

PROJECT REPORT

# Laptop Store

Semester: Programming Fundamentals

Class: PF13

Group: Group 6

Instructor Nguyen Xuan Sinh

Team members: Pham Cong Hung Ma Van Truong

Index

[Laptop Store 1](#_TOC_250000)

Index 2

1. Project introduction 4
2. Analyze System Requirements 6
   1. Use Case 7
      1. Use Case 7
      2. Login description 8
      3. Search laptops description 9
      4. View laptop details description 10
      5. Create order description 11
      6. Payment description 12
   2. Activity Diagram 14
      1. Login 14
      2. Search laptop 15
      3. Search laptop by id 16
      4. Create order 17
      5. Payment 18
3. Design Details 19
   1. UI Design 19
      1. Login 19
      2. Menu seller 19
      3. List of laptop 20
      4. Search laptop 20
      5. Show laptop information 21
      6. Create order 21
      7. Menu Accountant 22
      8. Show order list 22
      9. Payment 23
      10. Invoice 23
   2. Code Design (Class Diagram) 24
   3. Sequence Diagram 25
      1. Login 25
      2. Search laptops 26
      3. Search laptop by id 27
      4. Create order 28
      5. Payment 29
   4. Database Design 31
      1. Entity Relationship Diagram 31
      2. Database Design Details 32
4. Test 36
   1. Login test 36
      1. Login test 1 36
      2. Login test 2 37
   2. Search Laptops test 38
      1. Search Laptops test 1 38
      2. Search Laptops test 2 39
      3. Search Laptops test 3 39
      4. Search Laptops test 4 40
      5. Search Laptops test 5 40
      6. Search Laptops test 6 41
      7. Search Laptops test 7 41
   3. Get laptop by ID test 42
      1. Get laptop by id test 1 42
      2. Get laptop by id test 2 42
   4. Get Customer by phone test 43
      1. Get Customer by phone test 1 43
      2. Get Customer by phone test 2 43
   5. Create order test 44
      1. Create order test 1 44
      2. Create order test 2 44
      3. Create order test 3 45
      4. Create order test 4 45
      5. Create order test 5 46
      6. Create order test 6 46
      7. Create order test 7 47
      8. Create order test 8 47
      9. Create order test 9 48
      10. Create order test 10 48
   6. Get orders unpaid test 49
   7. Get order by ID test 49
      1. Get order by ID test 1 49
      2. Get order by ID test 2 50
   8. Change order status test 50
      1. Change order status test 1 50
      2. Change order status test 2 51
   9. Payment test 51
      1. Payment test 1 51
      2. Payment test 2 52
5. Task Assign (to each team member) 52
6. Installation Instructions 53
7. Project introduction

This system will provide some basic features including:

There are 2 main features include create order, conﬁrm payment or cancel order which includes additional features such as view laptop list, search laptop, view laptop details, view order list, view order details.

* 1. Proposed System
     + Creating a program to Seller create order
     + Creating a program to Accountant conﬁrm payment or cancel order
  2. The scope of the project to be applied

- Make for actor are Seller and Accountant

* 1. System Name

- Order management system

* 1. Deployment Environment

- Windows, Linux, Mac

* 1. Development Tools
* Visual Studio Code
* Draw.io Diagram
* MySQL Workbench 8.0 CE
  1. Customer Requirements
* Seller login
* Show laptops
* Create order
* Accountant login
* Show orders
* View order details
* Conﬁrm payment or Cancel order

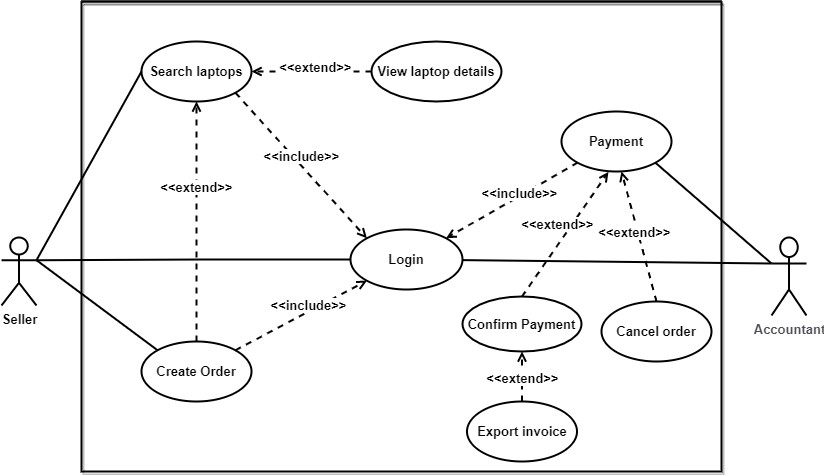
1. Analyze System Requirements

This system was created to help laptop stores manage orders more easily. It can help the seller to have more information about the laptops sold in the store, that information can include price, brand, type, information about hardware conﬁguration etc so that the seller

can give more speciﬁc advice to the buyer according to the needs of each person. The main function of this system for the seller is that it helps the seller to create the order

according to the buyer's request. As for accountants, this system allows them to view the information of unpaid orders and conﬁrm successful payment if enough money has been received from the buyer, cancel the order when the buyer changes his mind or pause payment in case of necessity.

* 1. Use Case
  2. Use Case



* 1. Login description

|  |  |
| --- | --- |
| Name | Login |
| ID | UC\_001 |
| Description | Used to access the program and deﬁne the feature accessible depending on the type of account. |
| Actor | Seller and Accountant |
| Organizational Beneﬁts | This feature was created to deﬁne the features and data that seller and accountant can access in the program. |
| Trigger | Login will be automatically activated as soon as we start the program. |
| Preconditions | Must enter the correct username and password, the database must be connected, accounts must be made available. |
| Postconditions | Success: Seller or accountant have logged into the system, the main menu of the program must be displayed.  Fail: Seller or accountant haven’t logged into the system |
| Main Course | 1. Input username and password 2. Verify login 3. Show main menu |
| Alternate Courses | AC 2.1: User enters wrong username, password   1. Display message error 2. Show the login screen again 3. Re-enter request   AC2.2: Username and password exists   1. Show successful login message 2. Show main menu |
| Exceptions | EX 1.1: Input Username/Password   1. Display: “Invalid username or password, please re-enter!” and ask to re-enter until correct 2. Request to re-enter username and password. EX 2.1: Disconnect to database 3. Step 1: Display “Error!!! Can’t connect to database” 4. Request to recheck internet connection |

* 1. Search laptops description

|  |  |
| --- | --- |
| Name | Search laptops |
| ID | UC\_002 |
| Description | Used to search for laptops |
| Actor | Seller |
| Organizational Beneﬁts | You can search laptops in the system |
| Trigger | Login to the program by the seller’s account and you can use this feature. |
| Preconditions | Must have at least one laptop previously created. |
| Postconditions | Success: Show list of valid laptops.  Fail: No valid laptop |
| Main Course | 1. Input laptop name/category/manufactory or laptop id you want to ﬁnd 2. Show list of laptop found |
| Alternate Courses | AC3.1: Data doesn't exist in the database.  Display “Laptop not found!” |
| Exceptions | EX1: Disconnect to database   1. Request to recheck internet connection. 2. Request program restart. |

* 1. View laptop details description

|  |  |
| --- | --- |
| Name | View laptop details |
| ID | UC\_003 |
| Description | Used to view laptop details |
| Actor | Seller |
| Organizational Beneﬁts | You can view all laptop information |
| Trigger | Login to the program by the seller's account and you can use this feature. |
| Preconditions | Must have at least one laptop previously created. |
| Postconditions | Success: Show all laptop information.  Fail: No laptop found. |
| Main Course | 1. Input laptop id you want to ﬁnd 2. If laptop id valid, show all laptop information |
| Alternate Courses | AC2.1: No data exists in the database.   1. Display msg: “Laptop not found!” 2. Requires pressing any key to go back |
| Exceptions | EX1: Disconnect to database   1. Request to recheck internet connection. 2. Request program restart. |

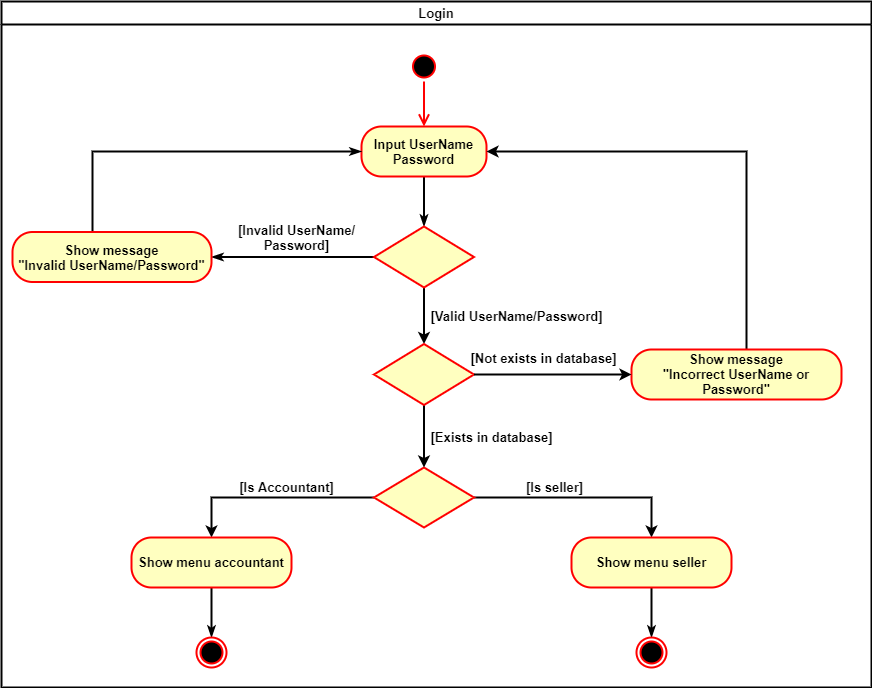
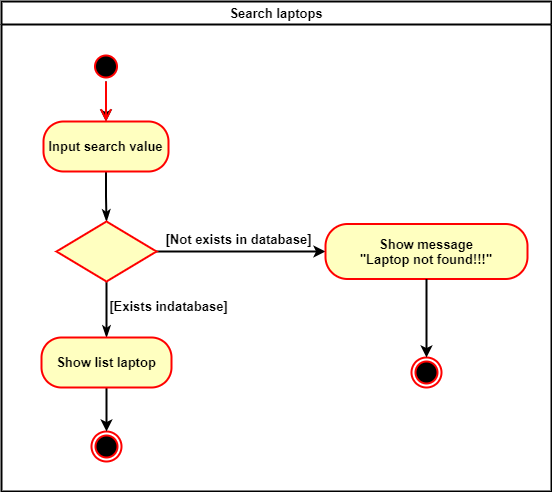
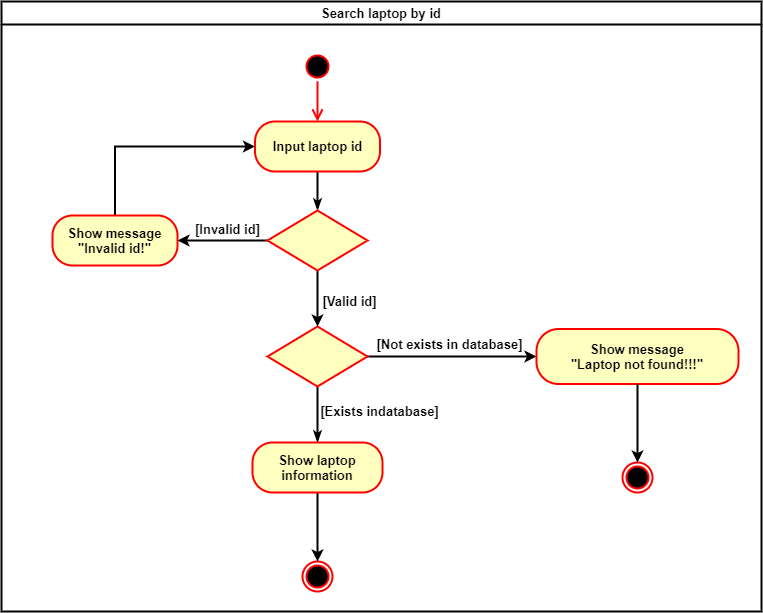
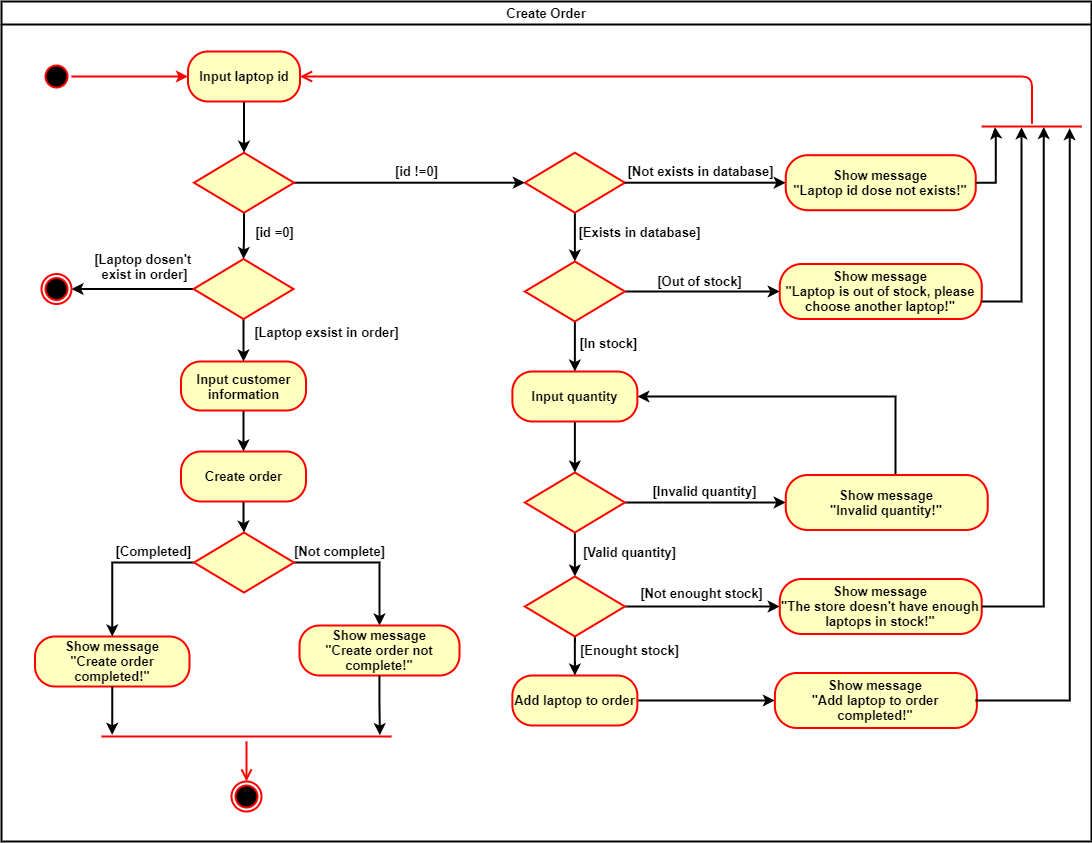
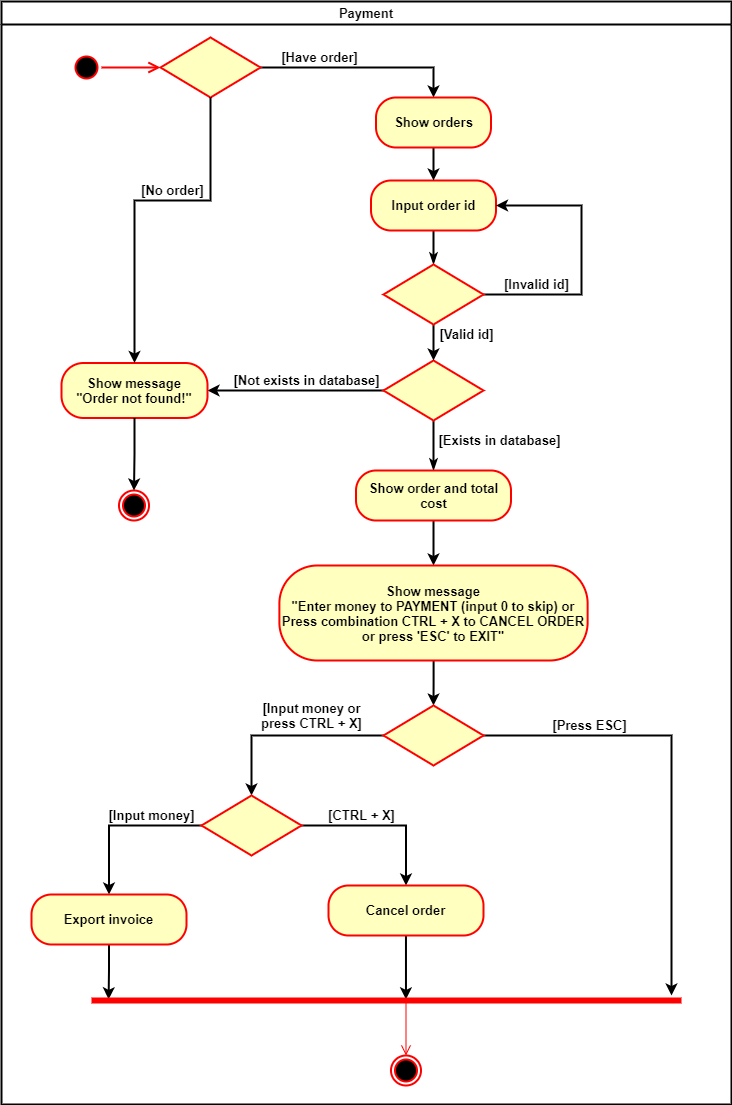
* 1. Create order description

|  |  |
| --- | --- |
| Name | Create order |
| ID | UC\_004 |
| Description | Used to create order |
| Actor | Seller |
| Organizational Beneﬁts | You can create one or more orders according to customer's request |
| Trigger | Login to the program by the seller's account and you can use this feature. |
| Preconditions | Must have at least one laptop previously created. |
| Postconditions | Success: Create order complete.  Fail: Create order not complete. |
| Main Course | 1. Input laptop id and quantity you want to add to order 2. Valid data 3. Input customer information 4. Create order complete. |
| Alternate Courses | AC 2.1: Invalid laptop id or quantity.   1. Display msg error 2. Request re-entry   AC 3.1: Customer information already exists.   1. You just need to enter the customer's phone number 2. The rest of the information will be entered automatically   AC 3.2: Invalid customer’s phone number   1. Display msg error 2. Request re-entry |
| Exceptions | EX1: Disconnect to database   1. Request to recheck internet connection. 2. Request program restart. |

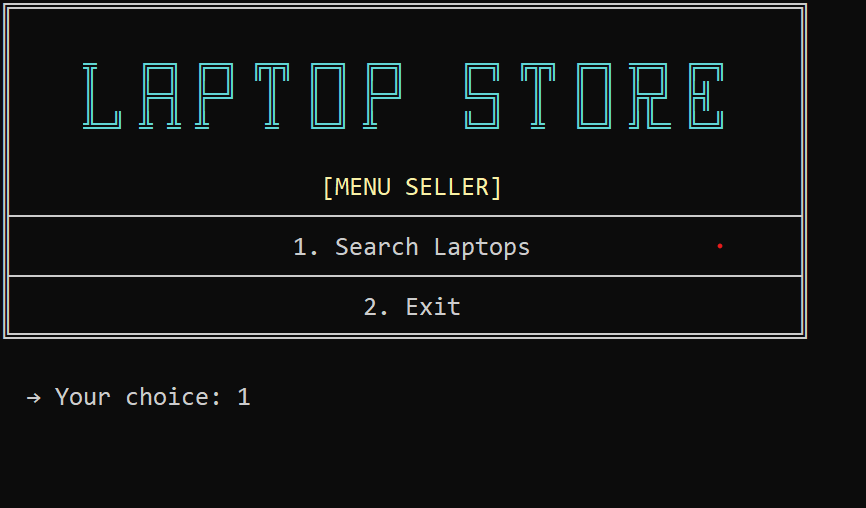
* 1. Payment description

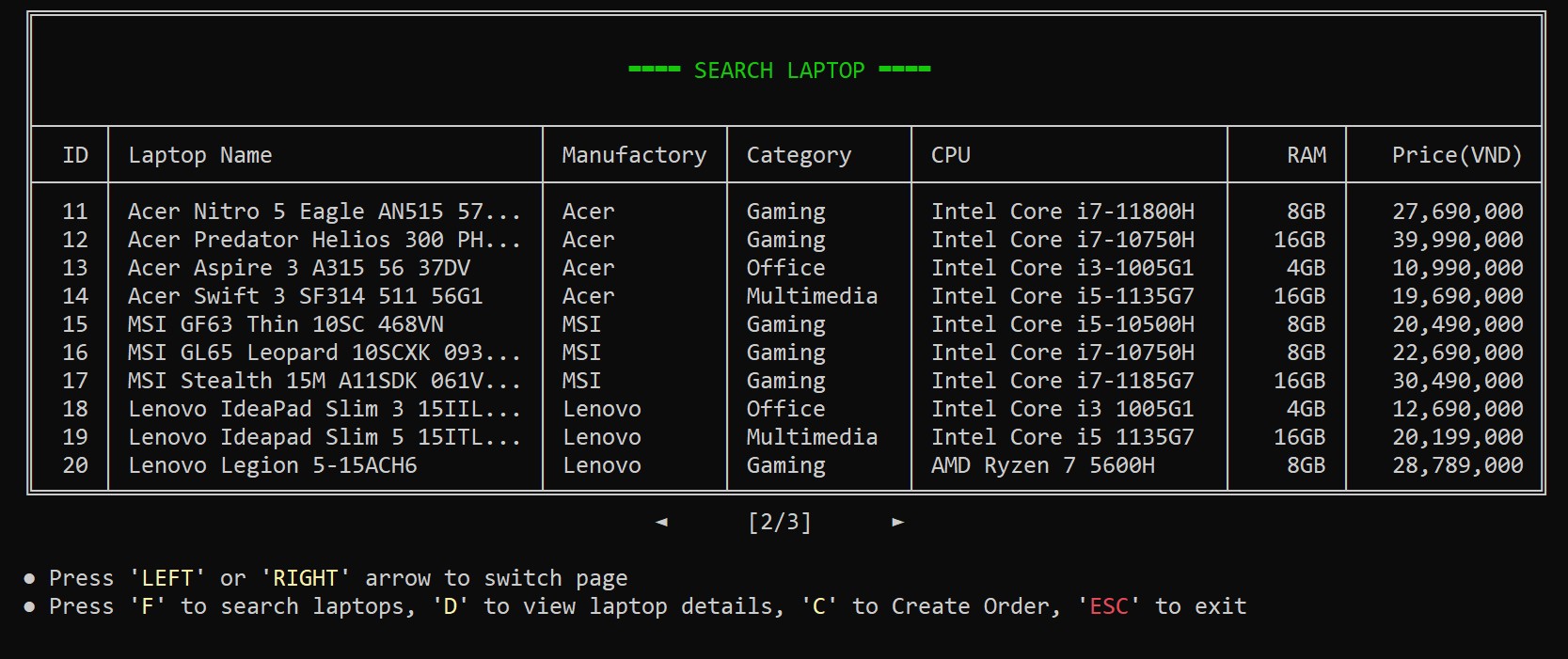
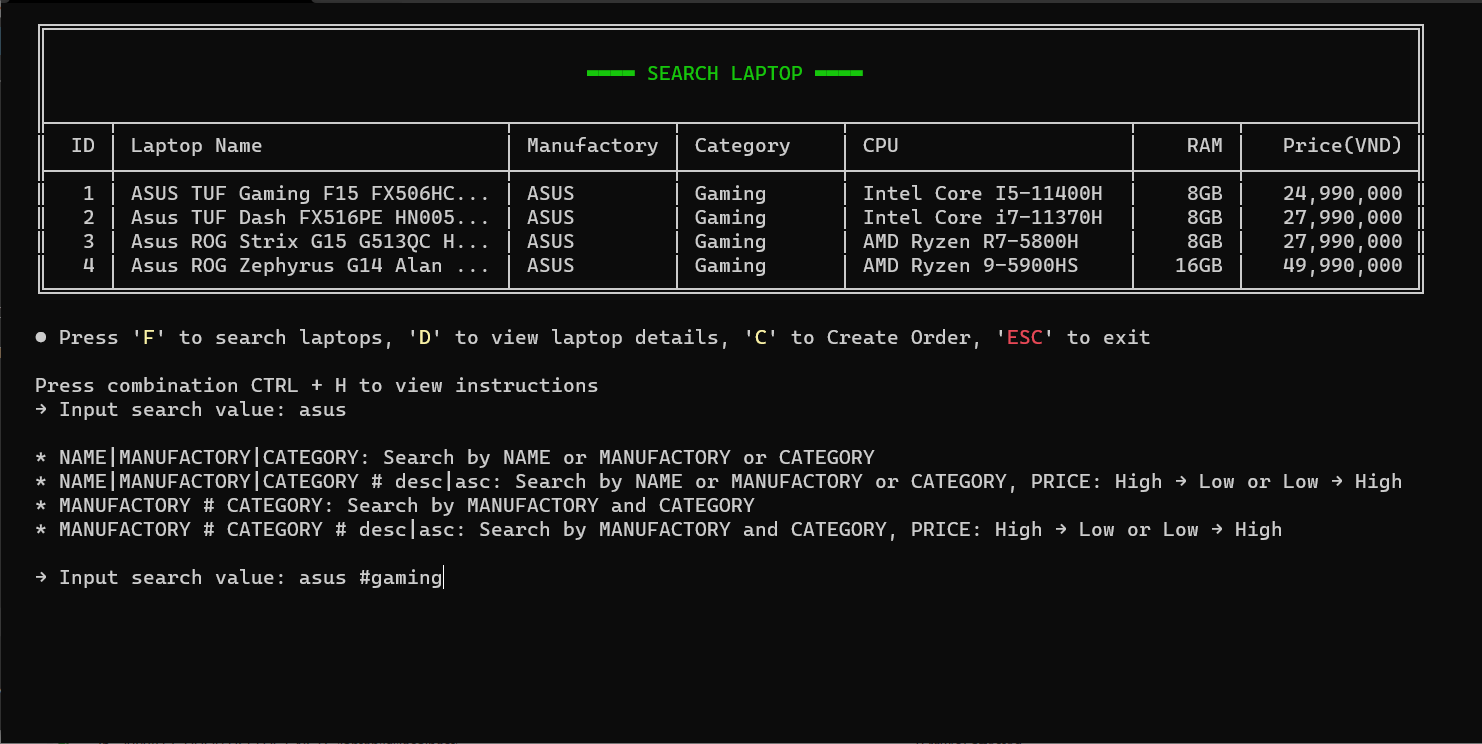
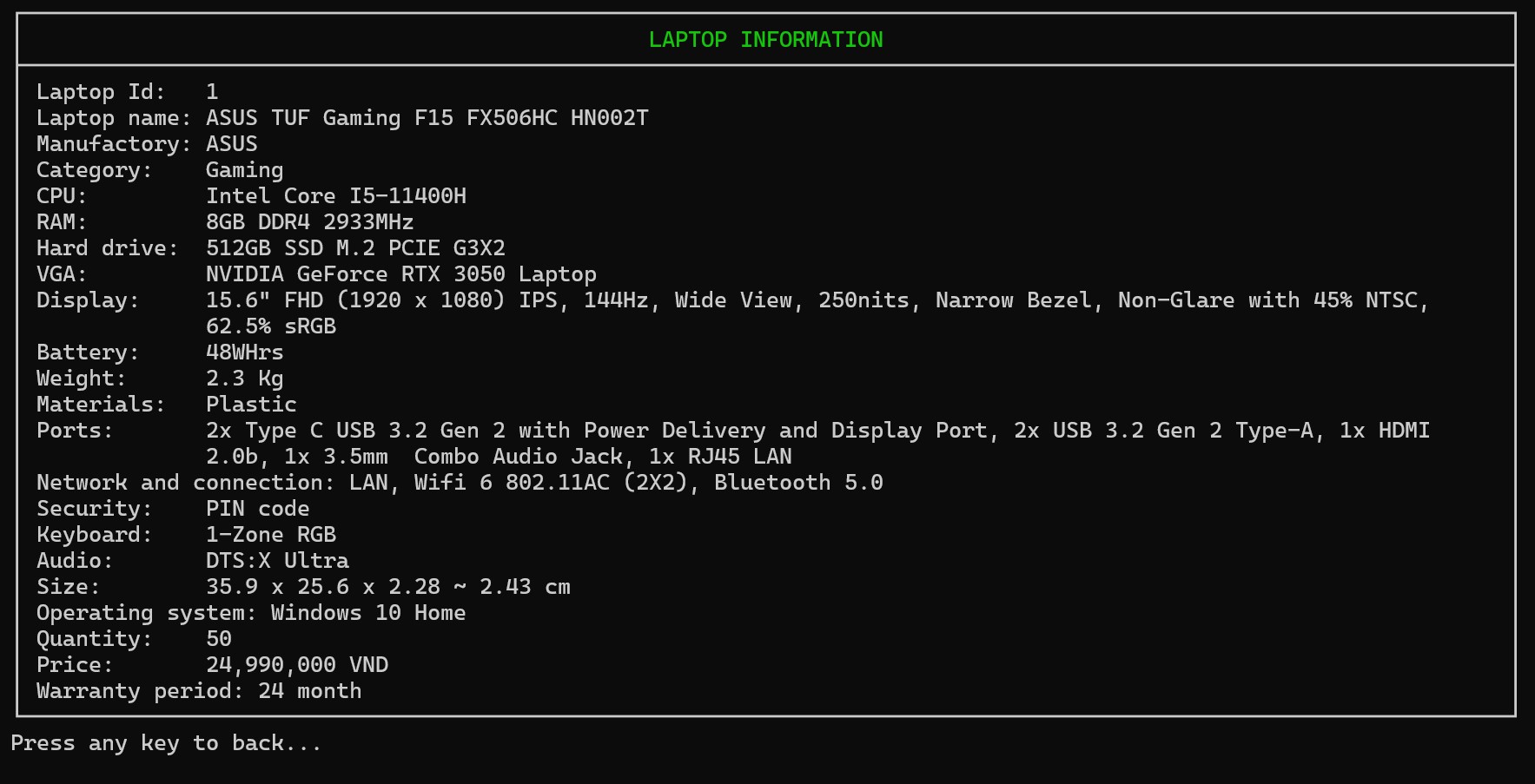
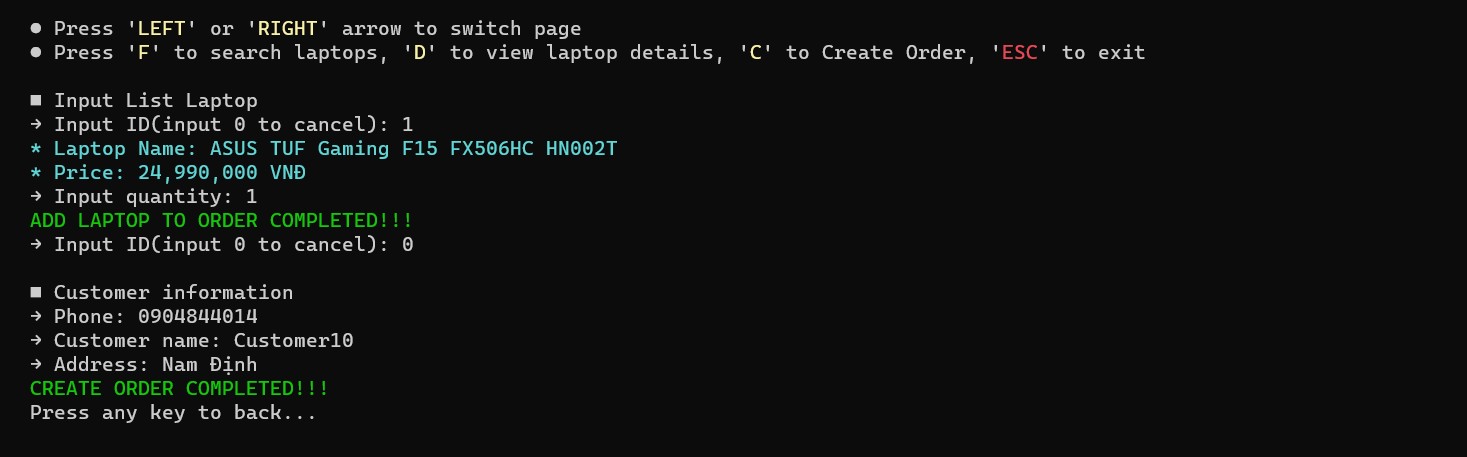
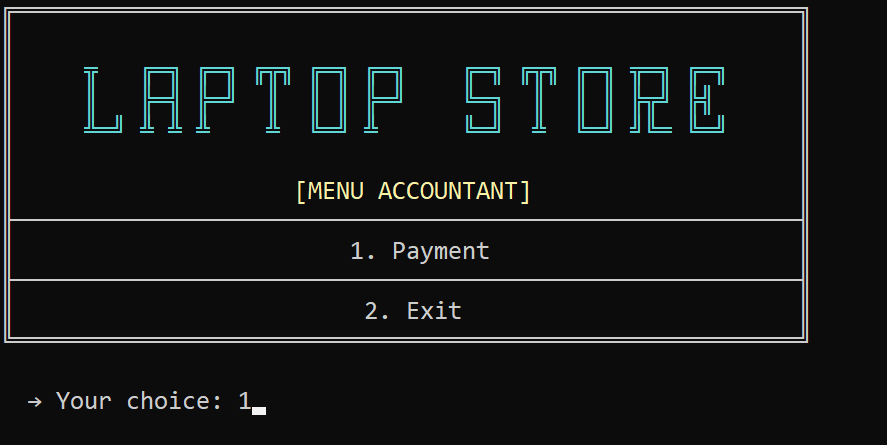
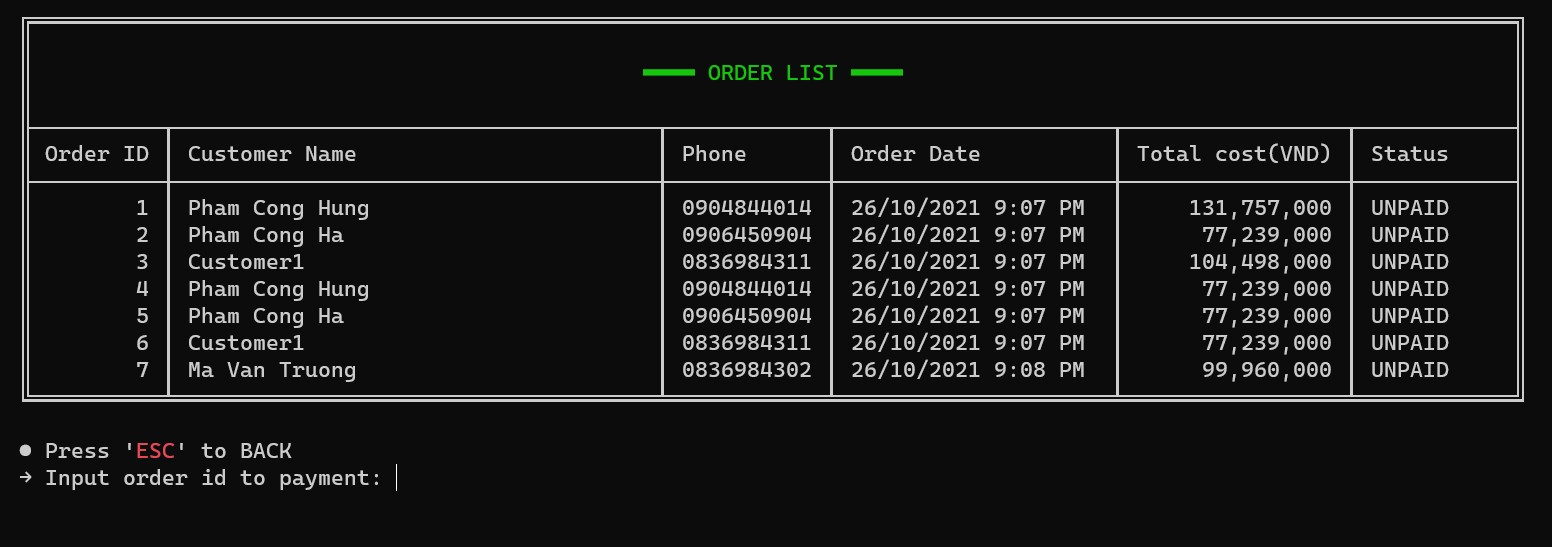
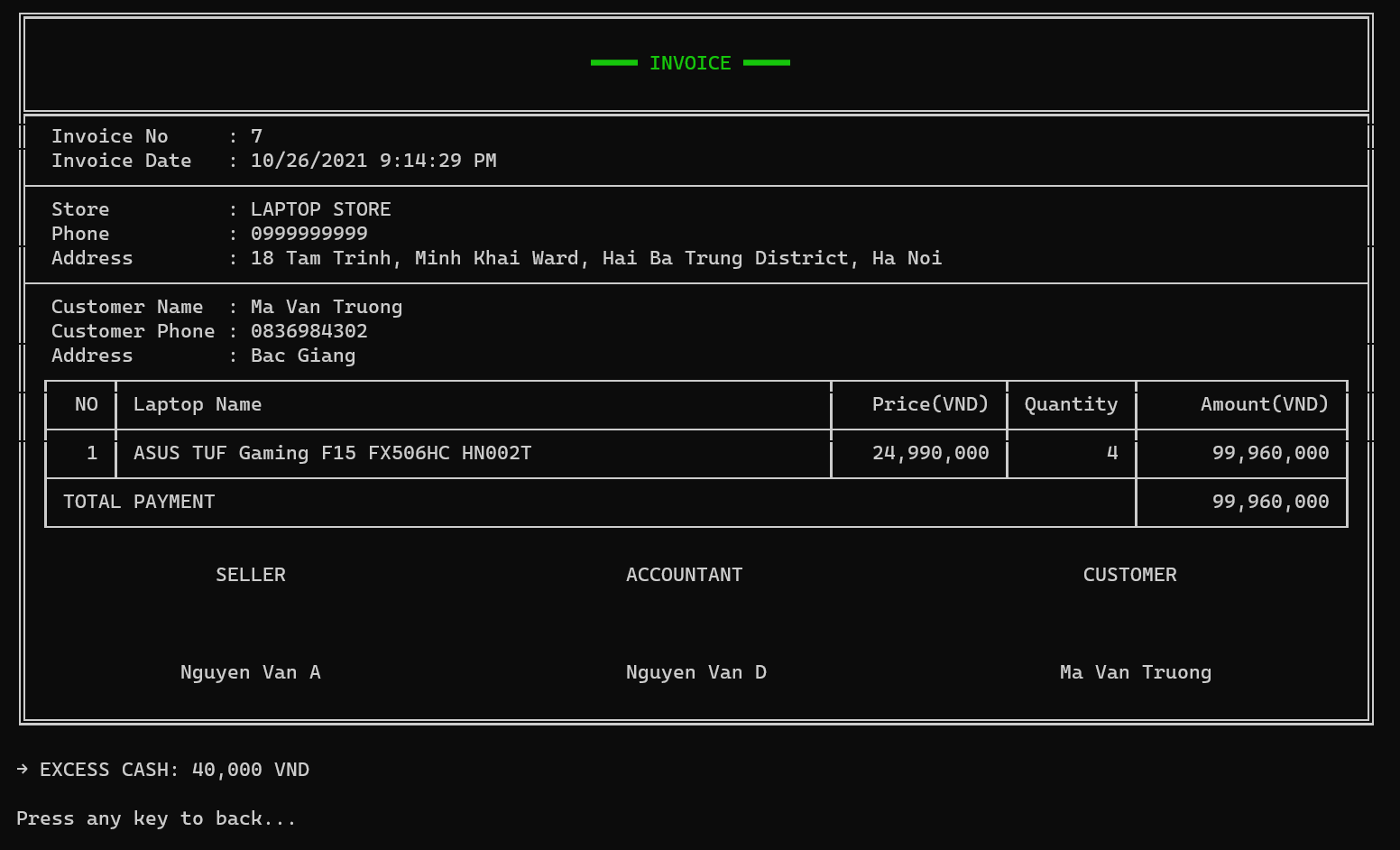
|  |  |
| --- | --- |
| Name | Payment |
| ID | UC\_005 |
| Description | Used to conﬁrm payment or cancel payment |
| Actor | Accountant |
| Organizational Beneﬁts | Allow accountants to conﬁrm paid orders or cancel orders |
| Trigger | Login to the program by the accountant's account and you can use this feature. |
| Preconditions | Must have at least one order in unpaid status. |
| Postconditions | Success: Conﬁrm payment/Cancel order complete or payment pause.  Fail: Conﬁrm payment or cancel order not complete. |
| Main Course | 1. Input order id you want 2. Show order information 3. Choose to payment, cancel order or exit 4. Request complete |
| Alternate Courses | AC 3.1: Conﬁrm payment   1. Enter the amount given by the customer (input 0 to skip) 2. Export invoice   AC 3.2: Cancel order   1. Press combination CTRL+X to cancel order 2. Display msg “Cancel order complete!’ 3. Requires pressing any movie to go back to list order   AC 3.3: Exit   1. Press ESCAPE to exit 2. Payment pause 3. Requires pressing any movie to go back to list order   AC 4.1: If payment or cancel order not complete   1. Display msg error 2. Requires pressing any key to go back to list order |
| Exceptions | EX1: Disconnect to database  3. Request to recheck internet connection. |

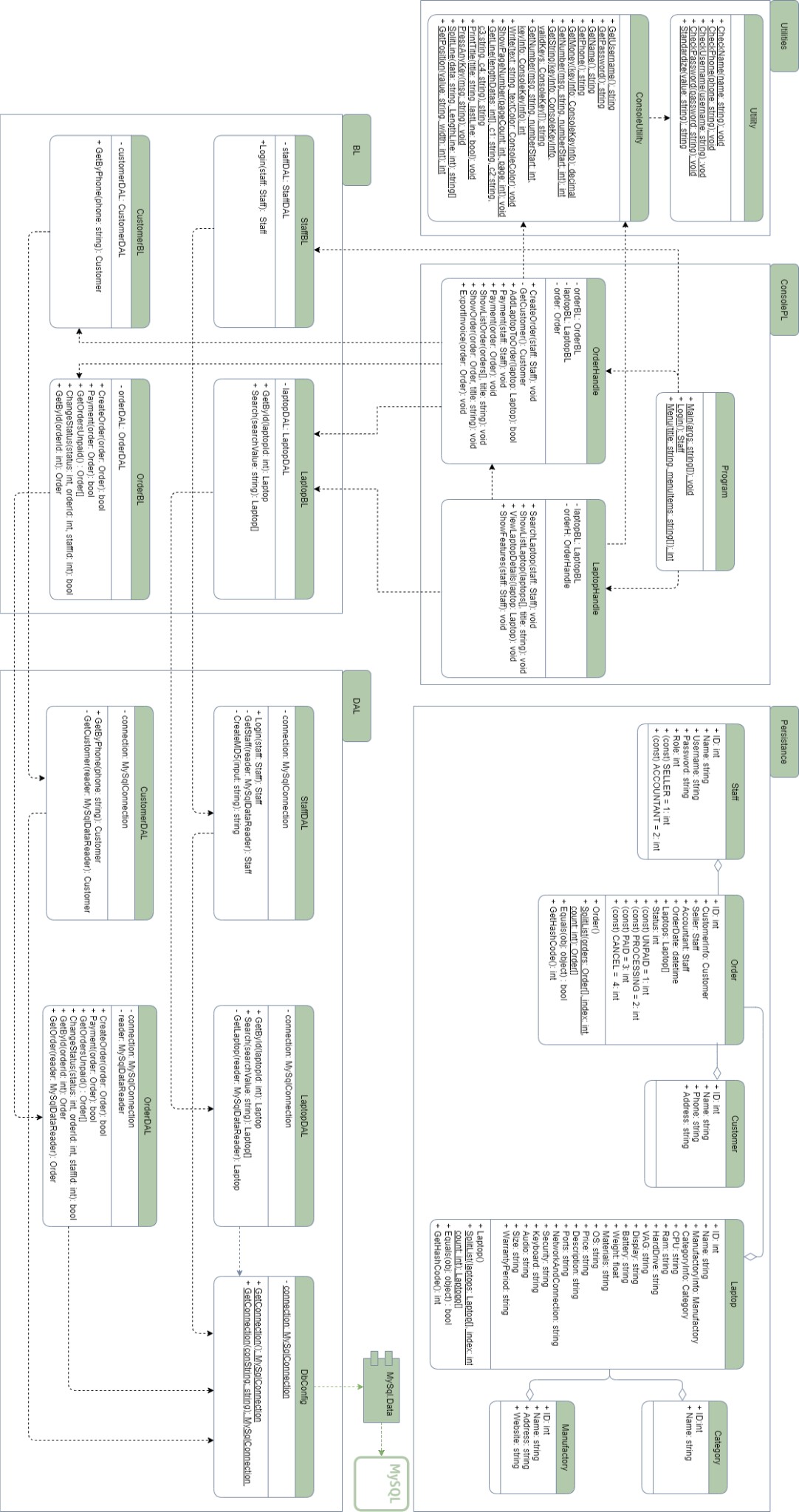
|  |  |
| --- | --- |
|  | 4. Request program restart. |

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

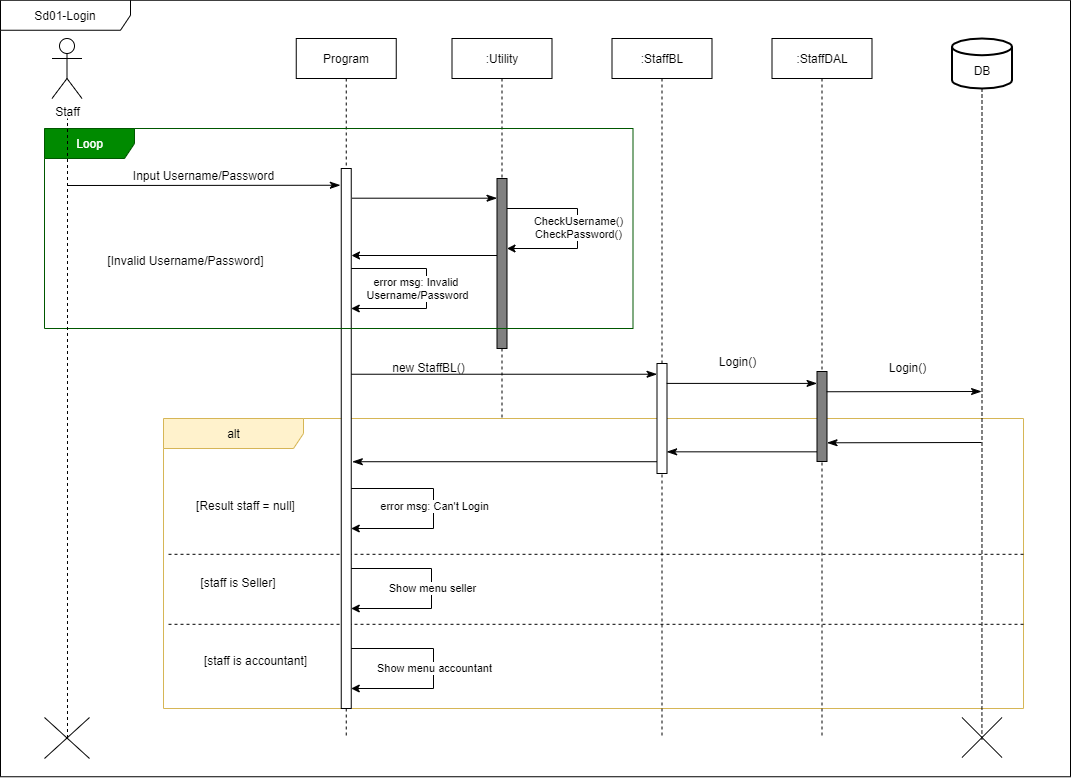
1. Design Details
   1. UI Design
   2. Login
   3. Menu seller



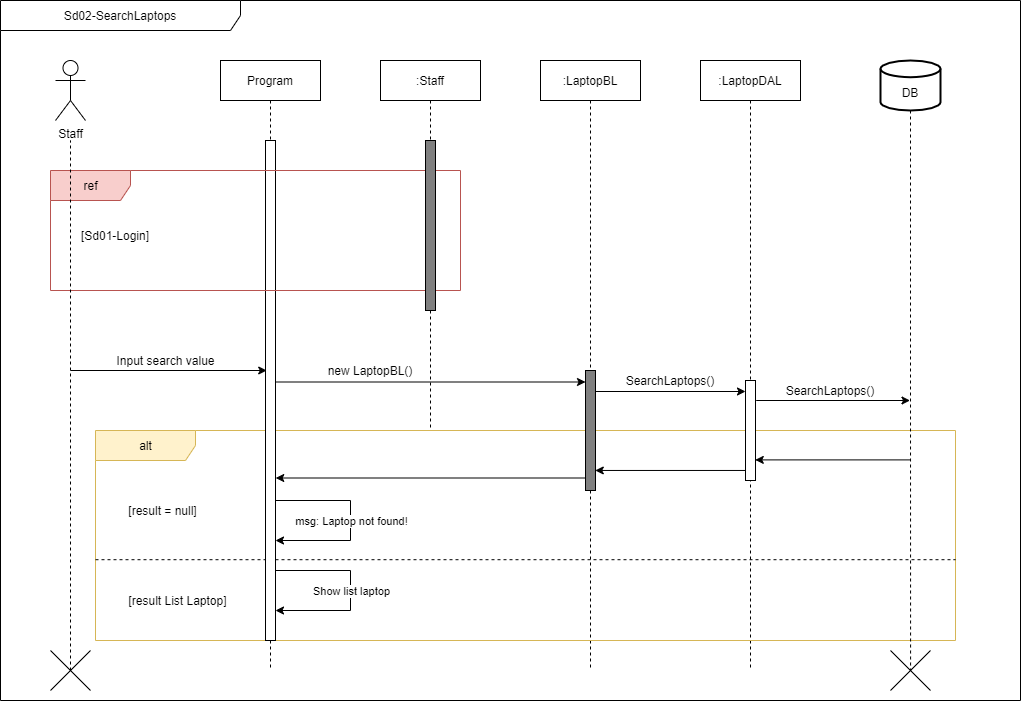
* 1. List of laptop
  2. Search laptop
  3. Show laptop information
  4. Create order
  5. Menu Accountant
  6. Show order list
  7. Payment
  8. Invoice
  9. Code Design (Class Diagram)



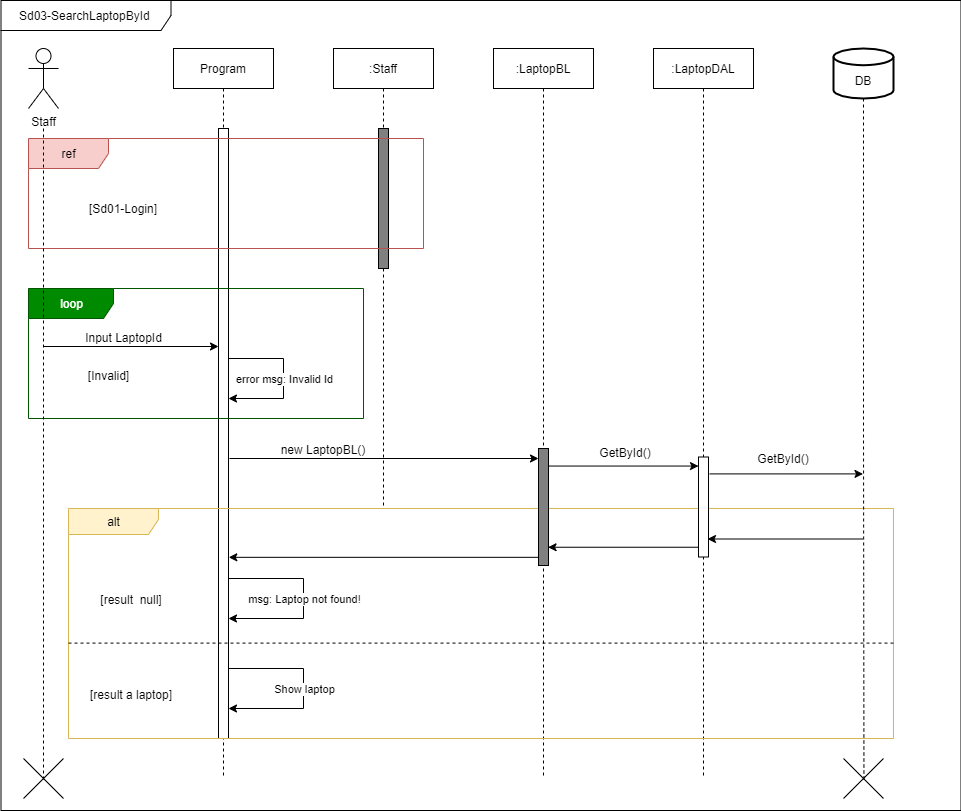
* 1. Sequence Diagram
  2. Login

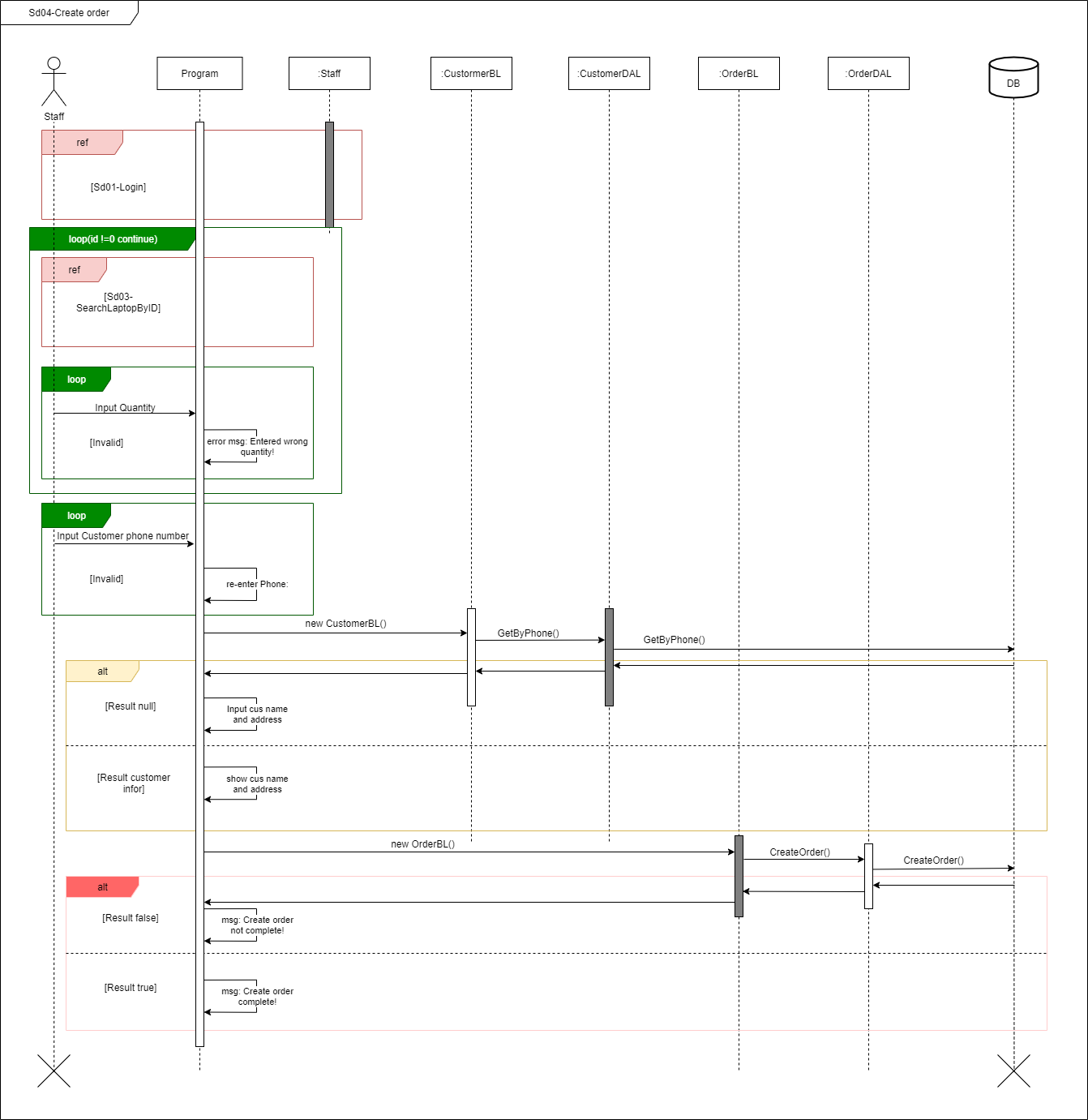
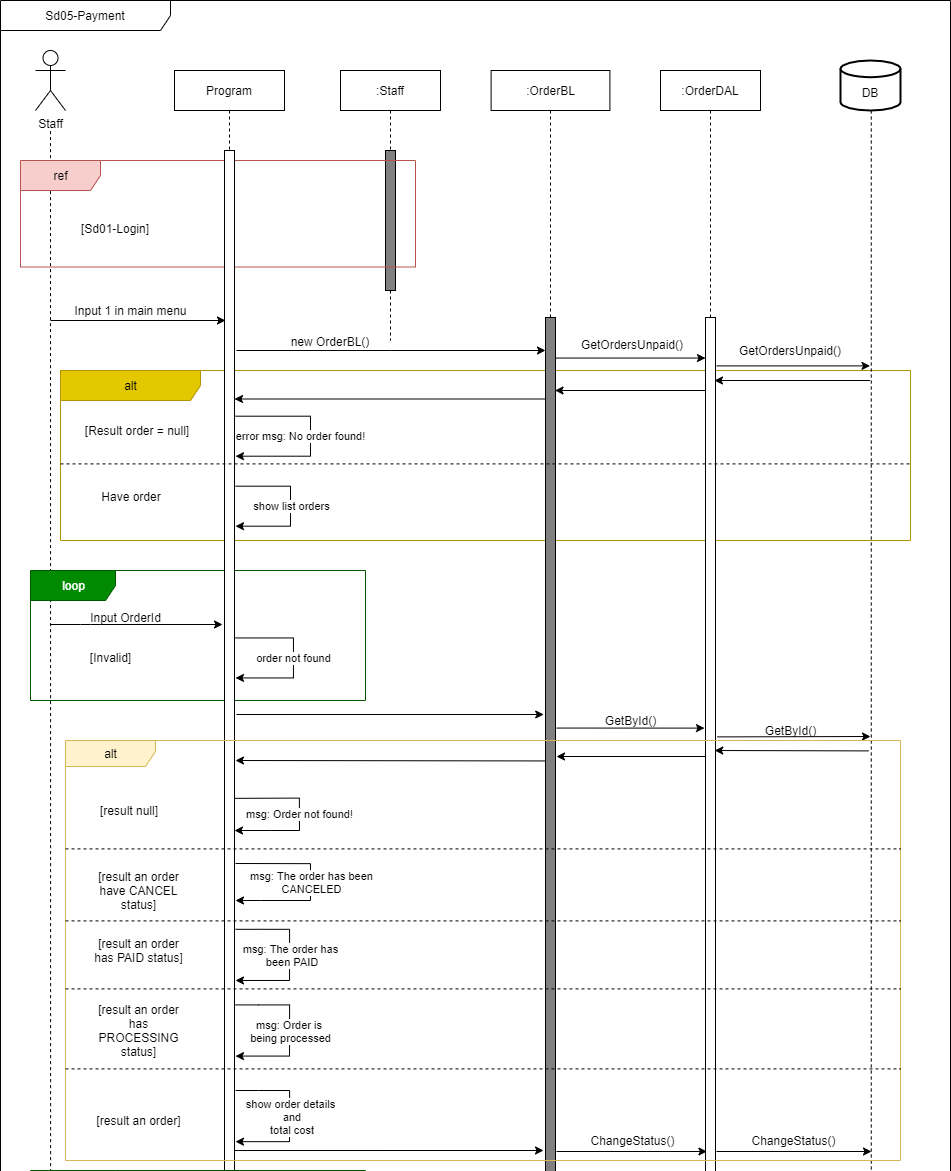


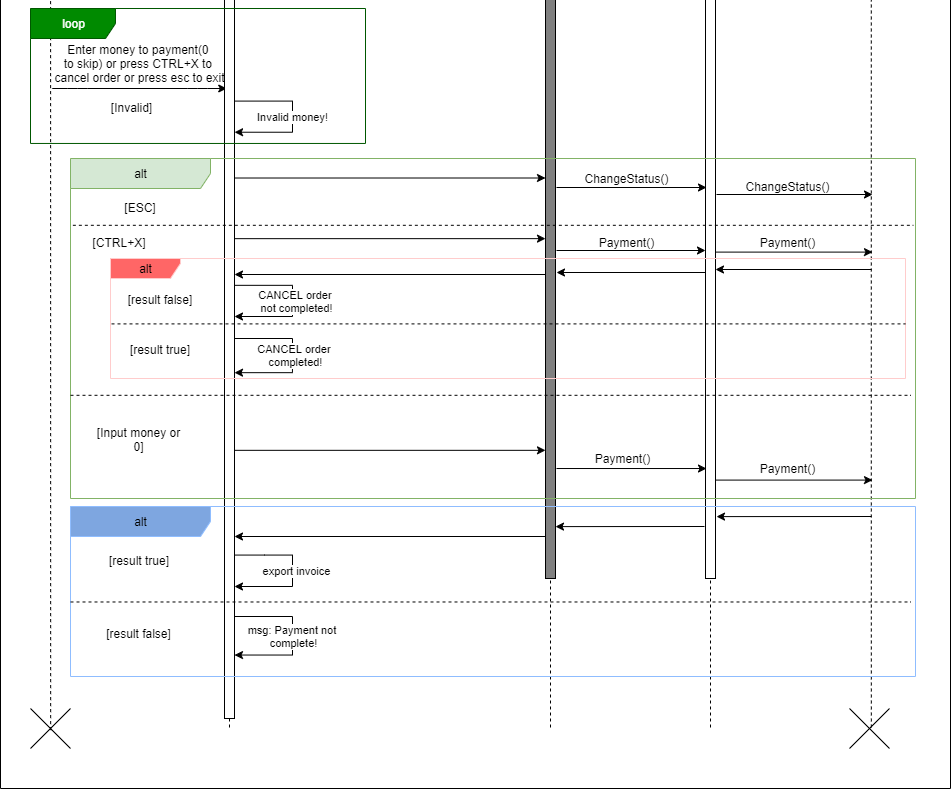
* 1. Search laptops



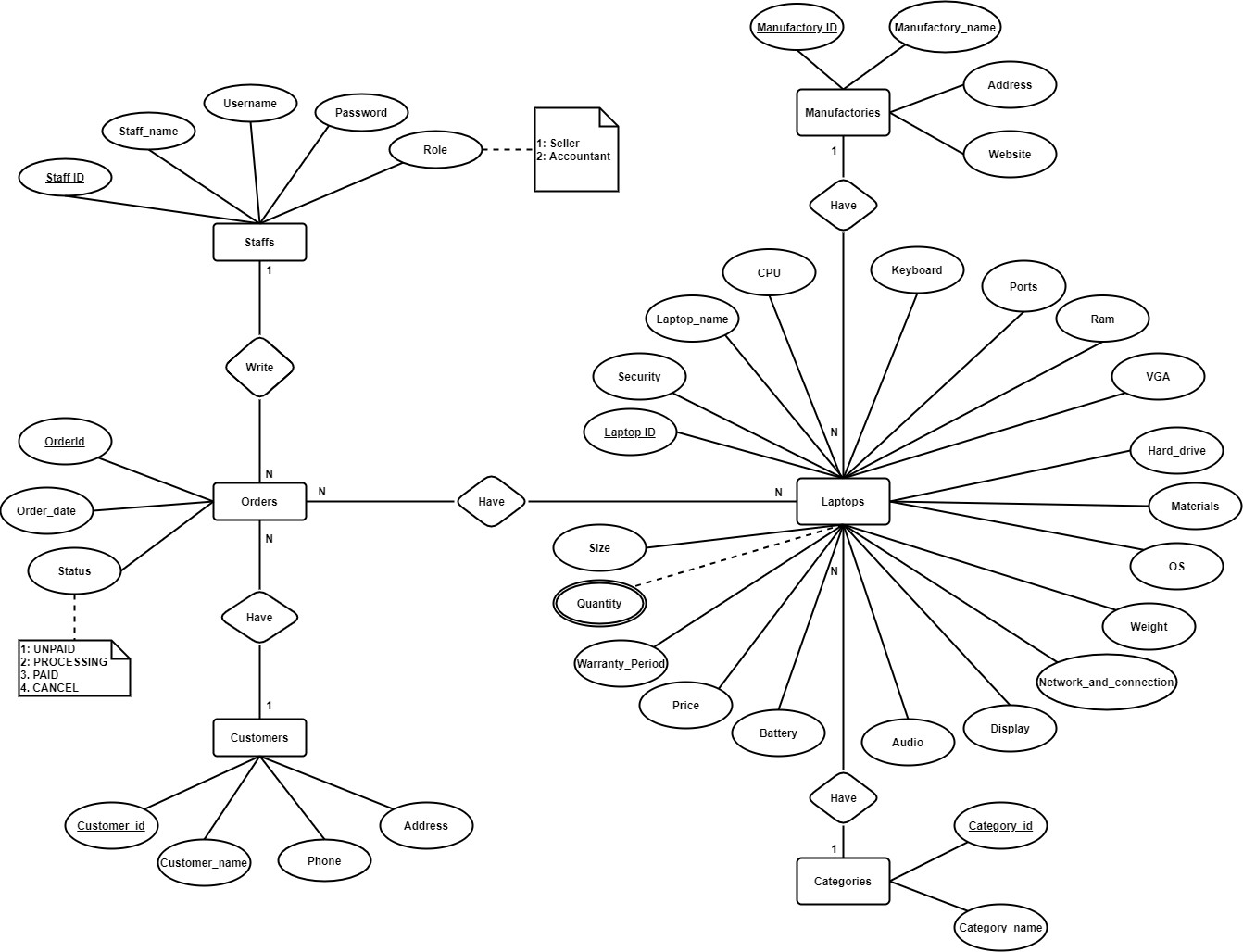
* 1. Search laptop by id



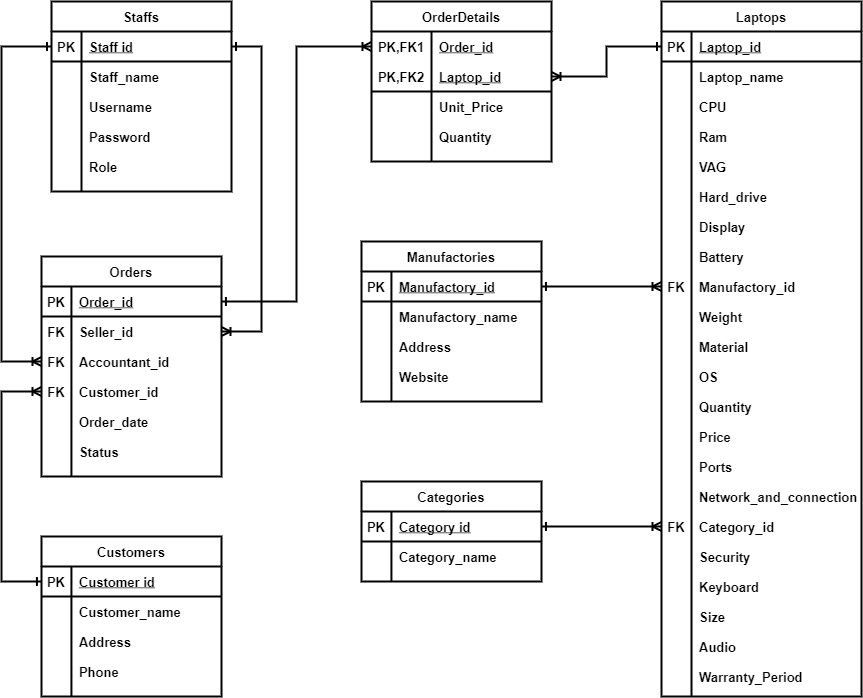
* 1. Create order
  2. Payment



* 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 |  |

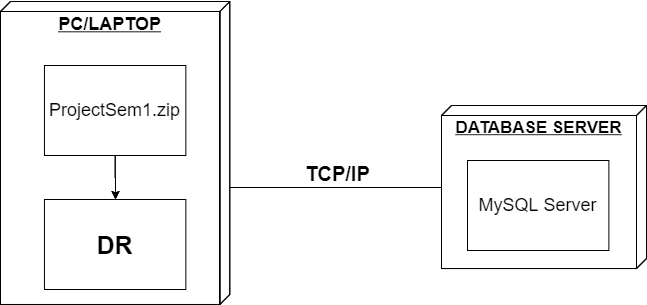
* 1. Payment test 2

|  |  |
| --- | --- |
| 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