

Requirement & Design Specification

**Laptop Shop**

**Version: 1.0**

– Hanoi, August 2024 –

# Record of Changes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Version** | **Date** | **A\* M, D** | **In charge** | **Change Description** |
| V1.0 | 15/05 | A | longnd | Create new SDS |
| V1.1 | 17/05 | A | longnd | Class diagram and specification, sequence diagram and database queries for user login function, home function |
| V1.2 | 17/05 | A | thangbq | Class diagram and specification, sequence diagram and database queries for blog list, blog details, product details function |
| V1.3 | 08/06 | A | thangbq | Class diagram and specification, sequence diagram and database queries for sales order list function |
| V1.4 | 08/06 | A | longnd | Class diagram and specification, sequence diagram and database queries for user authorization |
| V1.5 | 15/06 | A | thangbq | Class diagram and specification, sequence diagram and database queries for sales orderdetails function |
| V1.6 | 15/06 | A | longnd | Class diagram and specification, sequence diagram and database queries for cart details, cart information |
| V1.7 | 02/07 | A | longnd | Class diagram and specification, sequence diagram and database queries for product list |
| V1.8 | 02/07 | A | thangbq | Class diagram and specification, sequence diagram and database queries for order completion |

\*A - Added M - Modified D - Deleted

Contents

[**Record of Changes**](#_heading=h.gjdgxs) **2**

[**I. Overview**](#_heading=h.30j0zll) **7**

[1. User Requirements](#_heading=h.1fob9te) 7

[1.1 Actors](#_heading=h.3znysh7) 7

[1.2 Use Cases](#_heading=h.2et92p0) 8

[a. Diagram(s)](#_heading=) 8

[b. Descriptions](#_heading=) 13

[2. Overall Functionalities](#_heading=h.tyjcwt) 14

[2.1 Screens Flow](#_heading=h.3dy6vkm) 14

[2.2 Screen Descriptions](#_heading=h.1t3h5sf) 14

[2.3 Screen Authorization](#_heading=h.4d34og8) 15

[2.4 Non-UI Functions](#_heading=h.2s8eyo1) 15

[3. System High Level Design](#_heading=h.17dp8vu) 16

[3.1 Database Design](#_heading=h.3rdcrjn) 16

[a. Database Schema](#_heading=) 16

[b. Table Descriptions](#_heading=h.lnxbz9) 16

[3.2 Code Packages](#_heading=h.35nkun2) 18

[**II. Requirement Specifications**](#_heading=h.1ksv4uv) **19**

[1. Common Function](#_heading=h.h0dj1hjjc3rg) 19

[1.1 UC-1\_Login System](#_heading=h.pslthq57vzo) 19

[a. Functional Description](#_heading=) 19

[b. Business Rules](#_heading=h.cf9qijjyqn9e) 20

[1.2 UC-2 Home Pages](#_heading=h.8grwjoa5yd8o) 20

[a. Functional Description](#_heading=h.x9mhmnkbinuh) 20

[b. Business Rules](#_heading=h.2x1efgsinfwn) 21

[1.3 UC-3 Product Details](#_heading=h.cvj81xmu6hof) 21

[a. Functional Description](#_heading=h.vetyrt7l1pk) 21

[b. Business Rules](#_heading=h.asc6fehkd2uu) 22

[1.4 UC-4 List Blogs](#_heading=h.xcospclq92xf) 23

[a. Functional Description](#_heading=h.p49on1r3po2z) 23

[b. Business Rules](#_heading=h.r98tjqt8ffva) 23

[1.5 UC-5 Blog Details](#_heading=h.6y3u399vclx3) 24

[a. Functional Description](#_heading=h.b9idoi57qskp) 24

[b. Business Rules](#_heading=h.qervitj60nbl) 25

[1.7 UC-7\_Cart\_Detail](#_heading=h.aid8zpjvrovg) 25

[a. Functional Description](#_heading=h.jiu25ouv6r9o) 25

[b. Business Rules](#_heading=h.xw0gputnjlwb) 26

[1.8 UC-8\_Cart\_Information](#_heading=h.bkpuf4qu7kdk) 26

[a. Functional Description](#_heading=h.swqhxxiuwz9b) 26

[b. Business Rules](#_heading=h.bj9tbxrd0ajg) 27

[1.9 UC-10 Cart Completion](#_heading=h.rcvcvzid780i) 28

[a. Functional Description](#_heading=h.ik9bb3v20898) 28

[b. Business Rules](#_heading=h.87nx7elgduuk) 29

[1.10 UC-11 My Order](#_heading=h.lfywawktdns3) 29

[a. Functional Description](#_heading=h.5fvodvrlv05v) 29

[b. Business Rules](#_heading=h.2hi488ghha2t) 30

[1.11 UC-11 Order\_Information](#_heading=h.anxtcnhtlilh) 30

[a. Functional Description](#_heading=h.jpd0eyuz912t) 30

[b. Business Rules](#_heading=h.cpaviqst5z5b) 31

[2. Sales Functions](#_heading=h.js46ymo16h22) 31

[2.1 UC-9\_Order\_List](#_heading=h.ys00tploneev) 31

[a. Functional Description](#_heading=h.imum7qmg0ccc) 31

[b. Business Rules](#_heading=h.6kcsqh3988l0) 32

[2.2 UC-10: Order Dashboard](#_heading=h.kdtuht7rjvk6) 33

[a. Functional Description](#_heading=h.sxpzj2tls2hi) 33

[b. Business Rules](#_heading=h.xj4hxsrumiba) 34

[2.3 UC-12 Order Information](#_heading=h.vmieo2ywasgt) 34

[a. Functional Description](#_heading=h.kcolrvlwxl63) 34

[b. Business Rules](#_heading=h.m6qpsx29pr4d) 35

[3. Guest Function](#_heading=h.eft2zg7lwdsd) 35

[3.1, UC-6\_Register User Account](#_heading=h.diw1rdpdfie1) 35

[a. Functional Description](#_heading=h.e3qq3ud1nfwd) 35

[b. Business Rules](#_heading=h.8s4oxyiue7t8) 36

[3.2, UC-12\_Feedback](#_heading=h.1tkiti97j1k) 36

[a. Functional Description](#_heading=h.w4lx4a411psn) 36

[b. Business Rules](#_heading=h.rrdcl22ezfxz) 38

[4. Admin Function](#_heading=h.elqj1pkd7pf8) 38

[4.1 UC-9 User Authorization](#_heading=h.ffdwqknly4ju) 38

[a. Functional Description](#_heading=h.c8lofq4i52x3) 38

[b. Business Rules](#_heading=h.nvsd8dtojq73) 39

[**III. Design Specifications**](#_heading=h.1ci93xb) **39**

[1. Common function](#_heading=h.g3b9oxqaf6sb) 39

[a, User Login](#_heading=h.ganhgvgjn3bx) 39

[UI Design](#_heading=h.757qroy4mj3g) 40

[Database Access](#_heading=h.5r0tjsszq4n2) 40

[b, Home page:](#_heading=h.os1bgnfdz6sa) 40

[UI Screen:](#_heading=h.cogfz1rdvb8v) 41

[Database Access](#_heading=h.peuoud232o8m) 41

[c,Product Details](#_heading=h.j0txyg6fmr2m) 42

[UI Screen:](#_heading=h.w83uysckumni) 42

[Database Access](#_heading=h.qsui7wbfm8pz) 43

[d, Blog Lists](#_heading=h.g32unub92pnj) 43

[UI Screen:](#_heading=h.cb3dgsbvy7wv) 43

[Database Access](#_heading=h.rj4jxb3e1d3) 43

[e, Blog Detail](#_heading=h.r88sesrlvmtw) 44

[UI Screen:](#_heading=h.nuzwhm6gsdhq) 44

[Database Access:](#_heading=h.m53sx0ce9dh8) 44

[f, Cart Details](#_heading=h.40i53hix00ov) 45

[UI Screen:](#_heading=h.ufid81ixvqx1) 45

[Database Access:](#_heading=h.ptzjakj9s02u) 45

[h, Cart Information](#_heading=h.ttwy0id3d2v7) 46

[Ui Screen:](#_heading=h.m0wn6c31e8v4) 47

[Database Access:](#_heading=h.rq0s542uj0j0) 47

[i, Cart Completion](#_heading=h.uhc453yn6n8e) 48

[Ui Screen:](#_heading=h.ioc8iqahk142) 48

[Database Access](#_heading=h.divydkrnxo3i) 49

[j. My Order](#_heading=h.vbtse6m0h0mq) 49

[UI Screen](#_heading=h.ettz602g0rqm) 49

[Database Access:](#_heading=h.qb98fbbs9mcj) 51

[SQL Query](#_heading=h.vmzora6bendb) 51

[k, Order Information](#_heading=h.t6ui7dwxrgsw) 51

[UI Screen](#_heading=h.r2g9ax4xe1gt) 52

[Database Access:](#_heading=h.c2527ce8jovq) 52

[SQL Query](#_heading=h.u3m4yij0xoxf) 52

[2, Guest Function](#_heading=h.100aiwb3rl3u) 53

[e, User Register](#_heading=h.mcp5nbf3bnmo) 53

[UI Screen:](#_heading=h.2aqcuvcf2spm) 54

[Database Access:](#_heading=h.ttff7u4qoqwt) 54

[f, Feedback](#_heading=h.jutydstfarop) 55

[UI Screen:](#_heading=h.9k6pr6srw1t3) 56

[Database Access](#_heading=h.5xwtmor3mpjw) 56

[**3, Sales Function**](#_heading=h.c1nt869imh27) **57**

[g, Order Lists](#_heading=h.kbqtj6rtw4s2) 57

[UI Screen](#_heading=h.3p516i8avngb) 58

[Database Access](#_heading=h.u5i61xhqpbfi) 58

[Order Details:](#_heading=h.krb0iso4cbvy) 58

[UI Screen](#_heading=h.agdv3cr2qgjz) 59

[Database Access](#_heading=h.w4ttfztyarzi) 59

[h, Sale Dashboard](#_heading=h.7b8lpjj6pi7r) 60

[UI Screen:](#_heading=h.tj3nqppw5ss) 60

[Database Access](#_heading=h.ghwdoihgzgl6) 61

[**4, Admin Function**](#_heading=h.ygpt0lsshk5p) **61**

[User Authorization](#_heading=h.bdfkrj3uorw5) 61

[UI Screen:](#_heading=h.b87cio549a2f) 61

[Database Access](#_heading=h.2yav2bj7cufn) 62

[**IV. Appendix**](#_heading=h.dedf3rtxbiys) **63**

[4.1 External Interfaces](#_heading=h.lpfxmuz2bd3d) 63

[a. User Interfaces](#_heading=h.zeomw6ljwixi) 63

[b. Software interfaces](#_heading=h.2pobpsk1qfwl) 63

[c. Software interfaces](#_heading=h.4jn8vz1lkqxq) 63

[d. Security](#_heading=h.a9bu3ezg77gy) 63

[4.2 Quality Attributes](#_heading=h.x05ofpbvqm16) 63

[4.2.1 Usability](#_heading=h.t37gup40qzti) 63

[4.2.2 Reliability](#_heading=h.26in1rg) 64

[4.2.3 Performance](#_heading=h.osdiqvnjlcgb) 64

[5. Requirement Appendix](#_heading=) 64

[5.1 Business Rules](#_heading=h.4t91z5bbspzs) 64

[5.2 Common Requirements](#_heading=) 64

[5.3 Application Messages List](#_heading=) 65

[5.4 Other Requirements](#_heading=h.mw2qptf3etm7) 65

[Testing:](#_heading=h.ysumrk32wbq6) 65

# I. Overview

## 1. User Requirements

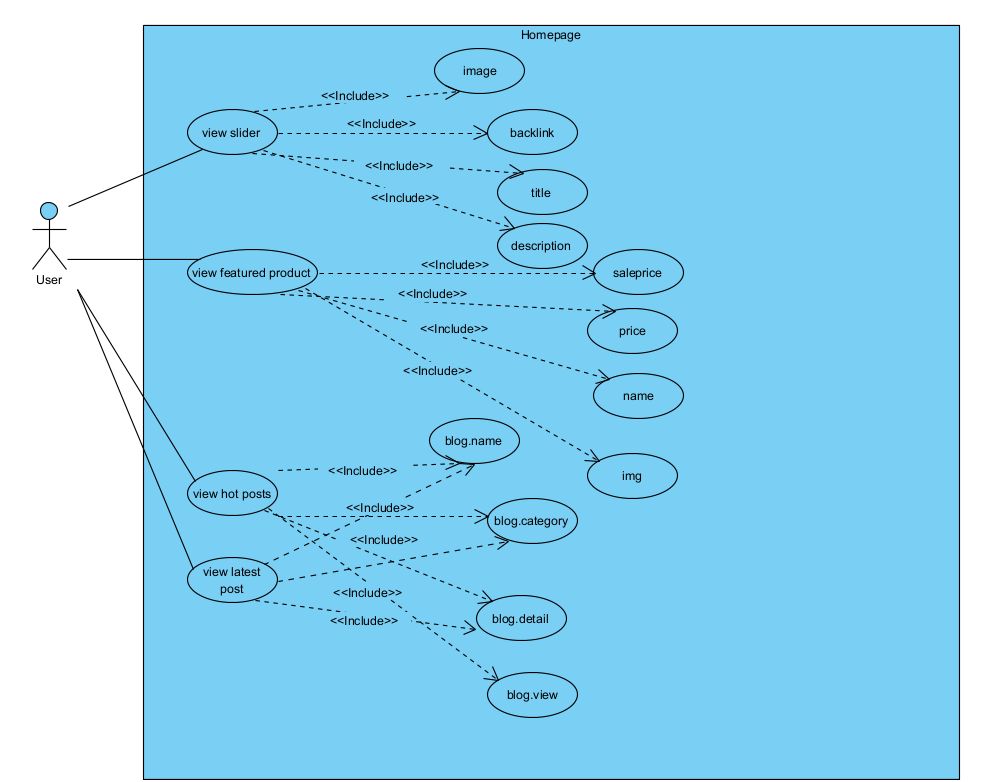
### 1.1 Actors

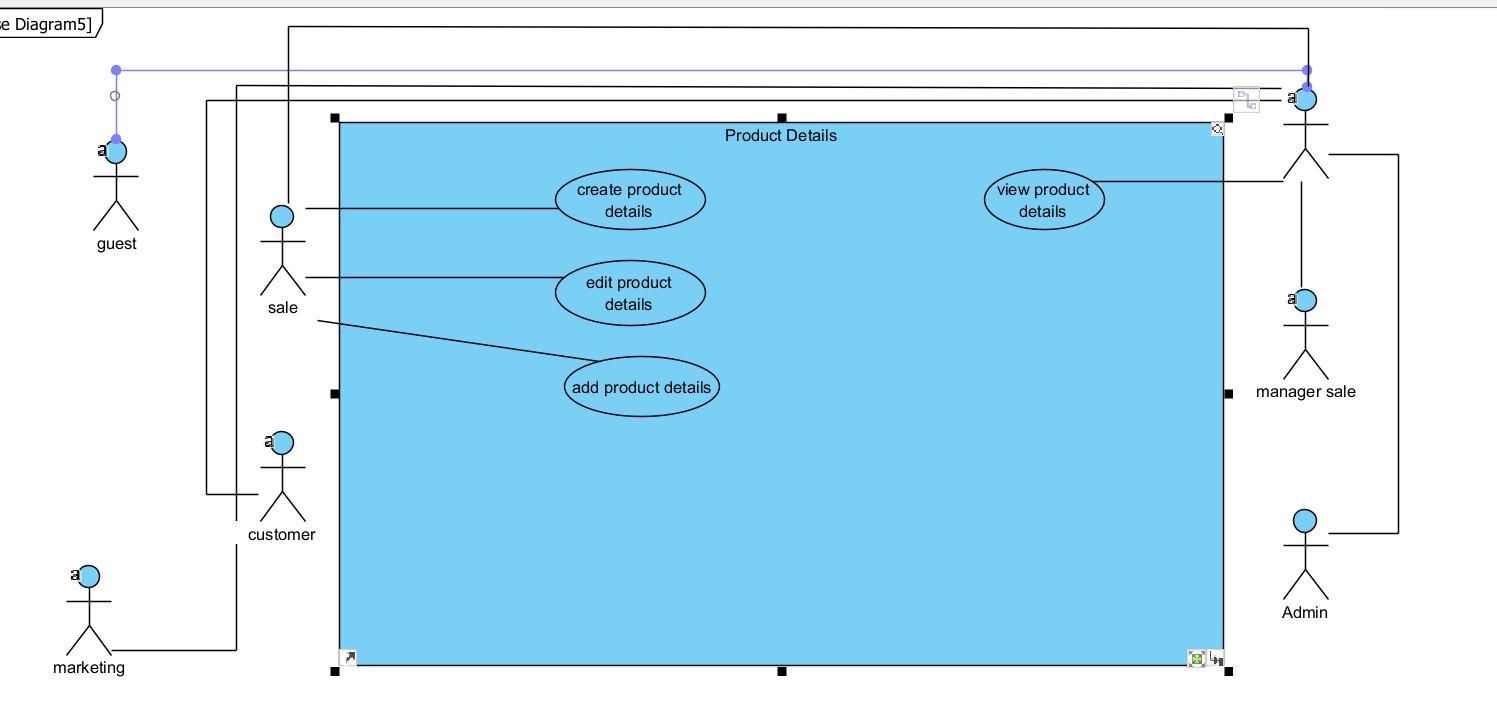
|  |  |  |
| --- | --- | --- |
| **#** | **Actor** | **Description** |
| 1 | Admin | * Is notified when something occurs within the system * Provides information or service to the system |
| 2 | Sale Manager | * Manage all order(s) on the system * Assign a Sale to one specific order * See the work dashboard of all and one specific sales |
| 3 | Marketing | * Manage blog in the system * Manage product in the system * Manage slider in the system * Manage user’s information |
| 4 | Sale | * Manage the order they got assigned * See work dashboard of all and self |
| 5 | Customer | * View the product, blog, slider * Can add the product to cart, add the product to checkout * Can view self’s order state |
| 6 | Guest | * View the product, blog, slider * Can add the product to cart, add the product to checkout |

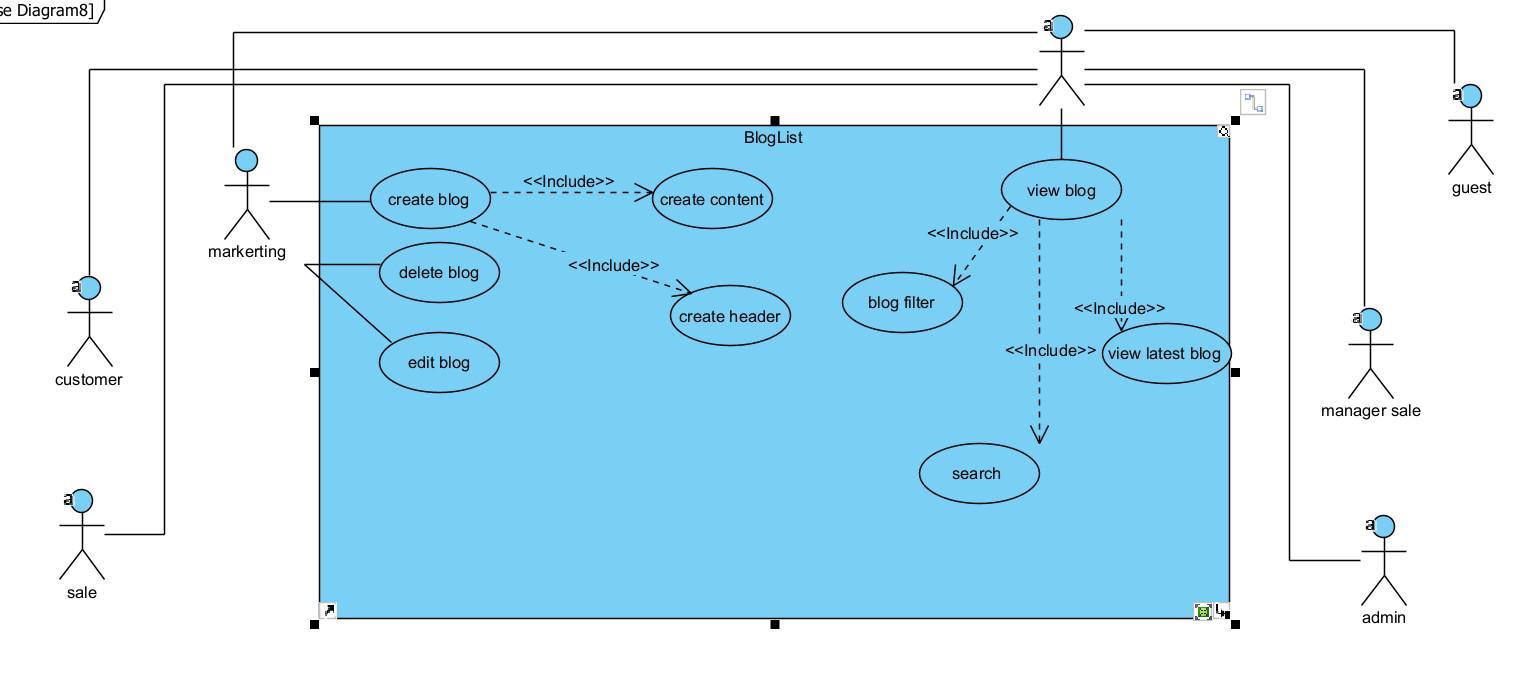
### 1.2 Use Cases

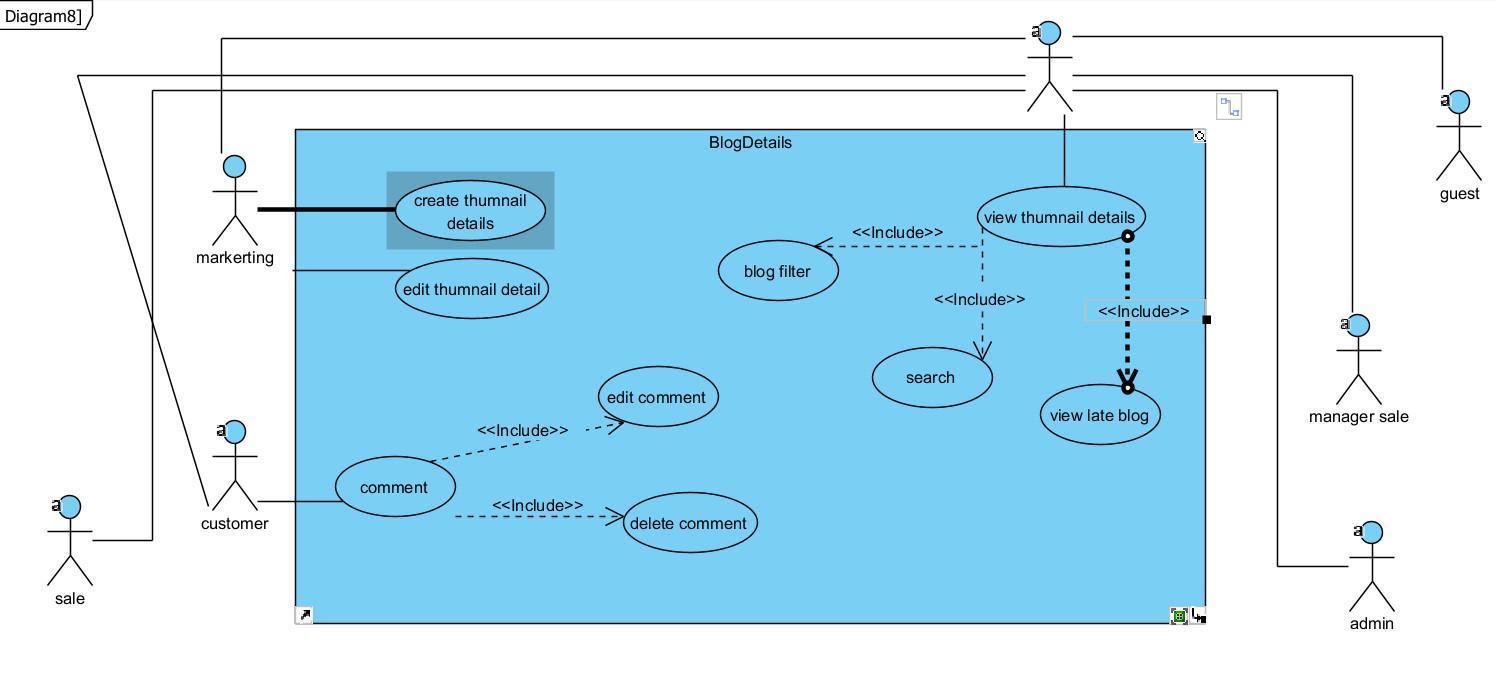
#### a. Diagram(s)

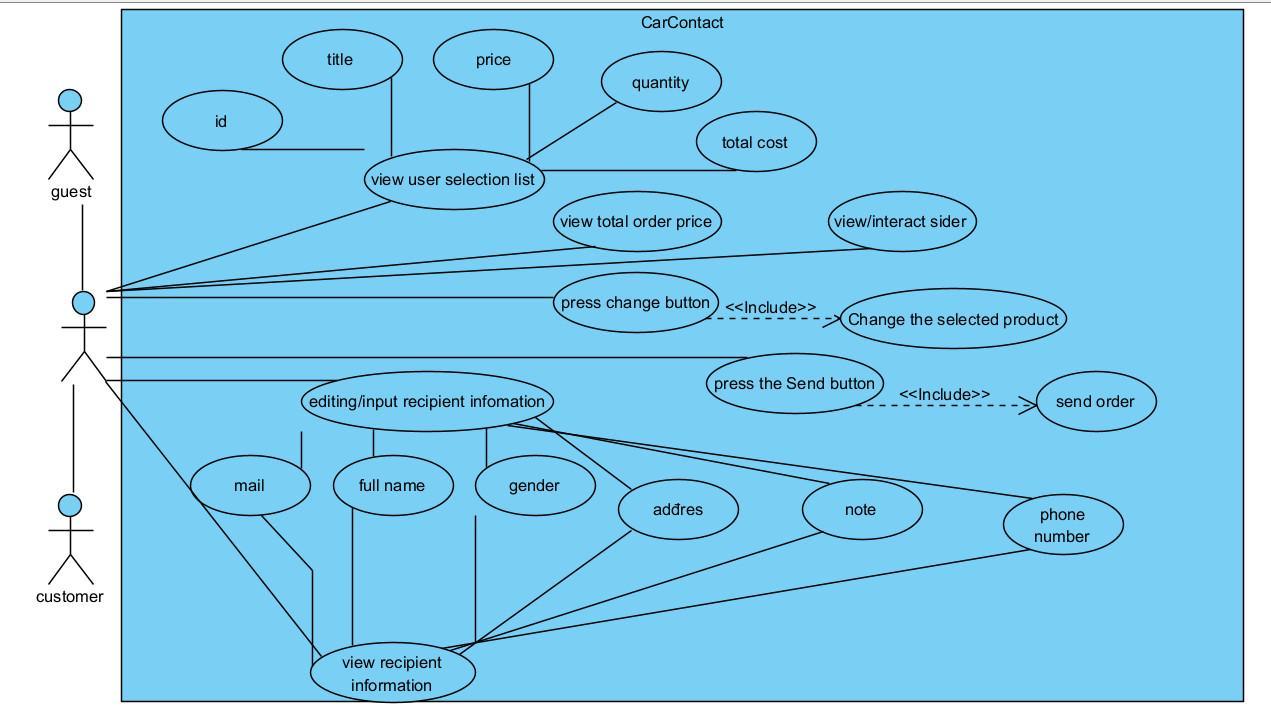
##### 



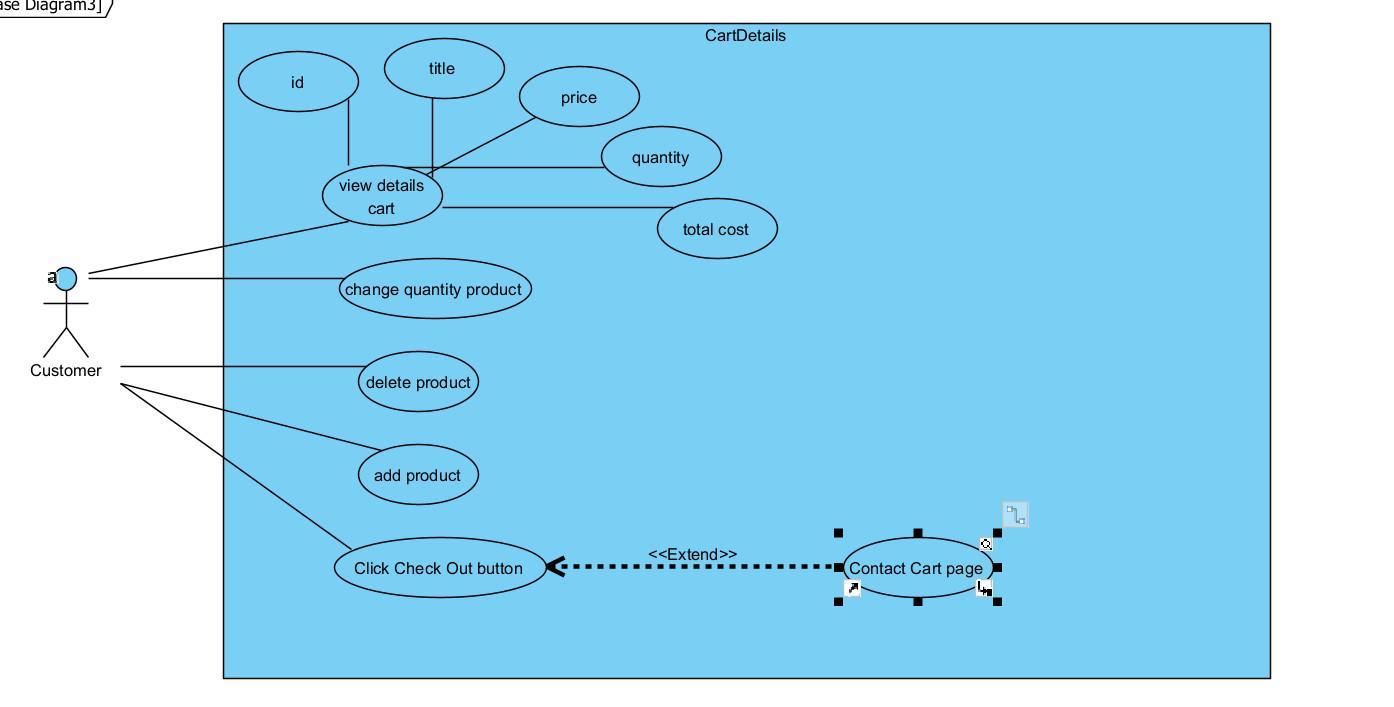


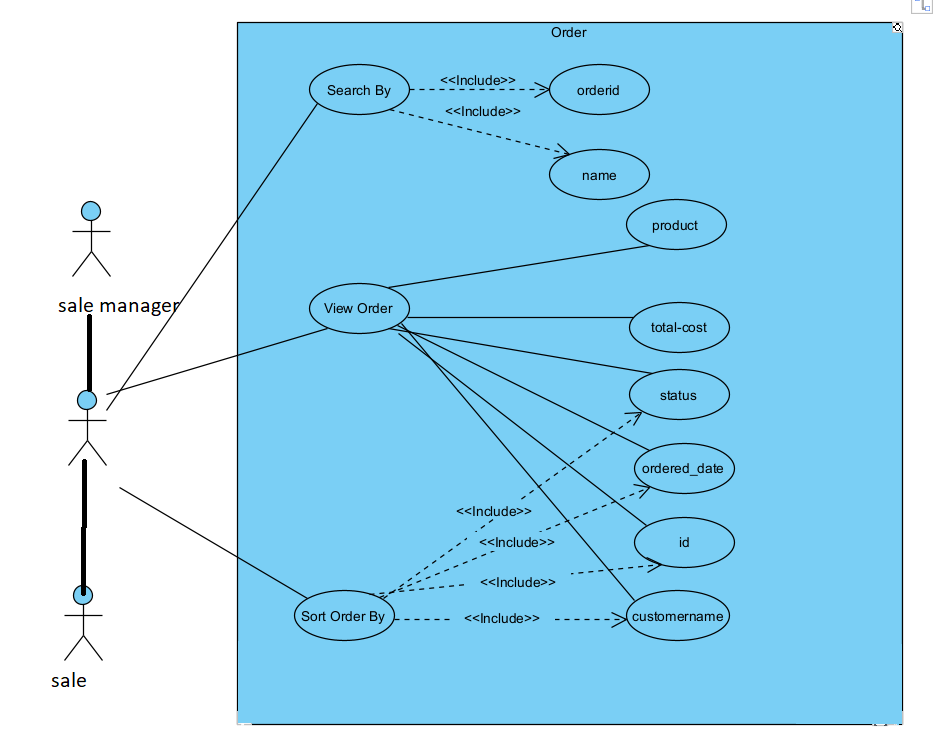


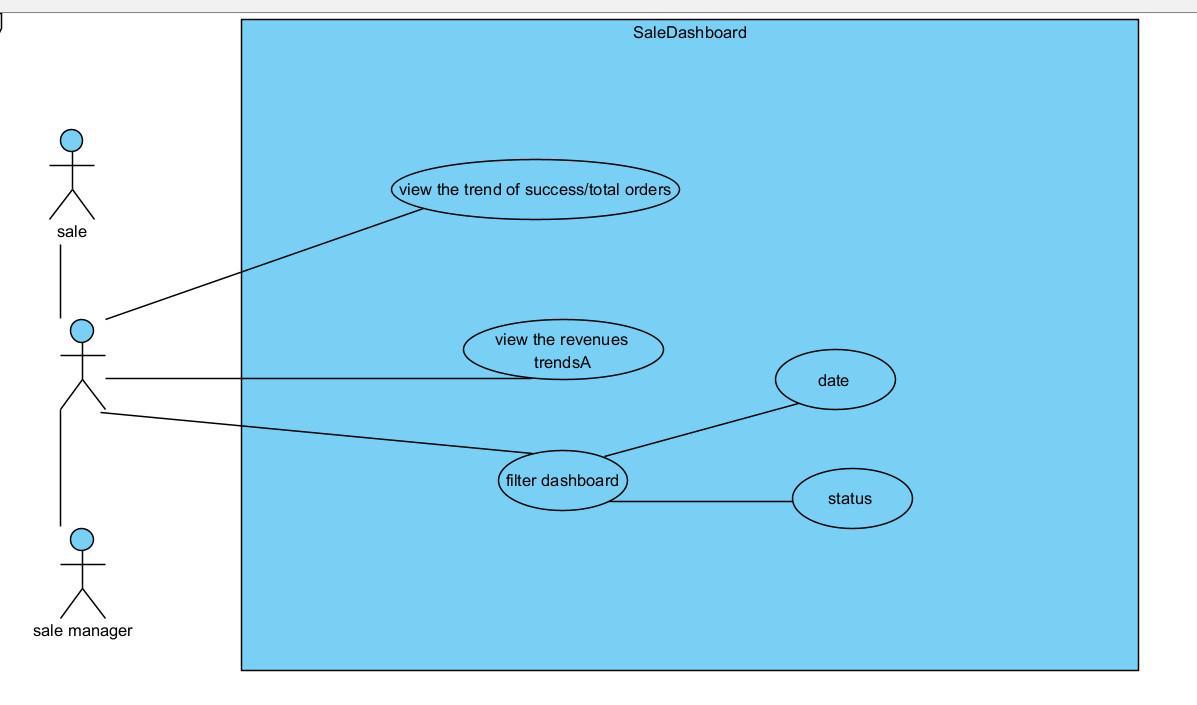


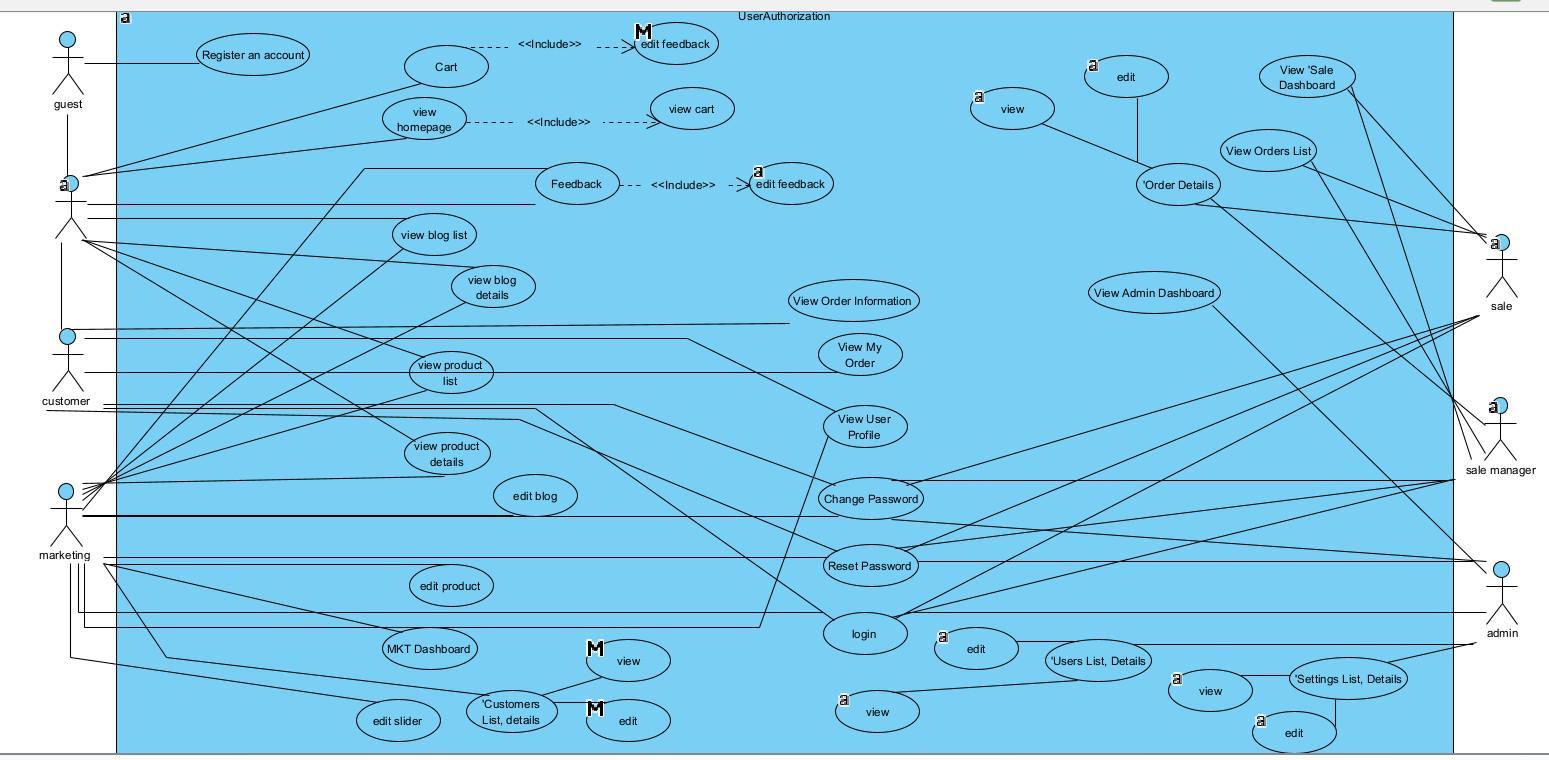


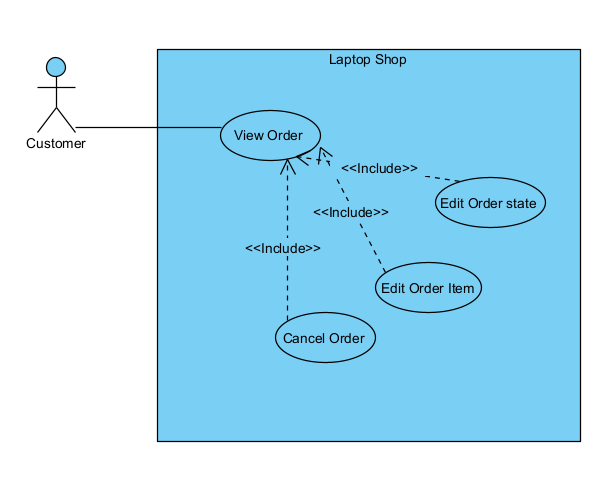
# 

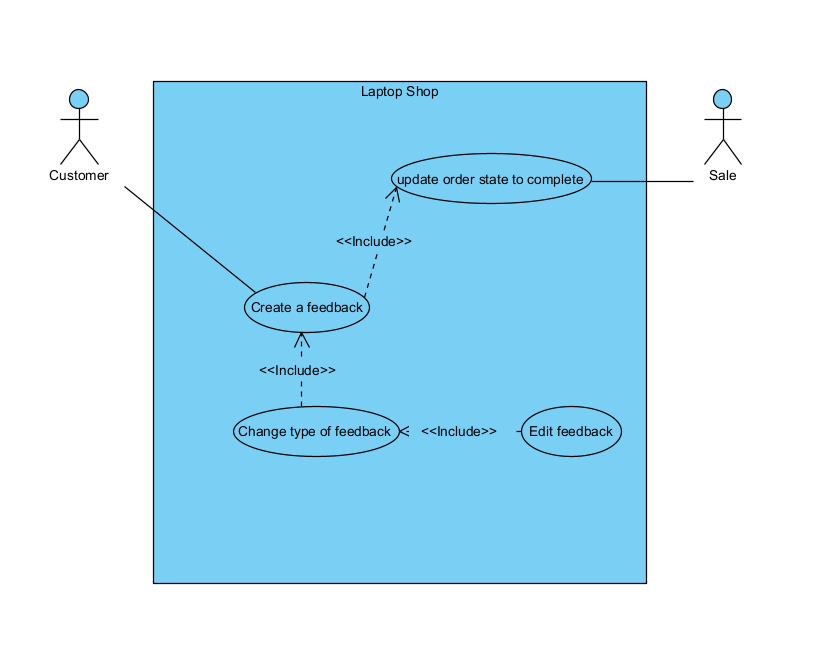












#### b. Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Feature** | **Use Case** | **Use Case Description** |
| 01 | Common | Login | Guest user input information and register with system an account for accessing more functions in system |
| 02 | Common | Home Page | A welcome-user page which allow user to get the basic views of the website |
| 03 | Common | Product Details | Allowing user to see the details of the product they choose |
| 04 | Common | Blog List | Allowing user to see all the blogs on the site |
| 05 | Common | Blog Details | Allowing user to see in-details for a blog |
| 06 | Common | Cart Contact | Allowing user to set up their information in-order to checkout |
| 07 | Guest | User Register | Allowing user to create a new account for further use |
| 08 | Common | Cart Details | Allowing user to see and modify their cart |
| 09 | Sales | Order Management | Manage the current orders |
| 10 | Sales | Dashboards | Allowing Sales to the the trend and revenue |
| 11 | Admin/Common | User Authorization | Allow admin to authorize a screen/prevent user without credential access that screen |
| 12 | Customer | Order Management | Allowing user to be edit their own order |
| 13 | Customer/Guest | Feedback Management | Allowing user to be feedback on their product upon successful |

## 2. Overall Functionalities

### 2.1 Screens Flow

### 2.2 Screen Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
| 01 | Common | Login | Guest user input information and register with system an account for accessing more functions in system |
| 02 | Common | Home Page | A welcome-user page which allow user to get the basic views of the website |
| 03 | Common | Product Details | Allowing user to see the details of the product they choose |
| 04 | Common | Blog List | Allowing user to see all the blogs on the site |
| 05 | Common | Blog Details | Allowing user to see in-details for a blog |
| 06 | Common | Cart Contact | Allowing user to set up their information in-order to checkout |
| 07 | Guest | User Register | Allowing user to create a new account for further use |
| 08 | Common | Cart Details | Allowing user to see and modify their cart |
| 09 | Sales | Order Management | Manage the current orders |
| 10 | Sales | Dashboards | Allowing Sales to the the trend and revenue |
| 11 | Admin/Common | User Authorization | Allow admin to authorize a screen/prevent user without credential access that screen |
| 12 | Common | Cart Completion | Allowing user to add their items to create an order |

### 2.3 Screen Authorization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Screen** | **Guest** | **Customer** | **Sales** | **Sales Manager** | **Marketing** | **Admin** |
| Login |  | X | X | X | X | X |
| Register | X |  |  |  |  |  |
| Product Details | X | X |  |  |  |  |
| Product Lists | X | X |  |  |  |  |
| Blog Details | X | X |  |  |  |  |
| Blog List | X | X |  |  |  |  |
| Cart Details |  |  |  |  |  |  |
| Cart Information |  |  |  |  |  |  |
| Cart Completion |  |  |  |  |  |  |
| Sales Dashboard |  |  | X | X |  |  |
| Check specific sales status |  |  |  | X |  |  |
| Order Lists |  |  | X | X |  |  |
| Order Details |  |  | X | X |  |  |
| Assign to other sales |  |  |  | X |  |  |
| User Authorization |  |  |  |  |  | X |

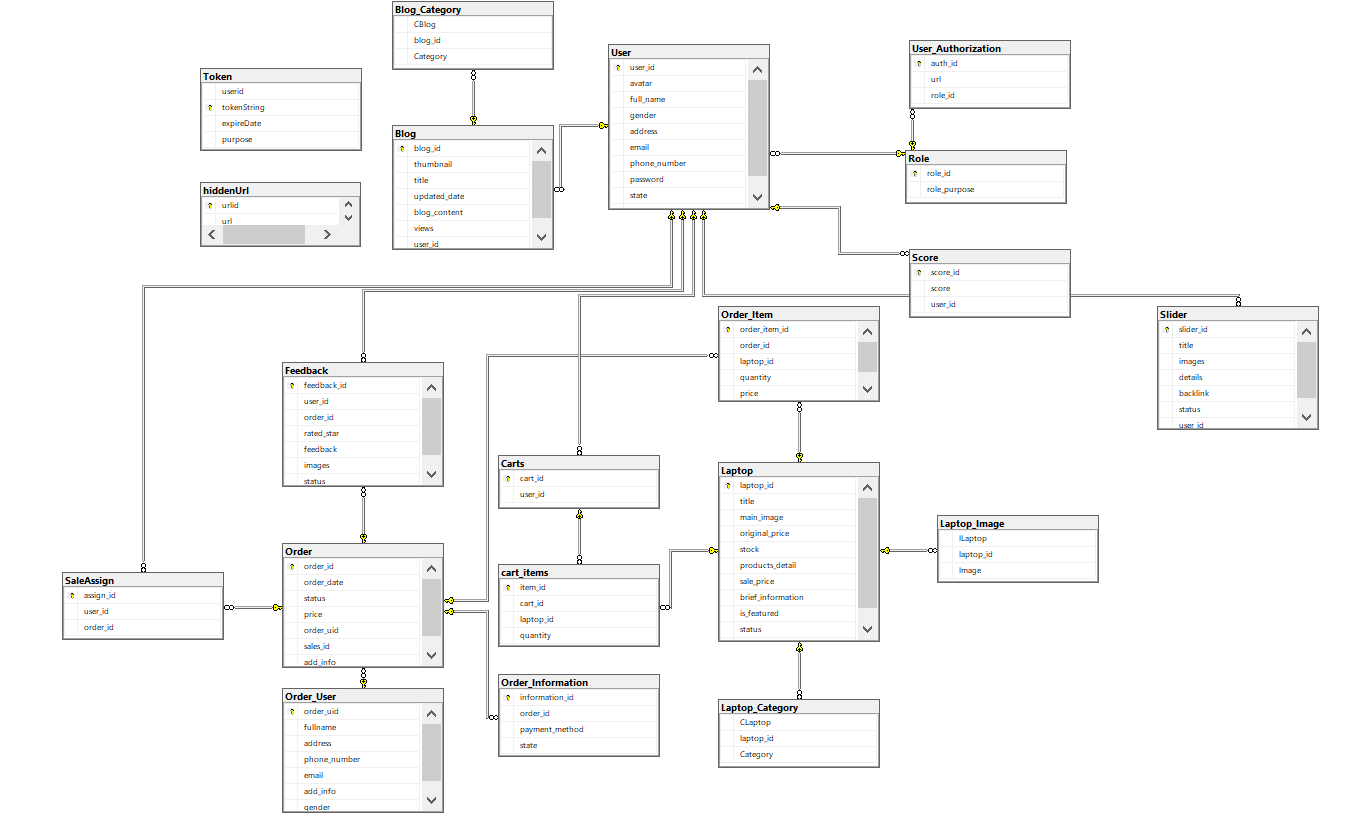
### 2.4 Non-UI Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| 1 | User Authorization (Common) | Non-UI Function | Implementation of authorization mechanism in the system, including the specify the role of logged-in user, his/her authorised page links (building the displayed menu items (in the front end) and preventing unauthorised access via enter the links directly). |
| 2 | Session Thrist | Non-UI Function | Automatically re-logged in for user even after they closed web browser if they tick “Remember me”, duration 1 hour |
| 3 | Cart | Non-UI Function | Auto fetch the cart cookie to the user |

## 3. System High Level Design

### 3.1 Database Design

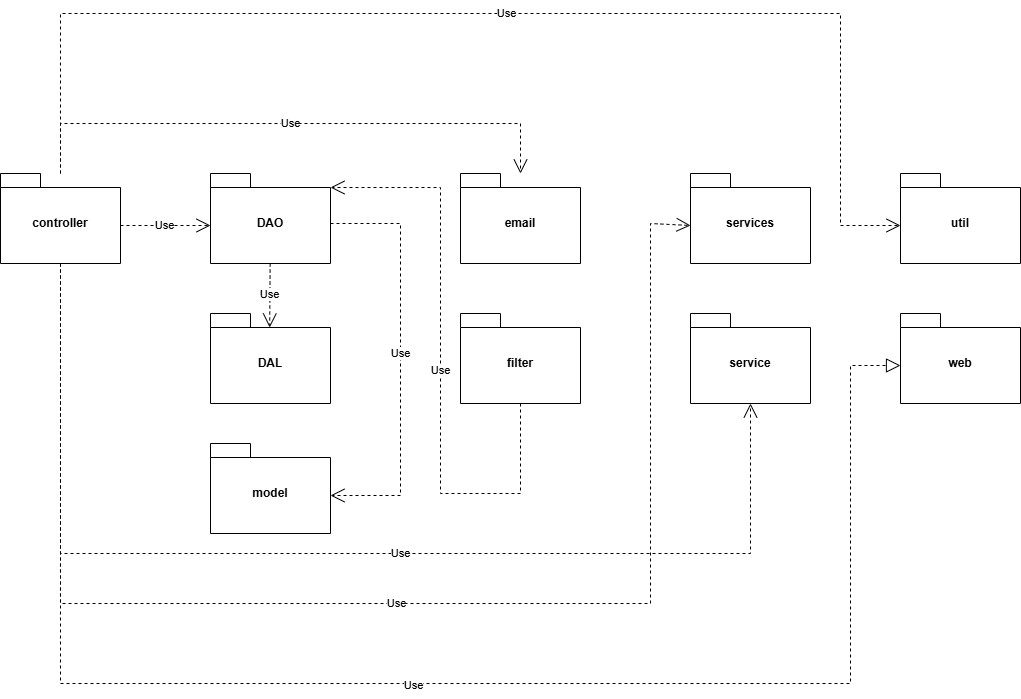
#### a. Database Schema



#### b. Table Descriptions

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| 01 | User | This table manage the user’s status  - Primary keys: **user\_id**  - Foreign keys: **role\_id** from **Role** |
| 02 | Role | This table manage the role from the project  - Primary keys: **role\_id** |
| 03 | Carts | This table manage the cart of a logged-in user, instead of using cookie  - Primary keys: **cart\_id** |
| 04 | cart\_items | This table manage the items of each logged-in cart  - Primary keys: **items\_id**  - Foreign keys: **cart\_id** from **Cart** |
| 05 | Order | This table manage the order(s) from the project  - Primary keys: **order\_id** |
| 06 | OrderItems | This table manage the items from each order  - Primary keys: **order\_item\_id**  - Foreign keys: **order\_id** from **Order** |
| 07 | OrderUser | This table manage the user who ordered said product  - Primary keys: **order\_uid**  - Foreign keys: **order\_id** from **Order** |
| 08 | Order Information | This table manage information from each product, including payment method  - Primary keys: **information\_id**  - Foreign keys: **order\_id** from **Order** |
| 09 | SaleAssign | This table saves the state of who assign who to which order  - Primary keys: **assign\_id**  - Foreign keys: + **order\_id** from **Order**  **+ user\_id** from **User** |
| 10 | Feedback | This table save the feedback for a product/for all service  - Primary keys: **feedback\_id**  - Foreign keys: + **order\_id** from **Order**  **+ user\_id** from **User** |
| 11 | Laptop | This table save the laptop item  - Primary keys: **laptop\_id** |
| 12 | Laptop\_Image | This table save the image for each laptop  - Primary keys: **ILaptop**  - Foreign keys: + **laptop\_id** from **Laptop** |
| 13 | Laptop\_Category | This table save the category for each laptop  - Primary keys: **CLaptop**  - Foreign keys: + **laptop\_id** from **Laptop** |
| 14 | Blog | This table save the list of blog  - Primary keys: **blog\_id** |
| 15 | Blog\_Category | This table save the category for each blog  - Primary keys: **CBlog**  - Foreign keys: + **blog\_id** from **Blog** |
| 16 | Slider | This table saves the slider  - Primary keys: **slider\_id**  - Foreign keys: + **user\_id** from **User** |
| 17 | Score | This table save each user’s score  - Primary keys: **score\_id**  - Foreign keys: + **user\_id** from **User** |
| 18 | Token | This table save token incase user forgot password  - Primary keys: **tokenString** |
| 19 | hiddenUrl | This table manage the url that should be hidden when showing the list of accessible url  - Primary keys: **urlid** |
| 20 | User\_Authorization | This table manage which user can access which url  - Primary keys: **authId**  - Foreign keys: **+ role\_id** from **Role** |

### 3.2 Code Packages



***Package descriptions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| 01 | controller | Contained all the servlet for the project |
| 02 | DAO | Contained Database Access Object based on the database table |
| 03 | DAL | Contains the DAL(Database Access Layer), which help the DAO to access database |
| 04 | filter | Contains filter, which run every time user visit certain url |
| 05 | model | Contains model class which used to get the data from the database or add then in |
| 06 | email | Contains class to send mail |
| 07 | service | Contain classes that help with direct problems |
| 08 | services | Contain classes that help with direct problems |
| 09 | util | Contain classes configurations |
| 10 | web | the front-end for the webpage |

# II. Requirement Specifications

## 1. Common Function

### 1.1 UC-1\_Login System

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-1\_Login System** | | |
| Created By: | LongND | Date Created: | 15/May/2024 |
| Primary Actor: | Customer(already registered) | Secondary Actors: | None |
| Trigger: | * User clicks the ‘Login’ button from the page header. * User attempts to access an authenticated feature directly via a link or by typing the URL into the address bar. | | |
| Description: | As a user, I want to be able to log into the system using my email and password so that I can access the system’s authenticated features and my personalised account. | | |
| Preconditions: | User account has been created & authorised | | |
| Postconditions: | * User logs in the system successfully * The system tracked successful login into the Activity Log | | |
| Normal Flow | **1.0 Login System**   1. The user is presented with a pop-up Login screen. 2. The user enters their email and password into the provided fields. 3. The user clicks the ‘Login’ button. 4. The system validates the entered credentials. 5. Upon successful validation, the system grants access to the authenticated features. 6. The system records the successful login in the Activity Log. 7. The user is redirected to the Home Page or the previously accessed page. | | |
| Alternative Flows: | None | | |
| Exceptions: | ***1.1.E1 System can’t authenticate the user***  1. The Error Message screen is shown to the user  2. User cancels the logging in *=> UC stops, change to UC-3\_View Home Page*  3. User clicks “Forgot Password?” link *=> change to UC-4\_Reset Password*  4. User clicks “Register” link *=> change to UC-2\_Register User Account*  ***1.1.E2 System can’t verify the user***   1. The Error Message is shown to the user 2. User cancel => *UC stops, change to UC-3\_View Home Page* 3. User click on “Verify Email” => change to *UC-4\_Verify\_Email* 4. User clicks “Register” link => change to *UC-2\_Register User Account* | | |
| Priority: | Must Have | | |
| Frequency of Use: |  | | |
| Business Rules: | FR1 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Invalid Logging In | User can’t be authenticated to login the system if below cases   * His/her logging-in details are incorrect * His/her account has not been verified * His/her account has been locked or blocked |

### 1.2 UC-2 Home Pages

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-2\_Home Pages** | | |
| Created By: | LongNDHE181257 | Date Created: | 23/May/2024 |
| Primary Actor: | Customer | Secondary Actors: | Marketing, Admin |
| Trigger: | * User access the web via url directly | | |
| Description: | As a user, I want to view sliders, hot posts, featured products, and a sidebar with the latest posts, static contacts/links on the home page, so that I can easily navigate and explore the content based on my interests. | | |
| Preconditions: | The sliders, hot posts, and featured products must be available and visible to the reader. | | |
| Postconditions: | None | | |
| Normal Flow | 1. Users access the web via URL directly. 2. User is presented with the home page containing the following components:  * Sliders: Each slider displays an image and title. The user can click on a slider to be redirected to its backlink. * Hot Posts: Each hot post displays a thumbnail, title, and brief information. The user can click on a hot post to be redirected to its details. * Featured Products: Each featured product displays a thumbnail, title, and brief information. The user can click on a product to be redirected to its details. * Sidebar with Latest Posts: The sidebar displays a list of the latest posts with their thumbnails, titles, and brief information. The user can click on a post to be redirected to its details. * Static Contacts/Links: Static contacts or links are displayed on the home page for easy access by the user. | | |
| Alternative Flows: | None | | |
| Exceptions: | ***2.2.E1: No Posts Available***   * The home page does not display any posts due to the absence of records in the database. * The system detects that there are no posts available to display on the home page. * The system presents a message to the user informing them that there are currently no posts available on the home page. * The system may include additional instructions or suggestions for the user, such as exploring other sections of the website or returning to the home page at a later time. | | |
| Priority: | Must Have | | |
| Frequency of Use: | Significantly | | |
| Business Rules: | FR1 | | |
| Other Information: |  | | |
| Assumptions: | * The system has a functioning database that stores all blog post information, including their types. * The user has access to the internet and a compatible web browser. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Content Availability | Sliders, hot posts, and featured products must be available and visible to the reader on the home page. |

### 1.3 UC-3 Product Details

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-3\_ Product Details** | | |
| Created By: | ThangBQ | Date Created: | 19/May/2024 |
| Primary Actor: | Customer | Secondary Actors: | Sale, Admin |
| Trigger: | * User clicks on one product on the “Shop” button on the page header of the screen | | |
| Description: | As a user, I want to see a products in details so I can choose what to buy based on my preferences. I also want the site to took as little time to load as possible, by doing so it can keep user interact with the system | | |
| Preconditions: | * The products must being created and set state not ‘empty’ | | |
| Postconditions: | * User can see the details of a product * User can view feedback of the product * User can choose to add to cart the product | | |
| Normal Flow | * The user navigates to the shop section of the website. * The system displays a list of products with filter options. * The User selects a product to watch the details * User then can add the product to Cart * User can see the feedback on the product * If they already purchase, they can leave a feedback as well | | |
| Alternative Flows: | none | | |
| Exceptions: | ***2.2.E1 Fail to display product by invalid product\_id***   1. The system shown an error indicating that no product matches request 2. User then press “Return” button => UC end, go to UC-3: Home Page   ***2.2.E2 The Product exist but took too long to load***   1. *The system shown an error indicate that the server is busy at the moment* 2. User then press “Return” button => UC end, go to UC-3: Home Page | | |
| Priority: | Must Have | | |
| Frequency of Use: | Depending on the number of active users. | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: |  | | |
| Assumptions: | * The system has a functioning database that stores all blog post information, including their types. * The user has access to the internet and a compatible web browser. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Product Visibility | Only product that has ‘public’ state can access by user |
| FR2 | Time Out Prevention | The time between user select on a product and see the details should be as little as possible |

### 1.4 UC-4 List Blogs

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-4\_List Blog** | | |
| Created By: | ThangBQ | Date Created: | 16/May/2024 |
| Primary Actor: | Customer | Secondary Actors: | Marketing, Admin |
| Trigger: | * User clicks “Blog” button on the page header of the screen * User click “Filter” to attempt to search for certain type of blogs by their interests | | |
| Description: | As a user, I want to view a list of available blog posts, so that I can choose which ones to read based on my interests. | | |
| Preconditions: | The blog posts must be published and visible to the reader. | | |
| Postconditions: | * The user views a filtered list of blog posts. * The user can select a blog post to read more details. | | |
| Normal Flow | * The user navigates to the blog section of the website. * The system displays a list of blog posts with filter options. * The user selects a filter by type. * The system updates the list to show only the blog posts of the selected type. * The Blog Reader selects a blog post to read. | | |
| Alternative Flows: | none | | |
| Exceptions: | ***2.2.E1 Cannot display id, blog does not exist***  *The system displays an error*  *The user then presses the “Return” button => returns to UC-3* | | |
| Priority: | Must Have | | |
| Frequency of Use: | Depending on the number of active users. | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: |  | | |
| Assumptions: | The user has access to the internet and a compatible web browser. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Blog Post wait time | Blog execution time must be fast |
| FR2 | Display posts beautifully | Posts display exactly the way the creator want |

### 1.5 UC-5 Blog Details

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-5\_ Blog Details** | | |
| Created By: | ThangBQ | Date Created: | 18/May/2024 |
| Primary Actor: | Customer | Secondary Actors: | Marketing, Admin |
| Trigger: | * User clicks on one Blog on the Blog List page | | |
| Description: | As a marketing, I want to see a my blog being displayed beautifully so that it can help making people attract to said product | | |
| Preconditions: | Detailed blog posts must be published and visible to readers. | | |
| Postconditions: | * User can watch the blog they choosed in details | | |
| Normal Flow | * The user navigates to the blog section of the website. * The system displays a list of blog posts with filter options. * The user selects a blog post to read. | | |
| Alternative Flows: | none | | |
| Exceptions: | ***2.2.E1 Fail to display blog with corresponding id***   1. The system shown an error 2. User then press “Return” button => UC end, go to UC-3: Home Page   ***2.2.E2 Blog is exist but took too long to load***   1. The system shown an error that the system is being busy at the moment 2. User then press “Return” button => UC end, go to UC-3: Home Page | | |
| Priority: | Must Have | | |
| Frequency of Use: | Depending on the number of active users. | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: |  | | |
| Assumptions: | * The system has a functioning database that stores all blog post information, including their types. * The user has access to the internet and a compatible web browser. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Blog Post Visibility | Only blog posts that have been marked as ‘public’ by the Author or System Administrator should be visible to the user. |
| FR2 | Display posts beautifully | Posts display exactly the way the creator want |

### 

### 1.7 UC-7\_Cart\_Detail

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-7\_Cart\_Detail** | | |
| Created By: | LongND | Date Created: | 5/June/2024 |
| Primary Actor: | Guest | Secondary Actors: | None |
| Trigger: | * User click on “Cart” icon on the header | | |
| Description: | As a user, I want to be able to view all the product inside my cart so that I can view what did I order along with their price. | | |
| Preconditions: | None | | |
| Postconditions: | None | | |
| Normal Flow | **1.7.1: Cart Detail**  User clicks on the "Cart" icon.  The system retrieves and displays all products currently in the user's cart.  The system displays the product name, quantity, and price for each item.  The user can review the contents of the cart. | | |
| Alternative Flows: | None | | |
| Exceptions: | **2.1.E1 System showns error**   1. Caused by network connectivity issues., database errors or insufficient system resources. 2. Display an error message indicating the issue. 3. Log the error for system administrators to investigate. | | |
| Priority: | Must Have | | |
| Frequency of Use: |  | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| **FR1** | Item Integirity | Ensure that only items added to the cart by the current user are displayed. |
| **FR2** | Accuration Prices | Prices should be accurate and reflect any discounts or promotions applied to the items. |

### 

### 1.8 UC-8\_Cart\_Information

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-8\_Cart\_Information** | | |
| Created By: | LongND | Date Created: | 5/June/2024 |
| Primary Actor: | Guest | Secondary Actors: | None |
| Trigger: | * User click on “Cart” icon on the header * User select “Checkout” button | | |
| Description: | As a user, I want to be able to view all the product that I have chosen, along with each quantity and total price. I also want to be able to enter the customer’s information to help with order without an account | | |
| Preconditions: | None | | |
| Postconditions: | None | | |
| Normal Flow | 1.9.1: Cart Information   * User click on the “Cart” icon * Cart shows a list of user’s selected product * User then select “Checkout” button | | |
| Alternative Flows: | None | | |
| Exceptions: | 2.1.E1 System showns error   1. Caused by network connectivity issues., database errors or insufficient system resources. 2. Display an error message indicating the issue. 3. Log the error for system administrators to investigate. | | |
| Priority: | Must Have | | |
| Frequency of Use: |  | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| **FR1** | Item Integirity | Ensure that only items added to the cart by the current user are displayed. |
| **FR2** | Accuration Prices | Prices should be accurate and updated once the price of said product update |

### 1.9 UC-10 Cart Completion

*a. Functional Description*

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-10 Cart Completion** | | |
| Created By: | ThangBQ | Date Created: | 23/May/2024 |
| Primary Actor: | User | Secondary Actors: | Sale/Sale Manager |
| Trigger: | * User access after the checkout page, fill in all information and press “Create Order” | | |
| Description: | As an user, I want the page to be able to add my order into the database so that the order can be saved and sales can manage them | | |
| Preconditions: | User fill in data in the Checkout page | | |
| Postconditions: | User successfully send the order, system display order status + estimated delivery date, user then can pay based on payment Method defined in checkout page | | |
| Normal Flow | * User access the web * User open the Checkout * User then fill in information * Pages show that order has been received, send a mail to the user, and navigate them to pay if they choosed pay with bank account in the checkout page * User then got redirected to VnPay payment page * User fill in information(cart number, ccv, name) and choose continue * User type in OTP they received * If the payment success, they will be redirected to “return” page. * After 5 second, redirect user to the Home page | | |
| Alternative Flows: | None | | |
| Exceptions: | 11.E1:No item found  If this occur, system will tell the user that their current cart is empty | | |
| Priority: | Must Have | | |
| Frequency of Use: | Significantly | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: |  | | |
| Assumptions: | * The system has a functioning database that stores all order. * The user has access to the internet and a compatible web browser. | | |

*b. Business Rules*

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Information Accuracy | All the information that displayed must be accurate with the one in the database |
| FR2 | Currency Accuracy | The currency converter must be as accurate as possible to prevent inflation |

### 1.10 UC-11 My Order

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-11 My Order** | | |
| Created By: | LongND | Date Created: | 23/May/2024 |
| Primary Actor: | Customer | Secondary Actors: | None |
| Trigger: | * User access by clicking on the “My Order” link on the header | | |
| Description: | As an user, I want to be able to see any order that i brought, I also want to check their current status, further edit if it wasn’t being delivered yet. I also want to re-buy my product if I find it helpful | | |
| Preconditions: | User Must login with customer account | | |
| Postconditions: | User can cancel if it didn’t delivered yet, or purchase the product(the order must be in pending state to do this) | | |
| Normal Flow | * User access the web * User log in with customer account * User then click on “My Order” * User received a page with table classified by their status * User can select the status to find the order they like | | |
| Alternative Flows: | None | | |
| Exceptions: | 11.E1:No order found  Caused By: SQL Exception, No order yet  System: Inform user that no order is found | | |
| Priority: | Should Have | | |
| Frequency of Use: | Significantly | | |
| Business Rules: | FR1 | | |
| Other Information: |  | | |
| Assumptions: | * The system has a functioning database that stores all order. * The user has access to the internet and a compatible web browser. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Information Accuracy | All the information that displayed must be accurate with the one in the database |

### 1.11 UC-11 Order\_Information

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-11 Order\_Information** | | |
| Created By: | LongND | Date Created: | 23/May/2024 |
| Primary Actor: | Customer | Secondary Actors: | None |
| Trigger: | * User access by clicking on any order on MyOrder page | | |
| Description: | As an user, I want to be able to see the details of the order I brought, I also want to be able to change the product quantity, cancel the order if emergency or have more control about what did I buy | | |
| Preconditions: | User Must login with customer account | | |
| Postconditions: | User can edit the order, and cancel if it didn’t delivered yet, can rebuy the product, or pirchase/delete if they aren’t want to purchase any more(the order must be in pending state to do this) | | |
| Normal Flow | * User access the web * User log in with customer account * User then click on “My Order” * User received a page with table classified by their status * User can select the status to find the order they like * User click on the id of an order * User receive a page contains order information * User can re-buy, or do any action that the current order’s state allowed | | |
| Alternative Flows: | None | | |
| Exceptions: | 11.E1:No order found  Caused By: SQL Exception, No order yet or the id didn’t match  System: Inform user that no order is found | | |
| Priority: | Should Have | | |
| Frequency of Use: | Significantly | | |
| Business Rules: | FR1 | | |
| Other Information: |  | | |
| Assumptions: | * The system has a functioning database that stores all order. * The user has access to the internet and a compatible web browser. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Information Accuracy | All the information that displayed must be accurate with the one in the database |

# 

## 2. Sales Functions

### 2.1 UC-9\_Order\_List

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-9: Order\_List** | | |
| Created By: | ThangBQ | Date Created: | 10/June/2024 |
| Primary Actor: | Sale | Secondary Actors: | None |
| Trigger: | * User(with Sale role) click on the Order button | | |
| Description: | As a sale, I want to be able to view a pagination list of orders, along with sort and search by, so that I can manage the order easier | | |
| Preconditions: | User must be in “Sale/Sale Manager” role | | |
| Postconditions: | System change the header to fit the sale | | |
| Normal Flow | **1.8.1: Order List**   * User login with a “Sale/Sale Manager” account * User click on the “Order” button on the header | | |
| Alternative Flows: | None | | |
| Exceptions: | **1.8.E1 System showns error**   1. Caused by network connectivity issues., database errors or insufficient system resources. 2. Display an error message indicating the issue. 3. Log the error for system administrators to investigate. | | |
| Priority: | Must Have | | |
| Frequency of Use: |  | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| **FR1** | Order Pagination | Order should be listed by paginated list |
| **FR2** | Price Updation | The total cost should being keep even if prices of each items inside an order change |

### 

### 2.2 UC-10: Order Dashboard

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-10\_Order\_Dashboard** | | |
| Created By: | ThangBQ | Date Created: | 5/June/2024 |
| Primary Actor: | Sales/Sales Manager | Secondary Actors: | None |
| Trigger: | * User login as “Sales/Sales Manager role” | | |
| Description: | As a sales, I want to be able to view the trends of success orders out of total orders, I also wants to be able to filter the date, as well as see how other member doing their work | | |
| Preconditions: | None | | |
| Postconditions: | None | | |
| Normal Flow | **1.10.1: Cart Information**   * User login with “Sales” or “Sales Manager” role account * System display dashboard of 7 previous day | | |
| Alternative Flows: | None | | |
| Exceptions: | **2.1.E1 System showns error**   1. Caused by network connectivity issues., database errors or insufficient system resources. 2. Display an error message indicating the issue. 3. Log the error for system administrators to investigate. | | |
| Priority: | Must Have | | |
| Frequency of Use: |  | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| **FR1** | Item Integirity | Ensure that all the information in the dashboard are displayed correctly |
| **FR2** | Accuration Prices | Each revenue keeping itself even if price of all the items in the order changed |

### 2.3 UC-12 Order Information

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-10 Order Information** | | |
| Created By: | ThangBQ | Date Created: | 23/May/2024 |
| Primary Actor: | Sale | Secondary Actors: | Sale/Sale Manager |
| Trigger: | * User access by clicking on the “view” URL on any order | | |
| Description: | As a sale manager, I want to be able to view in details any order details, including some basic information like username, address and password. I also want to assign the order to any sales to be able to edit it | | |
| Preconditions: | User Must login with sale/sale manager account | | |
| Postconditions: | None | | |
| Normal Flow | * User access the web * User log in with sale/sales manager role * User click on the Order link on the header * The Header shows the orders * User then click on “View” link on one of the order | | |
| Alternative Flows: | None | | |
| Exceptions: | 11.E1:No order with such id found  Caused By: SQL Exception, No order yet or the current id didn’t match with any order  System: Inform user that no order is found, and add a button to redirect the user back to home screen | | |
| Priority: | Should Have | | |
| Frequency of Use: | Significantly | | |
| Business Rules: | FR1 | | |
| Other Information: |  | | |
| Assumptions: | * The system has a functioning database that stores all order. * The user has access to the internet and a compatible web browser. | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Information Accuracy | All the information that displayed must be accurate with the one in the database |

## 3. Guest Function

### 3.1, UC-6\_Register User Account

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | UC-6\_Register User Account | | |
| Created By: | ThangBQ | Date Created: | 5/June/2024 |
| Primary Actor: | Guest | Secondary Actors: | None |
| Trigger: | * User clicks “Register” button on the page header * User clicks “Register” button on login page when attempts to access an authenticated feature directly via a link or by typing the URL into the address bar. | | |
| Description: | As a user, I want to be able to register my account into the system so that I can access the system’s authenticated features and create my own personalized account. | | |
| Preconditions: | User has not created an account before. | | |
| Postconditions: | * User successfully register new account into the system. * The system successful tracked new account creation into the Activity Log | | |
| Normal Flow | 2.0 Register User Account   1. User accesses the User Register screen 2. User types in user information details (full name, gender, email, mobile, address and password) 3. User clicks the “Register” button 4. System validates the register details (see 3.0.E1) 5. System sends verify link to provided email 6. System allow user to register account 7. System tracks user’s success register to the Activity Log 8. User is redirected to login page | | |
| Alternative Flows: | None | | |
| Exceptions: | 2.1.E1 System can’t register new user because user already exist in the system   1. The Error Message screen is shown to the user 2. User cancels the register => UC stops, change to UC-3\_View Home Page 3. User clicks “Login” link => change to UC-1\_Login System | | |
| Priority: | Must Have | | |
| Frequency of Use: |  | | |
| Business Rules: | FR1 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Password Encoding | User’s password must be encoded with MD5 hashing |

### 3.2, UC-12\_Feedback

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-12\_Feedback Management** | | |
| Created By: | ThangBQ | Date Created: | 5/June/2024 |
| Primary Actor: | Guest | Secondary Actors: | None |
| Trigger: | * User click on “Feedback” button on top of the header of the webpage. | | |
| Description: | As an user, I want to be able to feedback for my order so that the company can find out what’s wrong during to delivering process that caused un-satiisfaction to the user. I also want to be feedback for each one specific laptop, so that other can know that if this laptop worth the money. | | |
| Preconditions: | User has brought an order and the order status is “Completed” | | |
| Postconditions: | * User successfully send a feedback to the system * Said feedback, based on if it was general or directly to one specific item it will display on the productlist/product detail page respectfully. | | |
| Normal Flow | 3.2.1: Feedback   * User create an order for the webpage The order if being successfully purchase, Sales will update it, and the system call the user via email. * User go to feedback page * User entering their email used to order product * User gain a list of order that corresponding to this email * User select an order * System tell user to select General or a specific item * User rated by stars, sending images/media, and notes about the reason affect the rating result | | |
| Alternative Flows: | None | | |
| Exceptions: | **2.1.E1 System didn’t find any order relating to one email**   * Caused by said user cannot be found, or the email is wrong * System will tell user that no order with this user is found, please try again | | |
| Priority: | Must Have | | |
| Frequency of Use: | Alot(Depends on the amount of user on the system) | | |
| Business Rules: | FR1, FR2 | | |
| Other Information: |  | | |
| Assumptions: | User has a stable internet connection | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Order Integrity | Any order that didn’t belong to the user who typed the email in will result an error, prevent user from rating for a product they didn’t buy |
| FR2 | Rating sort | The rating with highest stars is placed on top so that user can see if it’s helpful |

## 4. Admin Function

### 4.1 UC-9 User Authorization

#### a. Functional Description

|  |  |  |  |
| --- | --- | --- | --- |
| UC ID and Name: | **UC-9 User Authorization** | | |
| Created By: | LongNDHE181257 | Date Created: | 23/May/2024 |
| Primary Actor: | Public | Secondary Actors: | Admin |
| Trigger: | * User access the web via url directly | | |
| Description: | As a administrator, I want to be able to edit the authorization of my webpages. I also want to shows to the user which page they can go, provided as a shortcut. I would also want if any user try bypassing by redirected URL, lead them back and inform them that current role has no authorization over | | |
| Preconditions: | None | | |
| Postconditions: | None | | |
| Normal Flow | * User access the web * User get the role “Guest” until login * User can visit any authorized website that shows in the “Page” list * After login, user can get the role of said logged-in account * On any site that the user has no authorizations over, caused by URL rewriting, lead user back to homepage and inform that current user has no authorization over | | |
| Alternative Flows: | **Admin Flow**   * Login as “Admin” account * Go to “User Authorization” page * Admin here can view and change the accessible role of a selected page | | |
| Exceptions: | None | | |
| Priority: | Must Have | | |
| Frequency of Use: | Significantly | | |
| Business Rules: | FR1, FR2, FR3 | | |
| Other Information: |  | | |
| Assumptions: |  | | |

#### b. Business Rules

|  |  |  |
| --- | --- | --- |
| **ID** | **Business Rule** | **Business Rule Description** |
| FR1 | Role Integrity | The role must be keep with an user after a web session, and changed upon user login |
| FR2 | Web Displayed Accuracy | The webpage displayed must be accuracy as the one in the webpage, so that said admin knows what this page is about |
| FR3 | Web Navigation | The web that shows to the admin must not be able to redirect to any other site |

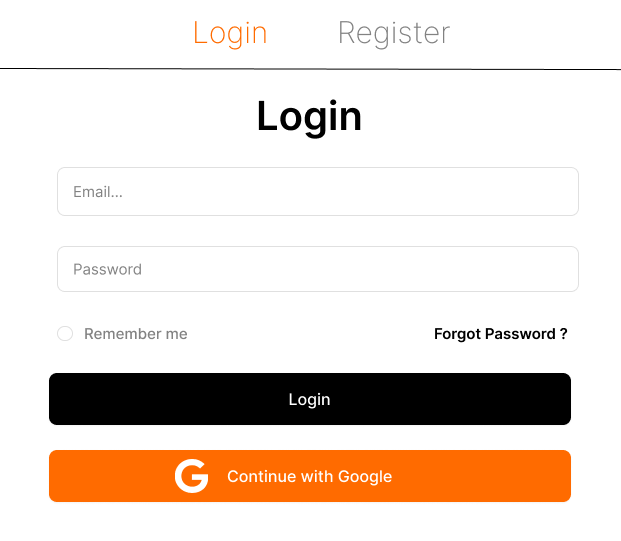
# III. Design Specifications

## 1. Common function

#### a, User Login

This screen allows users to be authenticated to the system screens/functionalities.

##### UI Design



##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| User | R | Verify email & Password information |
| Role | R | Specify the authorizations of the logged-in user |

***SQL Commands:***

1/ Verify Email & Password information

SELECT \*

FROM [user] WHERE email = ? AND password = ?

2/ Verify User role

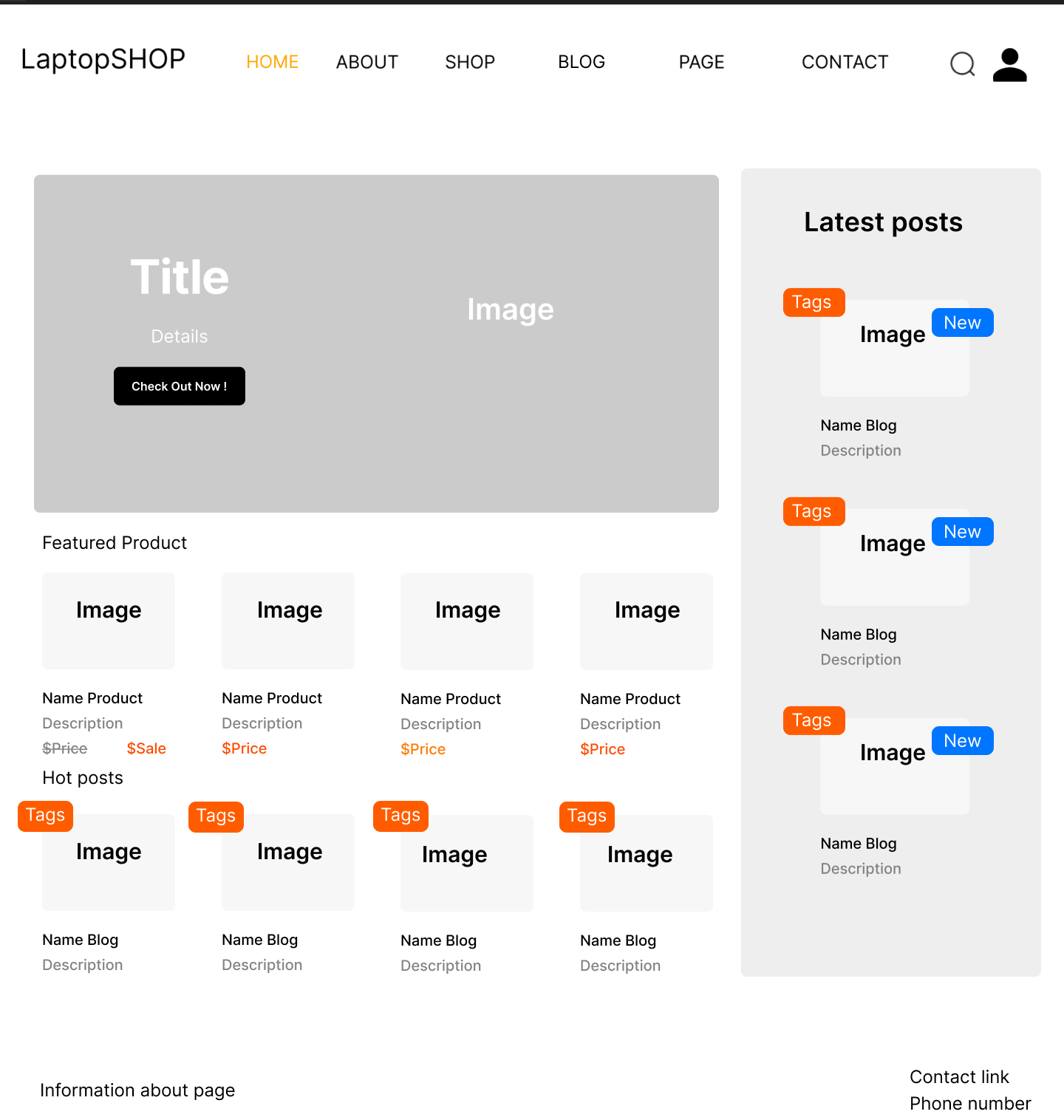
SELECT \*

FROM [user] INNER JOIN [role] on user.role\_id = role.role\_id  WHERE user.id = ?

#### b, Home page:

*This screen allows user to see Featured Products, Hot posts and Latest Post*

##### UI Screen:



##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Product | R | Display featured products |
| Blog | R | Display hot & latest blogs |
| Slider | R | Display sliders |

***SQL Commands***

**1/ Display Featured Products**

SELECT Top 4 \* FROM Laptop WHERE is\_featured = 1

**2/ Display Latest Blog**

SELECT TOP 4 \* from Blog ORDER BY updated\_date DESC

**3, Display Slider**

SELECT \* from Slider WHERE slider\_id IN (?, ?, ?)

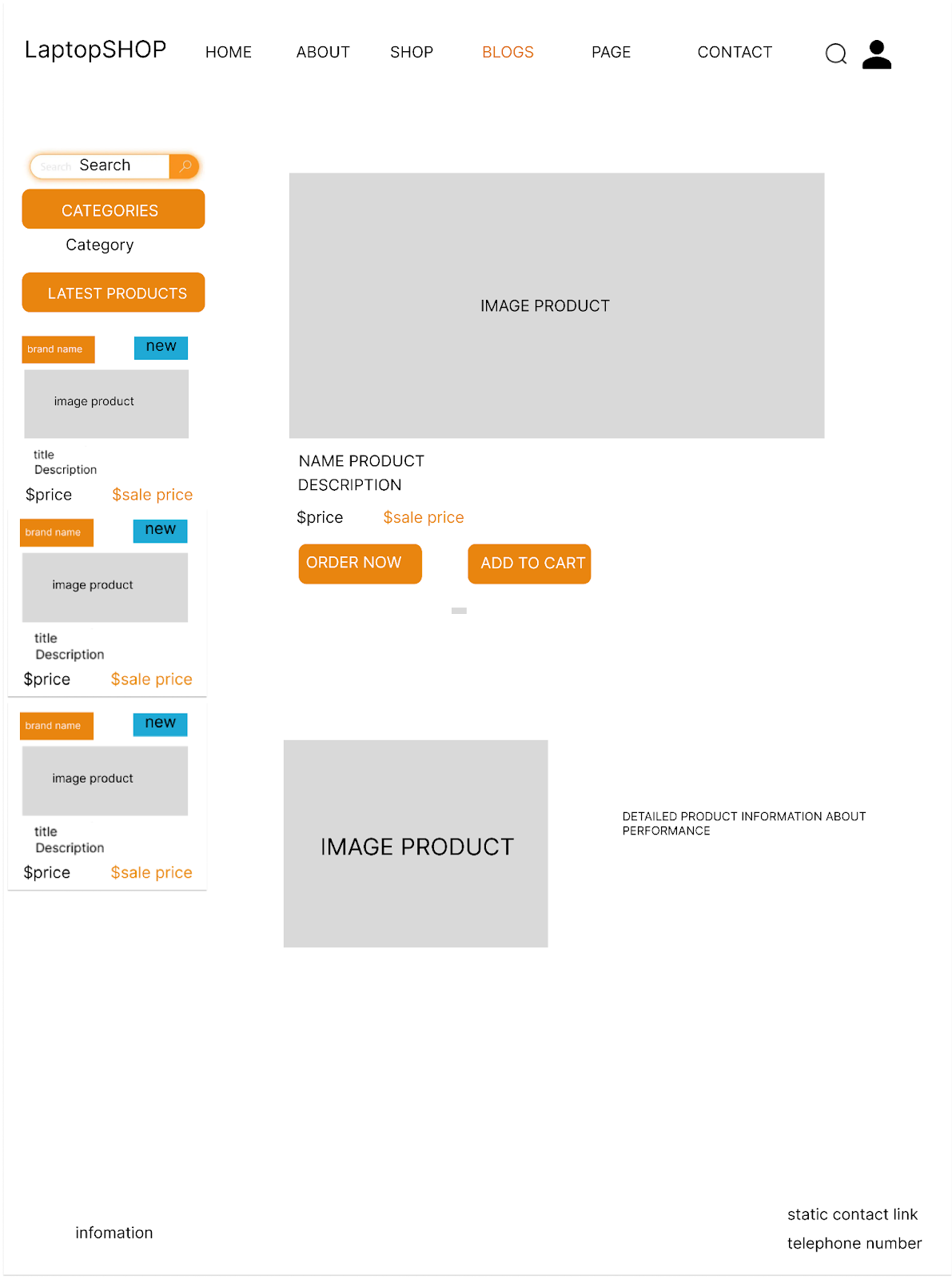
**4, Display Hot blog**

SELECT TOP 3 \* Blog ORDER BY [view]

#### c,Product Details

*This screen allow user to see the detail of a specific product*

##### UI Screen:



# 

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Laptop | R | Display a specific product based on id |

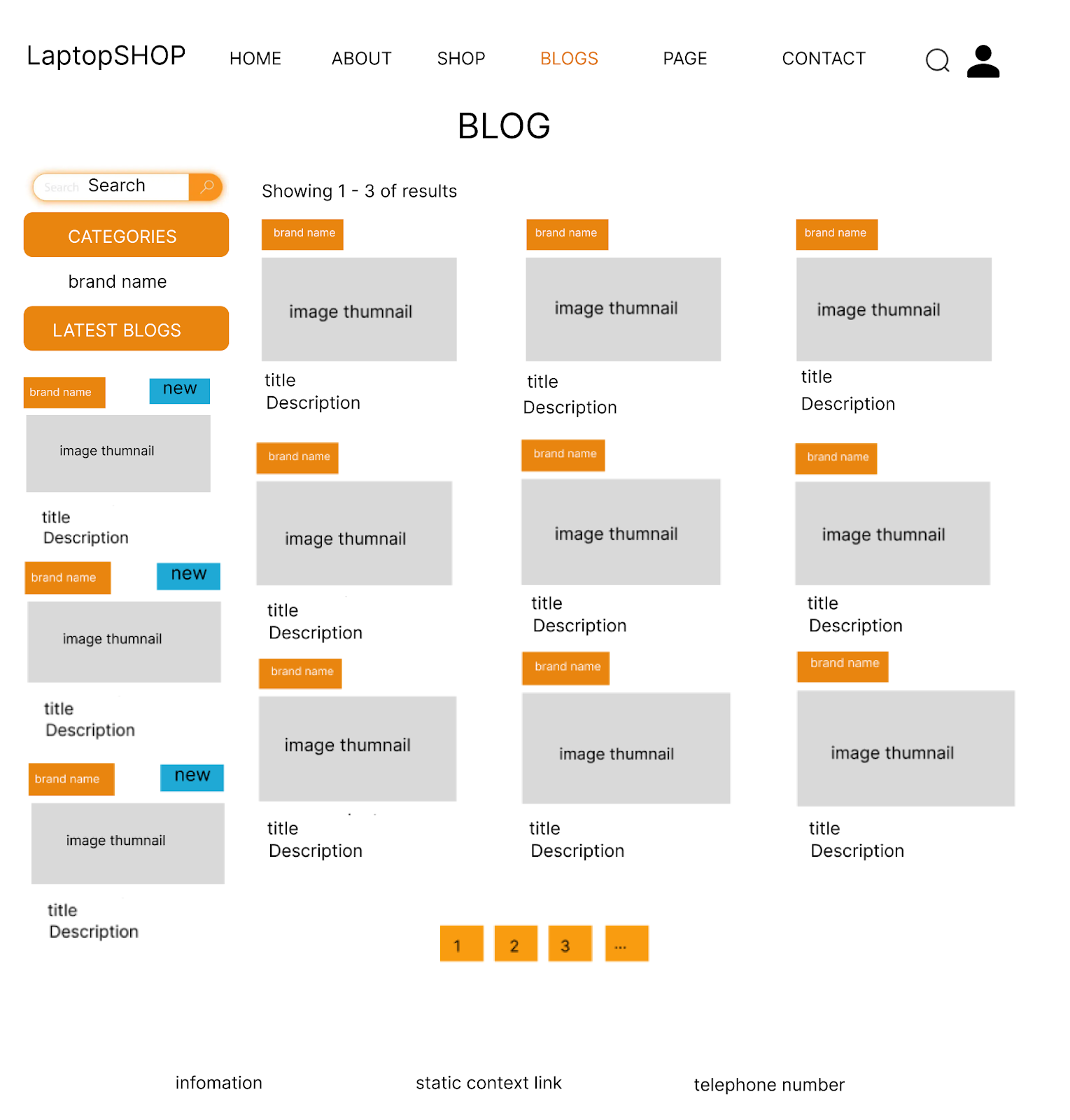
***SQL Commands***

Select \* from Laptop where Laptop.laptop\_id = ?

### d, Blog Lists

*This screen allow user to see the detail of a specific product*

##### UI Screen:



##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Blog | R | Display a specific list based on current page and all request data |

***SQL Commands***

select \* from Blog INNER JOIN Blog\_Category ON Blog.blog\_id = Blog\_Category.blog\_id WHERE category IN (?. ?. ?) AND title LIKE ? OFFSET ? ROWS FETCH NEXT ? ROWS ONLY

### e, Blog Detail

*This screen allow user to see the detail of a specific blog*

##### UI Screen:

##### 

##### Database Access:

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Blog | R | Display a specific Blog based on id |

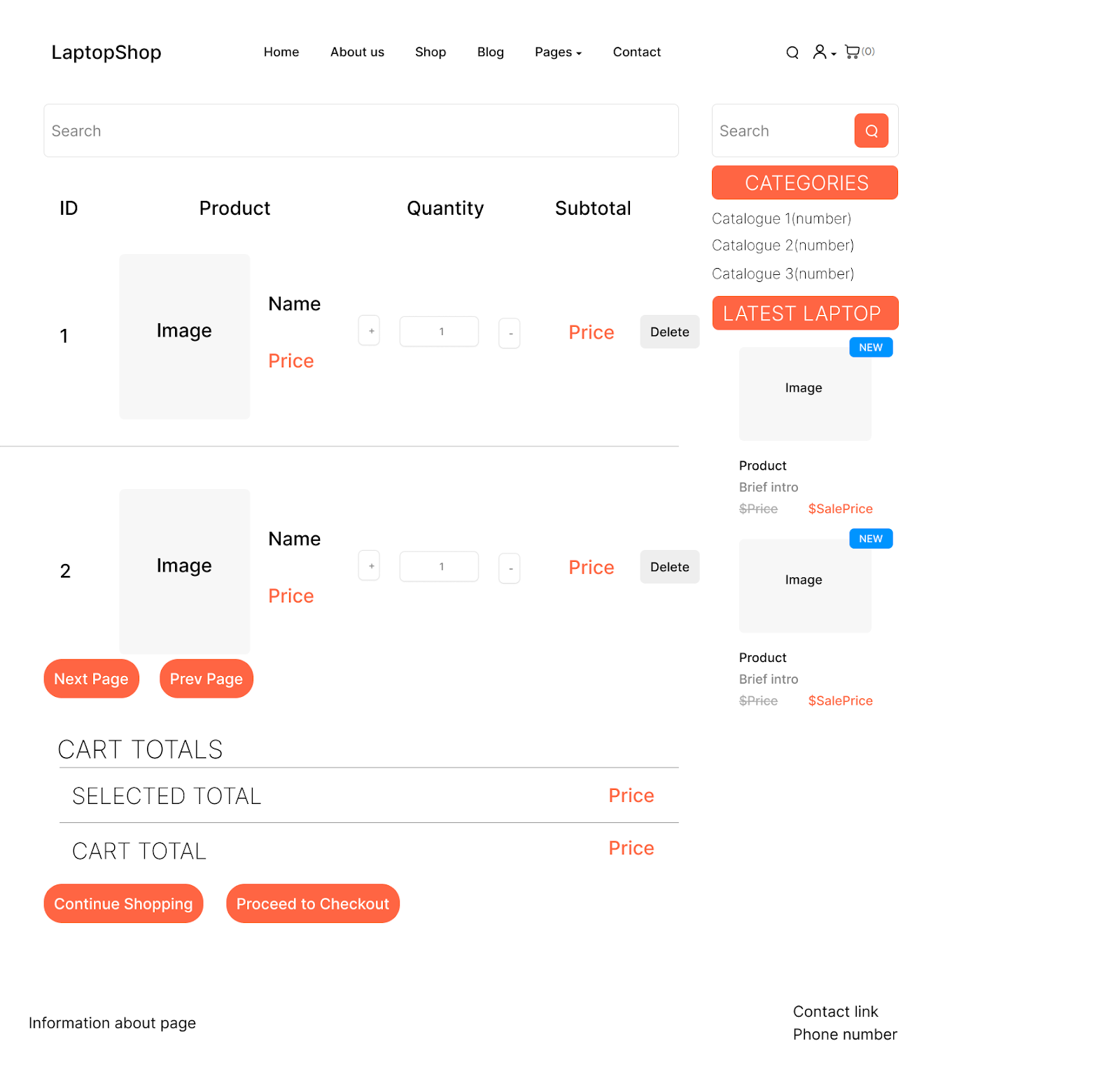
**SQL Query**

Select \* from Blog where Blog.blog\_id = ?

### f, Cart Details

*This screen allow user to view/edit cart*

##### UI Screen:



# 

##### Database Access:

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Cart | R | Takes the cart from said user(if account is being created), otherwise, use cookie |
| Cart\_Item | R | Take all the product from cart belong to user |

**SQL Query**

**1/ Fetch cart**

SELECT cart\_id FROM Carts WHERE user\_id = ?

INSERT INTO Carts (user\_id) VALUES (?)  
SELECT quantity FROM cart\_items WHERE cart\_id = ? AND laptop\_id = ?

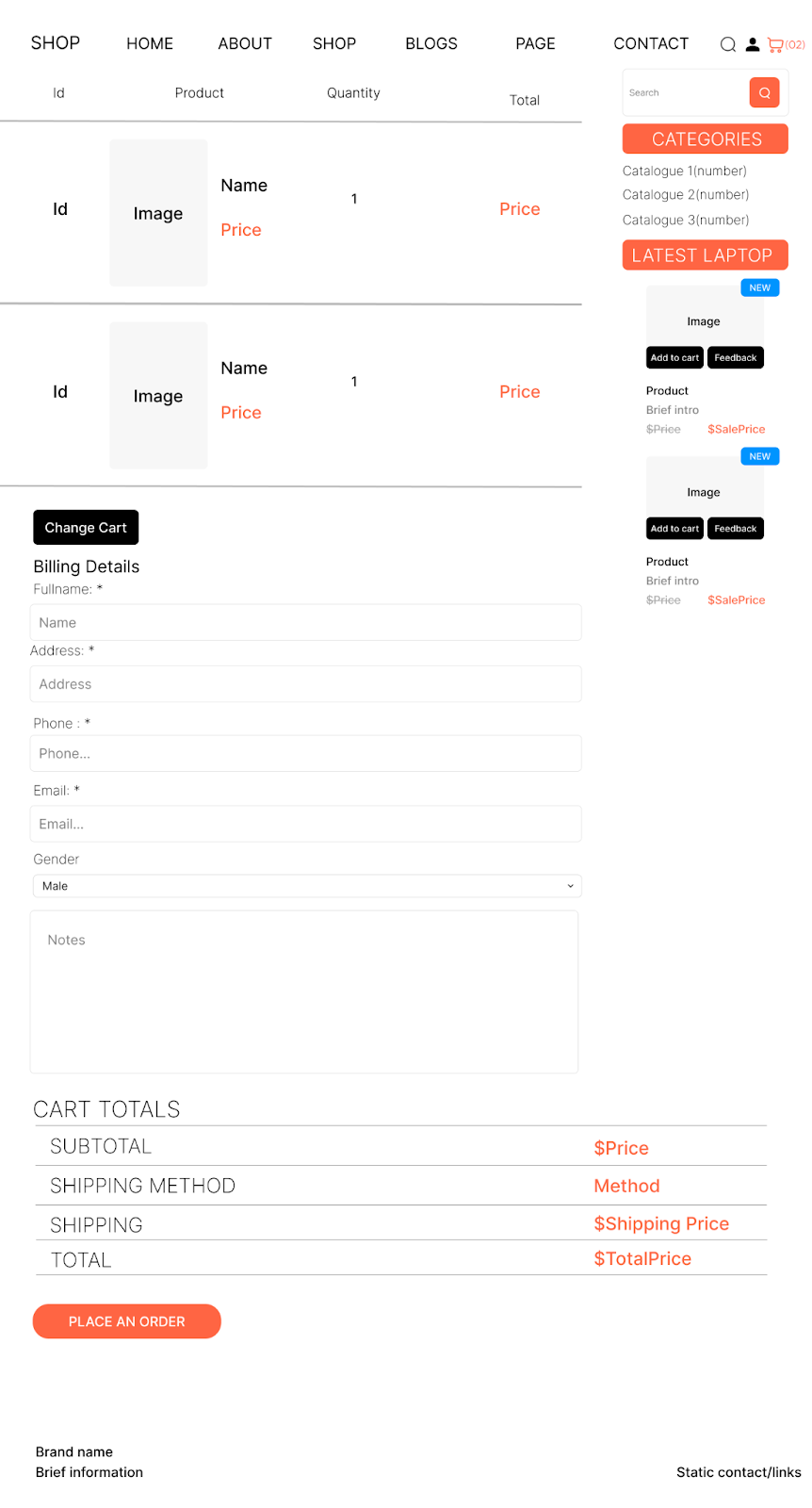
UPDATE cart\_items SET quantity = ? WHERE cart\_id = ? AND laptop\_id = ?

INSERT INTO cart\_items (cart\_id, laptop\_id, quantity) VALUES (?, ?, ?)  
UPDATE cart\_items SET quantity = ? WHERE cart\_id = ? AND laptop\_id = ?  
DELETE FROM cart\_items WHERE cart\_id = ? AND laptop\_id = ?  
SELECT \* FROM cart\_items WHERE cart\_id = ?

### h, Cart Information

This screen allow user to see their cart and edit their information before checkout

##### Ui Screen:



##### Database Access:

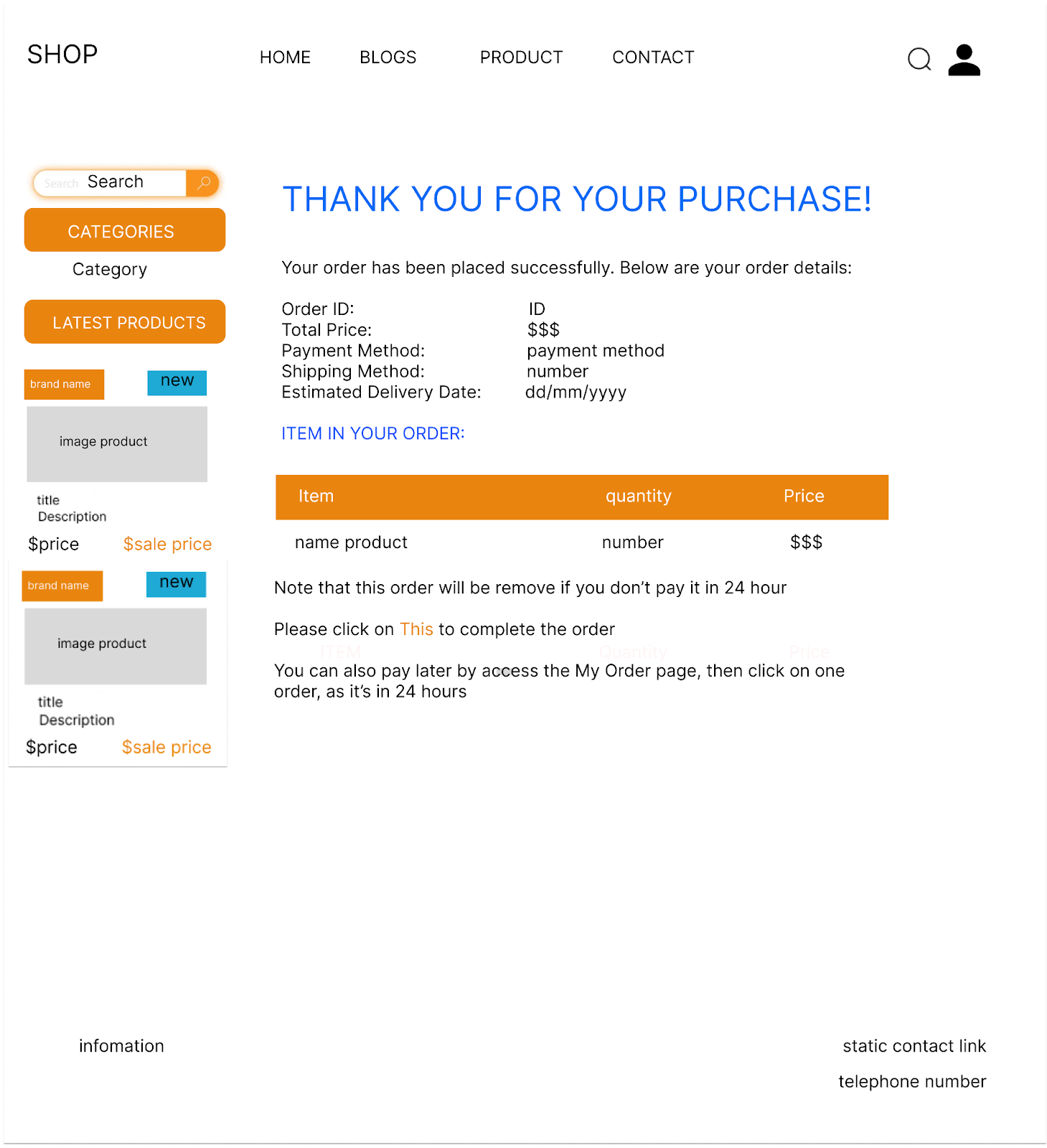
| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Laptop | R | Read newest laptop |

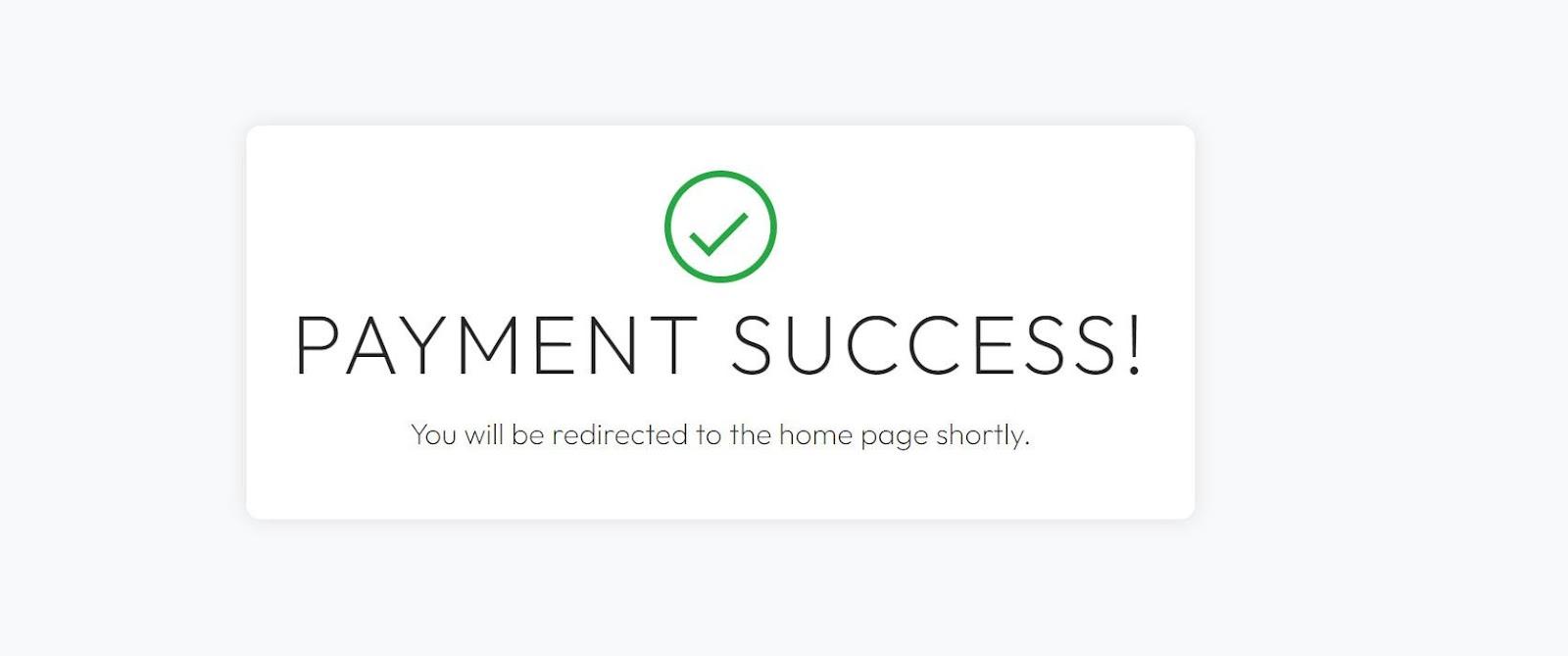
### 

### i, Cart Completion

This screen allow user to complete the order

##### Ui Screen:

****

****

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Order | C, U | Create the order Update the order if payment method are paid online + the order has been purchased |
| Order\_User | C | Create the user, or not if one already found by email |
| Order\_Item | C | Create the order based on the select in the cart |

**/SQL Command**

/INSERT INTO [Order] values (?,?,?,?,?,?)  
/INSERT INTO [Order\_User] values (?,?,?,?,?)

/INSERT INTO [Order\_Item] values (?,?,?,?)

/Update [Order] SET status = ? WHERE id = ?

### 

### j. My Order

This screen shows to user about their order, their state and basic action

##### UI Screen

# 

# 

##### Database Access:

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Order | RU | Fetch order, then classified it  Update to cancel order |
| OrderItem | R | Fetch all items from an order |
| OrderUser | R | Validate that all of these order belong to said account |

##### SQL Query

SELECT \* FROM [Order] where order\_uid = ?

SELECT \* FROM [Order\_Item] WHERE order\_id = ?

SELECT \* FROM Order\_User WHERE order\_uid = ?

### k, Order Information

This screen shows to user about their order, their state and basic action

##### UI Screen

# 

##### Database Access:

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Order | RU | Fetch order, then classified it  Update to cancel order |
| OrderItem | R | Fetch all items from an order |
| OrderUser | R | Validate that all of these order belong to said account |

##### SQL Query

SELECT \* FROM [Order] where order\_uid = ?

SELECT \* FROM [Order\_Item] WHERE order\_id = ?

SELECT \* FROM Order\_User WHERE order\_uid = ?

# 

## 2, Guest Function

#### e, User Register

*This screen allow user to create a new account*

##### UI Screen:

# 

##### Database Access:

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| User | R | Find if an email has been registered or not |
| User | C | Create a new account |

**SQL Query**

**1/ Check if an email with an user exist**

Select 1 from [User] where email = ?

**2/ Check if a phone number with an user exist**

Select 1 from [User] where phone\_number = ?

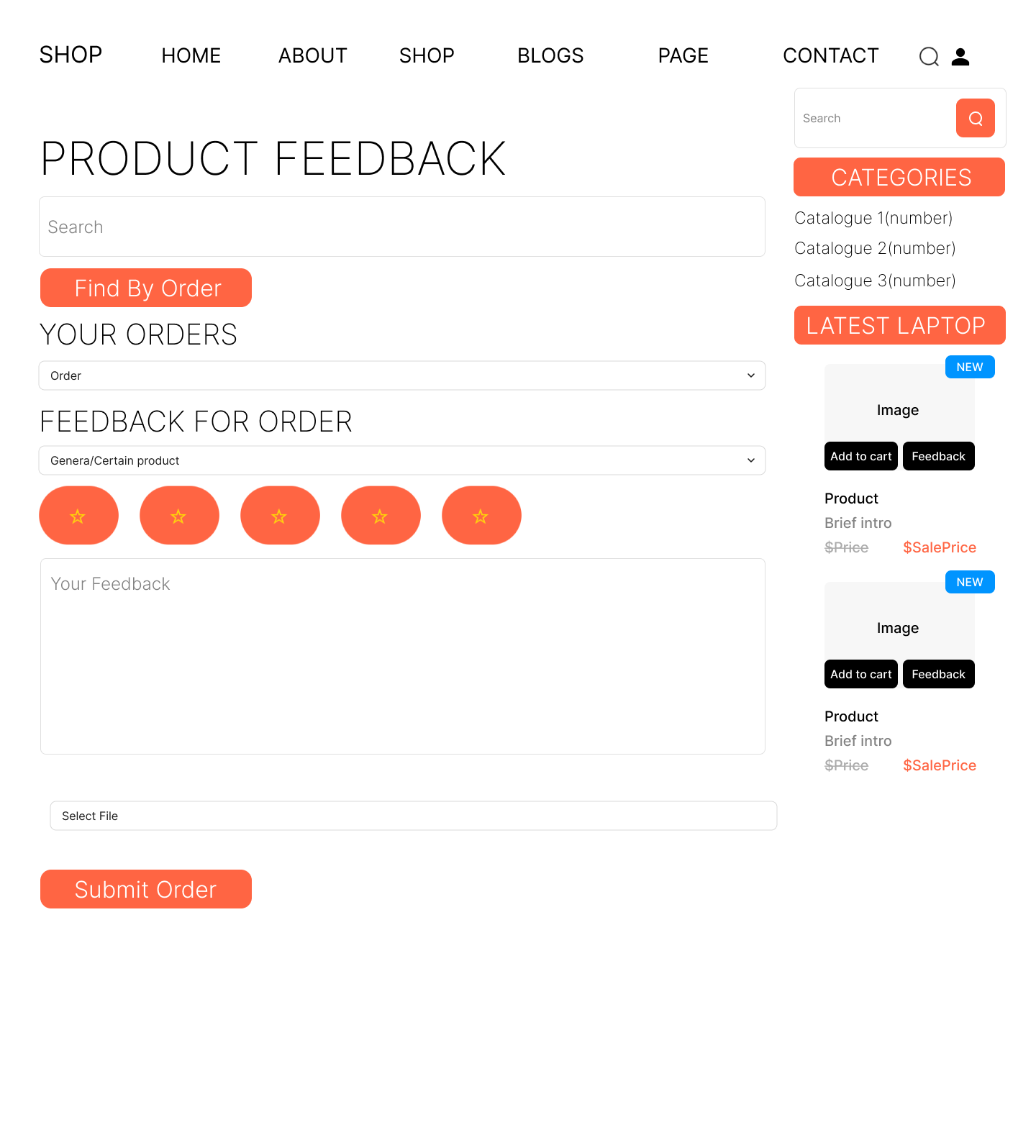
**3/Create a new account**

Insert into [User] values(?,?,?,?,?,?,?,?,?)

#### f, Feedback

*This screen allow user to feedback for an order, or each product inside*

##### UI Screen:



##### Database Access

| **Table** | **CRUD** | **Description** |
| --- | --- | --- |
| Feedback | CRUD | Read the feedback for update  Create a new feedback |
| FeedabackImage | CRD | Add/Delete images in the feedback |

### 

**SQL Query**

**1/ Check if an email with an user exist**

INSERT INTO Feedback (order\_uid, order\_id, rated\_star, laptop\_id, feedback, status) VALUES (?, ?, ?, ?, ?, ?)

UPDATE Feedback SET rated\_star = ?, feedback = ? WHERE feedback\_id = ?

**2/Update Feedback**

SELECT \* FROM Feedback WHERE laptop\_id = ?

**3/General Feedabck**

SELECT \* FROM Feedback WHERE order\_id = ? and laptop\_id = ? and order\_uid = ?

**4/ Display General Feedback**

SELECT \* FROM Feedback WHERE laptop\_id is null

**5/ Add Images**

INSERT INTO Feedback\_Image (image, feedback\_id) VALUES (?, ?)

**6/Delete Images**

DELETE FROM Feedback\_Image WHERE fimages = ?

**7/Get Images**

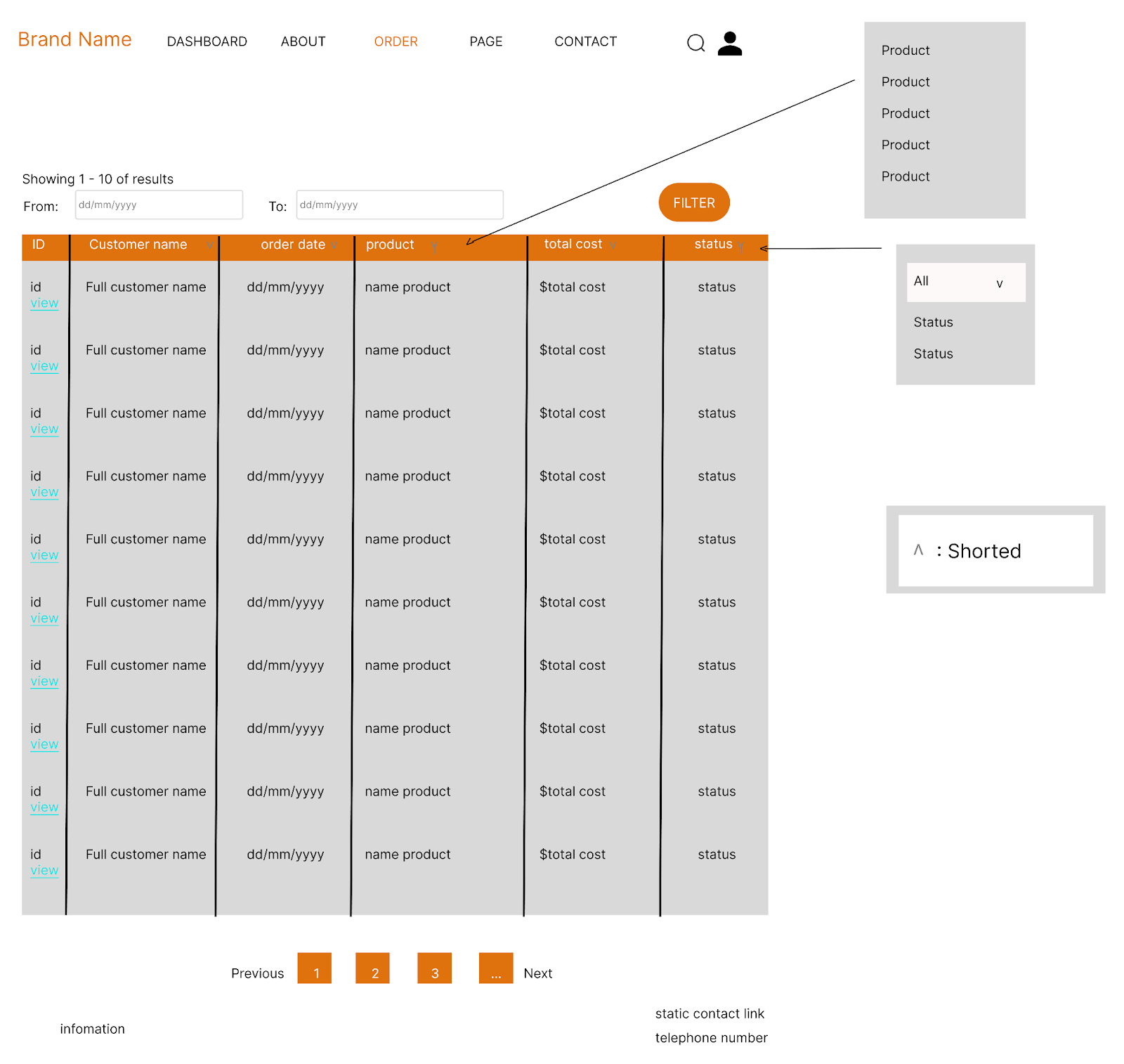
SELECT \* FROM Feedback\_Image WHERE feedback\_id = ?

# 3, Sales Function

### g, Order Lists

This screen allow user to see the list of orders:

##### UI Screen



##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Order | R | Show pagination list of order |
| Order\_Item | R | Show list of items being ordered |

**SQL Query**

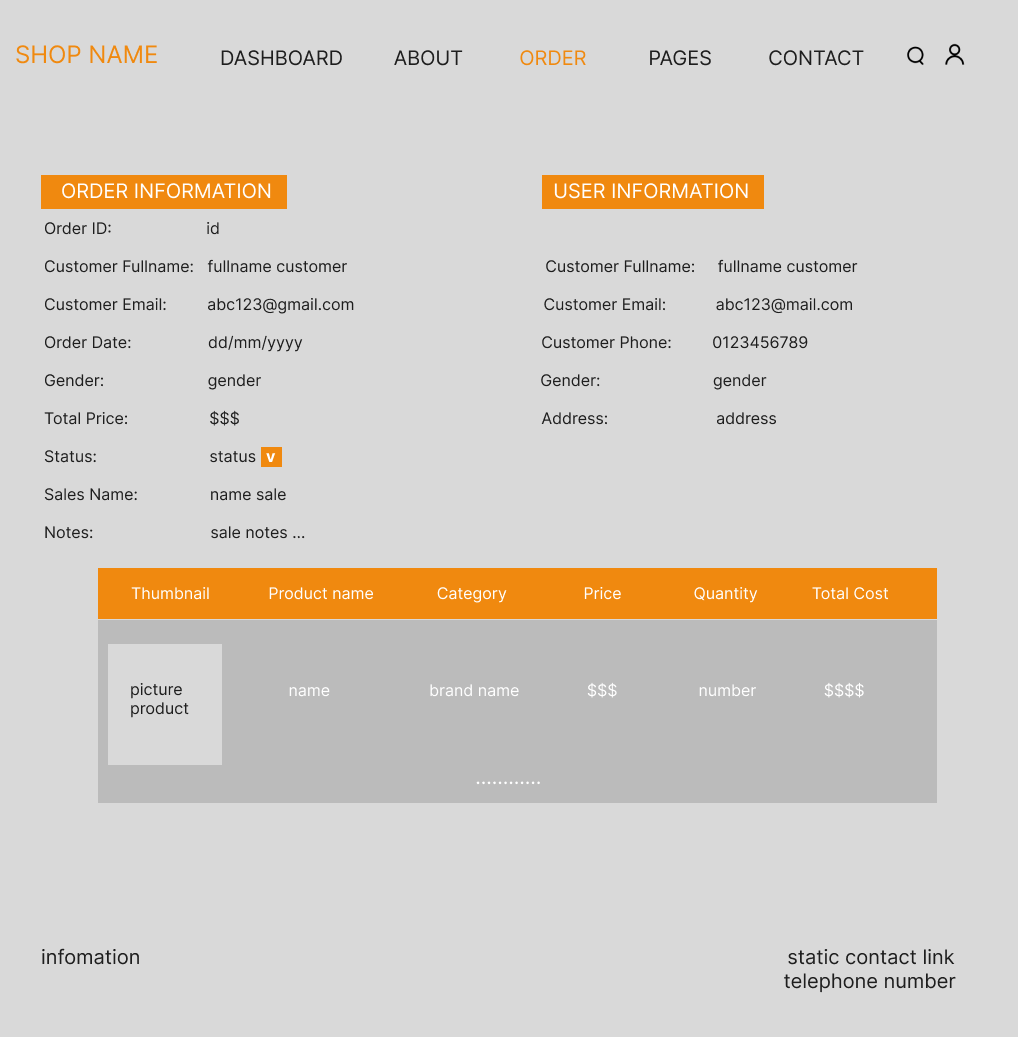
**1/Show pagination**

SELECT \* FROM [Order] INNER JOIN Order\_User ON [Order].order\_uid = Order\_User.order\_uid INNER JOIN [Order\_Item] on [Order].order\_id = Order\_Item.order\_id AND order\_id = ? AND Order\_User.fullname LIKE ? AND order\_date >= ? AND order\_date < ? AND order\_date < ? ORDER BY ? ASC OFFSET ? ROWS FETCH NEXT ? ROWS ONLY

### Order Details:

This screen allow user to see the details of an order

##### UI Screen



##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Order | R, U | Fetch the order based on ID |
| Order\_User | R | Fetch the User of said order, by order id |
| Order\_Item | R | Fetch all the items from this order, by orderid |
| Assign | C,R,U | Save the assign from sale manager to sales on 1 orderId |

**/SQL Command**

**Select order by ID**

SELECT \* FROM [Order] where order\_id = ?

**/Select order\_user by id**

SELECT \* FROM [Order\_User] where order\_id = ?

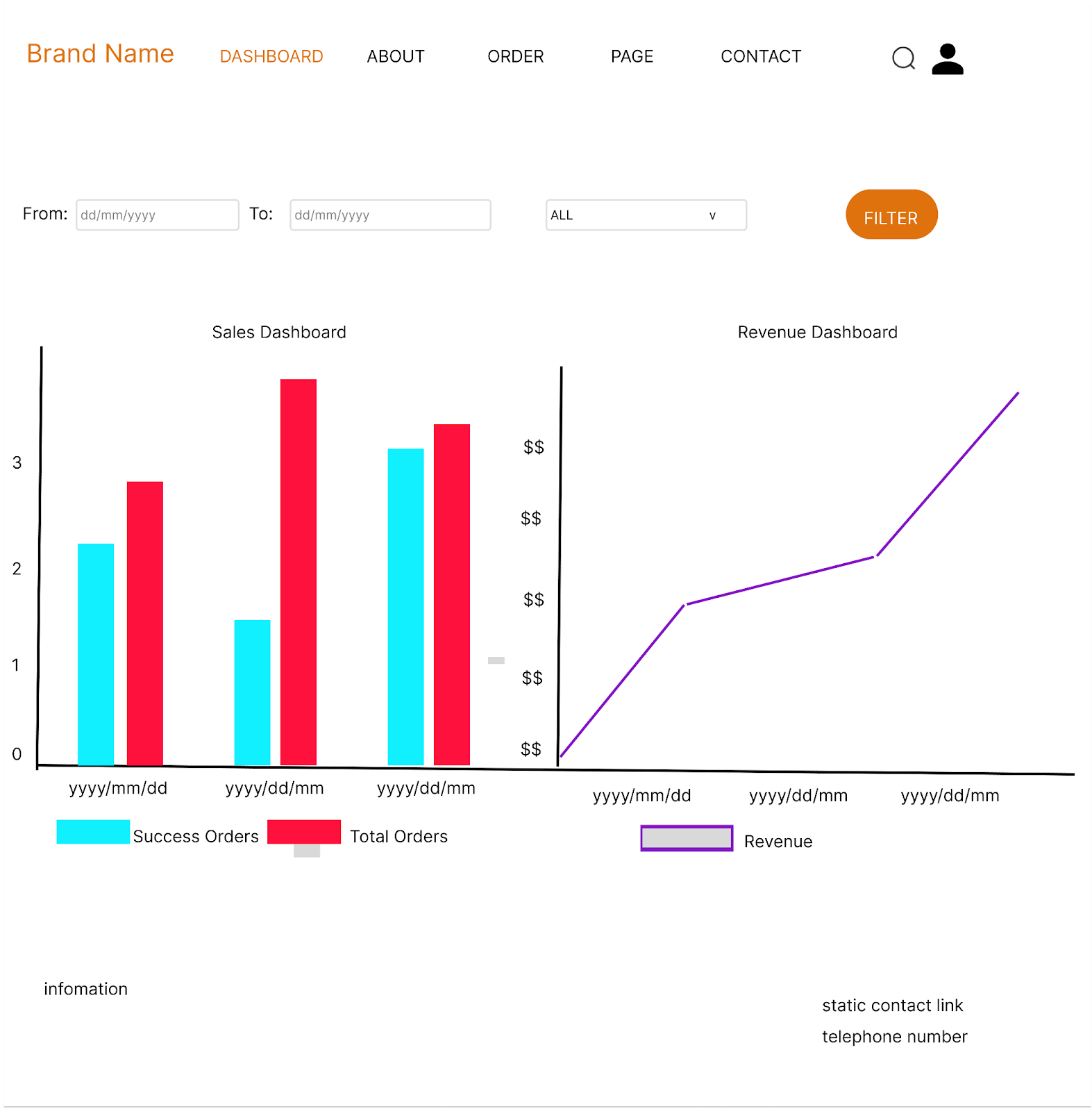
**/Select ordered Items by id**

SELECT \* FROM [Order\_Item] where order\_id - ?

### h, Sale Dashboard

*This screen allow user to see the detail of a specific product*

##### UI Screen:



##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| Order | R | Display all order, their price, their state and total Order, filter by days |

***SQL Commands*/Total**

SELECT \* FROM [Order] where Order.updated\_date between ? and ?  
**/Success**

SELECT \* FROM [Order] where Order.updated\_date between ? and ? and state == 3

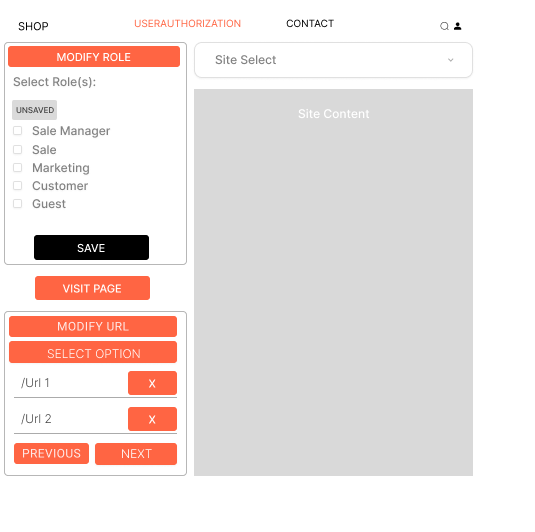
# 4, Admin Function

### User Authorization

This screen allows user to see any pages that they currently has role over

##### UI Screen:



****

##### Database Access

|  |  |  |
| --- | --- | --- |
| **Table** | **CRUD** | **Description** |
| User | R | Fetch the User based on the ID |
| Role | R | Get the role based on User’s RoleID |
| User\_Auth | C, R | Create and check if the current user is able to access the page |
| Hidden\_Url | C, R | Hidding any url that one user can access but have to hidden because of it’s the translation url |

***SQL Commands***

**1/ Find User with id**

SELECT \* FROM [User] where user\_id = ?

**2/ Find Role with id**

SELECT \* FROM [Role] where role\_id = ?

**3/Create authorization for URL page:**  
INSERT INTO User\_Authorization VALUES (?,?)

**4/HiddenUrl page**

INSERT INTO hidden\_url VALUES(?)

# IV. Appendix

### **4.1 External Interfaces**

#### a. User Interfaces

* ***Register and Login:*** Interfaces for new user registration and existing user login.
* ***User Profile:*** View and edit user information, including options to change the password.
* ***Blog Lists and Details:*** View blog posts and detailed content.
* ***Product List and Details:*** View available products and detailed information about a specific product. Allow user to add product to cart
* ***Home Page:*** Central hub for navigation and accessing various features.
* ***Cart Detail, Information, Completion:*** View and manipulate the cart, set the item for checkout process.

#### b. Software interfaces

* ***Database Connectivity:*** SQL Server for storing user data, quiz questions, and results, with JDBC used for database interactions.
* ***Server Communication:*** Servlets handle requests from JSP pages, processing data and interacting with the SQL Server database. AJAX is used for asynchronous data updates without refreshing the entire page.

#### c. Software interfaces

* ***HTTP/HTTPS:*** For secure web communication between clients and the Tomcat server.
* ***SMTP:*** For sending email notifications regarding quiz schedules, results, etc.

#### d. Security

* Include SSL/TLS encryption for data in transit and standard authentication protocols for secure access control.

### **4.2 Quality Attributes**

#### **4.2.1 Usability**

*Description:* The system should be user-friendly and accessible to a wide range of users.

* ***Ease of Use***: The UI should be intuitive, allowing users to navigate and perform tasks with minimal training.
* ***Consistency***: The UI should provide a consistent look and feel across all pages and functionalities.
* ***Error Handling***: Clear and informative error messages should be provided to help users resolve issues independently.

#### **4.2.2 Reliability**

*Description:* The system should be dependable and operate without failure.

* ***Availability***: The system should have an uptime of 99.9%, ensuring it is available for users most of the time.
* ***Fault Tolerance***: The system should be able to recover gracefully from hardware failures, network issues, and other unexpected problems.
* ***Data Integrity***: The system must ensure data accuracy and consistency, especially for user information, order details, blog details and product details.

#### **4.2.3 Performance**

*Description:* The system should be dependable and operate without failure.

* ***Response Time:*** The system should respond to user requests within 3-5 seconds under typical load conditions.

## **5. Requirement Appendix**

### **5.1 Business Rules**

|  |  |
| --- | --- |
| **ID** | **Rule Definition** |
| BR-01 | Overall web functionality must be fast and user-friendly |
| BR-02 | Must use POST method whenever the submission contains sensitive data |
| BR-03 | Overall data must be correctly retrieved from the database |
| BR-04 | The data that contains sensitivity information must be encrypted before doing action |
| BR-05 | Cart should be store as a cookie in-order for user to keep the cart for a long period |

### **5.2 Common Requirements**

* ***Authentication:*** 
  + Secure login and registration process with password encryption.
  + Validate user with their email

### **5.3 Application Messages List**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Message code** | **Message Type** | **Context** | **Content** |
| 1 | MSG01 | In red, above the header of the page | Login with wrong email or password | *Incorrect Email/Password* |
| 2 | MSG02 | In red, above the header of the page | Login with inactive account | *Account is not activated yet, please check your email for activation link* |
| 3 | MSG03 | In red, above the header of the page | Register with too short name | *Name must be between 3 and 30 characters!* |
| 4 | MSG04 | In red, above the header of the page | Register with invalid email format | *Invalid email format!* |
| 5 | MSG05 | In red, above the header of the page | Register with invalid phone number format | *Phone number should have 10 digits!* |
| 6 | MSG06 | In red, above the header of the page | Register with either too short(<20) and too long (> 255) address | *Address must be between 20 and 255 characters!* |
| 7 | MSG07 | In red, above the header of the page | Register with password and repassword not the same | *Passwords do not match!* |
| 8 | MSG08 | In red, above the header of the page | Register with invalid gender | *Invalid gender! Gender must be Male, Female, or Other.* |

### **5.4 Other Requirements**

#### Testing:

* Comprehensive testing to cover functionality, performance, security, and usability.
* User acceptance testing to ensure the system meets the needs of end-users.

## 

## 