

Online Medicine Shopping Website



The Pharmacy Management system otherwise referred to as the pharmacy information system stores data, systemizes and controls the use of the medication process with the pharmacies.

The Medical Technology and Information technology is growing day-by-day. The system developer makes use of this golden opportunity to help the pharmacist in controlling their stock and choosing the right medicine by using the computer program.

❖Functional requirements

- The system shall allow for on-line product ordering by either the customer or the sales agent. For customers, this will eliminate the current delay between their decision to buy and the placement of the order. This will reduce the time a sales agent spends on an order by x%. The cost to process an order will be reduced to \$y.
- The system shall reflect a new and changed product description within x minutes of the database being updated by the product owner. This will reduce the number of incidents of incorrectly displayed information by x%. This eliminates the current redundant update of information, saving \$y dollars annually.
- The system shall refill the stock automatically, This will reduce the inventory cost process to \$y.
- The system shall provide accounting with accurate purchase transaction data. This will improve customer service by reducing billing complaints by x% and save \$y in correcting inaccurate accounts.
- The system shall provide searching into the database to help the customers finding the product they want, faster and easily.
- The system shall provide product categories.
- The system shall provide both supplier information, customer information for the sells agent.
- The system shall allow the customer to know whether the product is available or out of stock.
- The system shall show the customer the total cost before placing the order, this will reduce returning orders, because the customer will be fully aware of what he will be paying.
- The system shall provide shipping with accurate order data. This will allow the order to be processed in x days and inventory to be updated within y hours.

❖ **Non-Functional requirements**

○ **Usability**

A user should know how to use the website from the first time.

A user who already knows what product he is interested in should be able to locate and view that page in x seconds.

The final payment transaction shall be done in one click.

○ **Security**

The system shall provide password protected access to web pages that are to be viewed only by employees.

Transaction data must be transmitted in encrypted form.

○ **Interfaces**

○ **The system must interface with**

The current Oracle database systems for product and order information.

The PayPal system.

- Login Module

- In order, to get personalized or restricted information, place orders or do other specialized transactions a user must login so that the system can determine his access level.

- There are two types of users:

- User.
- Admin.

1. User Level Features

- A user must have an account to complete the order process.
- if the user does not have an account he needs to register.

- Registration Form

- **User enters a valid user name at least 3 letters, and 15 most only in English letters.**
- **User enters a password 5 digits at least.**
- **User enters a valid email and not taken.**

- ✓ If registration was successful, user's data will be stored into the database and a registration succeed message will appear to the user.

- If the user already has an account, he must log in.
 - Log in form

- **User must enter an email that is already stored in the database.**
- **User enters a password that is already stored in the database with the same email.**

- ✓ If logging in was successful, user will be able to use all the user's
- ✓ features.

- **User Search/Look for a product and add it to the cart.**

User can either view the products in a random way from the Medicine page, can search for a particular product, or by categories.

User can read the product's description by placing the mouse on the product's scope.

- Search Form

- **User can search by the search bar, by entering the name of the product.**
- **If the product is found the system will display the product, if not found the system will display not found.**

- Categories Form

- **User can look in products by classifications according to the medical purpose.**
- **The classifications will be as follow, Heart, Allergy, Cholesterol etc.**
- **User will choose from the given classifications.**
- **The system will display all the products that belong to the same category.**

✓ If user have found the product he was looking for, he can place his order successfully.

- Placing Order Form

- **User can add to cart by clicking “add to cart.”**
- **User can add products of different types in the same order.**
- **User can not add more than 3 product of the same type per order.**
- If user add more than 3 the system will display an error.**

✓ If the user added to his cart successfully, he will be able to go to the next stage, to finalize his order.

• Delivery information and order confirmation

○ Order confirmation Form

- User can view it's cart.
- User can view it's cart, by clicking the cart mark.
- User can delete from the he's cart, by clicking the deletion mark.
- User can update the order quantity.

✓ If the user has updated his order successfully, he will go to the info/delivery page.

○ Delivery Information Form

○ Form

- User must enter a phone number, must be equal to 11 digits.
- The number should not be taken.
- User must enter location information not more than 100 letter.

○ Payment Information Form

○ Form

- User will be able to view the total cost order before payment.
- User must enter a valid Credit card number, equal to 16 digits. if the number was invalid the system will display an error.
- User must enter the card's name must be in English letter.
- User must enter the CVV number, not more than 3 digits. If the CVV number was invalid the system will display an error.
- User must click on "pay" in order for the process to be completed.
- Each order must decrease the product quantity in the inventory.

✓ If the payment process went successfully, bill will be printed.

○ Bill Information Form

○ Form

- Bill will contain; customer's information, order total cost, order time delivery time).
- Once the bill has been created, it will be stored in the database.
- Each bill has it's own id.
- User can view it's bill anytime.

2. Admin Level Features

- Admin Log in Form
 - Form

- **Admin must enter an email that is already stored in the database.**
- **Admin email must end with company name@companyName.**
- **Admin enters a password that is already stored in the database with the same email.**

✓ If logging in was successful, admin will be able to use all the admin's features.

- Admin View Users/Orders Form
 - Form

- **Admin can view the users details, their information.**
- **Admin can view the orders placed by the customers, their bills.**

- Admin Product Entry Form
 - Form

- **Admin can add new product by clicking on “add new product”.**
- **Admin must add a description to the new product by clicking on “Add description”.**
- **Each product has its own ID.**
- **Admin can update a product description, by clicking “update product’s description” and it will automatically reflect on the database.**
- **Product description will be stored in the database.**

✓ If adding/updating was successful, it will automatically reflect on the system and the database

- Admin Add/Update Supplier Form
- Form

- **Admin can add new supplier by clicking on “add new supplier”.**
- **Admin must add the supplier’s information**
Supplier’s name, email, phone number, that will store into the database.
- **The supplier’s name must be at least 3 letter and 20 letter at most.**
- **The supplier’s email must be valid.**
- **The supplier’s phone number must be equal to 11 digits.**
- **Admin can update supplier’s information by clicking on “update supplier”.**

✓ If adding/updating was successful, it will automatically reflect on the system and the database.

- Order from supplier Form
- Form

- **Admin can order from the supplier manually/automatically.**
- **The system must generate an automatic order when the inventory reach a certain quantity for each product.**
- **Admin can view the prices before ordering.**
- **Admin can generate an order by clicking on “place order”**
- **Admin must enter the quantity for the order”.**
- **The payment must be an automatic payment using a stored card.**

✓ If the order process went successfully, bill will be printed with the total cost.

❖ Use-Case Diagram



• Use-Case Description

Identifier and name: **Order a product**

Initiator: User \ Admin

Goal: a product reserved for a customer

Pre-condition: user must have an account to login

Post-condition: the user reserved his order

Assumptions: the expected initiator is customer (using web browser) or Admin

Main success scenario:

- 1 customer choose to order a product
- 2 customer search for a product
- 3 system provides details about the product
- 4 system provides availability and price of request
- 5 customer agrees to order
- 6 customer provides payment details
- 7 system creates an order and gives it an identifier
- 8 system reveals the identifier to the customer
- 9 system confirm the order

Exceptions:

1. product not available
2. customer may add the product to cart
3. customer has a limited number for ordering

• Use-Case Description

Mange product

Identifier and name: **Add product**

Initiator: Admin

Goal: Admin adds the product to the online website and store the product details into the database.

Pre-condition:

- Admin must have an account to login.
- The product must be available in stock.

Post-condition:

- The product must be shown on the website and stored successfully into the database.

Assumptions: the expected initiator is Admin (using web browser).

Main success scenario:

1. Admin must be authorized to add a product.
2. The product must be available on the website.
3. The system provides details about the product.
4. Customer must be able to add the product to its cart.
5. Product's details stored into the database

Exceptions:

1. Admin is not authorized to add a product.
2. Product is not available in the stock.
3. Customer can't view the product.
4. Product details is not stored into the database.

- **Use-Case Description**
Mange product

Identifier and name: **Update product**

Initiator: Admin

Goal: Admin update the product details and the new changes must reflects the online website and the new changes must be stored into the database.

Pre-condition:

- Admin must have an account to login.
- The product must be exists.

Post-condition:

- The new changes must reflect the online website and stored the new changes successfully into the database.

Assumptions: the expected initiator is Admin (using web browser).

Main success scenario:

- 1. Admin must be authorized to update a product.**
- 2. The system reflect the new changes about the product.**
- 3. Customer must be able to view the new product details.**
- 4. New changes stored into the database**

Exceptions:

1. Admin is not authorized to update a product.
2. Product is not stored in the database.
3. Customer can't view the new product details.

- **Use-Case Description**
Mange product

Identifier and name: **Delete product**

Initiator: Admin

Goal: Admin remove the product from the online website and remove its details from the database.

Pre-condition:

- Admin must have an account to login.
- The product must be exists.

Post-condition:

- The product and its details must be successfully removed from the website and from the database.

Assumptions: the expected initiator is Admin (using web browser).

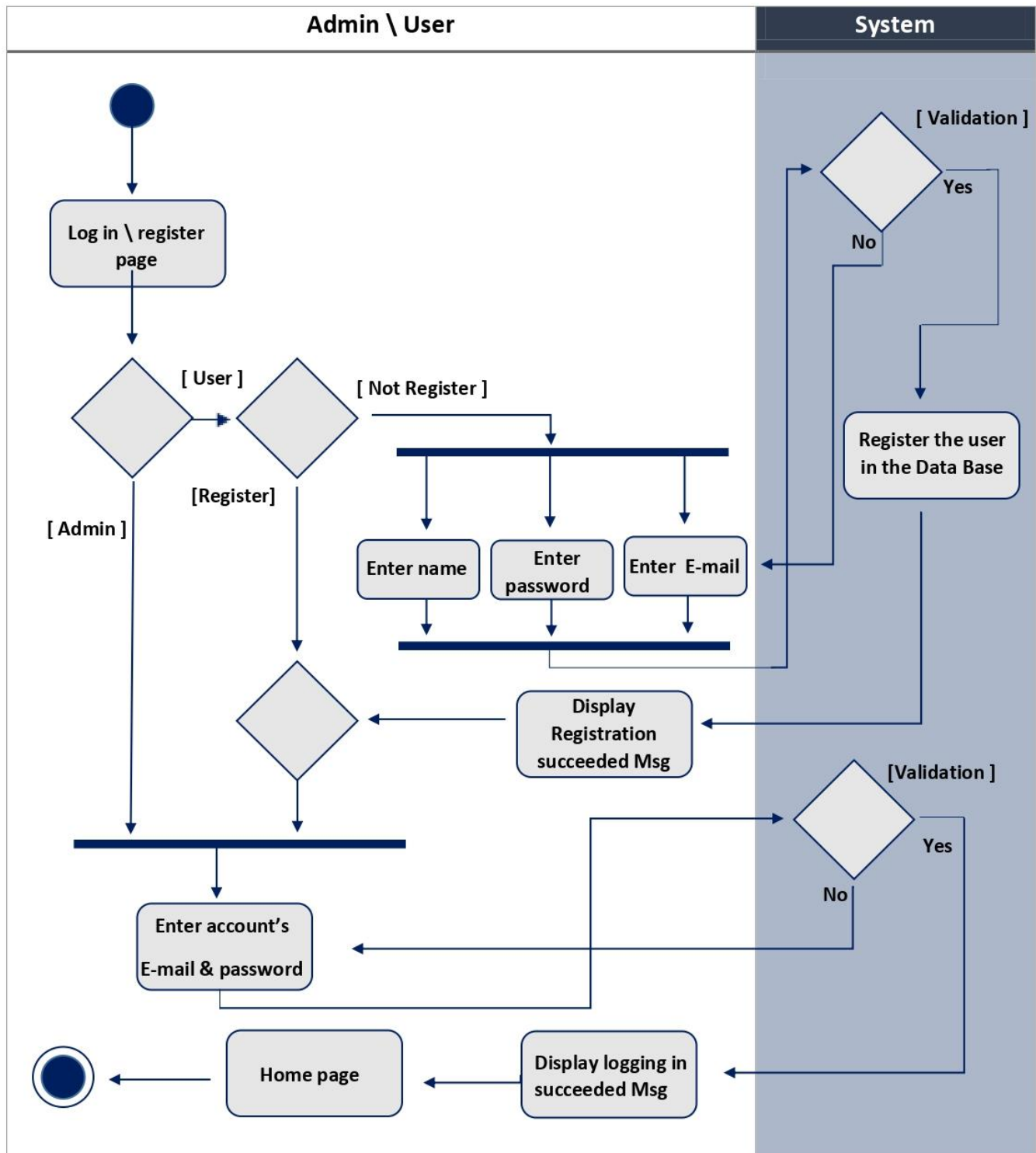
Main success scenario:

- 1. Admin must be authorized to delete a product.**
- 2. The product must be exists in the database.**
- 3. System removes the details about the product from the database.**
- 4. User can't view the product on the website.**

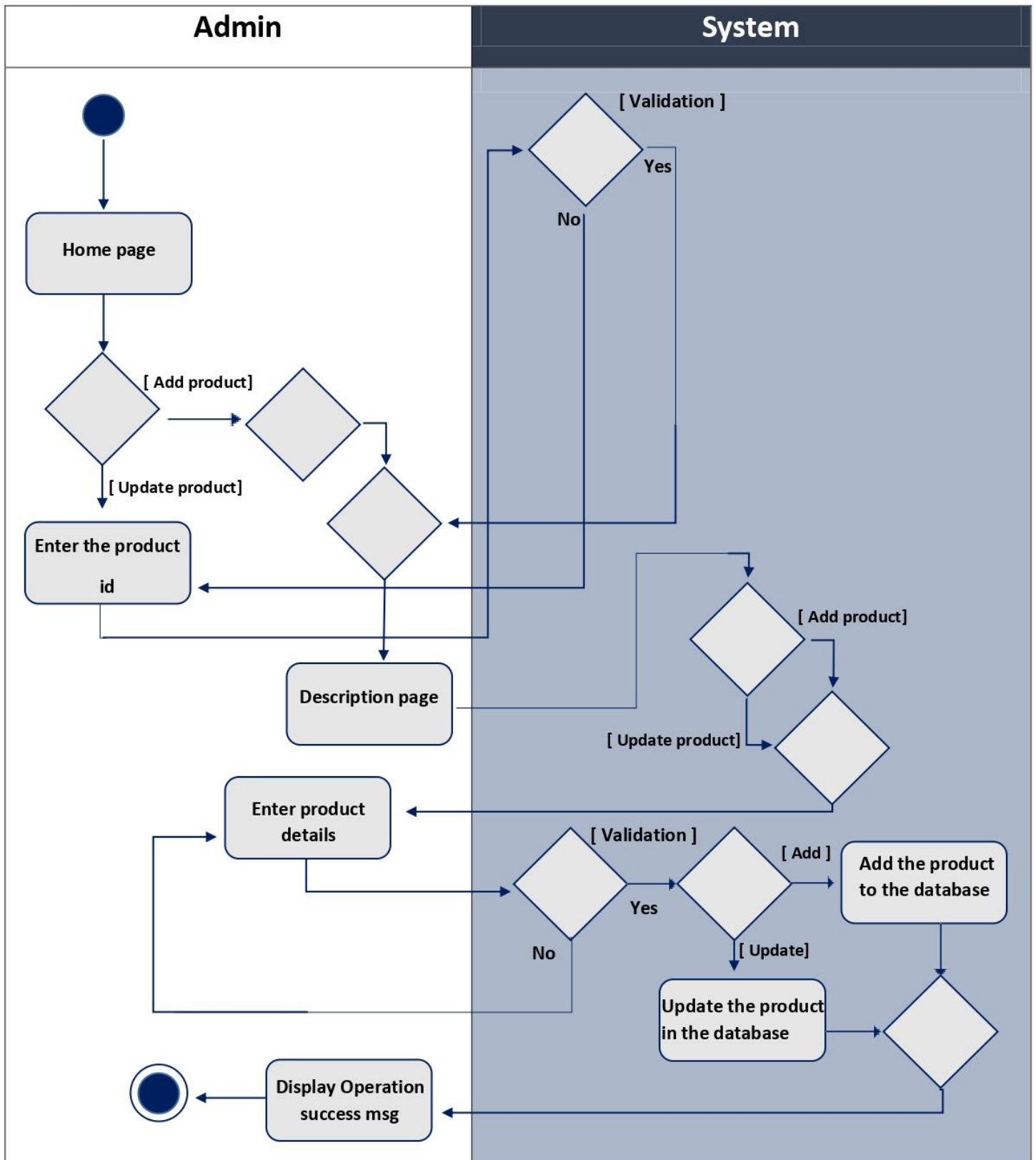
Exceptions:

1. Admin is not authorized to delete a product.
2. Product is not existed in the database.
3. Customer can view the product.
4. Product stored into the database.

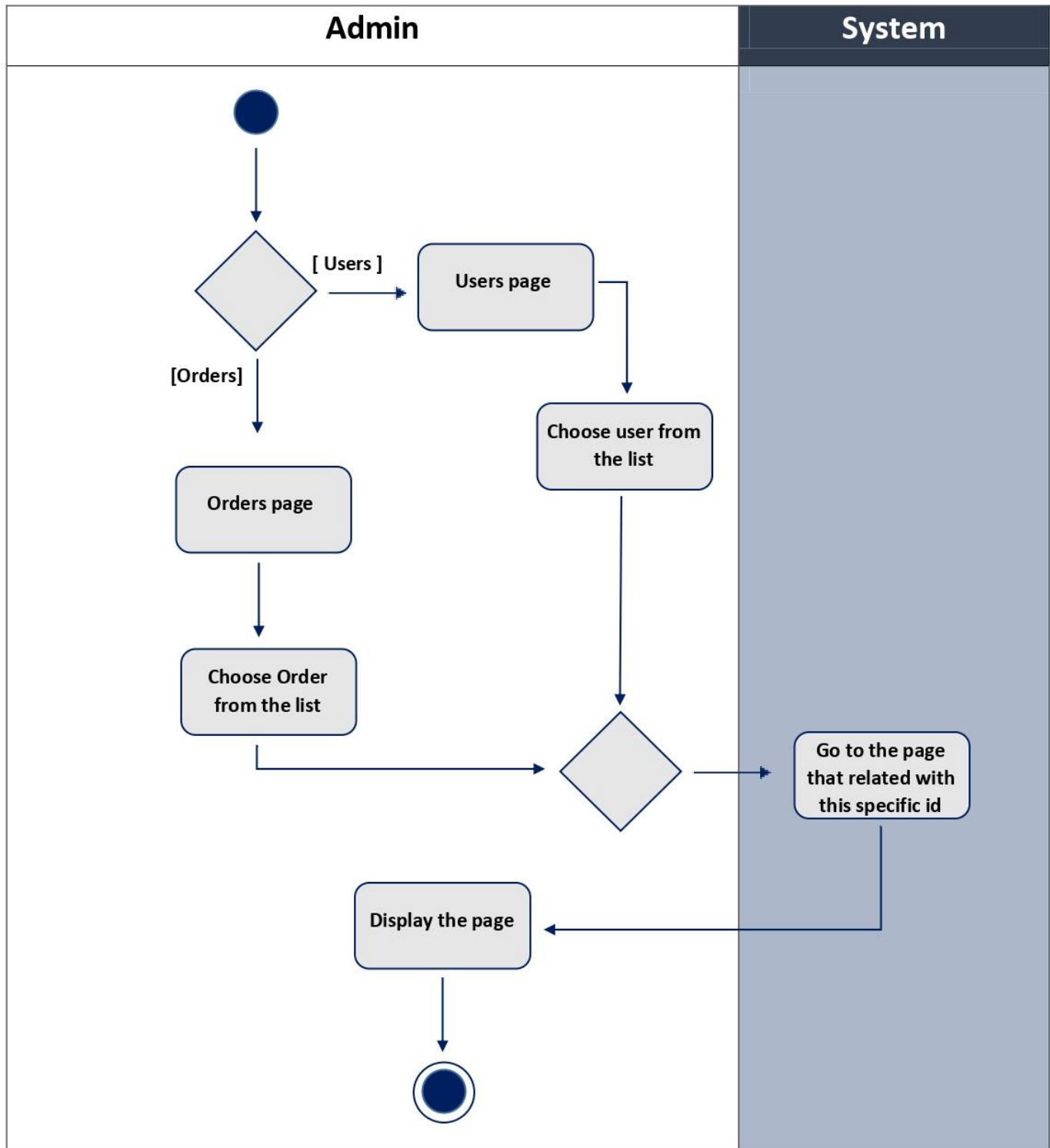
❖ Activity Diagram(s)



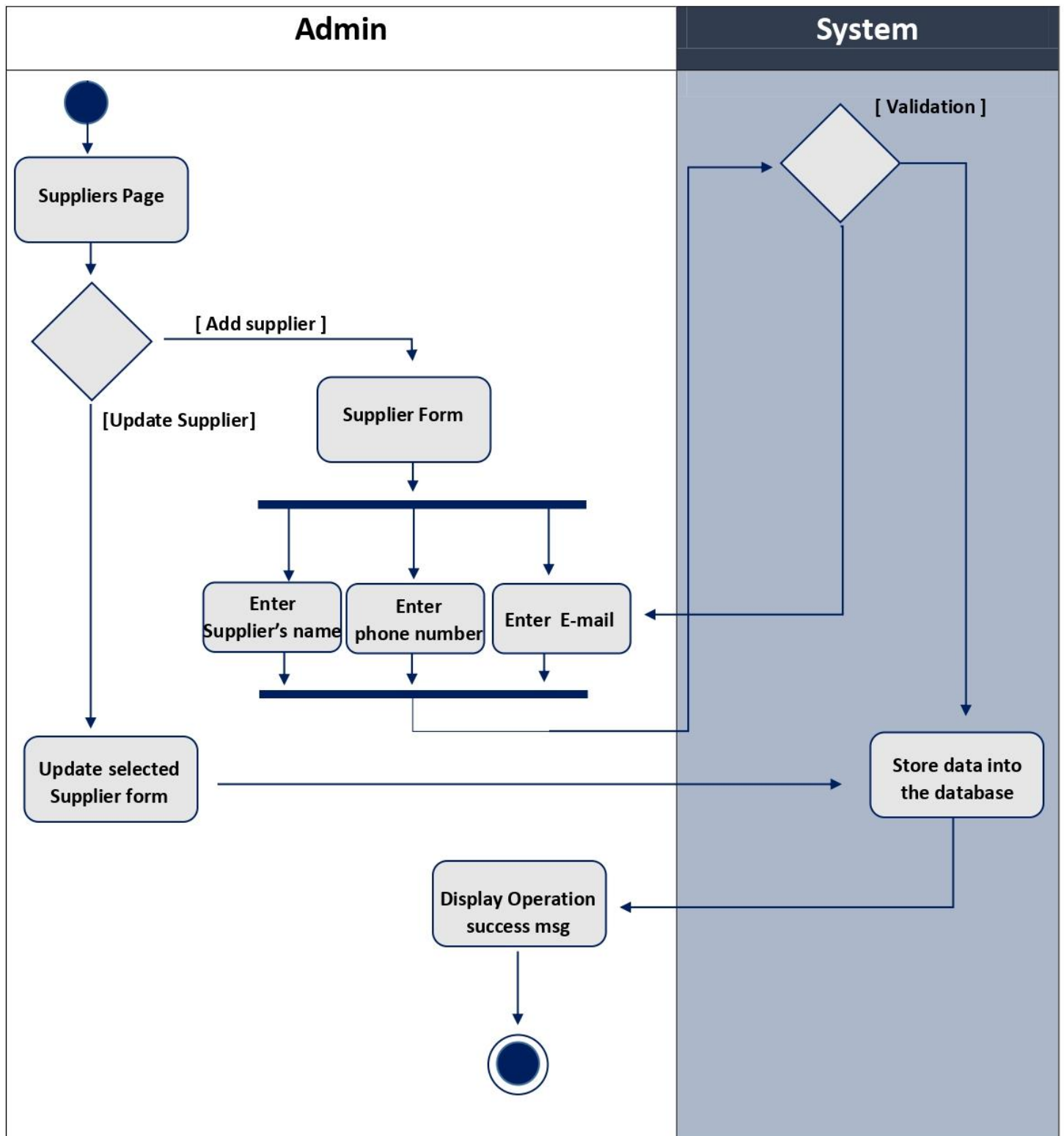
Admin product Entry



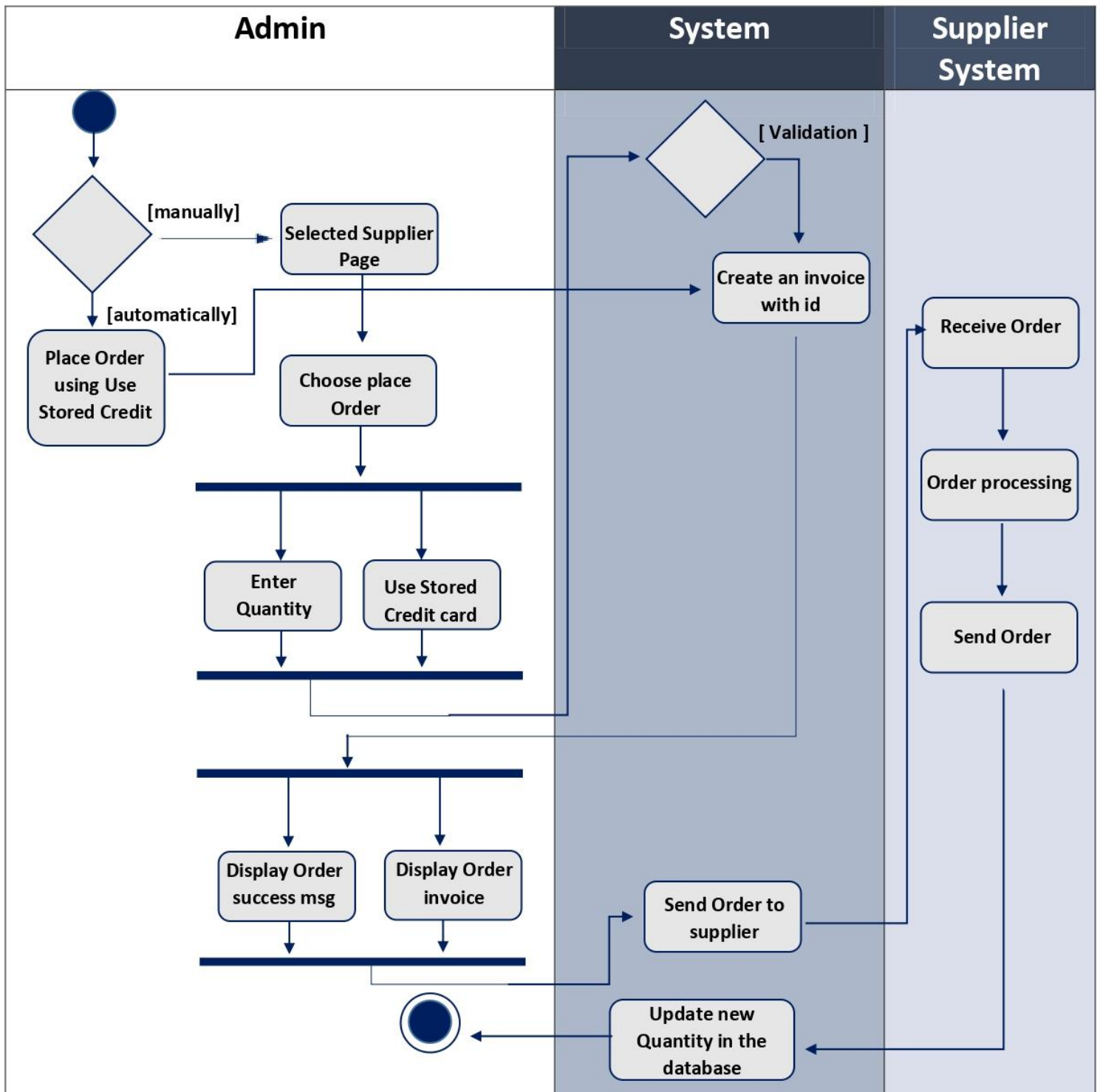
Admin view Orders\User details



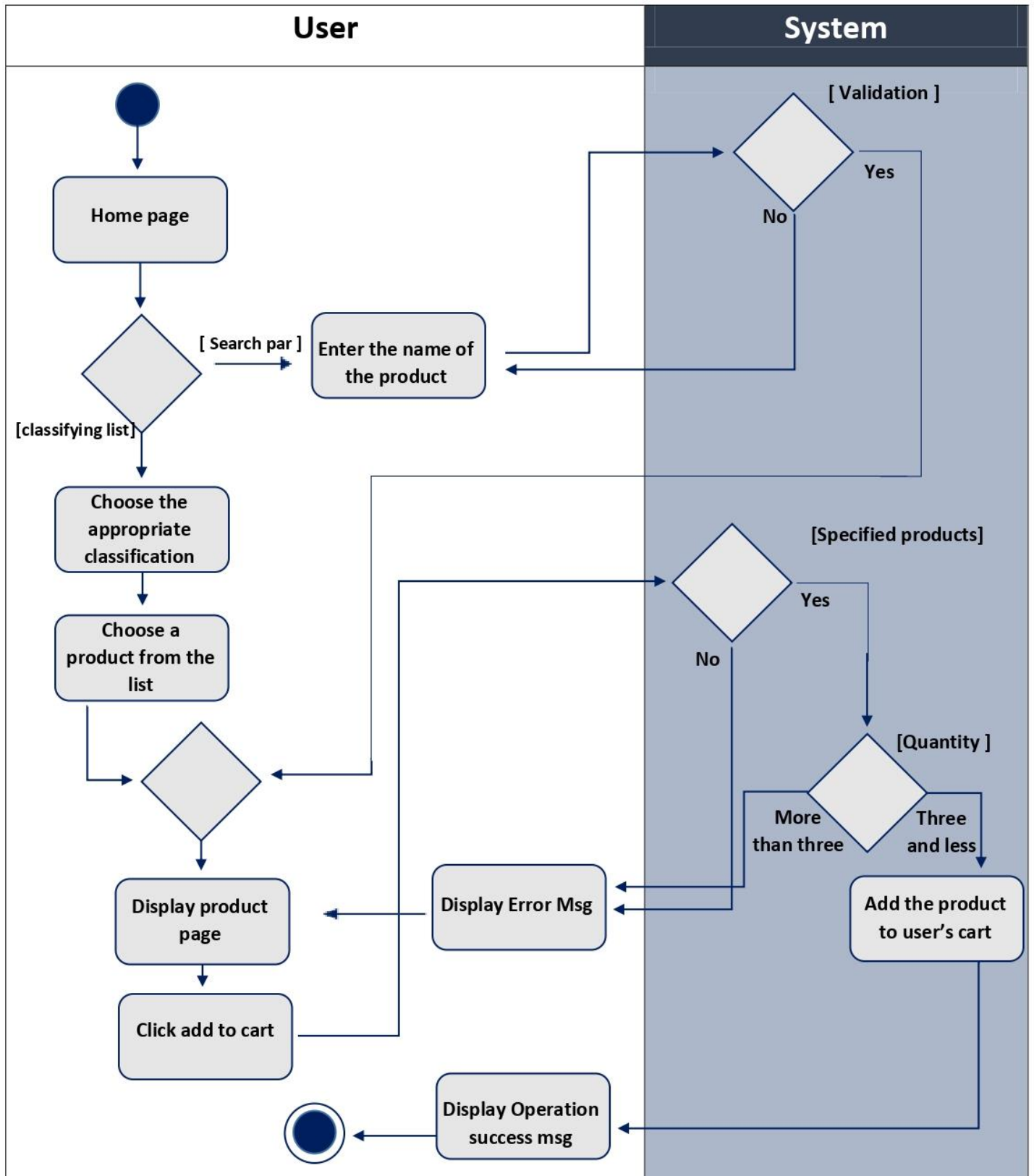
Admin Add/Update Supplier Form



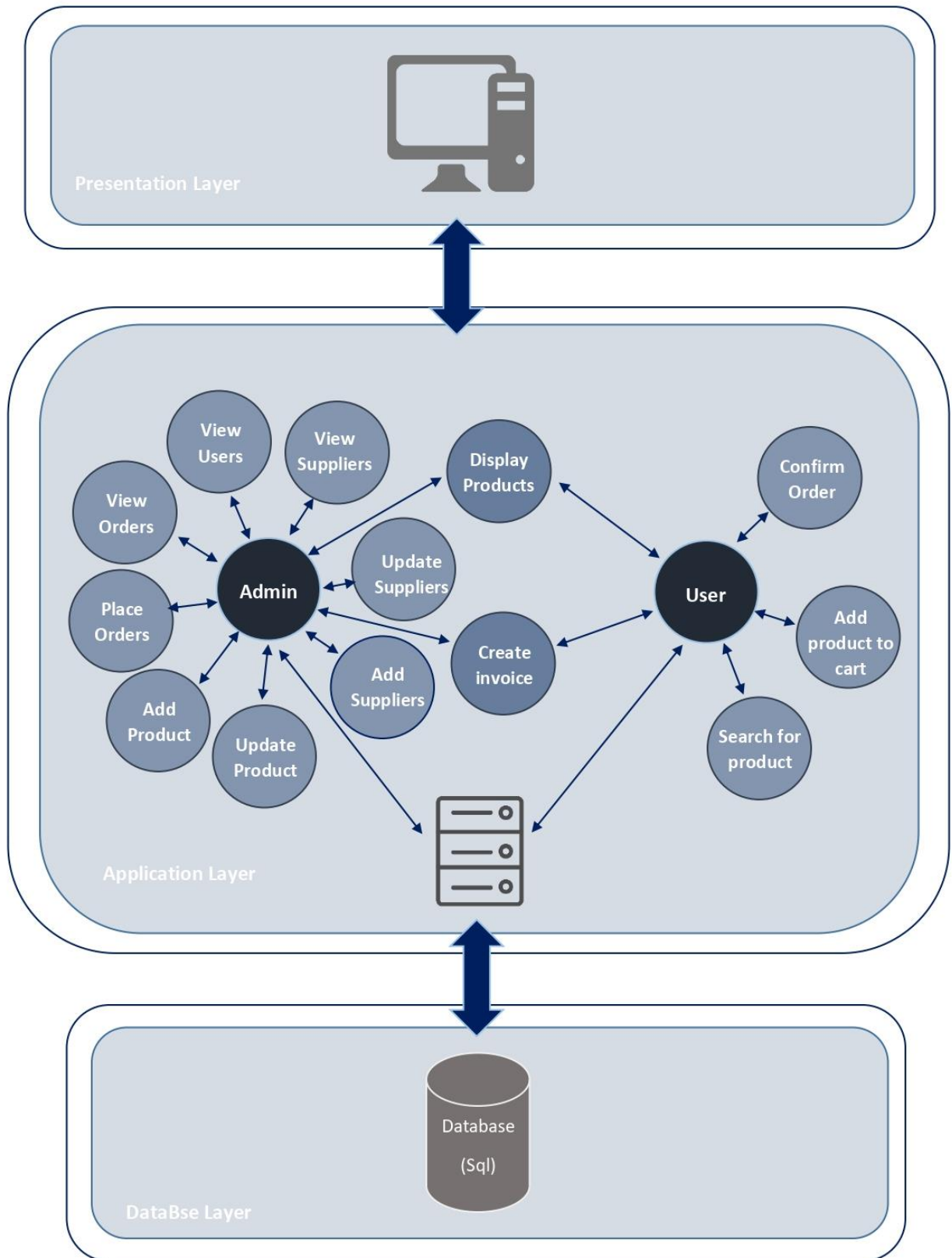
Admin Order from supplier Form



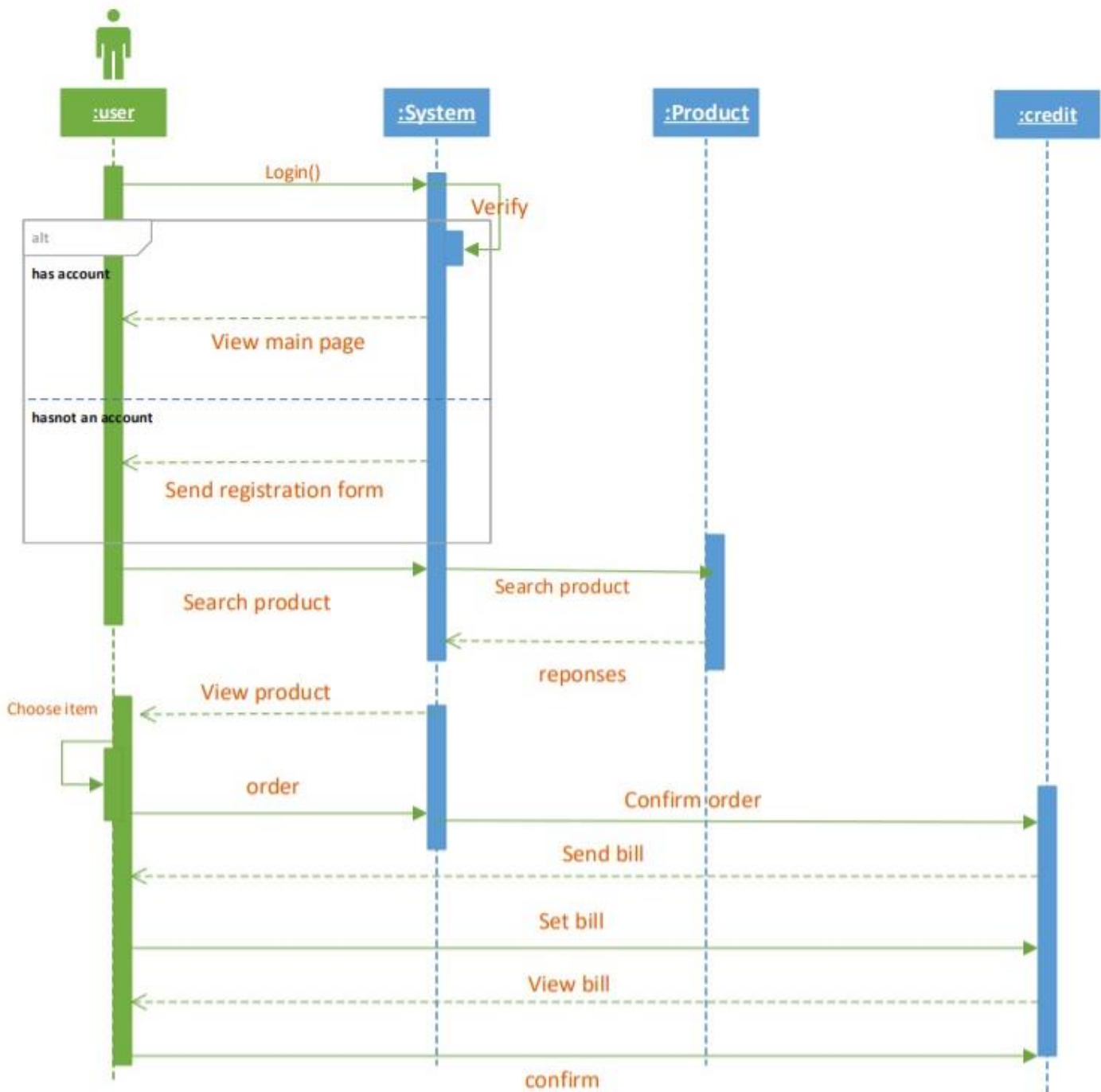
User's search for a product and add it to the cart

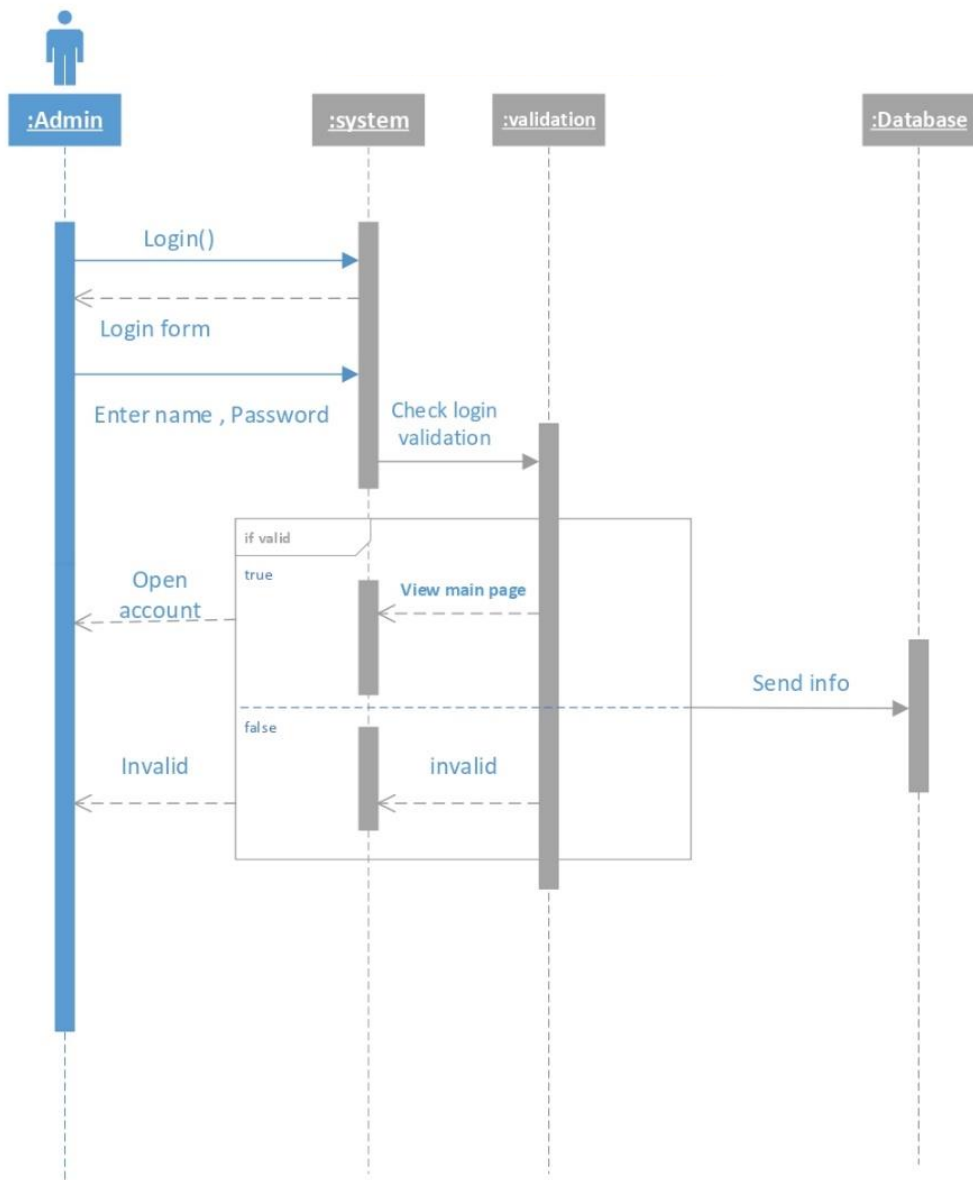


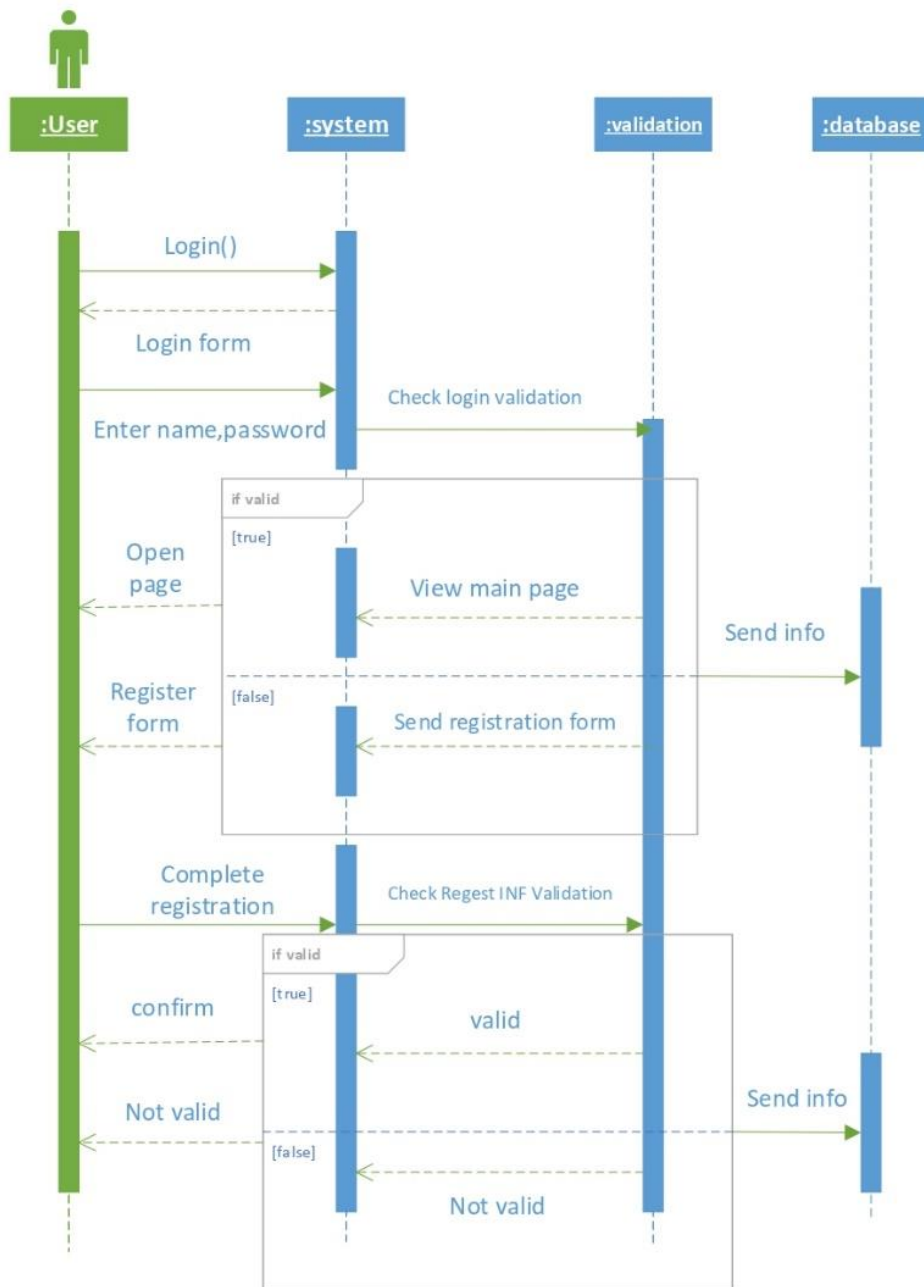
❖ System Architecture

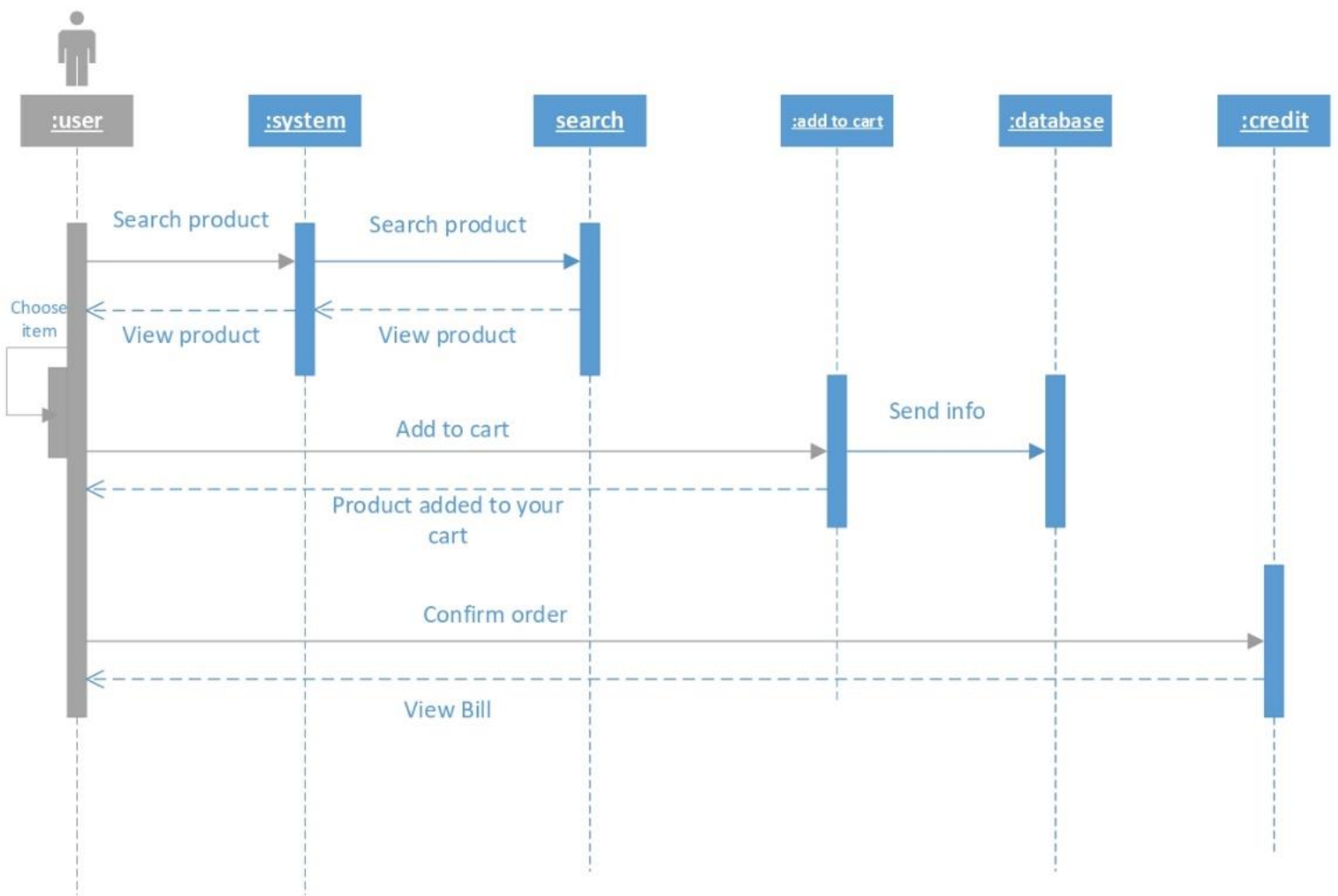


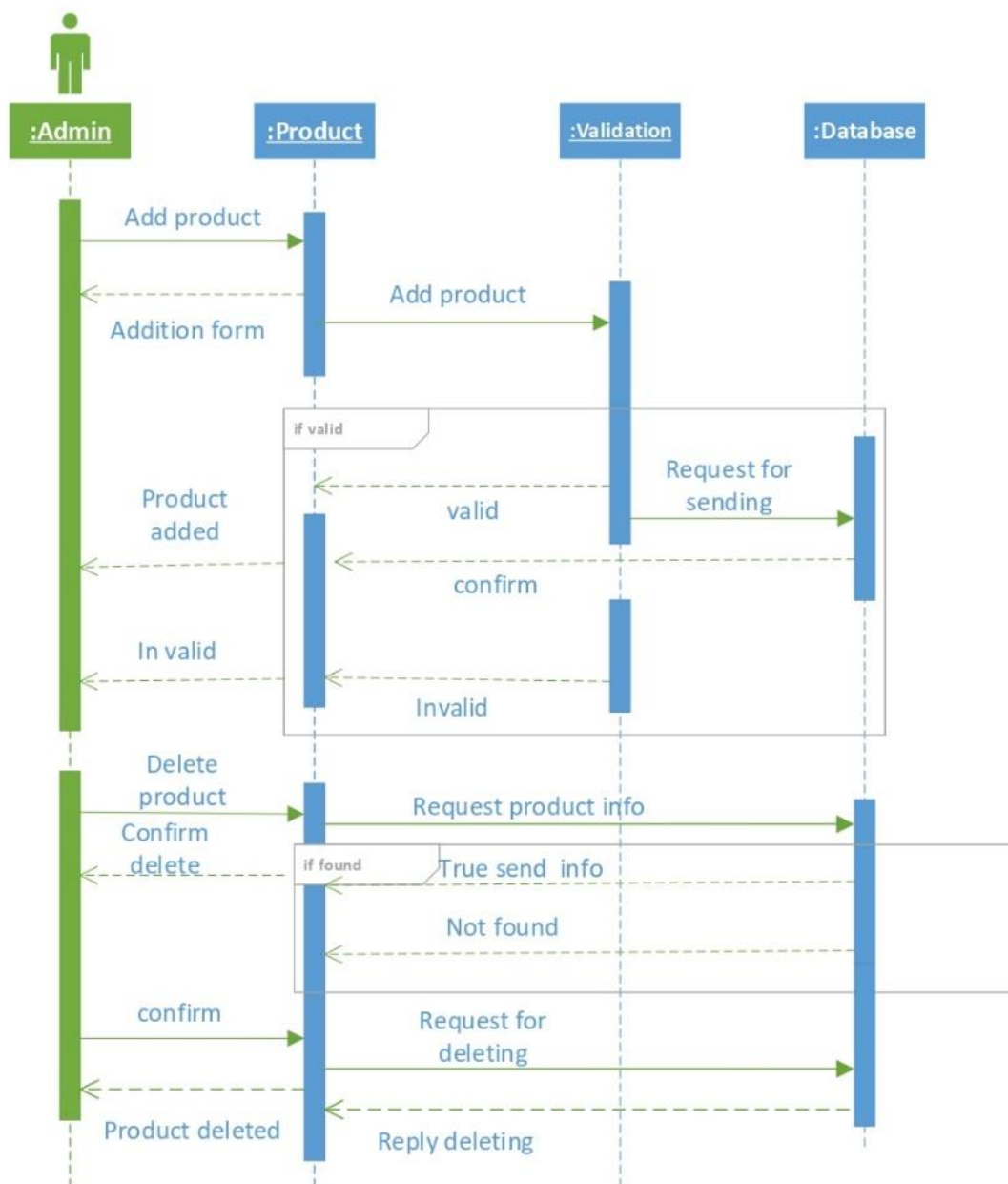
❖ Sequence Diagram(s)





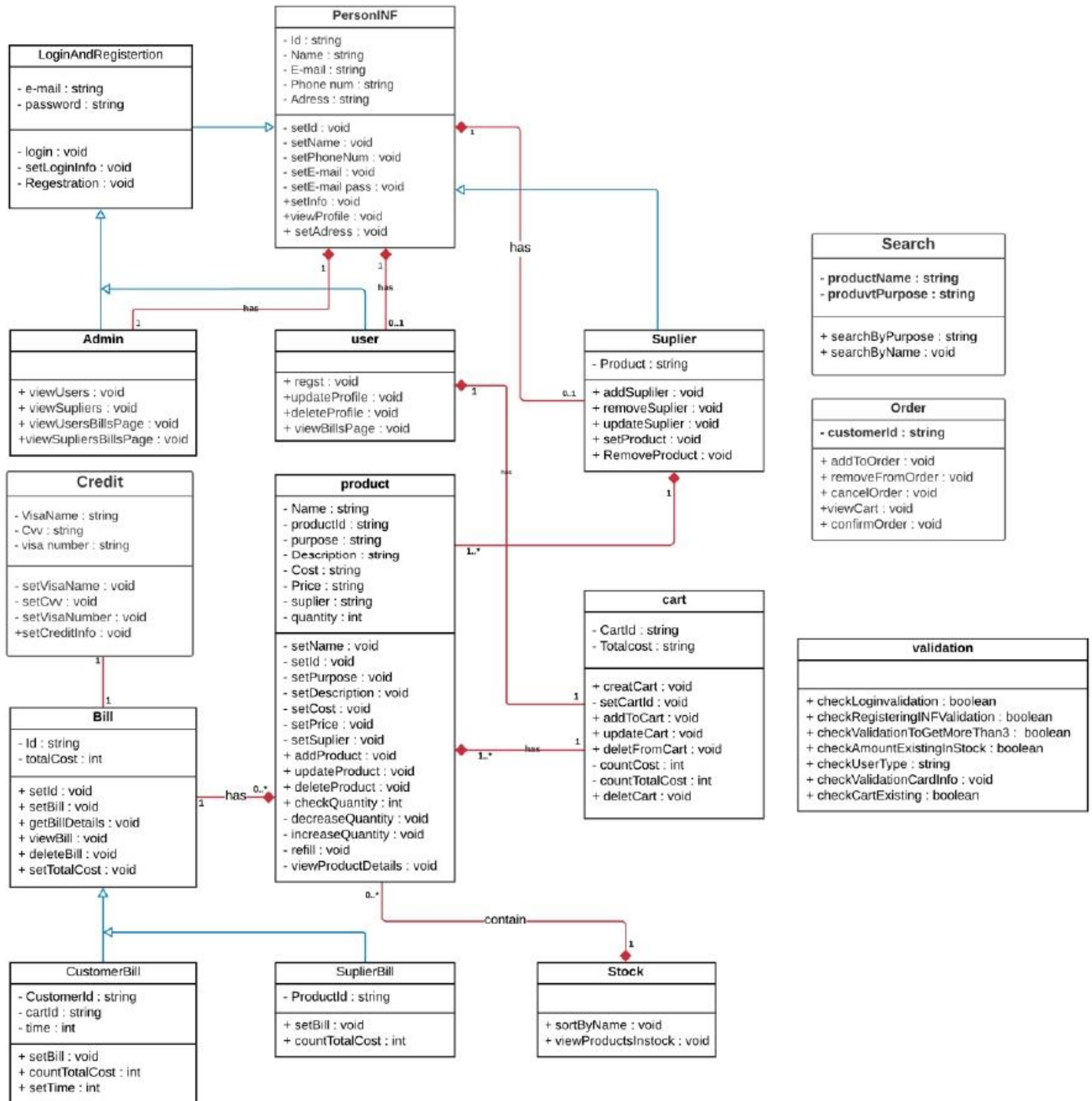




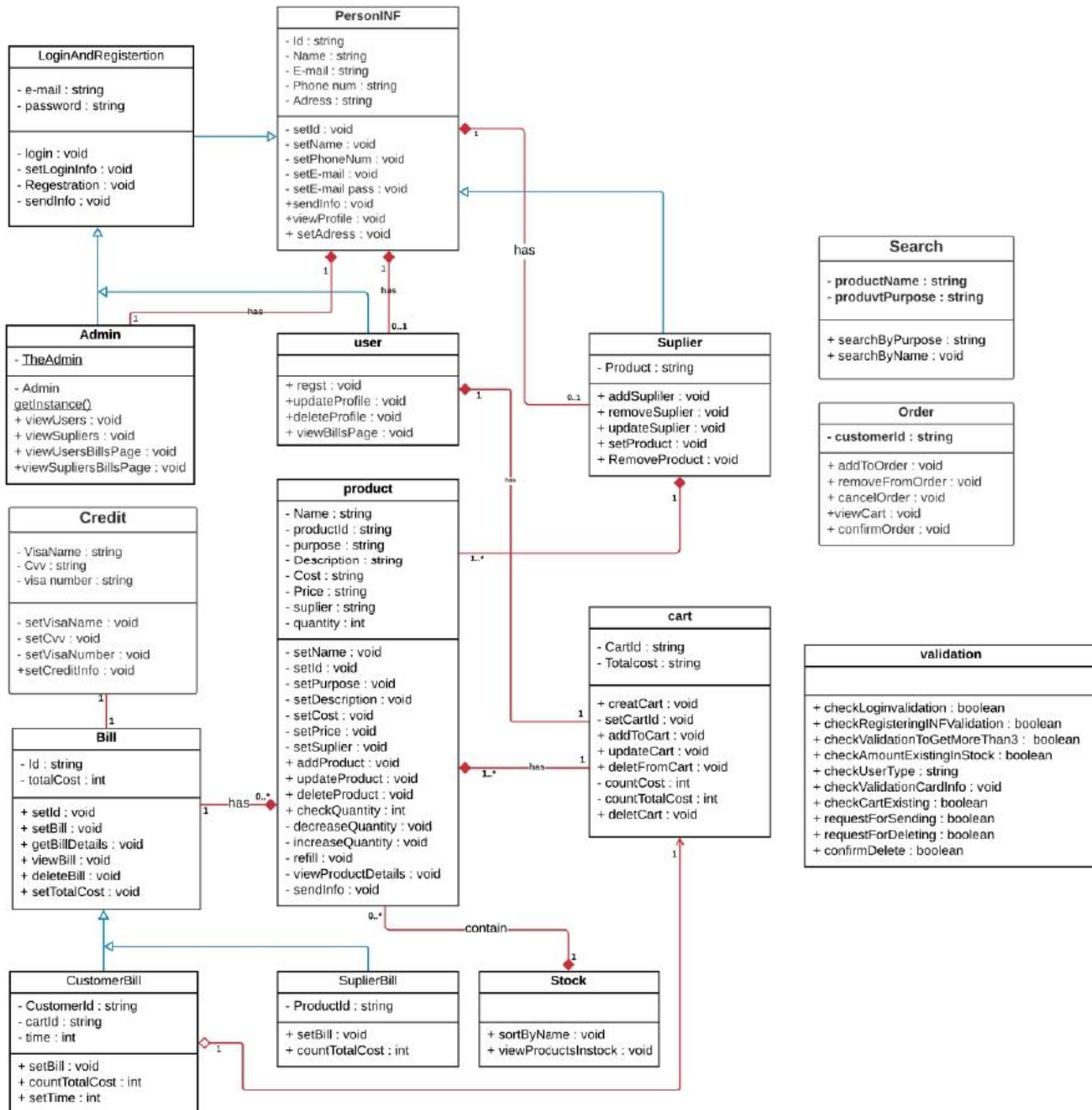


❖ Class Diagram

• Initial version

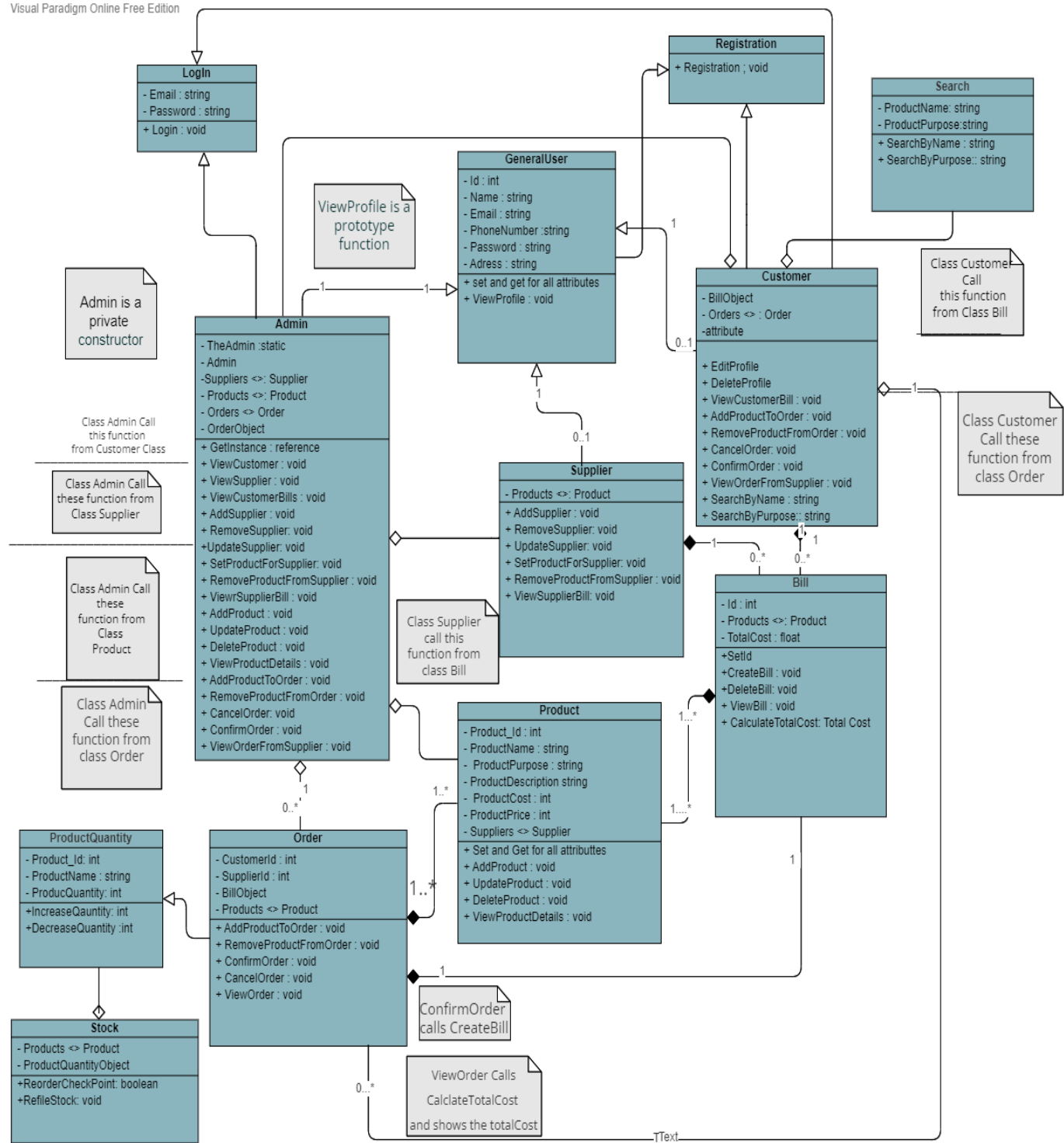


• Intermediate version



• Final version

Visual Paradigm Online Free Edition



❖Mandatory Design Pattern Applied

• SINGLETON PATTERN

- **Context:**

It is very common to find classes for which only one instance should exist (singleton).

Examples: Admin Class.

- **Problem:**

How do you ensure that it is never possible to create more than one instance of a Admin class, And provide a global point of access to it.

- **Forces:**

The use of a public constructor cannot guarantee that no more than one instance will Be created.

The singleton instance must also be accessible to all classes that require it therefore it Must often be public.

- **Solution:**

Have the constructor private to ensure that no other class will be able to create An instance of the Admin class
Define a public static method, The first time this method is called, it creates the Single instance of the class “Admin” and stores a reference to that object in a Static private variable.

Example:

Admin
<u>TheAdmin</u>
Admin()<<private>> <u>getInstance()</u>
If (TheAdmin==null) TheAdmin=newAdmin(); Return TheAdmin;

• DELEGATION PATTERN

- **Context:**

You are designing a method in a class. You realize that another class has a method which provides the required service. Inheritance is not appropriate

- Example:**

The customer_bill class could be created using an existing class called cart using delegation pattern. The CountTotalCost method of Customerbill calls the countTotalCost method of the Cart class .

- **Problem:**

How can you most effectively make use of a method that already exists in the other class?

- **Forces:**

You want to minimize development cost by reusing methods.

- **Solution:**

The CountTotalCost method in the CustomerBill class calls a countTotalCost method in the Cart class to perform the required task. An association must exist between the CustomerBill and cart classes.

- **IMMUTABLE PATTERN**

- **Context:**

An immutable object is an object that has a state that never changes after Creation.

Example: Admin Class.

- **Problem:**

How do you create a Admin class whose instances are immutable Forces

There must be no loopholes that would allow ‘illegal’ modification of an Immutable object.

- **Solution:**

Ensure that the constructor of the Admin class is the only place where The values of instance variables are set or modified.

Instance methods which access properties must not change instance Variables.