|  |  |
| --- | --- |
|  | **FPT ACADEMY INTERNATIONAL**  **FPT – APTECH COMPUTER EDUCATION** |

**Centre Name: ACE-HCMC-2-FPT.**

**Address: 590 Cach Mang Thang 8, District 3, Ho Chi Minh City, Viet Nam.**

**SHOPP**

|  |  |  |
| --- | --- | --- |
| **Supervisor:** | MR. HOANG DUC QUANG | |
| **Semester:** | 4 | |
| **Batch No:** | T1.2005.E1 | |
| **Group No:** | 2 | |
| **Order:** | **Full name** | **Roll No.** |
| 1. | NGUYEN HOANG PHU | Student1262866 |
| 2. | DAO LE DUONG |  |
| 3. | LE HONG SON | Student1263452 |
| 4. | NGUYEN MINH DUY | Student1141090 |

April 2022

**Contents:**

This is to certify that

Mr. Nguyen Hoang Phu

Mr. Le Hong Son

Mr. Dao Le Duong

Mr. Nguyen Minh Duy

Have successfully: Designed & Developed

SHOPP

Submitted by:

Mr. Hoang Duc Quang

Date of Issue:

Authorized Signature:



**REVIEW I 1**

**I. Acknowledgment 2**

**II. Problem Definition 3**

1. Introduction 3

2. Project Requirement 3

3. System Requirements 6

3.1. Server Requirements 6

3.2. Client Requirements 6

4. Development Software 6

5. Technology 6

**Task sheet review 1 8**

**REVIEW II 9**

Architecture & Design of the Project 10

**I. Data Flow Diagram** 12

1. Context Diagram 13

2. Level 0 Diagram 13

3. Level 1 Diagram 14

3.1 Login 14

3.2 Manage 14

3.3 Allot 15

3.4 Sale 15

**II. Use Case Diagram 16**

1. Company use case 16

2. Customer use case 17

3. Dealer use case 18

**III. Sequence Diagram 19**

1. Dealer login sequence 19

2. Dealer purchase vehicle from company sequence 21

3. Dealer sale to customer sequence 23

4. Dealer check reports sequence 25

**IV. Entity Relationship Diagram 26**

1. Logical ERD 26

2. Physical ERD 27

**V. Table Design 28**

1. Branch 28

2. Vehicle 28

3. Service 28

4. Status Order 29

5. Dealer 29

6. Customer 29

7. Showroom Vehicle 30

8. Warehouse Vehicle 30

9. Order 30

10. Order Detail 31

**Task Sheet Review II 32**

**REVIEW III 33**

**1. Login 34**

**2. Statistical 34**

**3. Order 35**

3.1 Order list 35

3.2 Create order sale 35

3.3 Create order purchase 36

3.4 Update status 36

**4. Showroom vehicle 37**

4.1 Showroom vehicle list 37

4.2 Register new showroom vehicle 37

**5. Company Vehicle 38**

**6. Dealer 38**

6.1 Dealer list 38

6.2 Dealer update 39

6.3 Register new dealer 39

**7. Customer 40**

7.1 Customer list 40

7.2 Customer update 40

**8. More 41**

**Task Sheet Review III 42**

**REVIEW 1**

**I Acknowledgement**

It is often said that “Gratitude is the sign of noble souls”. That’s why we devote this chapter to express our deep gratitude towards those who have supported and guided us to accomplish this Project successfully.

First and foremost, we appreciate the invaluable assistance that our beloved instructor – Mr.Hoang Duc Quang, has offered us. Whenever problems come up, he’ll always the one that we can count on for solutions. Moreover, there were moments when our team decided to give up our project, and Mr. Nguyen gave us a pep talk to keep us back on the track. Therefore, once again, we truly value your dedication and guidance, Mr. Nguyen.

Next, it would have been impossible to carry out and complete this project without the collaboration among our team members. Although we have gone through many obstacles during the project implementation, eventually we gained what we aimed at by our hard work, our strong commitment and of course, our brotherhood. Finally, thanks to Aptech’s Project assignment, we had a golden opportunity to apply what we have learnt into reality and gain hands-on experience which is essential for our future career.

**II. Problem Definition**

**1. Introduction**

Currently, the era of technology is growing, online shopping and transactions have become too popular in modern society. Therefore, the demand for sales and purchases between people is growing and diversifying. So we created a project to serve that need – Shopp.

We believes in the evolving power of technology and wishes to contribute to making the world a better place by connecting the community of buyers and sellers through providing an e-commerce platform. Shopp project platform is built to provide users with an easy, safe and fast experience when shopping online through a strong operating and payment support system. The project provides users with a simple, easy and enjoyable online shopping experience.

**2. Project Requirement**

**2.1 Customer's Requirement Specifications**

The website will be an e-commerce platform 24/7 to provide customers with information about products, post sales, order online at the website at any time.

Client functional:

* Sign up
* Sign in
* Forgot password

Client (Buyer) functional:

* Manage Information
* Update profile
* Update address
* Update card Information
* View products
* Filter products
* Filter by price
* Filter by local
* Filter by name
* Filter by category
* Filter by rates
* Manage orders
* View list orders
* View detail orders
* Follow orders
* Payment
* Payment offline
* Payment online by card
* Rate products
* View rates
* Manage cart
* Add to cart
* Delete product in cart
* Edit quantity product in cart

Client (Seller) functional:

* Manage shop
* Register shop
* Update shop’s profile
* Manage shop’s transaction
* Withdraw shop’s fund
* Manage products
* Post products
* Delete products
* Update products
* Manage orders
* View orders
* Update, tracking status orders

Admin functional:

* Report revenue
* Report shops
* Follow customers’ order

**3. System Requirements**

**3.1 Server Requirements**

Hardware

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| **CPU** | Processor type: Pentium IV-compatible processor or faster Processor speed: Recommended: 2.0 GHz or faster |
| **OS** | Microsoft Windows XP or higher with IIS |
| **Memory (RAM)** | RAM: Minimum: 512 MB Recommended: 2 GB or more |
| **Hard Drive** | Free space: Minimum: 200 MB Recommended: 50 GB or more Maximum: Operating system maximum |

Software

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| **Java Development Kit** | JDK 11 |
| **RDBMS** | MySQL Server |

**3.2 Client Requirements**

Hardware

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| CPU | Processor type: Pentium III-compatible processor or faster Processor speed: Recommended: 1.0 GHz or faster |
| Memory (RAM) | RAM: Minimum: 512 MB Recommended: 1 GB or more Maximum: Operating system maximum |
| Hard Drive | Free space: Minimum: 10 MB |

Software

|  |  |
| --- | --- |
| **Component** | **Requirement** |
| **OS** | All OS(Window ,Linux ,Android ,Mac OS …) |
| **Web Browser** | IE 4.0 , Firefox 3.0,Chrome or Higher….. |

**4. Development Software**

* BDeaver
* Jetbrains Intellij
* Microsoft Visio 2016.
* Java Development Kit (JDK) 11
* Tomcat Server

**5. Technology**

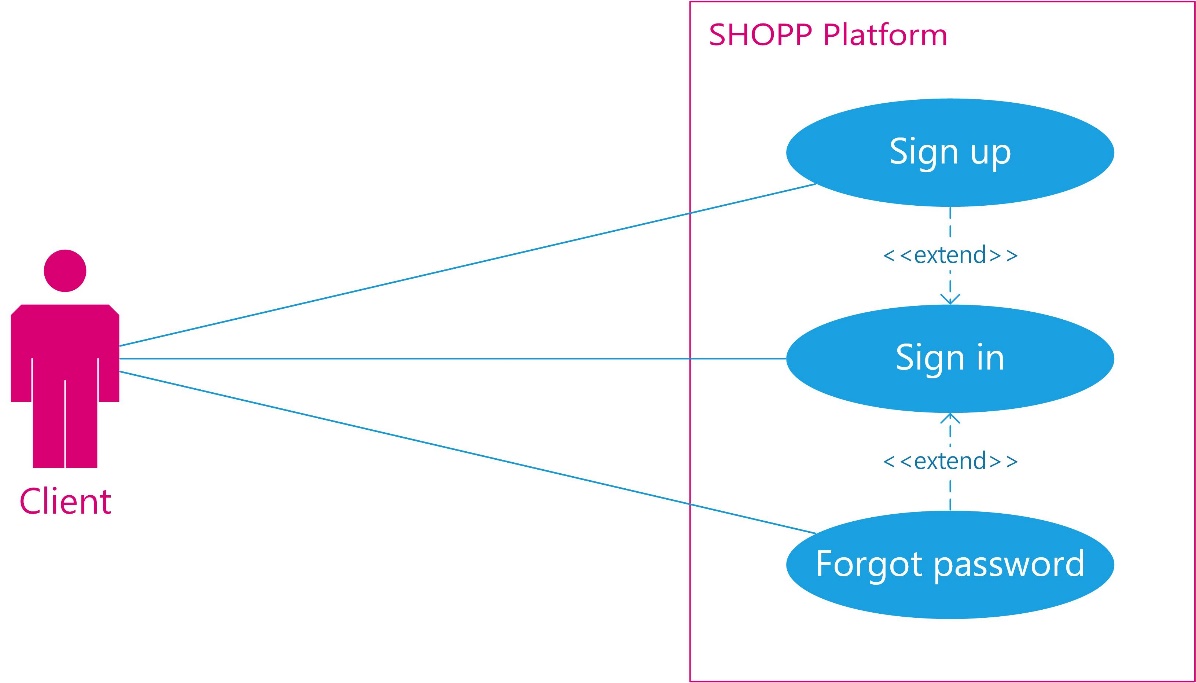
* Spring Boot 2.6.3
* Lombok, Spring Security, ModelMapper
* MySQL Server
* HTML, CSS, Javascript
* Bootstrap
* Jquery
* Ajax
* Stripe API
* GHN API

**TASK SHEET REVIEW 1**

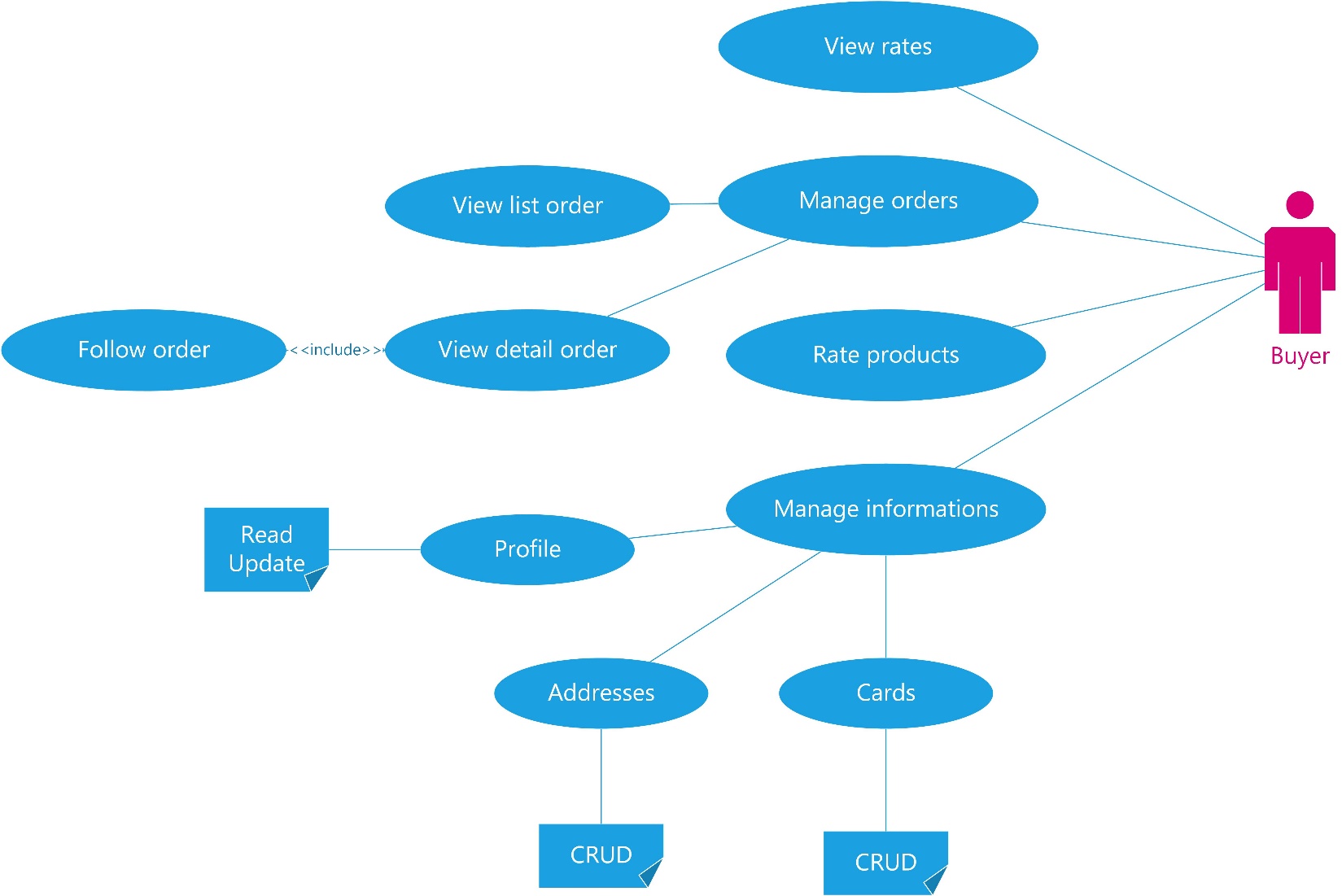
**REVIEW 2**

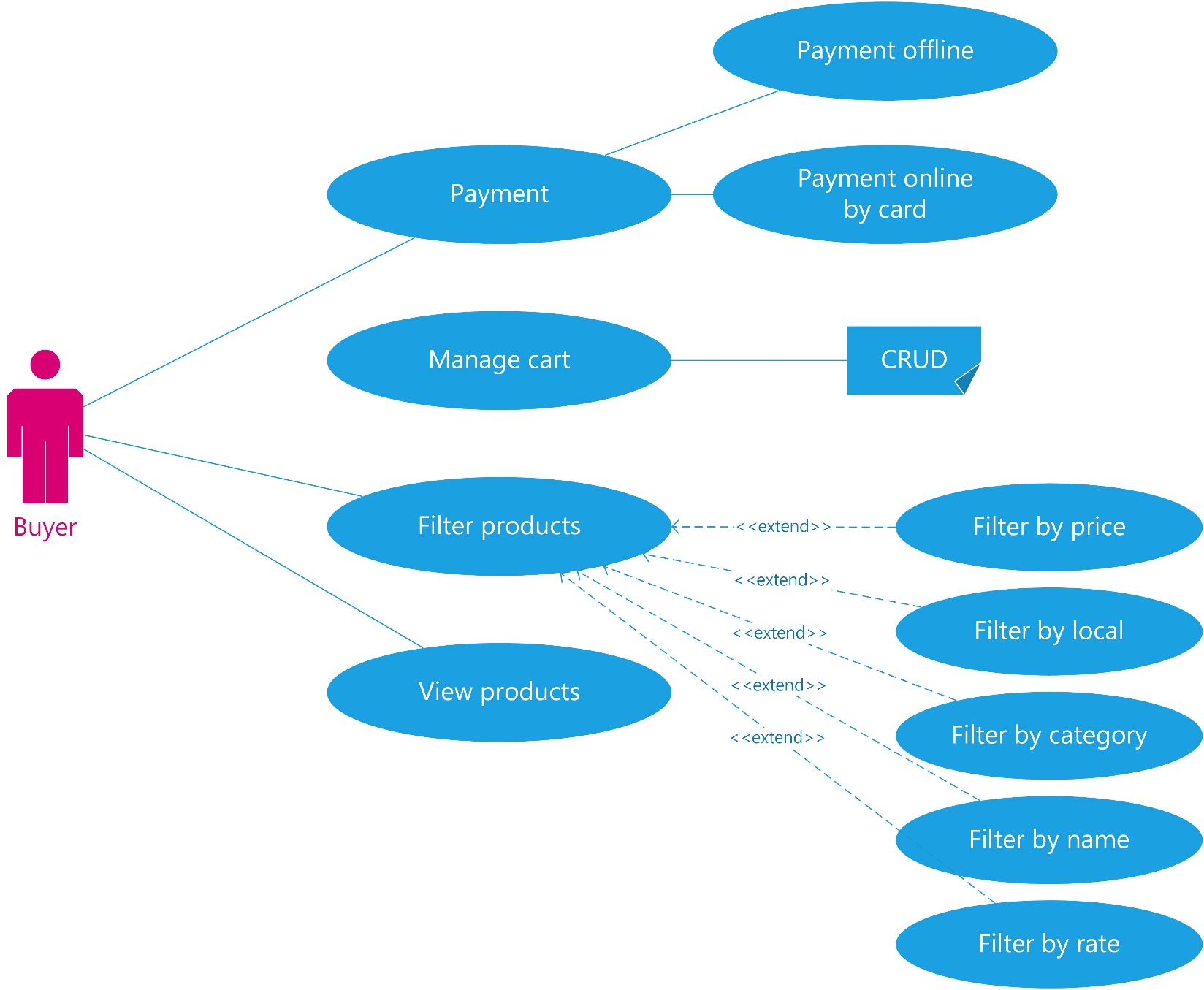
**I. Use Case Diagram**

**1. Client**

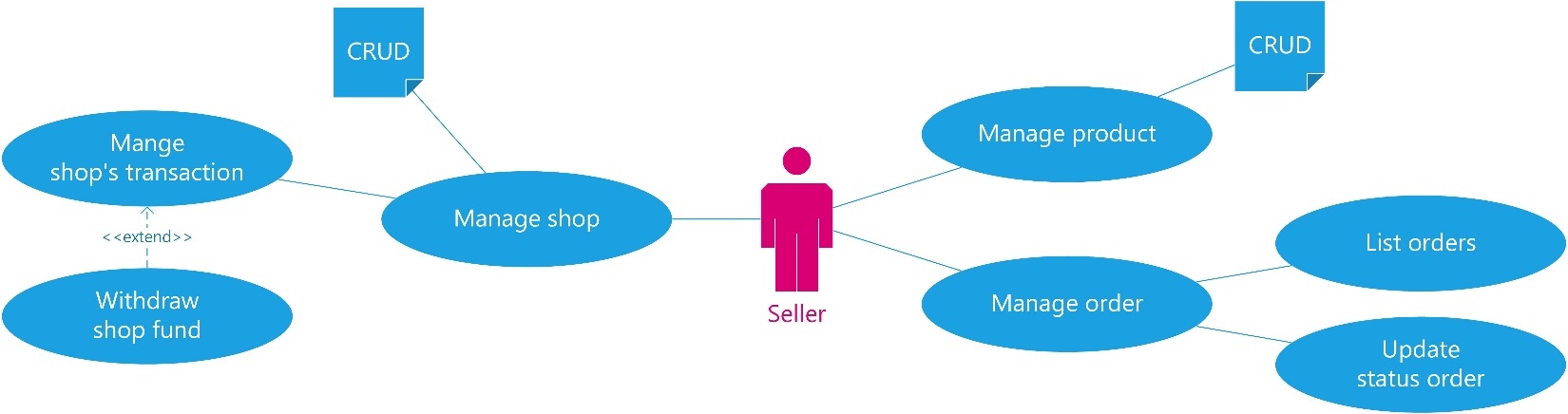
****

**2. Buyer**

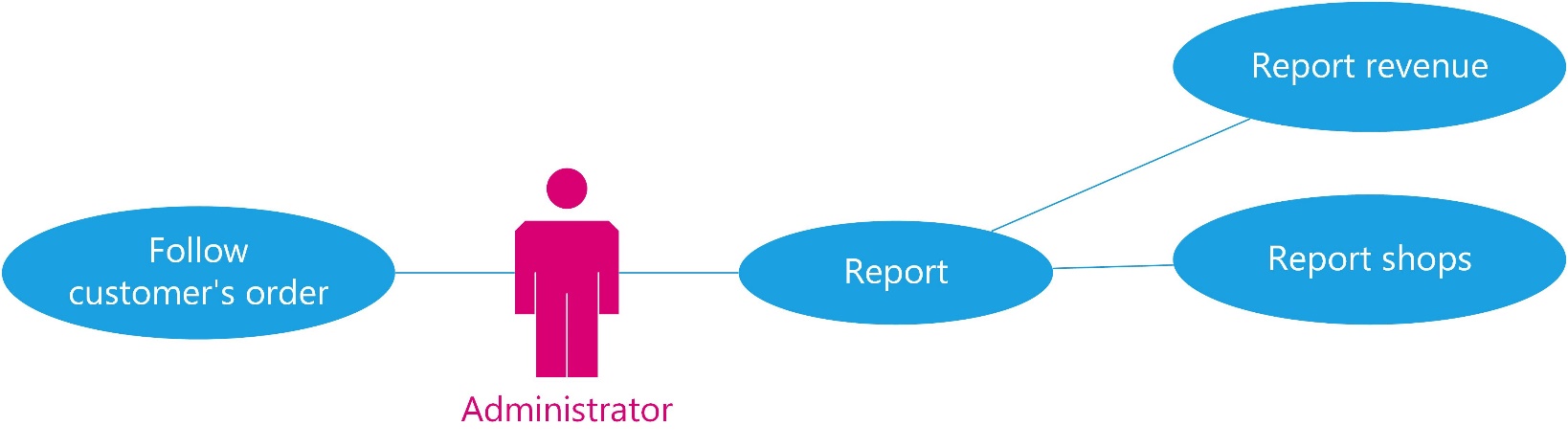
****

****

**3. Seller**

****

**4. Admin**

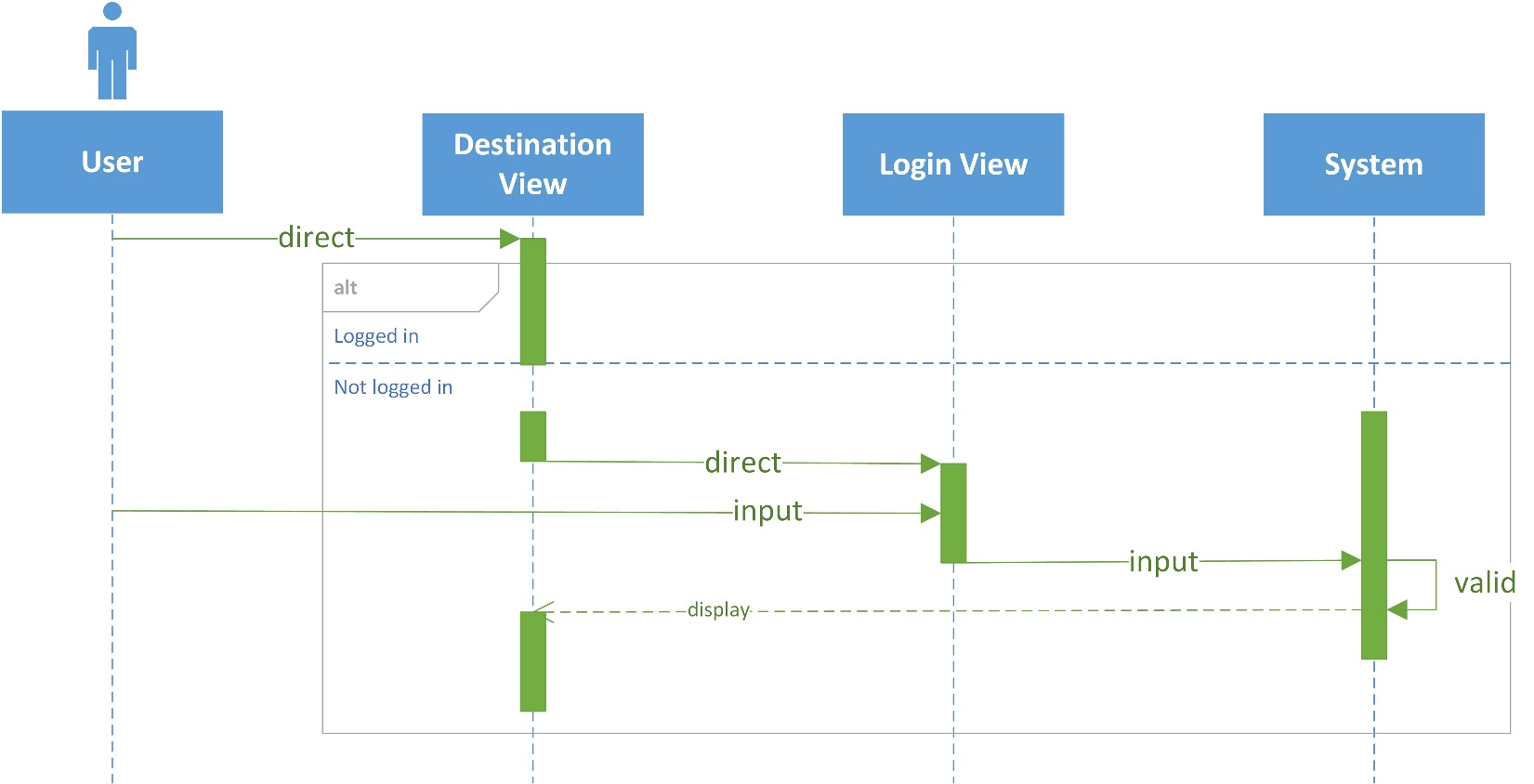
****

**II.Sequence Diagram**

**1. Client**

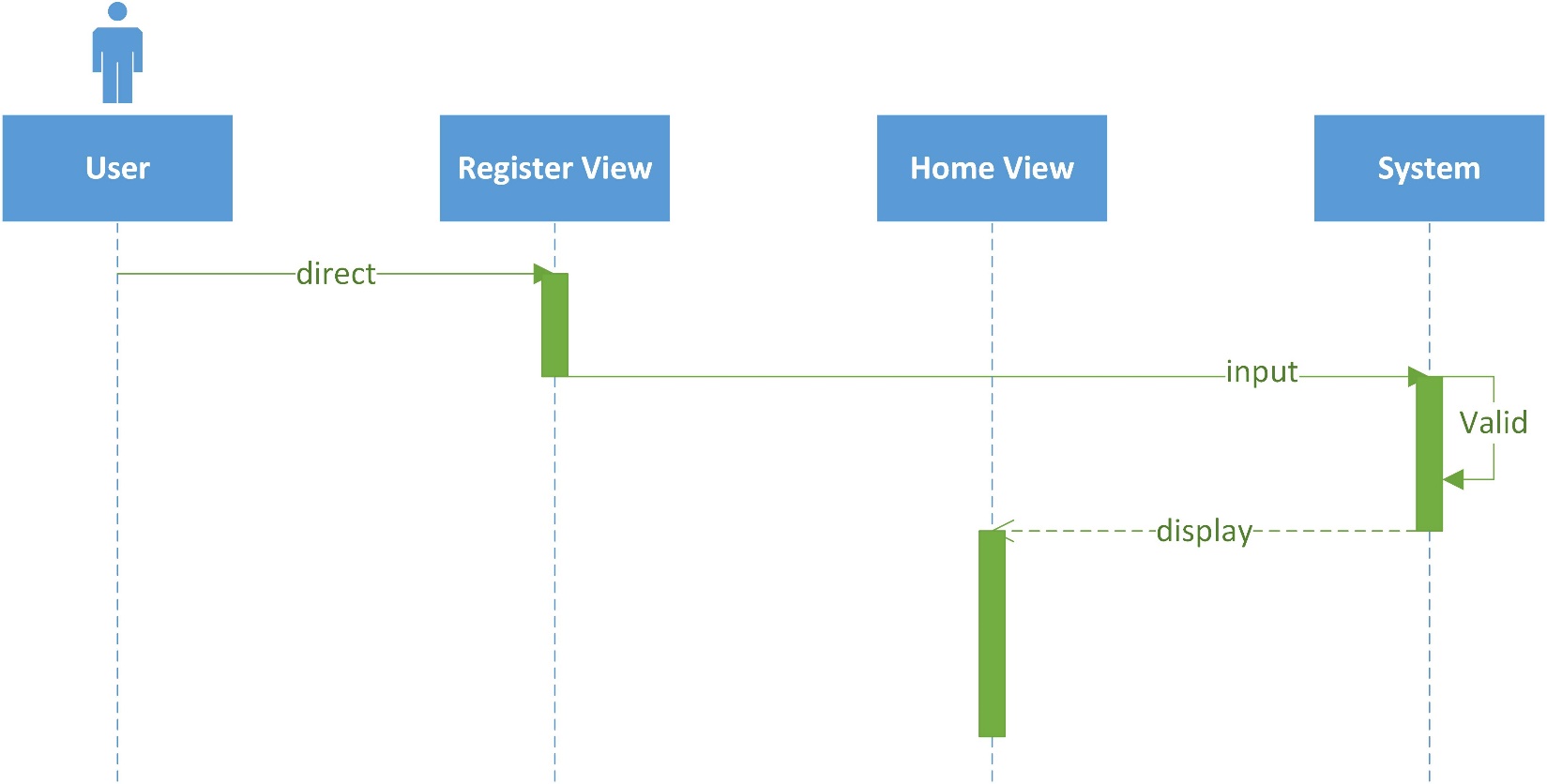
**1. Sign in**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Minh Duy | |
| **Diagram Name** | User Sign in | |
| **Actors** | Guests | |
| **Description** | Display Login pages and after login the system allows user perission function of platfrom | |
| **Requirements** | Request the system have user date. | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor sent request to the system. | 2. The system information processing user.  3. the system return information user if sign in success or return message if login fail |
| **Alternative Flow** |  |  |

****

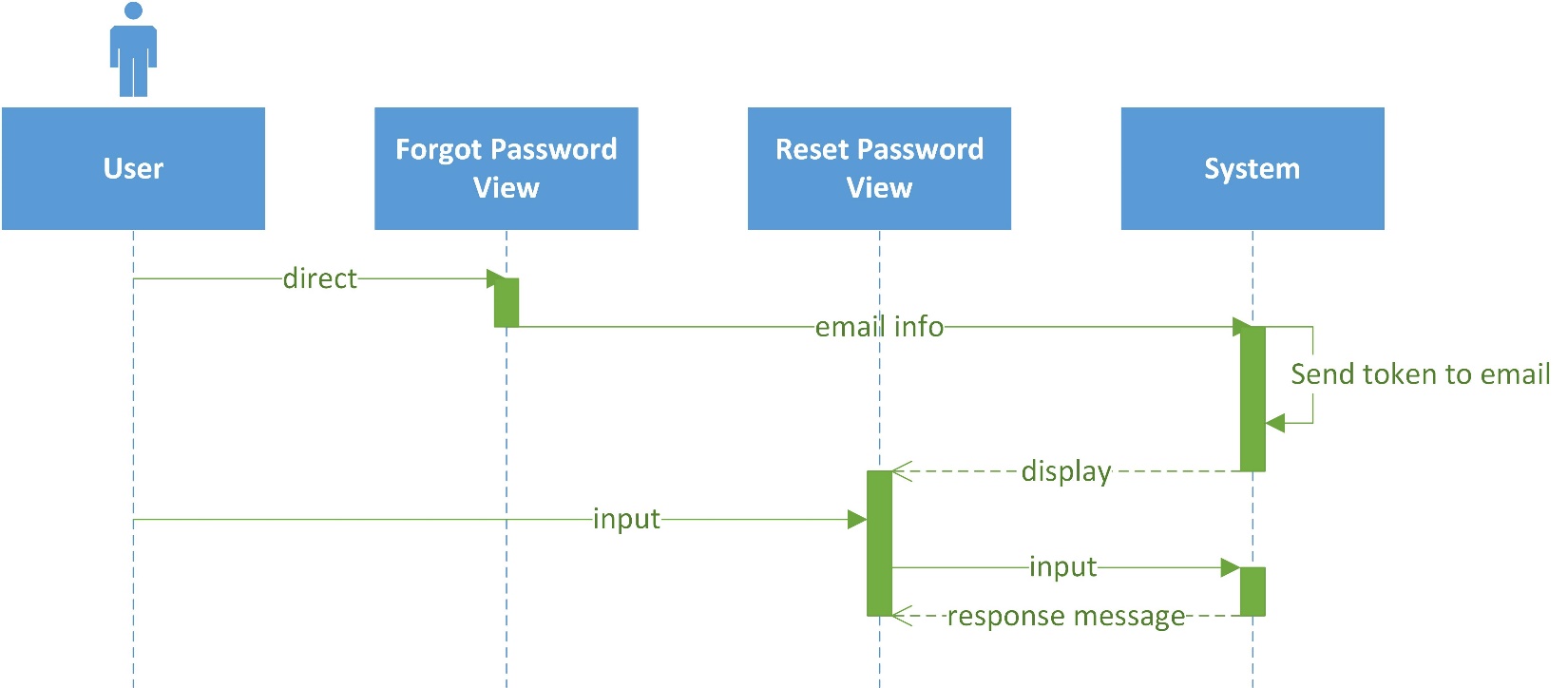
**2. Sign up**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Minh Duy | |
| **Diagram Name** | Display Sign up | |
| **Actors** | Gusest | |
| **Description** | Display Sign in Pages and allow guests user. | |
| **Requirements** |  | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor sent request to the system.  3. Actor input profile and submit profile. | 2. the system return sign up page.  4. The system information processing user.  5. Return status after processing. |
| **Alternative Flow** |  |  |

****

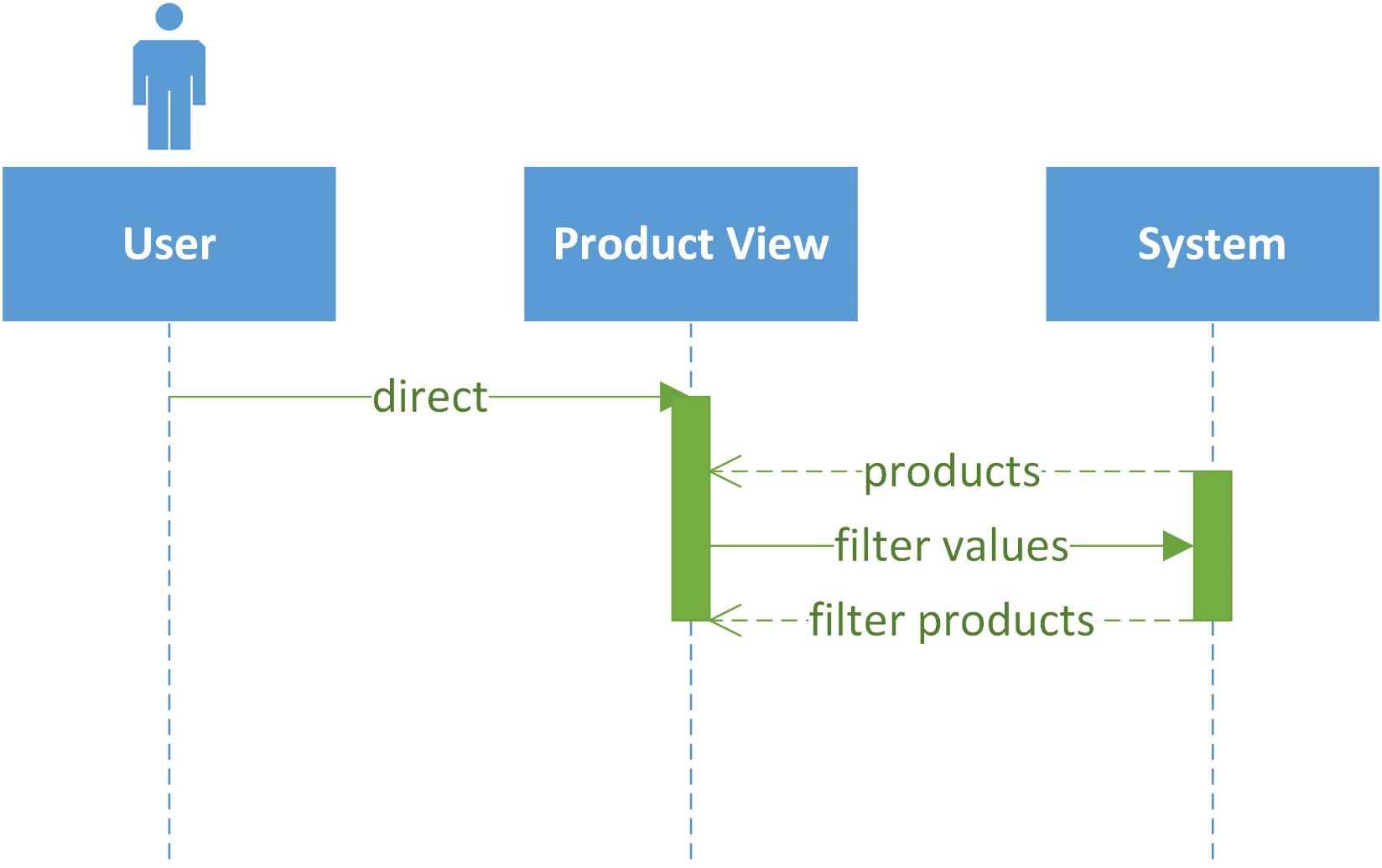
**3. Reset password**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Minh Duy | |
| **Diagram Name** | Reset password | |
| **Actors** | User | |
| **Description** | Show Reset Password pages and sent mail to user to update password. | |
| **Requirements** | Request the system have user date. | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor sent request to the system.  3. actor input password and sent new password | 2. the system return reset pasword page.  4. The system information processing user.  5. Return status after processing. |
| **Alternative Flow** |  |  |

****

**4. Search products**

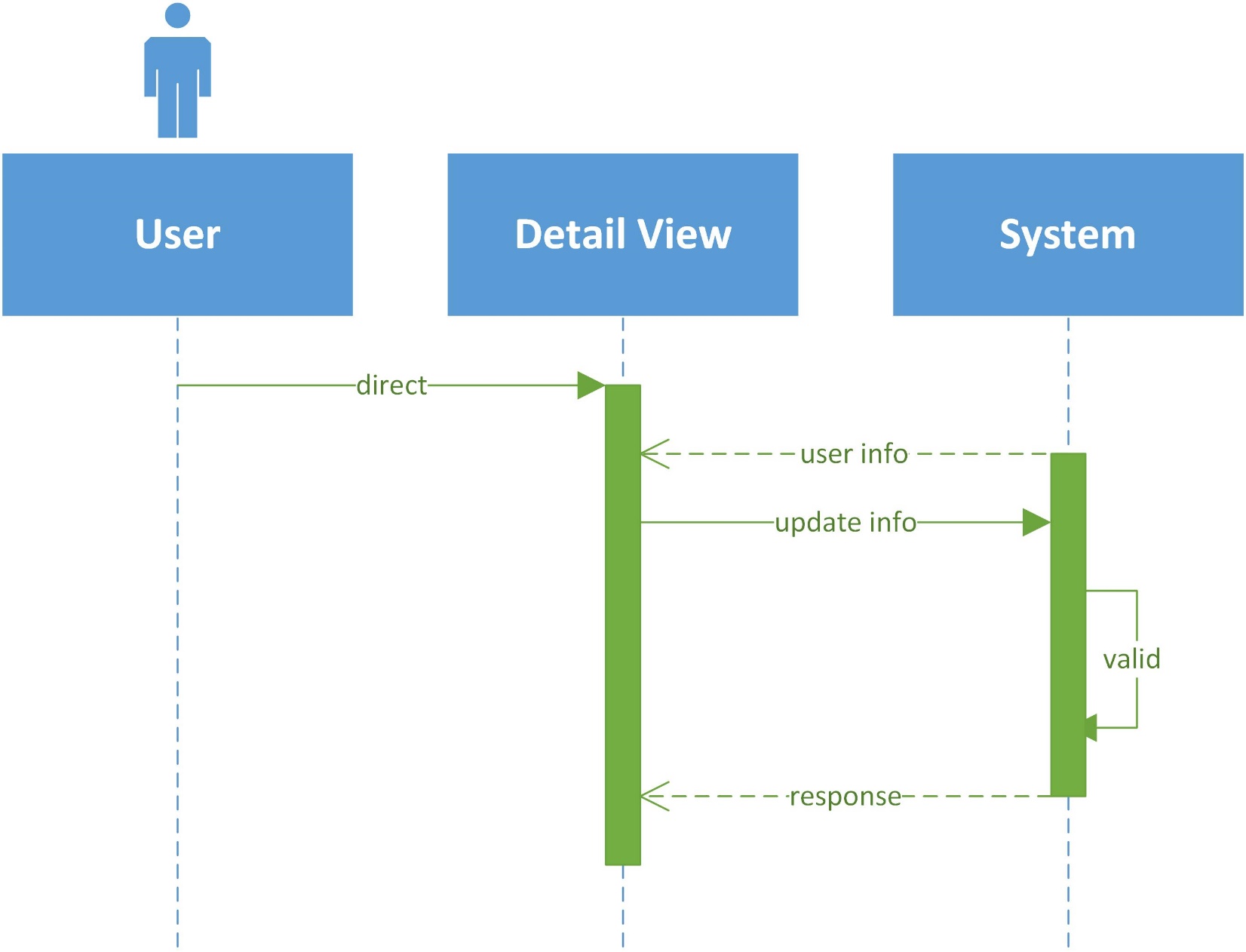
|  |  |  |
| --- | --- | --- |
| **Author** | Le Hong Son | |
| **Diagram Name** | Search products | |
| **Actors** | Guests & Customer | |
| **Description** | Display products which have names contain values in search field | |
| **Requirements** | Search result appear in the panel right below to search field, which content name & product’s image | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor direct to index product page  4. Actor input value filter and send request to system | 2. System return index product page  3. System show default products  5. System handle request, process and return list of products filtered to actor |
| **Alternative Flow** |  |  |

****

**2. User**

**1. Update profile**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Minh Duy | |
| **Diagram Name** | Update Account | |
| **Actors** | User | |
| **Description** | Show Update Account pages and update new profile. | |
| **Requirements** | Request the system have user date. | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor sent request to the system.  3. actor input new profile | 2. the system return update account page.  4. The system information processing user.  5. Return status after processing. |
| **Alternative Flow** |  |  |

****

**2. Update Address**

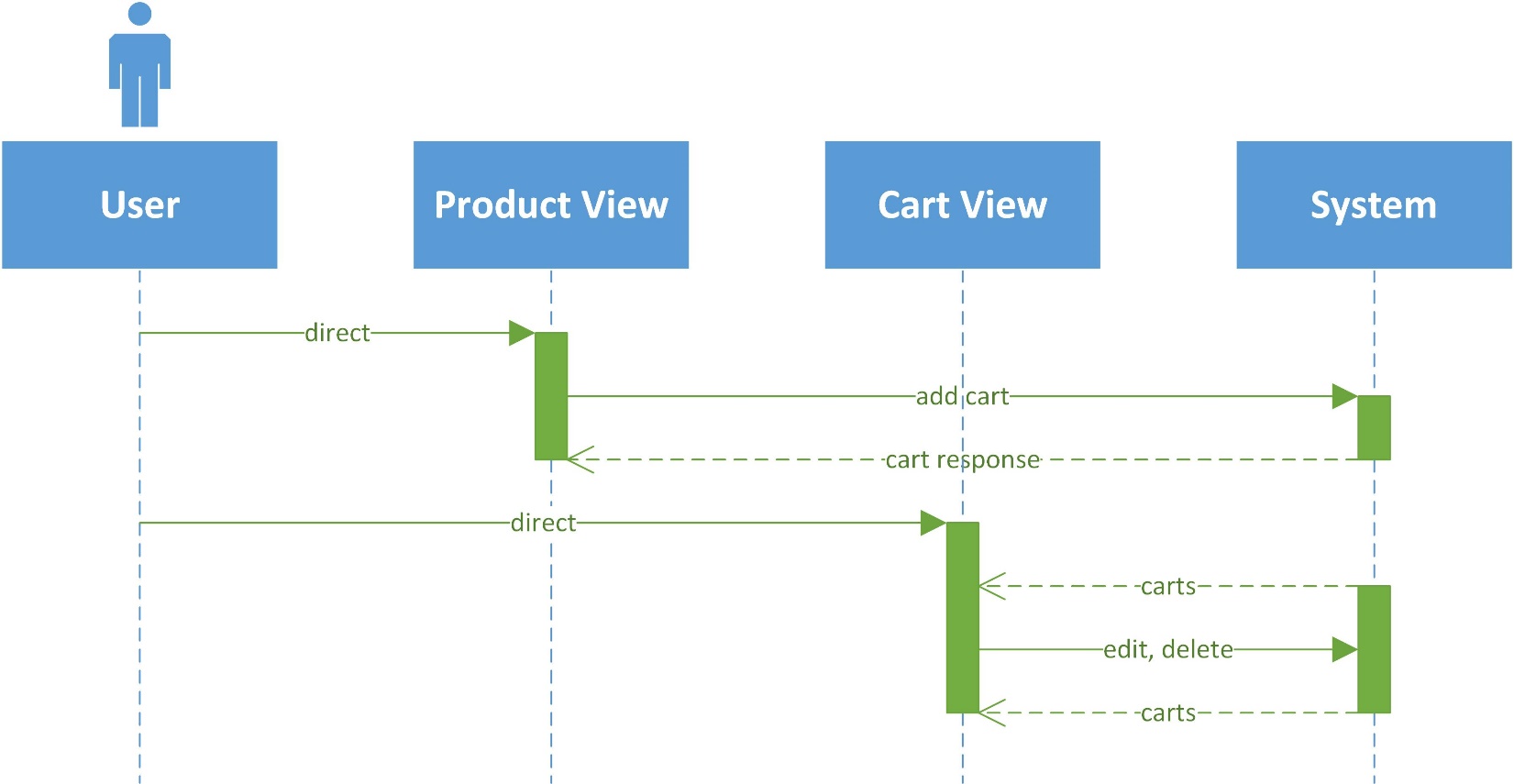
Same as above (2.1 Update profiles)

**3. Update Card**

Same as above (2.1 Update profiles)

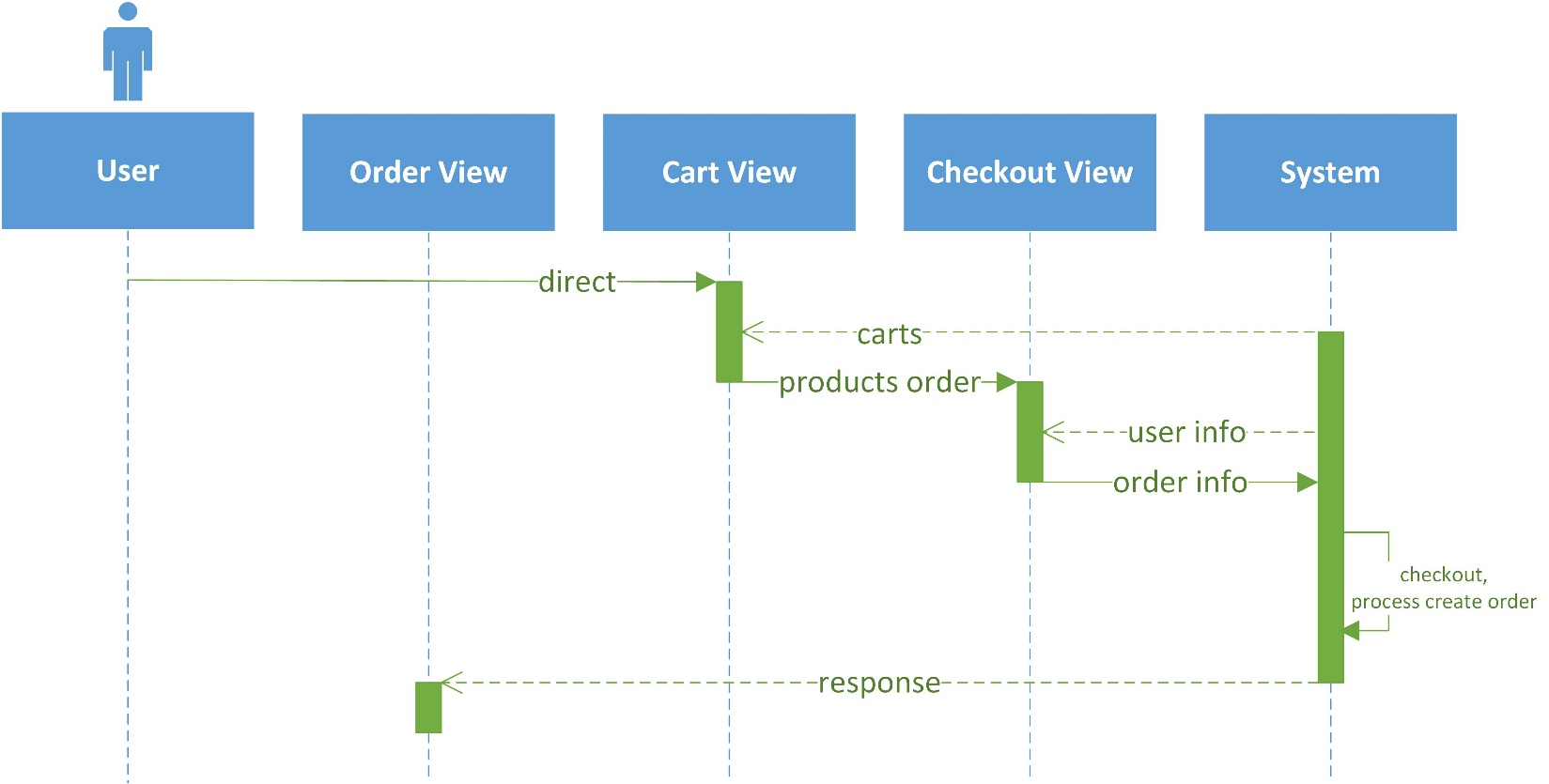
**4. Manage cart**

|  |  |  |
| --- | --- | --- |
| **Author** | Le Hong Son | |
| **Diagram Name** | Manage cart | |
| **Actors** | Customer | |
| **Description** | Customer add to cart before checkout and edit | |
| **Requirements** | Customer must be select product first | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor filter product and choose product which actor want to buy  3. Actor select quantity and click add to cart  5. Actor redirect to cart index page  6. Actor can edit or delete cart | 2. System redirect to product detail page  4. System add product to cart of actor  7. System process request of actor  8. System return carts to view of actor |
| **Alternative Flow** |  |  |

****

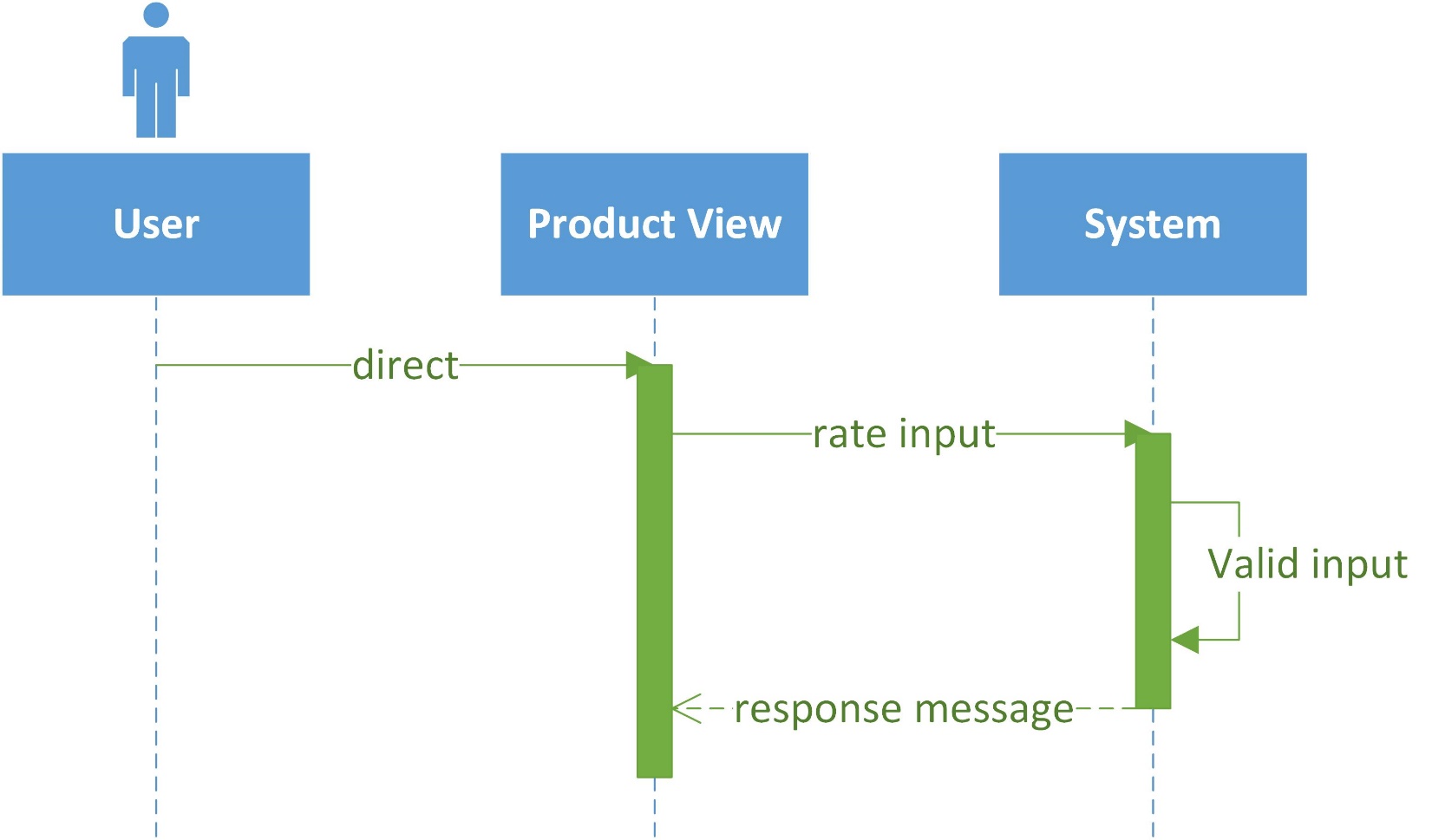
**5. Payment**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Phu | |
| **Diagram Name** | Checkout | |
| **Actors** | Customer | |
| **Description** | Customer buy product and checkout | |
| **Requirements** | Customer must has card info in their account and has products in their cart | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor direct to cart page  3. Actor choose product to make order  5. Actor confirm order and agree to payment, checkout | 2. System return carts to view  4. System return actor’s info, and create order info back to view of actor  6. System valid and accept payment of actor  7. System create order and return order manage view to actor |
| **Alternative Flow** | Payment offline  1. Actor choose ship COD | Payment offline  2. System valid and accept payment of actor  3. System create order and return order manage view to actor |

****

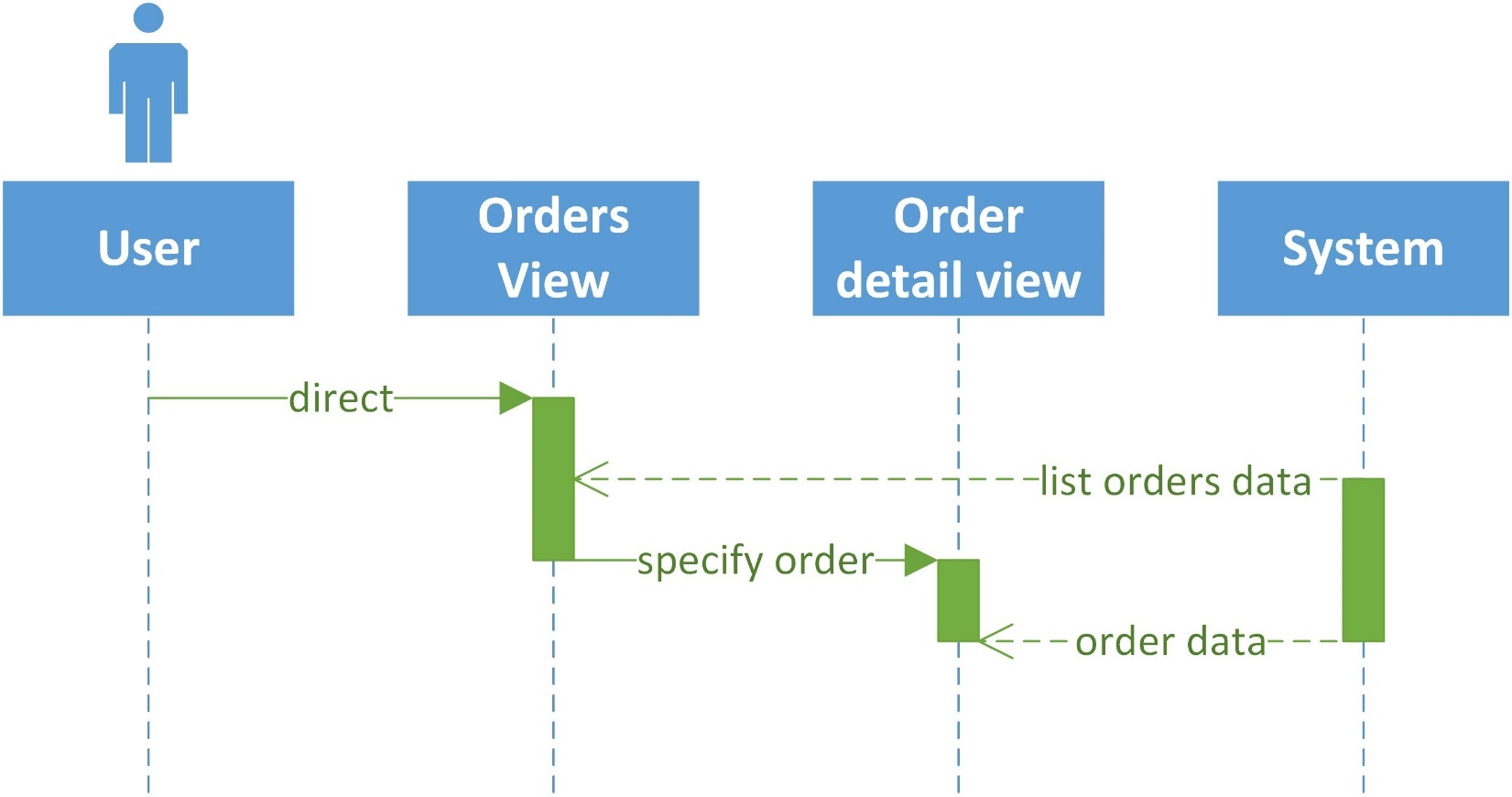
**6. View and Rate product**

|  |  |  |
| --- | --- | --- |
| **Author** | Le Hong Son | |
| **Diagram Name** | Rate | |
| **Actors** | Customer | |
| **Description** | Customer review product then the others can refer and decided to buy product | |
| **Requirements** | Customer must buy product before | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. After buy product, actor direct to product detail view  3. Actor write comment and up picture of product to rate | 2. System return product detail with all other actor’s rates  4. System receive request and save review info to this product  5. System return response message to product detail view |
| **Alternative Flow** |  |  |

****

**7. Orders management**

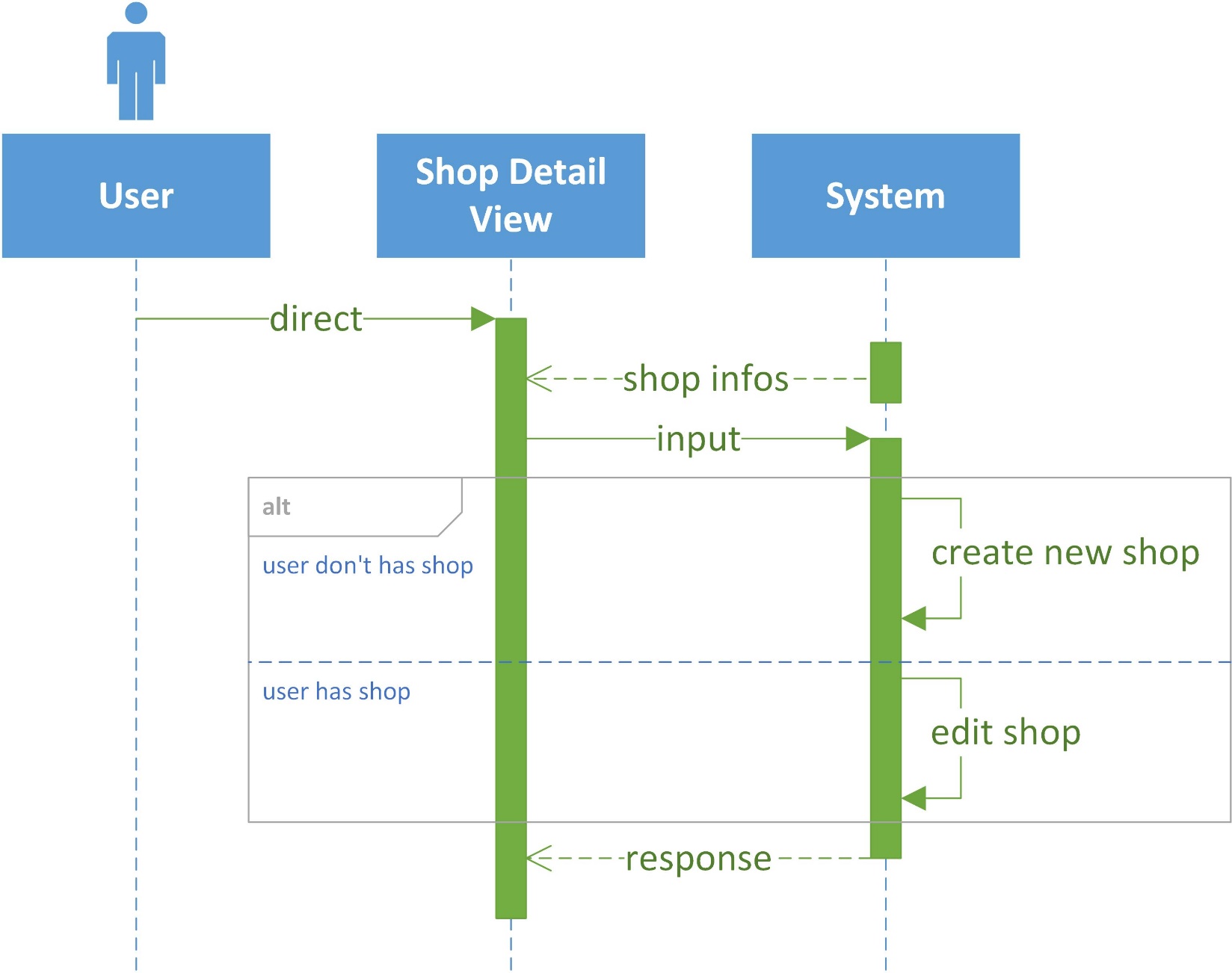
|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Phu | |
| **Diagram Name** | Orders Management | |
| **Actors** | Customer | |
| **Description** | Check details and status of orders | |
| **Requirements** | Customer must have at least one order | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor direct to orders management view  3. Actor choose order their want to follow | 2. System return orders data  4. System return specify order data to actor. |
| **Alternative Flow** |  |  |

****

**3. Seller**

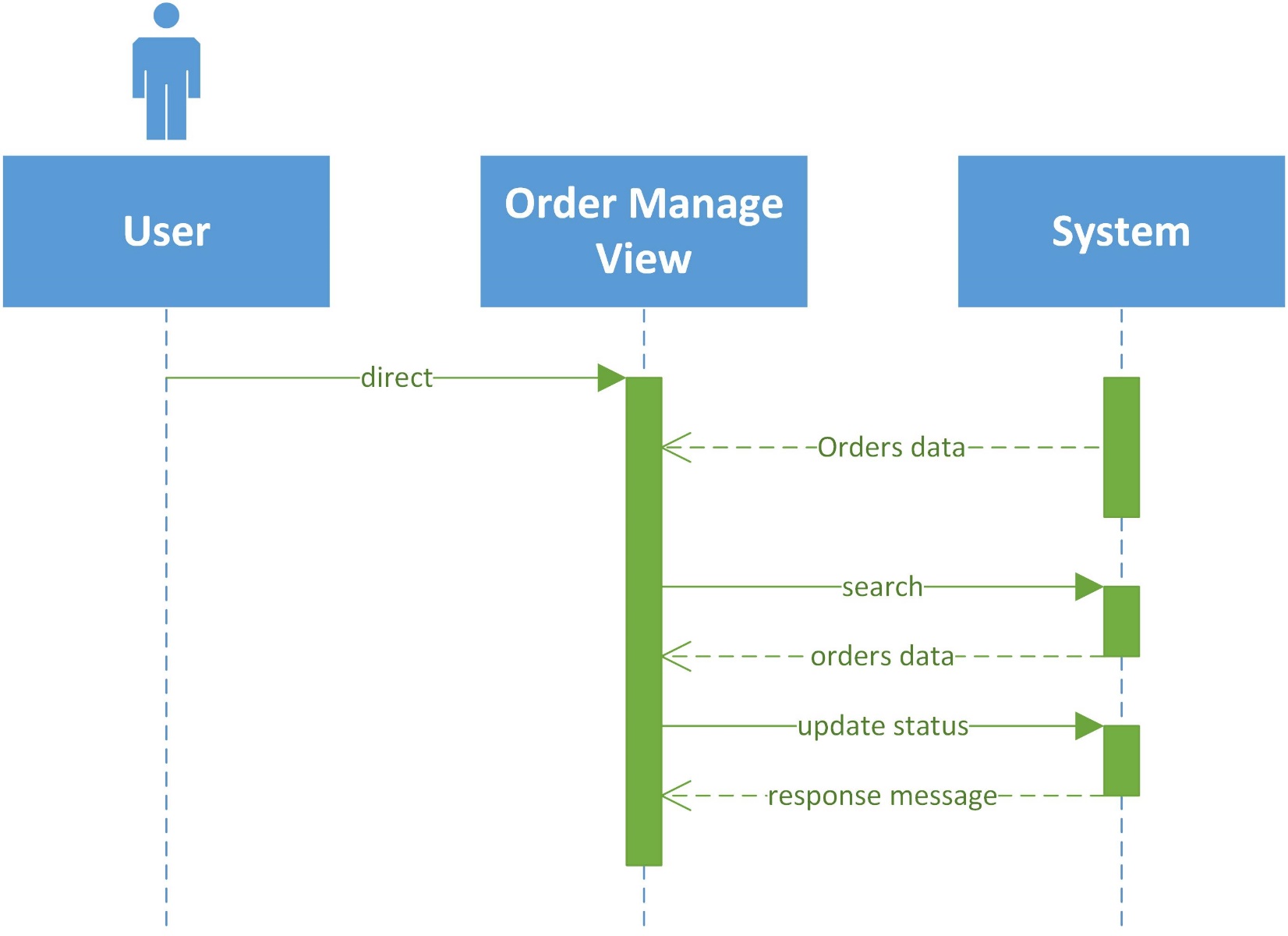
**1. Create, update info**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Phu | |
| **Diagram Name** | Shop | |
| **Actors** | Shop owner | |
| **Description** | Customer want to post product for sale must be register their shop first | |
| **Requirements** | If update info, Shop owner must have their shop first | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor direct to portal manage of their shop  3. Actor edit shop info and submit | 2. System return their shop info  4. System receive request and update their shop  5. System return message response to actor |
| **Alternative Flow** | **Actor don’t have their shop** | |
| 1. Actor direct to portal to manage their shop  3. Actor fill the info and submit | 2. System return form for actor to register new shop  4. System valid shop info, confirm and create shop for actor  5. After create shop success, system return message response to actor |

****

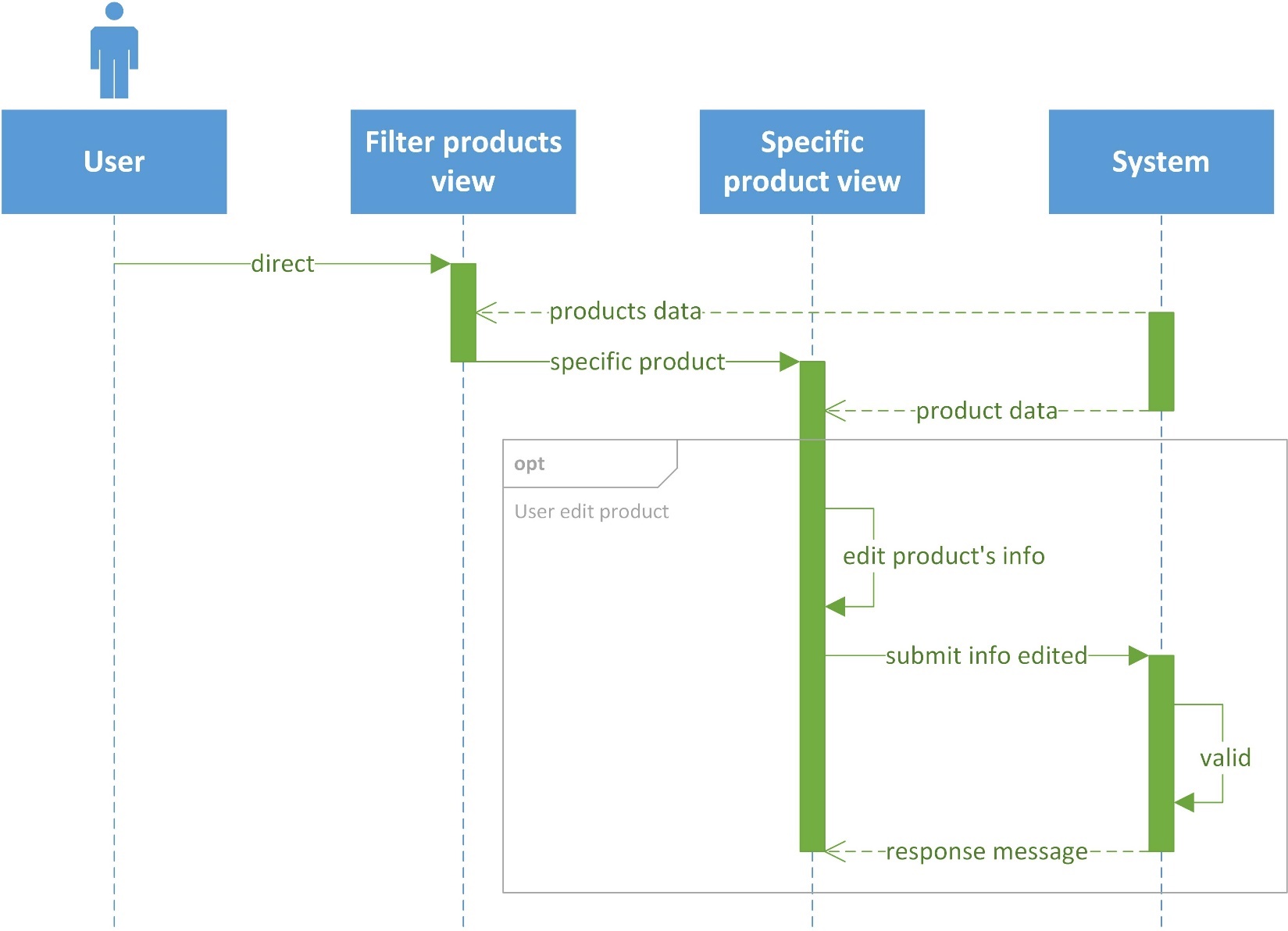
**2. Orders management**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Phu | |
| **Diagram Name** | Orders Management | |
| **Actors** | Shop owner | |
| **Description** | Check amount, status, customer of orders | |
| **Requirements** | Shop owner must have shop and orders buy by customer, | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor direct to portal orders management view  3. Actor can filter, search order their want to access  5. Actor update new status to specific order | 2. System return orders data  4. System return orders filtered to actor  6. System receive request and process to change status of specific order  7. System return response message to actor |
| **Alternative Flow** |  |  |

****

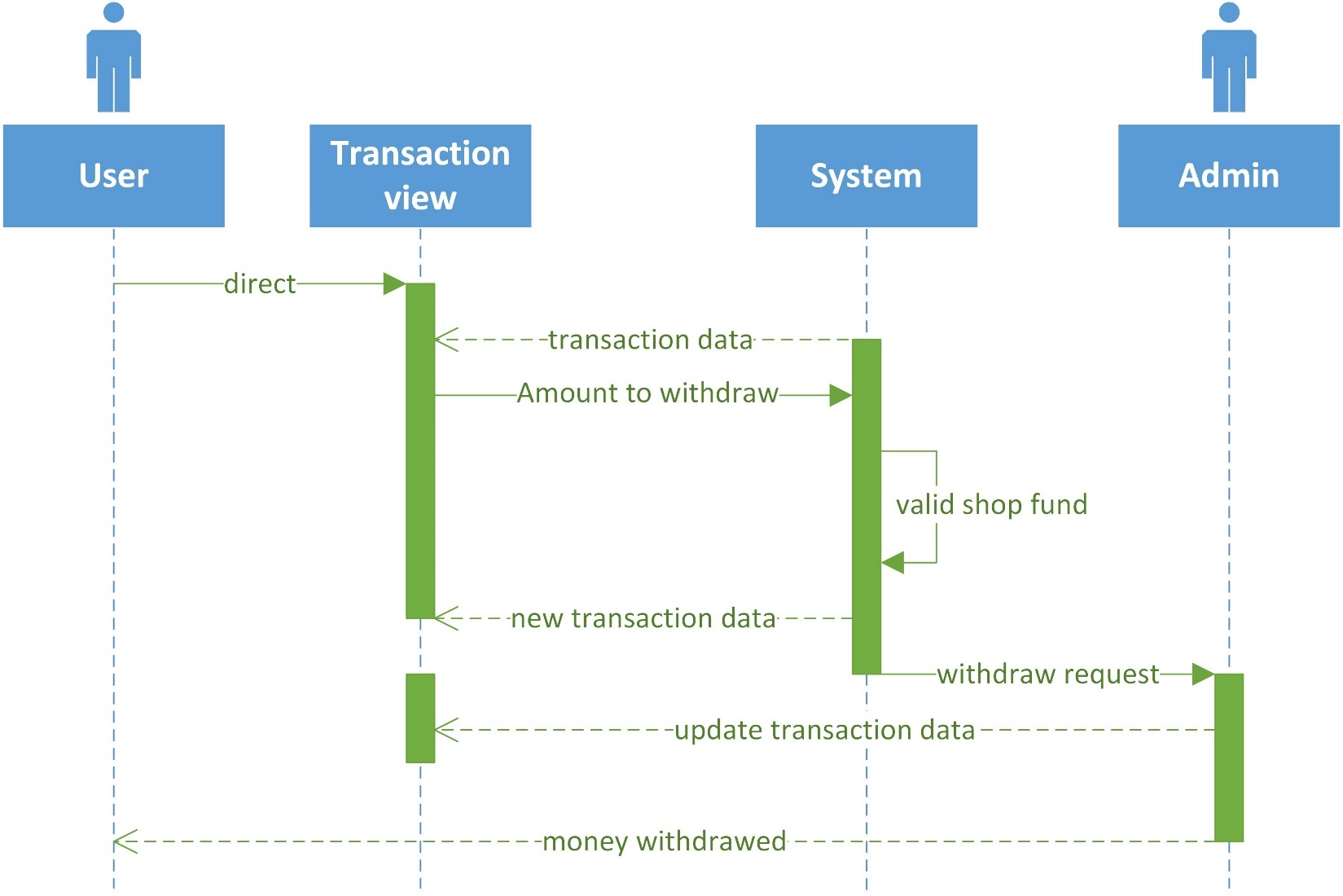
**3. Products management**

|  |  |  |
| --- | --- | --- |
| **Author** | Le Hong Son | |
| **Diagram Name** | Products Management | |
| **Actors** | Shop owner | |
| **Description** | CRUD products of shop | |
| **Requirements** |  | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor direct to products view  3. Actor choose product their want to see and edit | 2. System return products to actor  4. System return information view of specific product |
| **Alternative Flow** | - Edit product  1. At specific product view, actor edit product’s information and submit to system  - Create product  1. Actor click ‘Thêm sản phẩm’ button.  2. Actor redirect to create product view  3. Actor fill info of new product and submit to system | - Edit product  2. System valid and update product  3. System return response message to actor  - Create product  4. System valid and create new product  5. System return response message to actor. |

****

**4. Withdraw from shop’s fund**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Phu | |
| **Diagram Name** | Withdraw money | |
| **Actors** | Shop owner | |
| **Description** | Withdraw money from shop’s fund | |
| **Requirements** | Must be money in the shop | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor direct to withdraw view  3. Actor fill in the amount to withdraw and submit | 2. System return fund data to actor  4. System valid and process  5. System return new transaction info to actor  6. System send request withdraw of actor to admin  7. Admin decided transfer money to actor and update transaction data of actor |
| **Alternative Flow** |  |  |

****

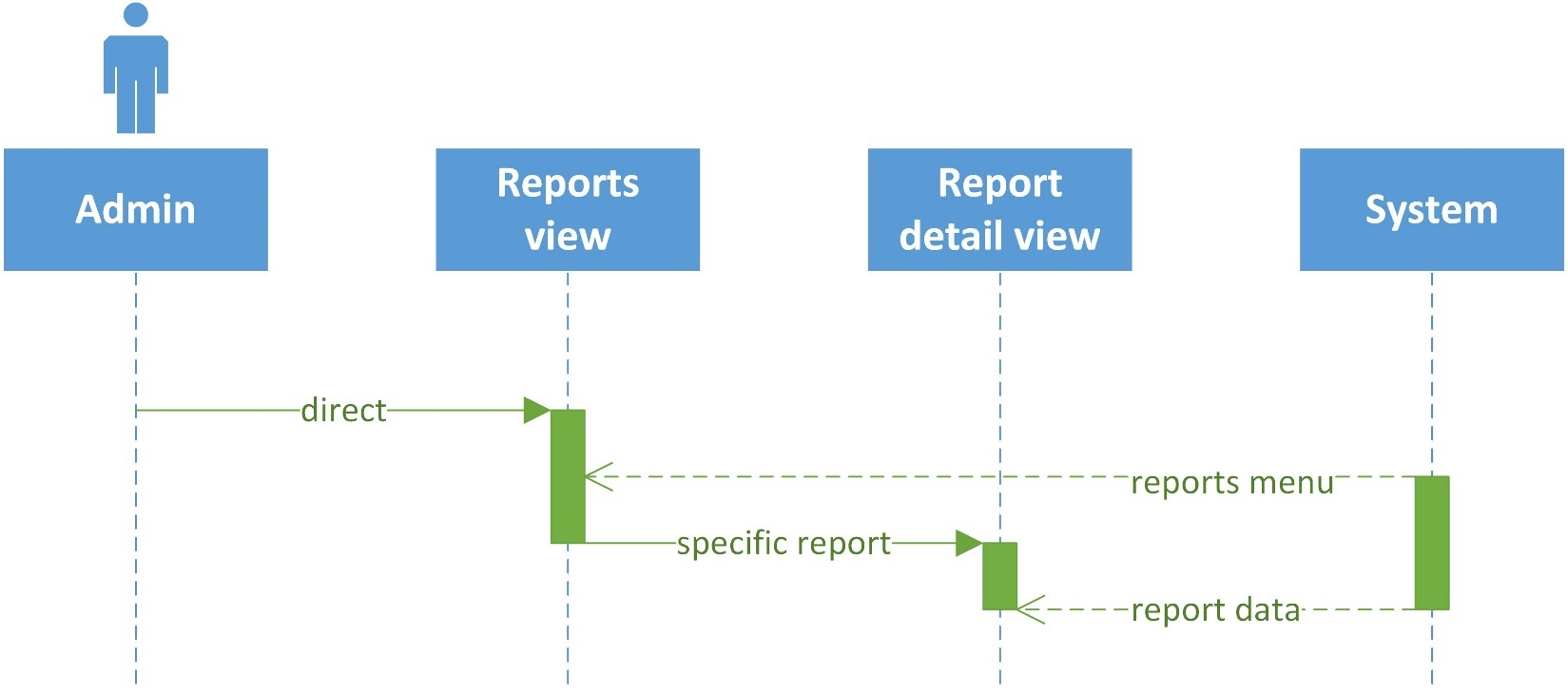
**4. Admin**

**1. Follow order customer**

Same as above (2.7 Order management)

**2. Report**

|  |  |  |
| --- | --- | --- |
| **Author** | Nguyen Hoang Phu | |
| **Diagram Name** | Admin report | |
| **Actors** | Admin | |
| **Description** | Admin view reports and statistics | |
| **Requirements** |  | |
|  | **Actors Action** | **System Response** |
| **Basic Flow** | 1. Actor direct to report pages  2. Actor choose report their want to see | 3. System calculate and return charts, statistics table to actor |
| **Alternative Flow** |  |  |

****

**II. Data Flow Diagram (DFD)**

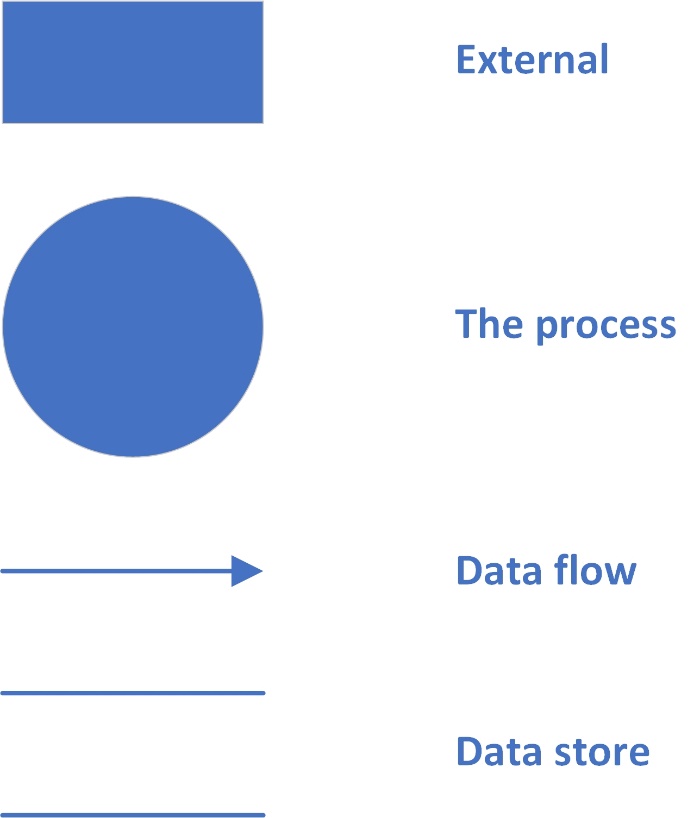
**Define:** Data Flows Diagram (DFD) describes the information flow in the system. The next step of system analysis is to consider in detail the information necessary for the implementation for functions discussed above and the one necessary for the improvement of the functions. Modelling tool frequently used for this purpose is DFD. DFD will support 4 main activities:

**Analysis:** DFD is used to determine requirement of users. Design: DFD is used to map out plan and illustrate solution to analysis and users while designing a new system.

**Communication:** one of the strength of DFD is its simplicity and ease to understand to analysts and users.

**Document:** DFD is used to provide special description of requirement and system design. DFD provide an overview of key functional components of the system but it does not provide any detail on these components. We have to use other tools like database dictionary, process specification to get an idea of which information will be exchanged and how.

The main components of Context Diagram:

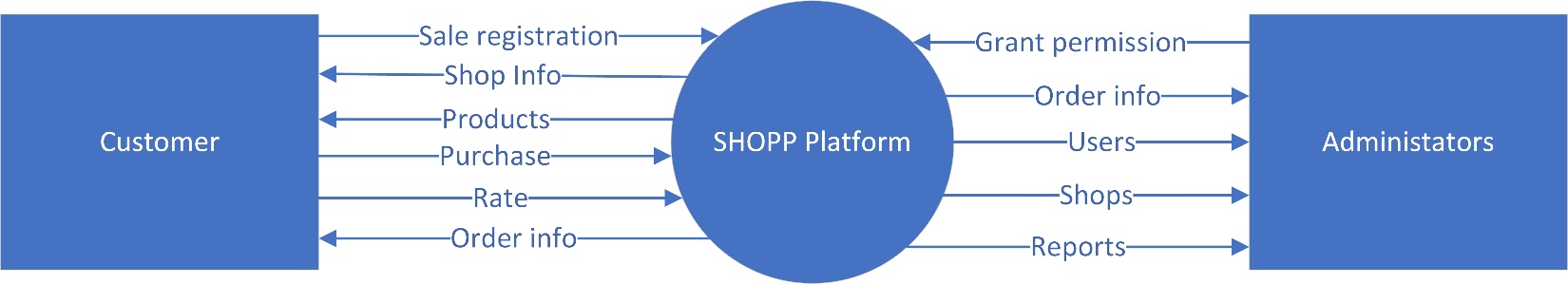
**The external factors**: External factors can be a person, a group of persons or an organization that are sources of information for the systems and are where system products are transferred to.

**The process**: Shows the common function of system

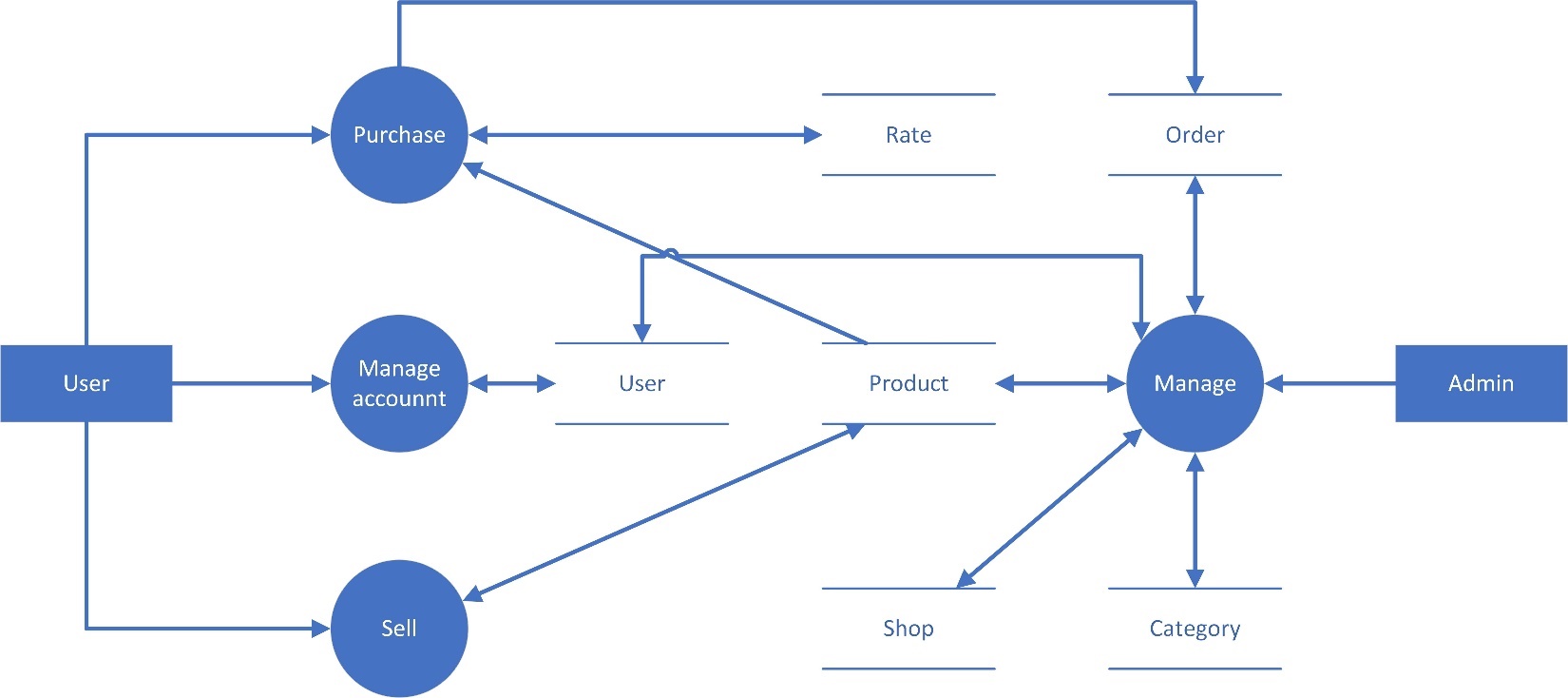
**The data flow**: Describe the movement of information from one part of the system to another.

**The data store**: The Data Store is used to model a collection of data packets at rest. A store is represented graphically by two parallel lines. The name of a Data Store that identifies the store is the plural of the name of the packets that are carried by flows into and out of the Data Store

**1. Context Level Diagram**

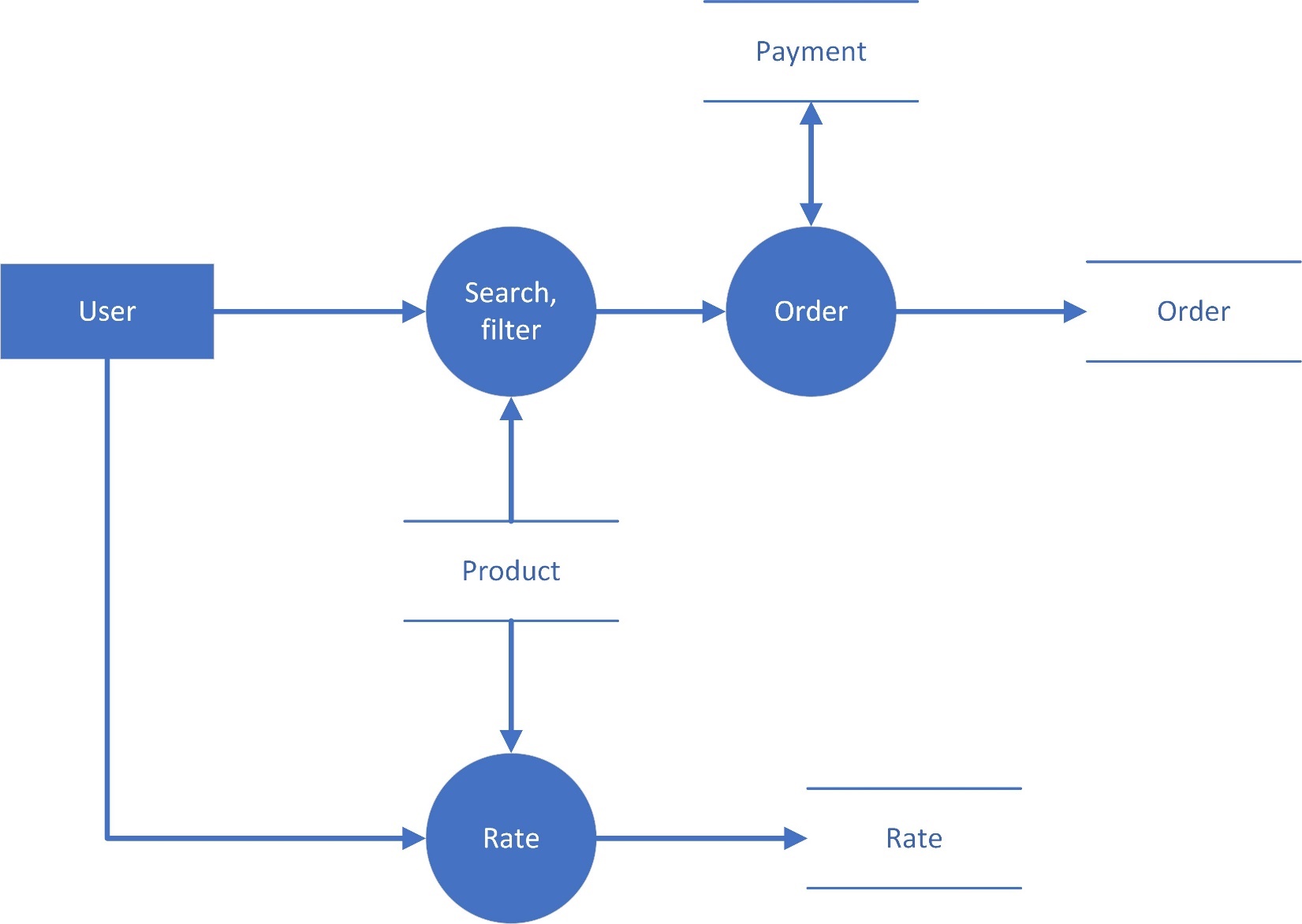


**2. Level 0 DFD**

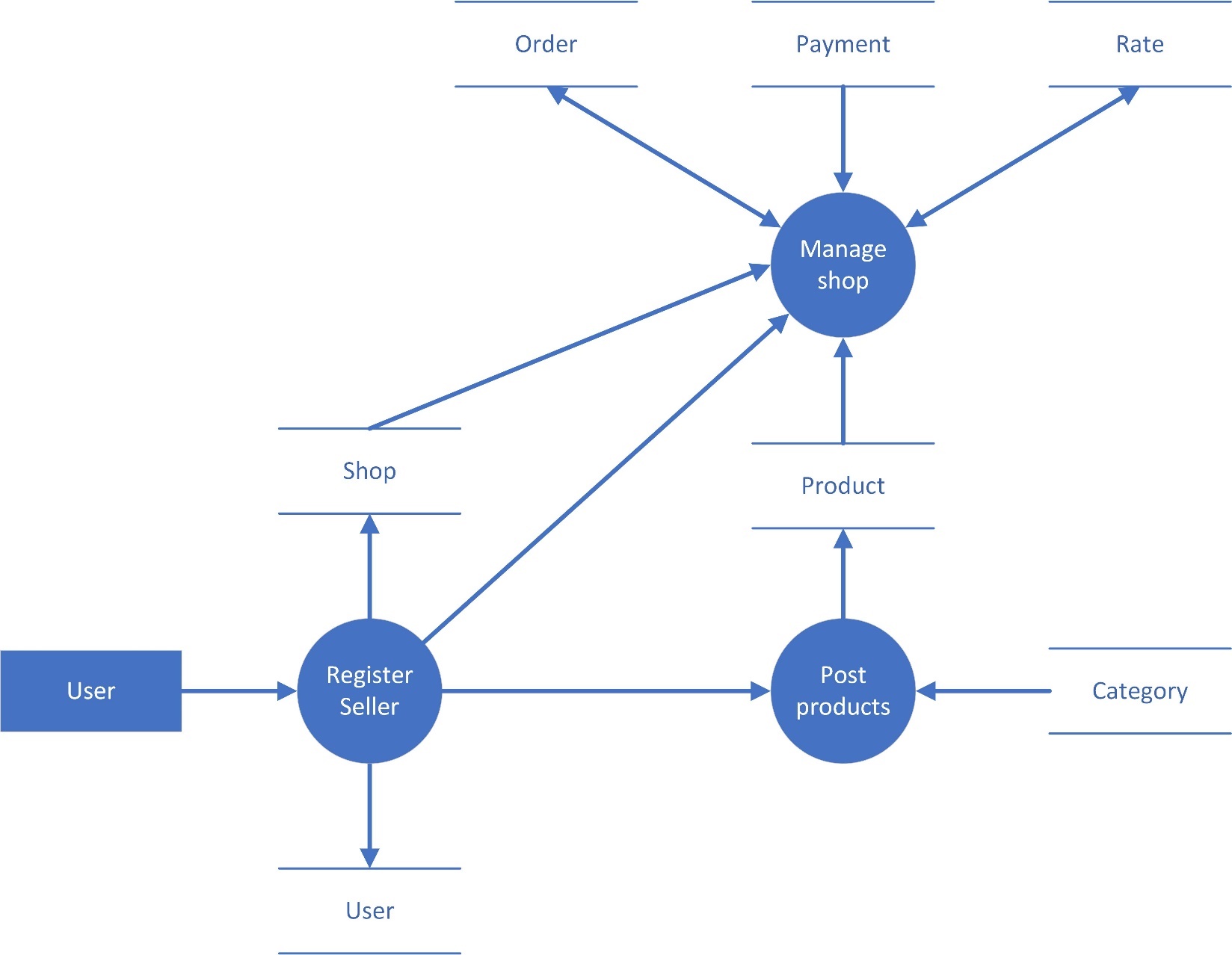
****

**3. Level 1 DFD**

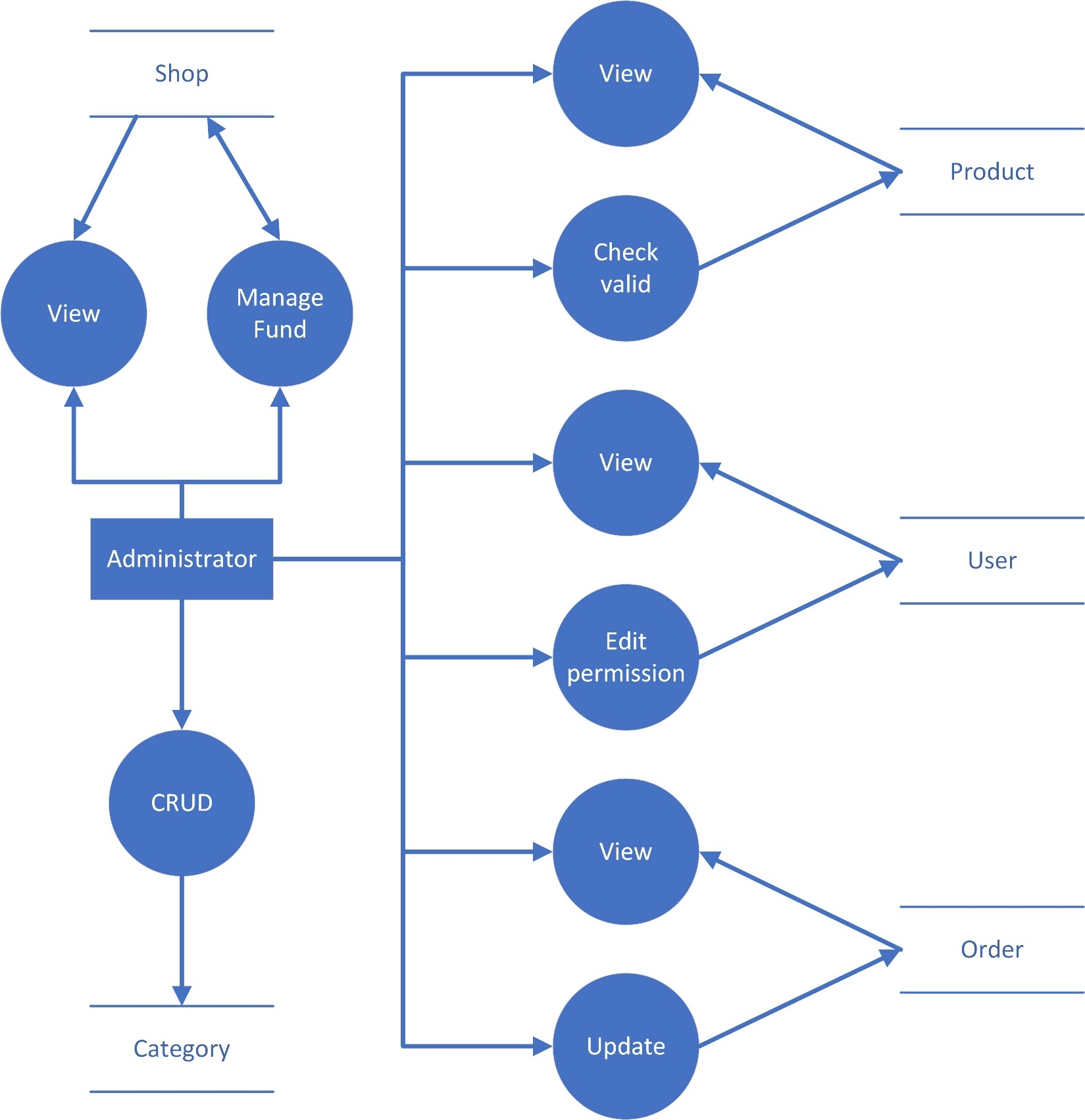
**3.1 Purchase**

****

**3.2 Sell**

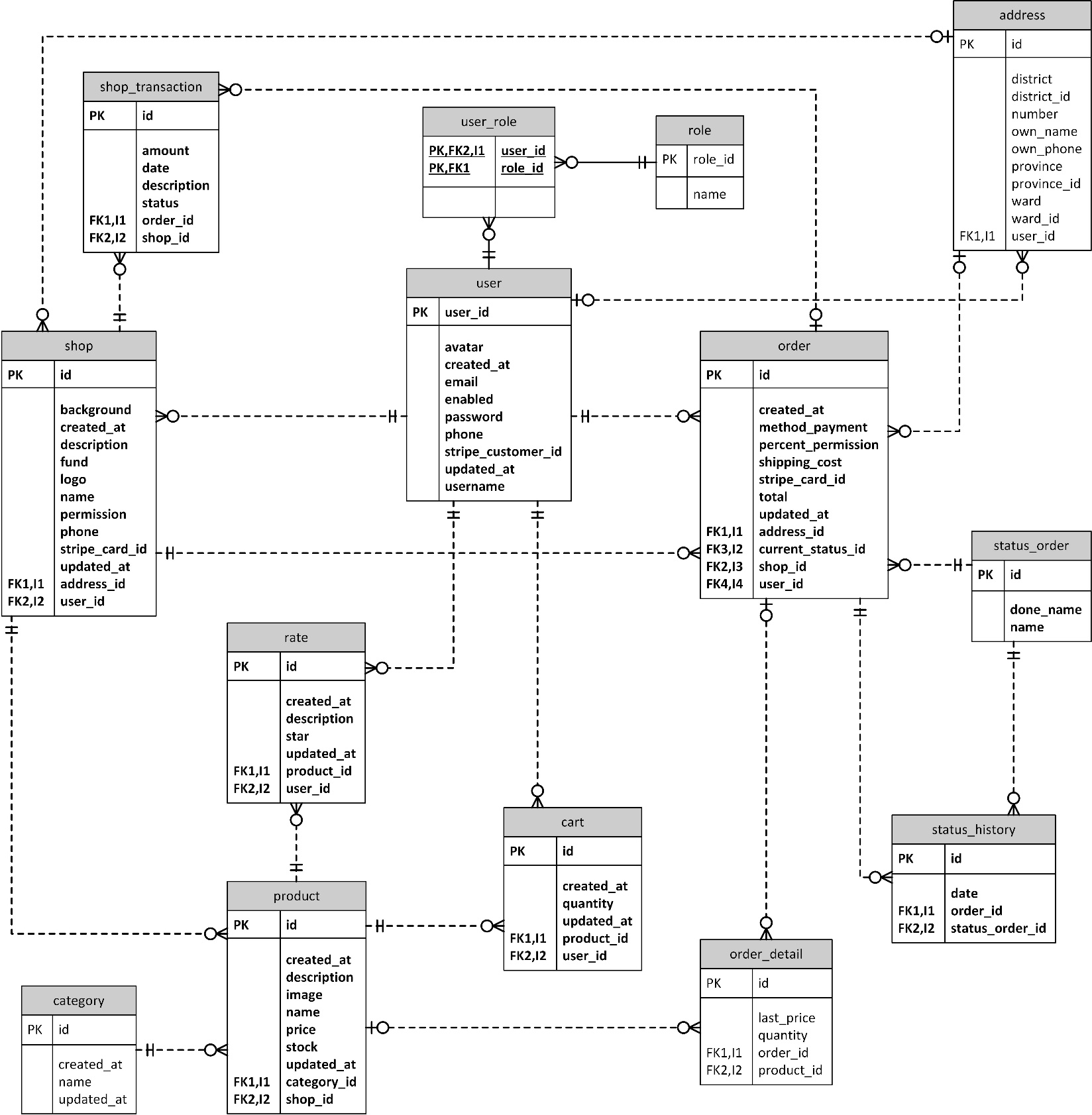
****

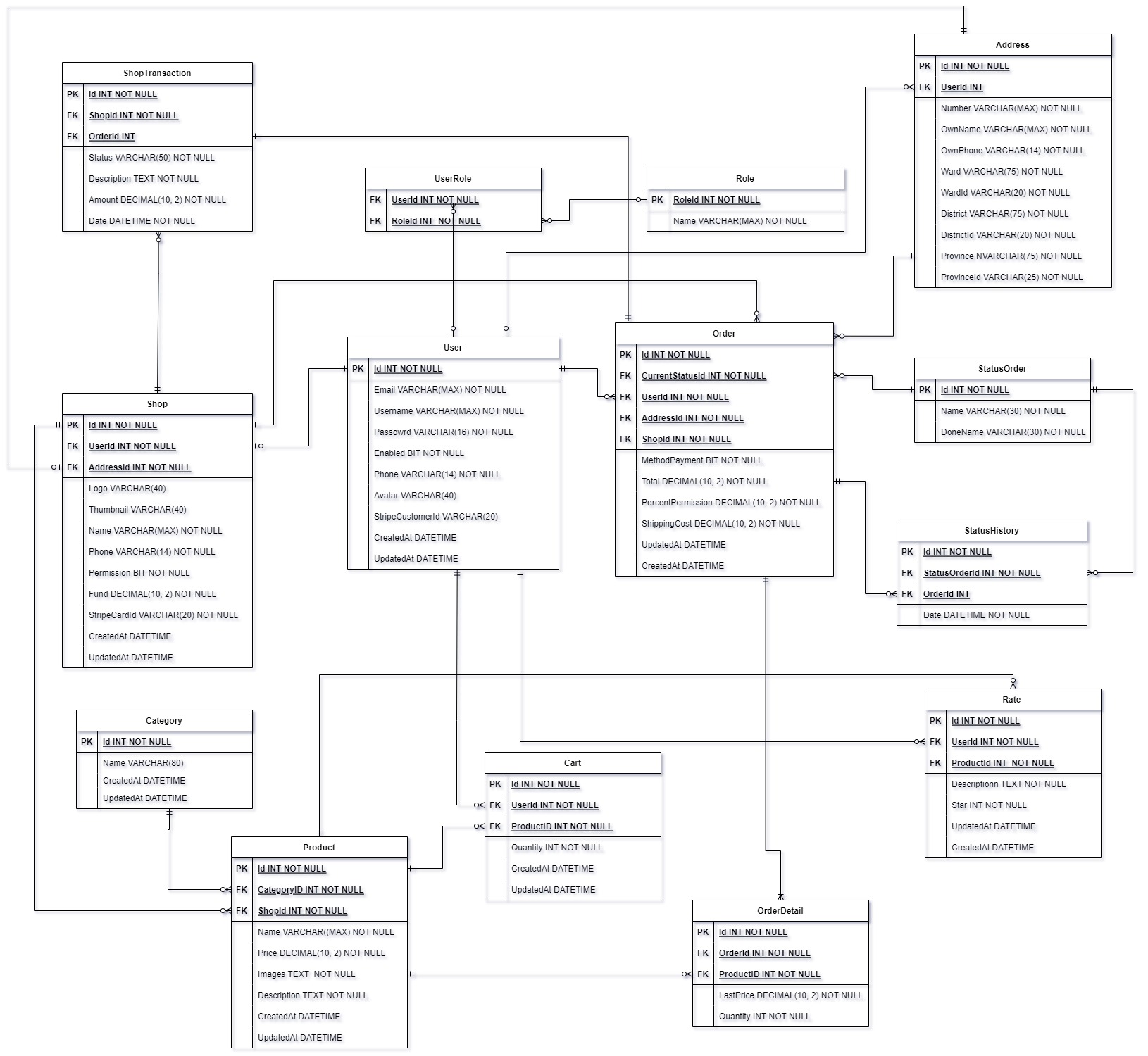
**3.3 Admin management**



**III. Entity Relationship Diagram (ERD)**

**1. ERD**

****

****

**2. Table Definition**

**2.1 User**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
|  | Email | varchar(max) |  |  |  |
|  | Username | varchar(max) |  |  |  |
|  | Password | varchar(max) |  |  |  |
|  | Enabled | bit |  | 1 |  |
|  | Phone | varchar(14) |  |  |  |
|  | Avatar | varchar(40) | 🗸 |  |  |
|  | StripeCustomerId | varchar(20) | 🗸 |  |  |
|  | CreatedAt | datetime | 🗸 | now() |  |
|  | UpdatedAt | datetime | 🗸 | now() |  |

**2.2 Role**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | RoleId | int |  |  |  |
|  | Name | varchar(max) |  |  |  |

**2.3 User role**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| FK | UserId | int |  |  |  |
| FK | RoleId | int |  |  |  |

**2.4 Rate**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
| FK | UserId | int |  |  |  |
| FK | ProductId | int |  |  |  |
|  | Description | text |  |  |  |
|  | Star | int |  |  |  |
|  | CreatedAt | datetime | 🗸 | now() |  |
|  | UpdatedAt | datetime | 🗸 | now() |  |

**2.5 Address**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
| FK | UserId | int |  |  |  |
|  | Number | varchar(max) |  |  |  |
|  | OwnName | varchar(max) |  |  |  |
|  | OwnPhone | varchar(14) |  |  |  |
|  | Ward | varchar(75) |  |  |  |
|  | WardId | varchar(20) |  |  |  |
|  | District | varchar(75) |  |  |  |
|  | DistrictId | varchar(20) |  |  |  |
|  | Province | varchar(75) |  |  |  |
|  | ProvinceId | varchar(20) |  |  |  |

**2.6 Cart**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
| FK | UserId | int |  |  |  |
| FK | ProductId | int |  |  |  |
|  | Quantity | Int |  | 1 |  |
|  | CreatedAt | datetime | 🗸 | now() |  |
|  | UpdatedAt | datetime | 🗸 | now() |  |

**2.7 Shop**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
| FK | UserId | int |  |  |  |
| FK | AddressId | int |  |  |  |
|  | Logo | varchar(40) | 🗸 |  |  |
|  | Thumbnail | varchar(40) | 🗸 |  |  |
|  | Name | varchar(max) |  |  |  |
|  | Permission | bit |  | 0 |  |
|  | Phone | varchar(14) |  |  |  |
|  | Fund | decimal(10, 2) |  | 0 |  |
|  | StripeCardId | varchar(20) |  |  |  |
|  | CreatedAt | datetime | 🗸 | now() |  |
|  | UpdatedAt | datetime | 🗸 | now() |  |

**2.8 Category**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
|  | Name | varchar(80) |  |  |  |
|  | CreatedAt | datetime | 🗸 | now() |  |
|  | UpdatedAt | datetime | 🗸 | now() |  |

**2.9 Product**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
| FK | CategoryId | int |  |  |  |
| FK | ShopId | int |  |  |  |
|  | Images | text |  | [] |  |
|  | Name | varchar(max) |  |  |  |
|  | Price | decimal(10, 2) |  |  |  |
|  | Description | text |  |  |  |
|  | CreatedAt | datetime | 🗸 | now() |  |
|  | UpdatedAt | datetime | 🗸 | now() |  |

**2.10 Order**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
| FK | CurrentStatusId | int |  |  |  |
| FK | UserId | int |  |  |  |
| FK | AddressId | int |  |  |  |
| FK | ShopId | int |  |  |  |
|  | MethodPayment | bit |  | 0 |  |
|  | Total | decimal(10, 2) |  |  |  |
|  | PercentPermission | decimal(10, 2) |  |  |  |
|  | ShippingCost | decimal(10, 2) |  |  |  |
|  | CreatedAt | datetime | 🗸 | now() |  |
|  | UpdatedAt | datetime | 🗸 | now() |  |

**2.11 Order detail**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
| FK | OrderId | int |  |  |  |
| FK | ProductId | int |  |  |  |
|  | LastPrice | decimal(10, 2) |  |  |  |
|  | Quantity | int |  |  |  |

**2.12 Status Order**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
|  | Name | varchar(30) |  |  |  |
|  | DoneName | varchar(30) |  |  |  |

**2.13 Status History**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
| FK | OrderId | int |  |  |  |
| FK | StatusOrderId | int |  |  |  |
|  | Date | datetime |  | now() |  |

**2.14 Shop Transaction**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Key | Filed name | Data Type | Null | Default | Description |
| PK | Id | int |  |  |  |
| FK | OrderId | int |  |  |  |
| FK | ShopId | int |  |  |  |
|  | Status | varchar(50) |  |  |  |
|  | Description | text |  |  |  |
|  | Amount | decimal(10, 2) |  |  |  |
|  | date | datetime |  | now() |  |