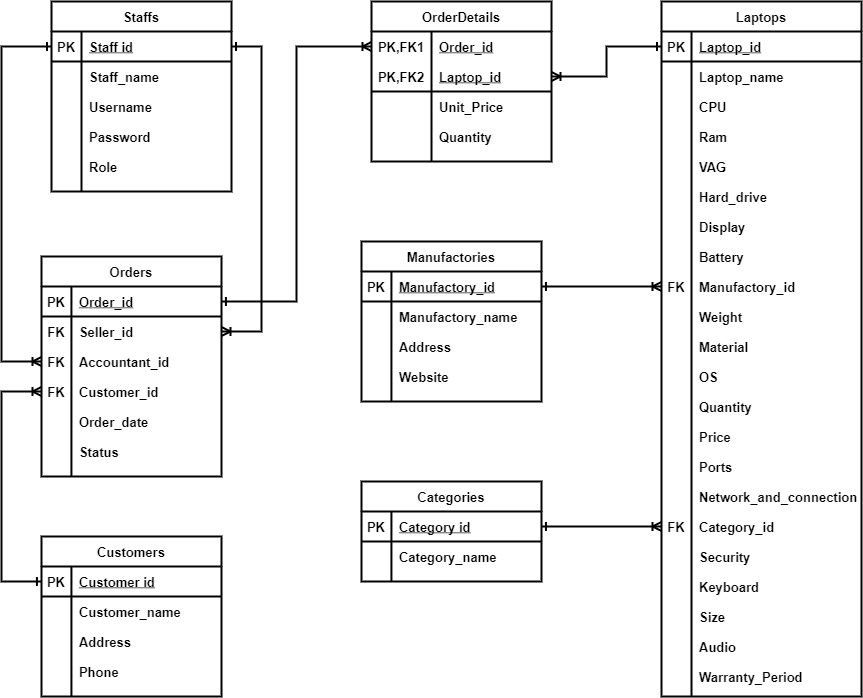
* 1. Create order
  2. Payment



* 1. Database Design
  2. Entity Relationship Diagram



* 1. Database Design Details



|  |  |  |  |
| --- | --- | --- | --- |
| staﬀs | | | |
| Column Name | Data Type | Constraints | Description |
| staﬀ\_id | int | PRIMARY KEY, AUTO  INCREMENT |  |
| username | varchar(255) | NOT NULL |  |
| password | varchar(255) | NOT NULL, UNIQUE |  |
| staﬀ\_name | varchar(50) | NOT NULL | Staﬀ’s name |
| role | int | NOT NULL | 1: Seller  2: Accountant |

|  |  |  |  |
| --- | --- | --- | --- |
| customers | | | |
| Column Name | Data Type | Constraints | Description |
| customer\_id | int | PRIMARY KEY, AUTO  INCREMENT |  |
| customer\_name | varchar(50) | NOT NULL | Customer name |
| Phone | varchar(10) | NOT NULL, UNIQUE | Customer’s Phone  number |
| address | varchar(100) |  | Customer’s address |

|  |  |  |  |
| --- | --- | --- | --- |
| manufactories | | | |
| Column Name | Data Type | Constraints | Description |
| manufactory\_id | int | PRIMARY KEY, AUTO  INCREMENT |  |
| manufactory\_name | varchar(50) | NOT NULL, UNIQUE | Manufacturer name |
| website | varchar(255) |  | Manufacturer's website |
| address | varchar(255) |  | Manufacturer’s address |

|  |  |  |  |
| --- | --- | --- | --- |
| categories | | | |
| Column Name | Data Type | Constraints | Description |
| category\_id | int | PRIMARY KEY, AUTO  INCREMENT |  |
| category\_name | varchar(50) | NOT NULL, UNIQUE | Laptop category |

|  |  |  |  |
| --- | --- | --- | --- |
| orders | | | |
| Column Name | Data Type | Constraints | Description |
| order\_id | int | PRIMARY KEY, AUTO  INCREMENT |  |
| customer\_id | int | NOT NULL |  |
| seller\_id | int | NOT NULL |  |
| accoutant\_id | int |  |  |
| order\_date | datetime | NOT NULL, DEFAULT  CURRENT DATE | Order creation date |
| status | int | NOT NULL | 1: Unpaid  2: Processing  3: Paid  4: Cancel |

|  |  |  |  |
| --- | --- | --- | --- |
| order\_details | | | |
| Column Name | Data Type | Constraints | Description |
| order\_id | int | PRIMARY KEY, FOREIGN  KEY |  |
| laptop\_id | int | PRIMARY KEY, FOREIGN  KEY |  |
| unit\_price | Decimal | NOT NULL | Laptop’s unit price |
| quantity | int | NOT NULL, DEFAULT 1 | Laptop’s Quantity |

|  |  |  |  |
| --- | --- | --- | --- |
| laptops | | | |
| Column Name | Data Type | Constraints | Description |
| laptop\_id | int | PRIMARY KEY, AUTO  INCREMENT |  |
| laptop\_name | varchar(100) | NOT NULL | Laptop’s name |
| category\_id | int | NOT NULL | Laptop’s category |
| manufactory\_id | int | NOT NULL | Laptop’s manufacturer |
| CPU | varchar(100) | NOT NULL | Laptop’s CPU |
| Ram | varchar(100) | NOT NULL | Laptop’s Ram |
| hard\_drive | varchar(100) | NOT NULL | Laptop’s Hard\_drive |
| VGA | varchar(100) | NOT NULL | Laptop’s VGA |
| display | varchar(255) | NOT NULL | Laptop’s Display |
| battery | varchar(255) | NOT NULL | Laptop’s Battery |
| weight | ﬂoat | NOT NULL | Laptop’s Weight,  unit: kilogram |
| materials | varchar(100) | NOT NULL | Laptop’s Materials |
| ports | varchar(255) | NOT NULL | Laptop’s Ports |
| network\_and\_co  nnection | varchar(100) | NOT NULL | Laptop’s Network and  connection |
| security | varchar(100) | NOT NULL | Laptop’s Security |
| keyboard | varchar(100) | NOT NULL | Laptop’s Keyboard |
| audio | varchar(100) | NOT NULL | Laptop’s Audio |
| size | varchar(100) | NOT NULL | Laptop’s Size |
| OS | varchar(100) | NOT NULL | Laptop’s OS |
| warranty\_period | varchar(100) | NOT NULL | Laptop’s Warranty period |
| price | Decimal | NOT NULL, DEFAULT 0 | Laptop’s Price |
| quantity | int | NOT NULL, DEFAULT 0 | Laptop’s Quantity |

1. Test
   1. Login test
   2. Login test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | LoginTest1() |
| Test Case Description | This test case checks username and password correct or not |
| Preconditions | Username and password must exist in the database |
| Test Case Input | “seller001”, “12345678”  “accountant003”, “12345678” |
| Test Case Expected Output | A Staﬀ |
| Test Case Steps | New Staﬀ  New StaﬀDAL  Call Login method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Login test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | LoginTest2() |
| Test Case Description | This test case checks username and password correct or not |
| Preconditions |  |
| Test Case Input | "seller00fdsj", "12345678"  "seller002", "ddf12345678" "SEller002", "12345678" |
| Test Case Expected Output | Null |
| Test Case Steps | New Staﬀ  New StaﬀDAL  Call Login method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Search Laptops test
  2. Search Laptops test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | SearchLaptopsTest1() |
| Test Case Description | Search for products in the system |
| Preconditions | Logged in |
| Test Case Input | “a”  “oﬃce” “macbook” “asus” |
| Test Case Expected Output | List laptop |
| Test Case Steps | New LaptopDAL Call Search method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Search Laptops test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | SearchLaptopsTest2() |
| Test Case Description | Search for products in the system |
| Preconditions | Logged in |
| Test Case Input | “a # desc”  “oﬃce # desc” “asus # desc” |
| Test Case Expected Output | List laptop |
| Test Case Steps | New LaptopDAL Call Search method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Search Laptops test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | SearchLaptopsTest3() |
| Test Case Description | Search for products in the system |
| Preconditions | Logged in |
| Test Case Input | “a # asc” “oﬃce # asc”  “asus # asc” |
| Test Case Expected Output | List laptop |
| Test Case Steps | New LaptopDAL Call Search method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Search Laptops test 4

|  |  |
| --- | --- |
| Test Case Number | 004 |
| Test Case Name | SearchLaptopsTest4() |
| Test Case Description | Search for products in the system |
| Preconditions | Logged in |
| Test Case Input | “asus # gaming” “gaming # acer”  “apple # multimedia” |
| Test Case Expected Output | List laptop |
| Test Case Steps | New LaptopDAL Call Search method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Search Laptops test 5

|  |  |
| --- | --- |
| Test Case Number | 005 |
| Test Case Name | SearchLaptopsTest5() |
| Test Case Description | Search for products in the system |
| Preconditions | Logged in |
| Test Case Input | “asus # gaming # desc” “gaming # acer # desc”  “apple # multimedia # desc” |
| Test Case Expected Output | List laptop |
| Test Case Steps | New LaptopDAL Call Search method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Search Laptops test 6

|  |  |
| --- | --- |
| Test Case Number | 006 |
| Test Case Name | SearchLaptopsTest6() |
| Test Case Description | Search for products in the system |
| Preconditions | Logged in |
| Test Case Input | “asus # gaming # asc” “gaming # acer # asc”  “apple # multimedia # asc” |
| Test Case Expected Output | Null |
| Test Case Steps | New LaptopDAL Call Search method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Search Laptops test 7

|  |  |
| --- | --- |
| Test Case Number | 007 |
| Test Case Name | SearchLaptopsTest7() |
| Test Case Description | Search for products in the system |
| Preconditions | Logged in |
| Test Case Input | “sdfasjhfj”  “asus # dell #” “a # # asus”  “dell # mac # # desc” |
| Test Case Expected Output | Null |
| Test Case Steps | New LaptopDAL Call Search method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Get laptop by ID test
  2. Get laptop by id test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | GetLaptopByIdTest1() |
| Test Case Description | Search for laptop in the system |
| Preconditions | Logged in |
| Test Case Input | 1  5  27 |
| Test Case Expected Output | A laptop |
| Test Case Steps | New LaptopDAL  Call GetById method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Get laptop by id test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | GetLaptopByIdTest2() |
| Test Case Description | Search for laptop in the system |
| Preconditions | Logged in |
| Test Case Input | 0  29 |
| Test Case Expected Output | Null |
| Test Case Steps | New LaptopDAL  Call GetById method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Get Customer by phone test
  2. Get Customer by phone test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | GetCustomerByPhoneTest1() |
| Test Case Description | Search for customer in the system |
| Preconditions | Logged in |
| Test Case Input | “0836984311”  “0836984312”  “0836984313” |
| Test Case Expected Output | A customer |
| Test Case Steps | New CustomerDAL  Call GetByPhone method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Get Customer by phone test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | GetCustomerByPhoneTest2() |
| Test Case Description | Search for customer in the system |
| Preconditions | Logged in |
| Test Case Input | “”  “0836984310” |
| Test Case Expected Output | Null |
| Test Case Steps | New CustomerDAL  Call GetByPhone method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test
  2. Create order test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | CreateOrderTest1() |
| Test Case Description | Create an order with at least one laptop and customer information |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | CreateOrderTest2() |
| Test Case Description | Create an order with at least one laptop and customer information |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 3

|  |  |
| --- | --- |
| Test Case Number | 003 |
| Test Case Name | CreateOrderTest3() |
| Test Case Description | Create an order with at least one laptop and customer information |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 4

|  |  |
| --- | --- |
| Test Case Number | 004 |
| Test Case Name | CreateOrderTest4() |
| Test Case Description | Create an order with at least one laptop and customer information |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 5

|  |  |
| --- | --- |
| Test Case Number | 005 |
| Test Case Name | CreateOrderTest5() |
| Test Case Description | Create an order with at least one laptop and customer information |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 6

|  |  |
| --- | --- |
| Test Case Number | 006 |
| Test Case Name | CreateOrderTest6() |
| Test Case Description | Create an order with at least one laptop and customer information |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 7

|  |  |
| --- | --- |
| Test Case Number | 007 |
| Test Case Name | CreateOrderTest7() |
| Test Case Description | Create an order with at least one laptop and customer information |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 8

|  |  |
| --- | --- |
| Test Case Number | 008 |
| Test Case Name | CreateOrderTest8() |
| Test Case Description | Create an order with at least one laptop and customer information |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 9

|  |  |
| --- | --- |
| Test Case Number | 009 |
| Test Case Name | CreateOrderTest9() |
| Test Case Description | Create an order with at least one laptop and customer information |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Create order test 10

|  |  |
| --- | --- |
| Test Case Number | 010 |
| Test Case Name | CreateOrderTest10() |
| Test Case Description | Create an order with at least one laptop and customer information |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | False |
| Test Case Steps | New Order  New OrderDAL  Call CreateOrder method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Get orders unpaid test

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | GetOrderUnpaidTest1() |
| Test Case Description | Take order with status as unpaid |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | List Order |
| Test Case Steps | New OrderDAL  Call GetOrdersUnpaid method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Get order by ID test
  2. Get order by ID test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | GetOrderByIdTest1() |
| Test Case Description | Take order by insert order id |
| Preconditions | Logged in |
| Test Case Input | 1  2  3 |
| Test Case Expected Output | An order |
| Test Case Steps | New OrderDAL  Call GetOrderById method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Get order by ID test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | GetOrderByIdTest2() |
| Test Case Description | Take order by insert order id |
| Preconditions | Logged in |
| Test Case Input | 0  1000 |
| Test Case Expected Output | null |
| Test Case Steps | New OrderDAL  Call GetOrderById method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Change order status test
  2. Change order status test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | ChangeOrderStatusTest1() |
| Test Case Description | Change order status |
| Preconditions | Logged in |
| Test Case Input | 1, 1, 5  3, 2, 4  4, 3, 6 |
| Test Case Expected Output | True |
| Test Case Steps | New OrderDAL  Call ChangeOrderStatus method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Change order status test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | ChangeOrderStatusTest2() |
| Test Case Description | Change order status |
| Preconditions | Logged in |
| Test Case Input | 1, 1, 100  3, 2, 0  4, 1000, 6 |
| Test Case Expected Output | False |
| Test Case Steps | New OrderDAL  Call ChangeOrderStatus method  Compare output with expected output |
| Default Value Preverving |  |

* 1. Payment test
  2. Payment test 1

|  |  |
| --- | --- |
| Test Case Number | 001 |
| Test Case Name | PaymentTest1() |
| Test Case Description | Conﬁrm payment or cancel order |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL  Call GetById method Call Payment method  Compare output with expected output |
| Default Value Preverving |  |

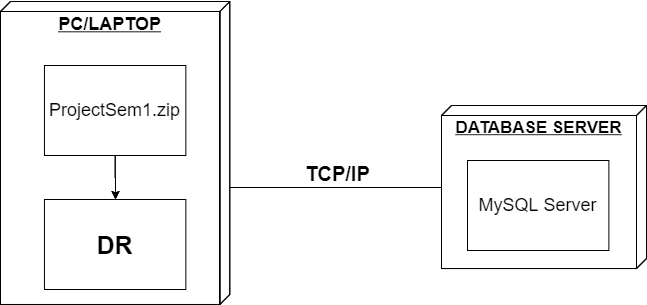
* 1. Payment test 2

|  |  |
| --- | --- |
| Test Case Number | 002 |
| Test Case Name | PaymentTest2() |
| Test Case Description | Conﬁrm payment or cancel order |
| Preconditions | Logged in |
| Test Case Input |  |
| Test Case Expected Output | True |
| Test Case Steps | New Order  New OrderDAL Call ById method  Call Payment method  Compare output with expected output |
| Default Value Preverving |  |

1. TaskAssign (to each team member)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Group  6 | Laptop Store | | | | | |
| No | Task name | Descriptio n | Start Date | End Date | Member | Self assessme  nt |
| 1 | Write report |  | 2 Aug | 29 Aug | Hung & Truong |  |
| 2 | Code |  | 15 Aug | 10 Sep | Truong |  |
| 3 | Code |  | 05 Sep | 30 Sep | Hung |  |
| 4 | Code |  | 25 Sep | 13 Oct | Truong |  |
| 5 | Control |  | 13 Oct | 19 Oct | Hung & Truong |  |

1. Installation Instructions
   1. Deployment Diagram



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