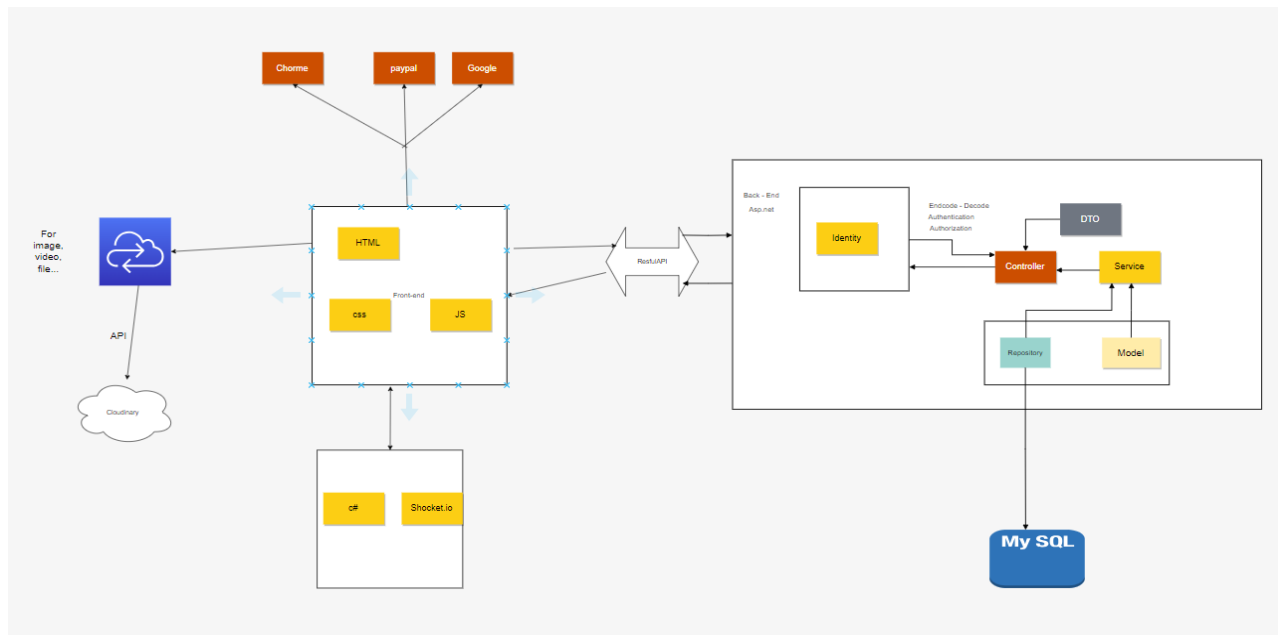


Shop Online

Software Design Document

I. Design Description

1. Overall Architecture



2. System Architecture

2.1. Component Explanation.

a. Front-end:

- HTML/CSS



HTML (Hypertext Markup Language) is a markup language used to create the structure and content of web pages. It uses various tags to define elements such as headings, paragraphs, images, links, and forms.

CSS (Cascading Style Sheets) is a language used to define the presentation and styling of HTML documents. It allows developers to control the visual appearance of web pages by specifying properties such as colors, fonts, layout, and positioning. CSS can apply to HTML documents in various ways, including through inline styles, embedded styles, and external style sheets.

- Javascript



JavaScript is a popular programming language used to create interactive and dynamic web pages. It is primarily used on the client-side (in the browser) but can also be used on the server side. JavaScript is a high-level, object-oriented language with a syntax similar to C and Java. It is used for a variety of tasks, including validating user input, modifying the content of a webpage, creating animations, and responding to user actions. JavaScript can also be used with HTML and CSS to create rich, interactive web applications. In recent years, JavaScript has expanded beyond the web and is used in areas such as serverless computing, machine learning, and Internet of Things (IoT) devices.

b. Back-end:

- Mysql



MySQL is a widely used open-source relational database management system (RDBMS) that is used to manage and store data. It is a popular choice for web applications because it is fast, reliable, and scalable. MySQL is owned by Oracle Corporation and is a key component of the LAMP (Linux, Apache, MySQL, and PHP/Python/Perl) stack.

MySQL is used to manage data in a structured format, allowing you to organize and store data in tables that are connected by relationships.

- Csharp



In an ASP.NET Core web app, C# plays a significant role in developing the back-end portion of the application. Here are some key points about using C# in ASP.NET Core:

1. **Object-Oriented Programming:** C# is a powerful object-oriented programming language, allowing you to build objects and classes to organize your source code in a logical and manageable way. It supports concepts such as inheritance, polymorphism, abstraction, and interfaces.
2. **Razor Pages and MVC:** C# is used to write code in Razor Pages and controllers in the MVC (Model-View-Controller) pattern of ASP.NET Core. Razor Pages provides an easy-to-use approach for building web pages, while MVC provides a structure for separating logic and presentation.
3. **Dependency Injection (DI):** ASP.NET Core supports Dependency Injection to reduce dependencies between components in the application. C# is used to register and resolve dependencies in the DI container, helping manage dependencies and instantiate necessary objects at runtime.
4. **Entity Framework Core:** C# along with Entity Framework Core makes interacting with databases straightforward. Entity Framework Core provides an ORM (Object-Relational Mapping) approach to interact with the database, allowing you to use C# to perform queries and manipulate data.
5. **Data Processing and Business Logic:** C# in ASP.NET Core enables you to process data from requests and perform business logic of the application. Using C#, you can perform CRUD operations (Create, Read, Update, Delete), handle errors, authenticate and authorize users, and execute complex calculations and logic.

c. Program Environment

- **Visual studio code**



Visual Studio Code is a text editing program, released by Microsoft. It is an Integrated Development Environment (IDE) that offers users the ability to edit different text and code. In this project, Visual Studio Code is used to build the front-end application.

d. API

- PayPal
- Facebook

II. Entity-Relationship Diagram

1. Entities Description

1.1. Users

Entity Name:	Users
Description:	Storage basic information of user

Attribute:	<ul style="list-style-type: none">• UserName (PrimaryKey)• NormalizedUserName• Email• NormalizedEmail• EmailConfirmed• PasswordHash• SecurityStamp
------------	--

	<ul style="list-style-type: none"> • ConcurrencyStamp • PhoneNumber • PhoneNumberConfirmed • TwoFactorEnabled • LockoutEnd • LockoutEnabled • AccessFailedCount
--	--

1.2. Addresses

Entity Name:	Addresses
Description:	Store all information about the user's shipping address before placing an order.
Attribute:	<ul style="list-style-type: none"> • AddressId (Primary Key) • UserId (Foreign Key) • Province • Town • District • HomeStatus

1.3. UserLogins

Entity Name:	UserLogins
Description:	Storage all user information of user after login.
Attribute:	<ul style="list-style-type: none"> • LoginProvider (Primary Key) • ProviderKey (Primary Key) • ProviderDisplayName • UserId(Foreign Key)

1.4. UserRoles.

Entity Name:	UserRoles
Description:	Used to decentralize users to limit permissions in the system.
Attribute:	UserId (primary key) RoleId (primary key)

1.5. Roles.

Entity Name:	RolesR
Description:	This is where the information for authorization by admin will appear.
Attribute:	Id (primary key) Name NormalizedName ConcurrencyStamp

1.6. RoleClaims

Entity Name:	RoleClaims
Description:	Functions that the administrator can decentralize.
Attribute:	Id (Primary Key) RoleId (Forgein Key) ClaimType ClaimValue

1.7.UserTokens.

Entity Name:	UserTokens
Description:	Manage user access and authentication in systems, applications, or services
Attribute:	UserId (Foreign Key) LoginProvider (Primary Key) Name (Primary Key) Value

1.8. Bills.

Entity Name:	Bills
Description:	Record orders, manage payments, invoice customers, track order status, and generate financial reports.
Attribute:	BillId int (Primary Key) PaymentCode (Foreign Key) UserId (Foreign Key) TransportId (Foreign Key) PurchaseDate BillStatus PaymentMethod ShippingAddress

	Note ShippingFee
--	---------------------

1.9. Payment.

Entity Name:	Payments
Description:	Accept a variety of payment methods, secure payment information, payment history, and process returns and refunds.
Attribute:	PaymentCode (Primary Key) PaymentName PaymentStatus

1.10. Transports

Entity Name:	Transports
Description:	Manage and integrate with shipping services, charge shipping, track shipping, manage schedules, support returns and exchanges
Attribute:	TransportId int (Primary Key) TransportName TransportStatus

1.11. BillDetails

Entity Name:	BillDetails
Description:	Store and manage the details of each line item in a customer's invoice
Attribute:	BillDetailId int (Primary Key) BillId ProductId (Foreign Key) Quantity

1.12. Products.

Entity Name:	Products
Description:	Store and manage product details and integrate with shopping cart and order management.

Attribute:	ProductId int (Primary Key) ProductName ImportDate ProductDescription typeGender PerDiscount ProductQuantities ProductPrice BlogId (Foreign Key) IsAvailble HomeStatus SubCategoryID (Foreign Key) ListColor ImageMain ListSize
------------	---

1.13.SubCategory.

Entity Name:	SubCategory
Description:	Classify and subdivide different types of products.
Attribute:	SubCategoryId (Primary Key) Name CateogoryId (Foreign Key)

1.14. Categories.

Entity Name:	Category
Description:	Aggregate all clothes before sorting.
Attribute:	CategoryId (Primary Key) CategoryName

1.15. Feedbacks.

Entity Name:	Feedbacks
Description:	Users can rate products after experiencing them.
Attribute:	feedback_id (Primary Key) user_id (Foreign Key) feedback_date feedback_title feedback_detail feedback_answer feedback_status product_id (Foreign Key)

1.16. Blogs.

Entity Name:	Blogs
Description:	Users can rate products after experiencing them.
Attribute:	Blogid int (Primary Key) HomeStatus Content

1.17. ImageBlogs.

Entity Name:	ImageBlogs
Description:	The place where the images of the previous Blogs page appear to reinforce the quality of the products when sharing.
Attribute:	Blogid int (Primary Key) HomeStatus Content

1.18. UserClaims.

Entity Name:	UserClaims
Description:	Role that the user can get after being authorized
Attribute:	Id (Primary Key) UserId (Foreign Key) ClaimType ClaimValue

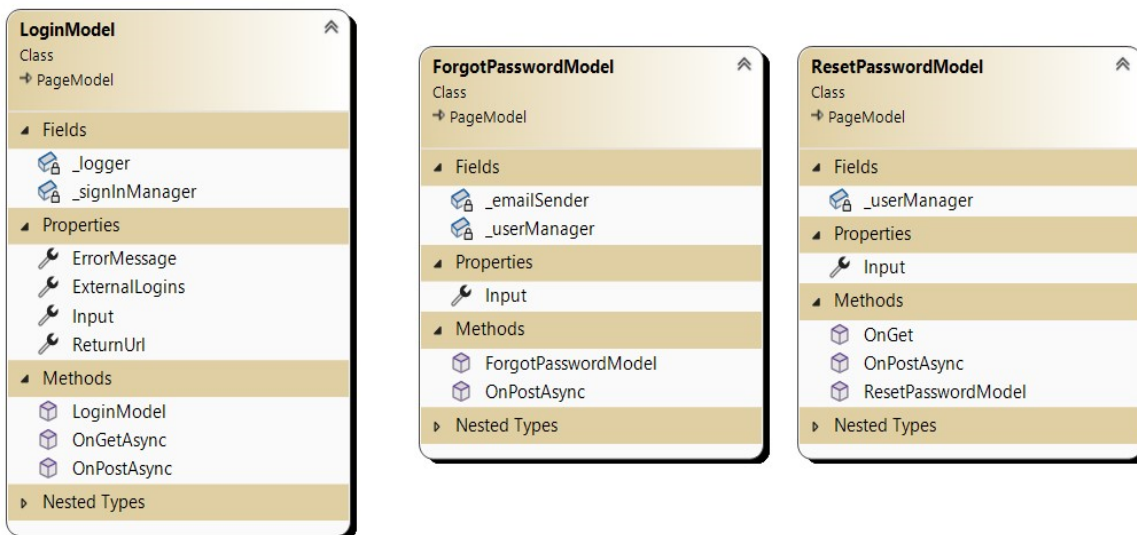
1.19. ImageProducts

Entity Name:	ImageProducts
Description:	Store images related to products to help buyers see the product in advance.
Attribute:	ImageProductId (Primary Key) ProductId (Foreign Key) ImageURL

III. Code Design

1. Login

a. Class Diagram



b. Class Specifications

Login class

No	Method	Description
01	login	This method allows the user to login to the system. Users need to enter Email, Password and click on the login button.

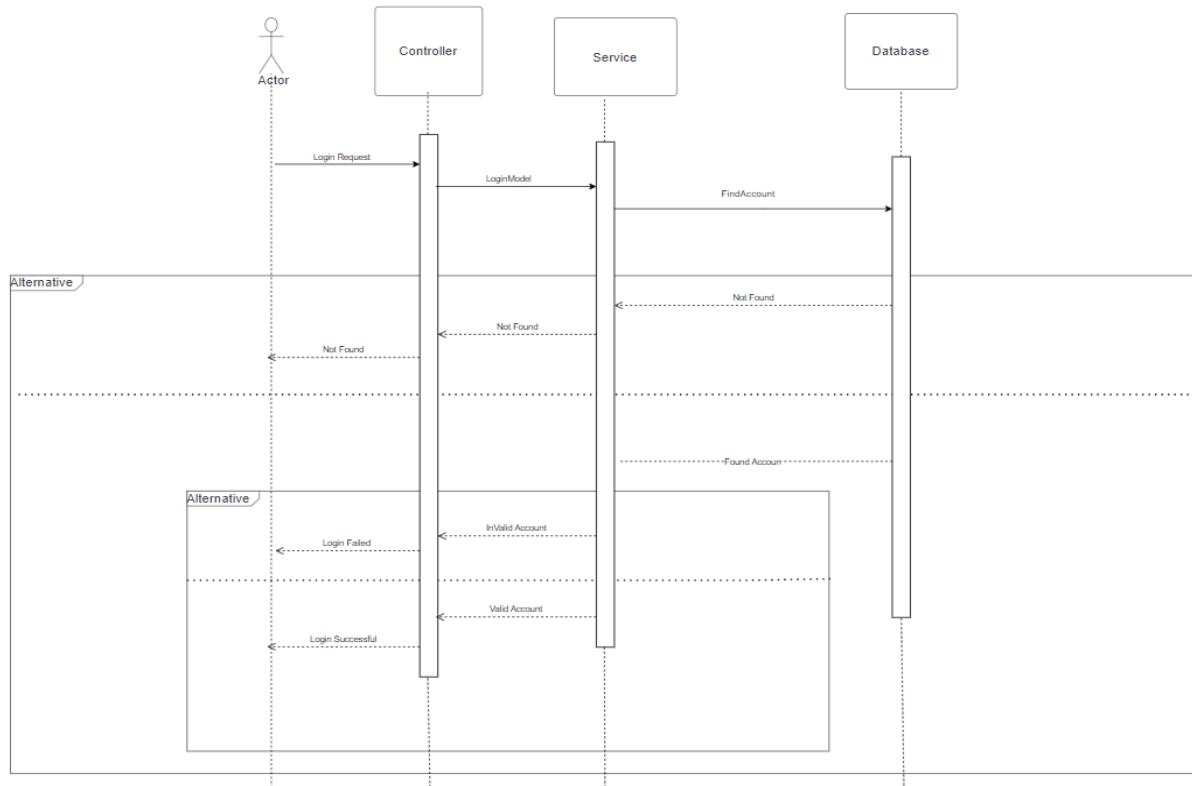
Forgot Password class

No	Method	Description
01	forgot password	This method allows the user to get back a new password if they forgot their password. Users need to enter email and click on the forgot password button after that user will get a new password in an email that the system sends to the user.

Reset Password class

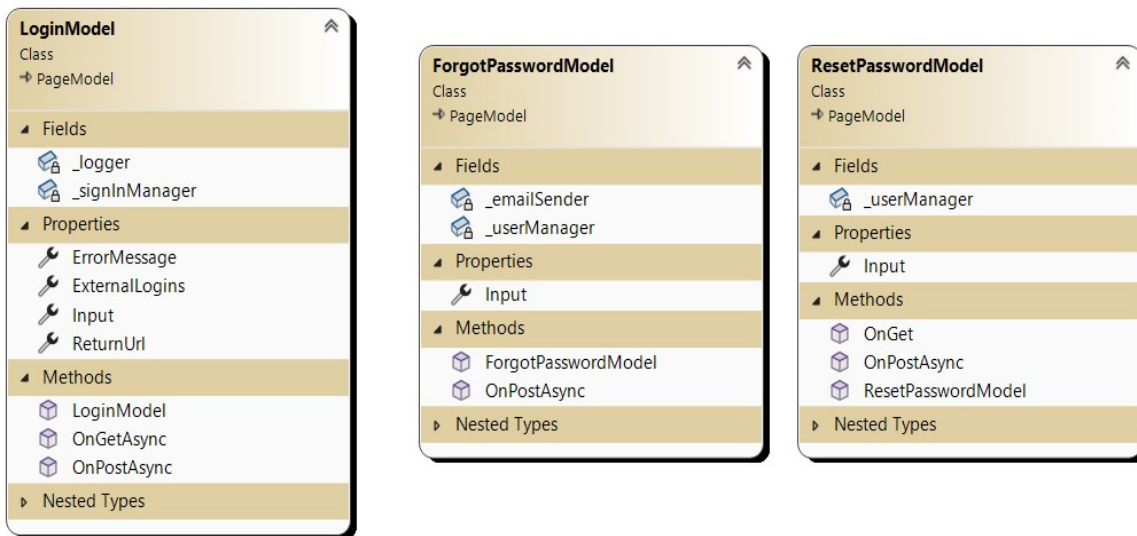
No	Method	Description
01	Reset password	This method allows the user to change their password. Users need to enter a password, re-password and click on the change password button.

c. Sequence Diagram(s)



2. Forgot Password

a. Class Diagram



b. Class Specifications

Login class

No	Method	Description
01	login	This method allows the user to login to the system. Users need to enter Email, Password and click on the login button.

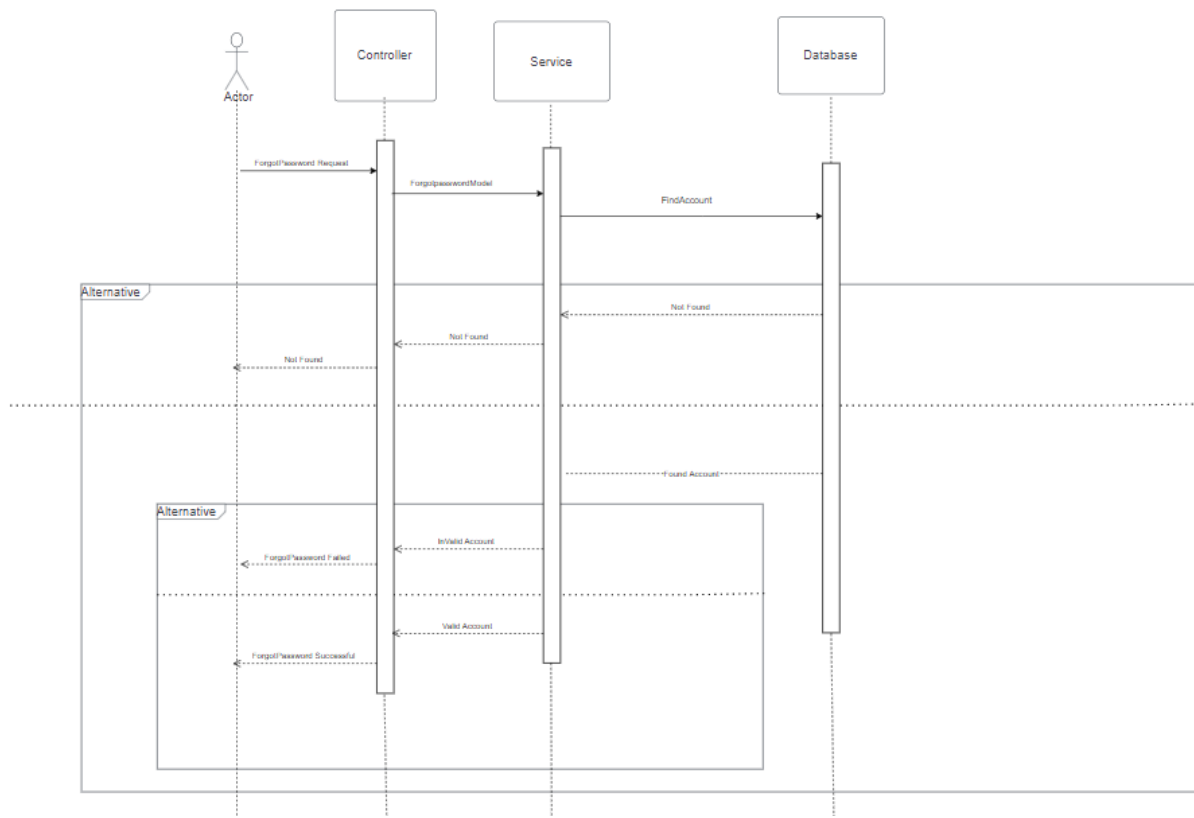
Forgot Password class

No	Method	Description
01	forgot password	This method allows the user to get back a new password if they forgot their password. Users need to enter email and click on the forgot password button after that user will get a new password in an email that the system sends to the user.

Reset Password class

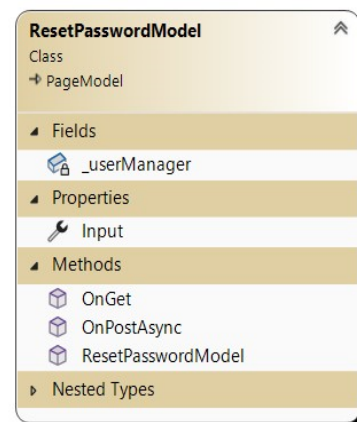
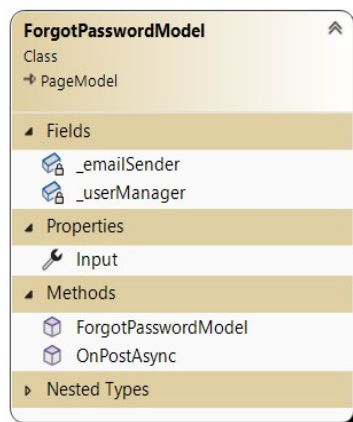
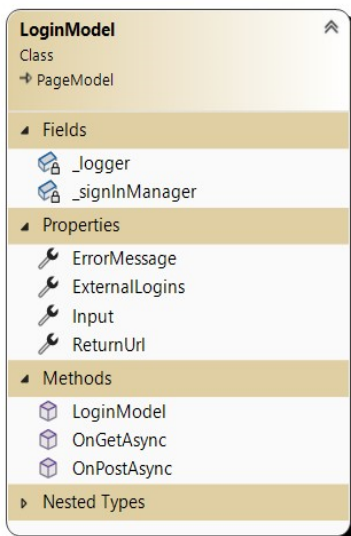
No	Method	Description
01	Reset password	This method allows the user to change their password. Users need to enter a password, re-password and click on the change password button.

a. Sequence Diagram(s)



3. Reset Password

a. Class Diagram



b. Class Specifications

Login class

No	Method	Description
01	login	This method allows the user to login to the system. Users need to enter Email, Password and click on the login button.

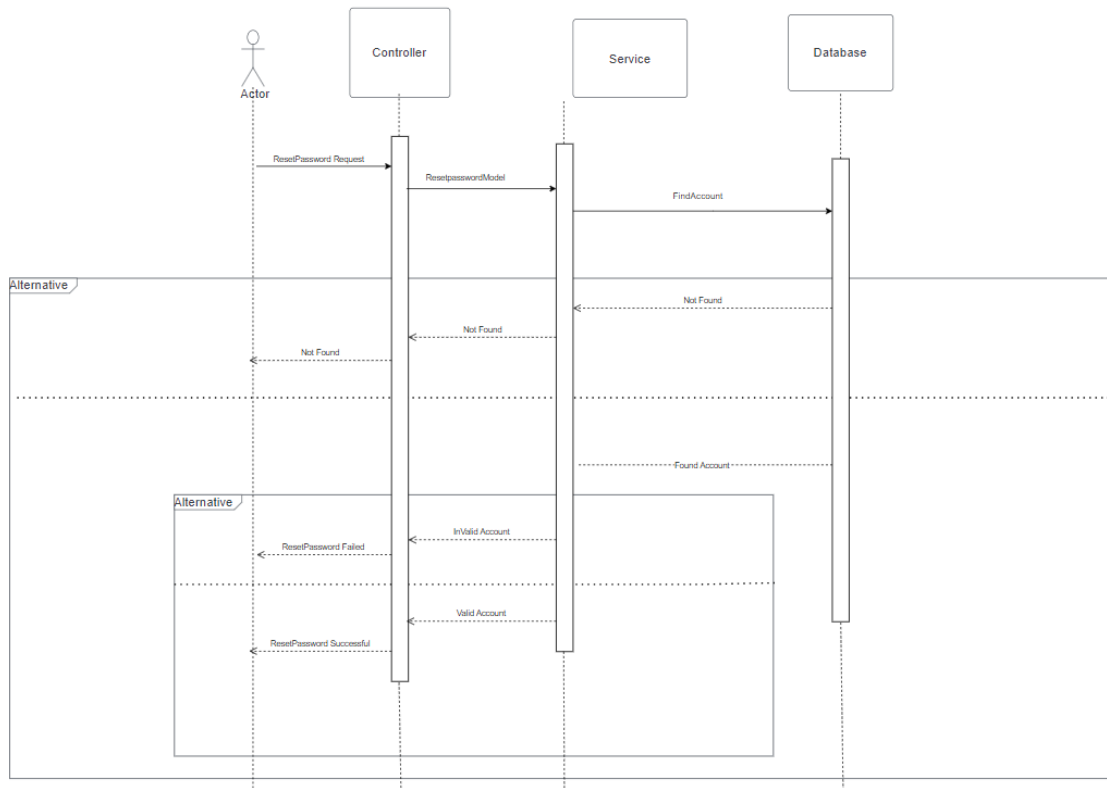
Forgot Password class

No	Method	Description
01	forgot password	This method allows the user to get back a new password if they forgot their password. Users need to enter email and click on the forgot password button after that user will get a new password in an email that the system sends to the user.

Reset Password class

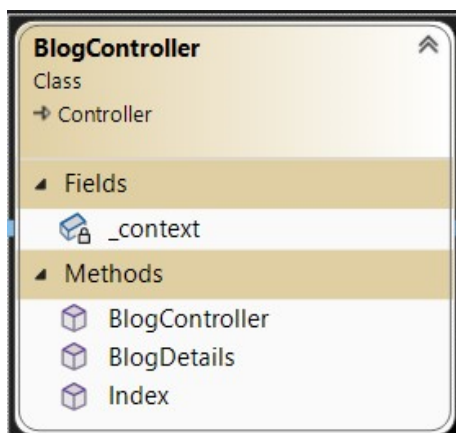
No	Method	Description
01	Reset password	This method allows the user to change their password. Users need to enter a password, re-password and click on the change password button.

a. Sequence Diagram(s)



4. Create new Blog

a. Class Diagram



b. Class Specifications

Create new course

No	Method	Description
01	Create new Blog	This method allows the admin or marketing expert to create a new course in the system

Edit Blog

No	Method	Description
01	Edit Blog	This method allows the admin or marketing to edit details of a blog with name, content, second content, and image. Admin or marketing click to View Blog and click update button and navigate to edit detail page, enter new information want to edit and click update to transfer to update image, after update image click update to finish.

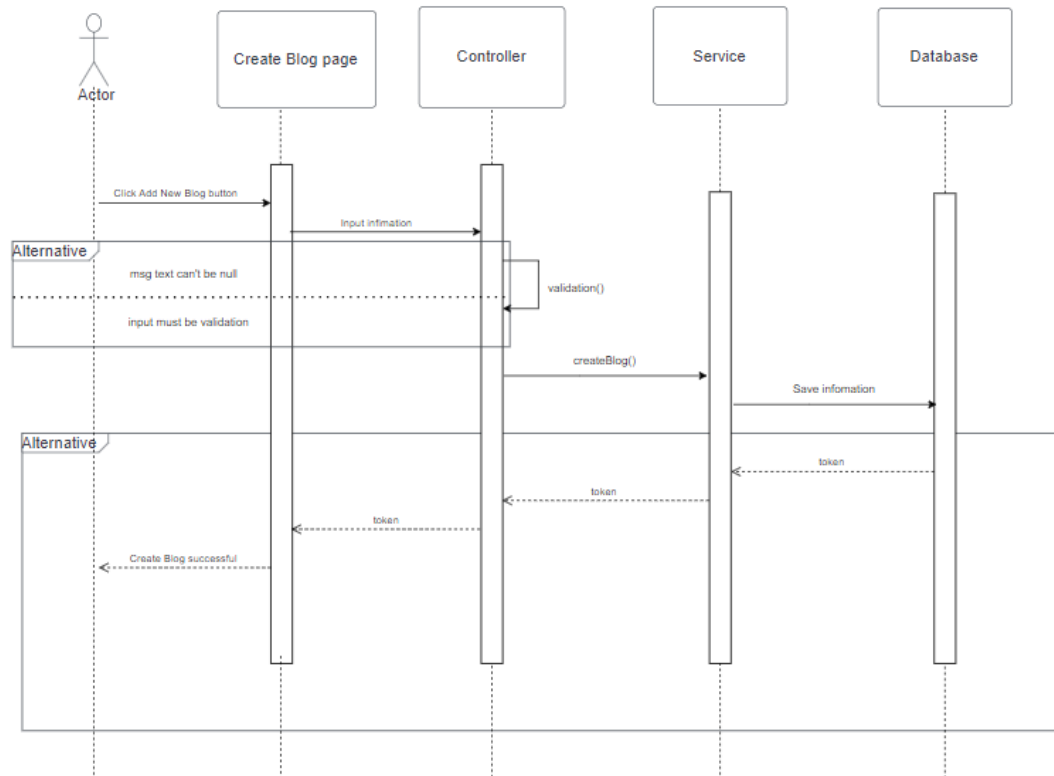
Delete Blog

No	Method	Description
01	Delete Blog	This method allows the admin or marketing to delete a blog. Admin or marketing click to View Blog and click on Delete to remove Blog out of webpage

Change status of Blog

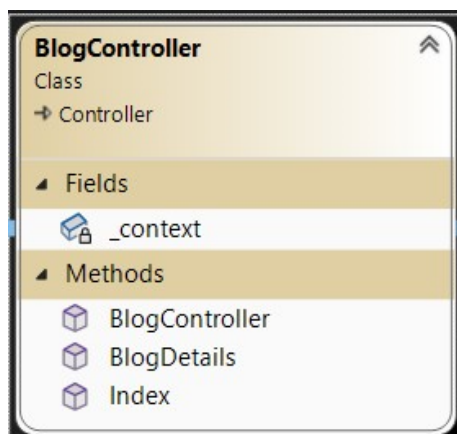
No	Method	Description
	Change status of Blog	This method allows the admin or marketing to change the active status of the Blog. Admin or marketing click to View Blog and click to the status of a Blog they wanna change status, if the status is Enable it will change to Disable and vice versa.

a. Sequence Diagram(s)



5. Edit Blog

a. Class Diagram



b. Class Specifications

Create new course

No	Method	Description
01	Create new Blog	This method allows the admin or marketing expert to create a new course in the system

Edit Blog

No	Method	Description
01	Edit Blog	This method allows the admin or marketing to edit details of a blog with name, content, second content, and image. Admin or marketing click to View Blog and click update button and navigate to edit detail page, enter new information want to edit and click update to transfer to update image, after update image click update to finish.

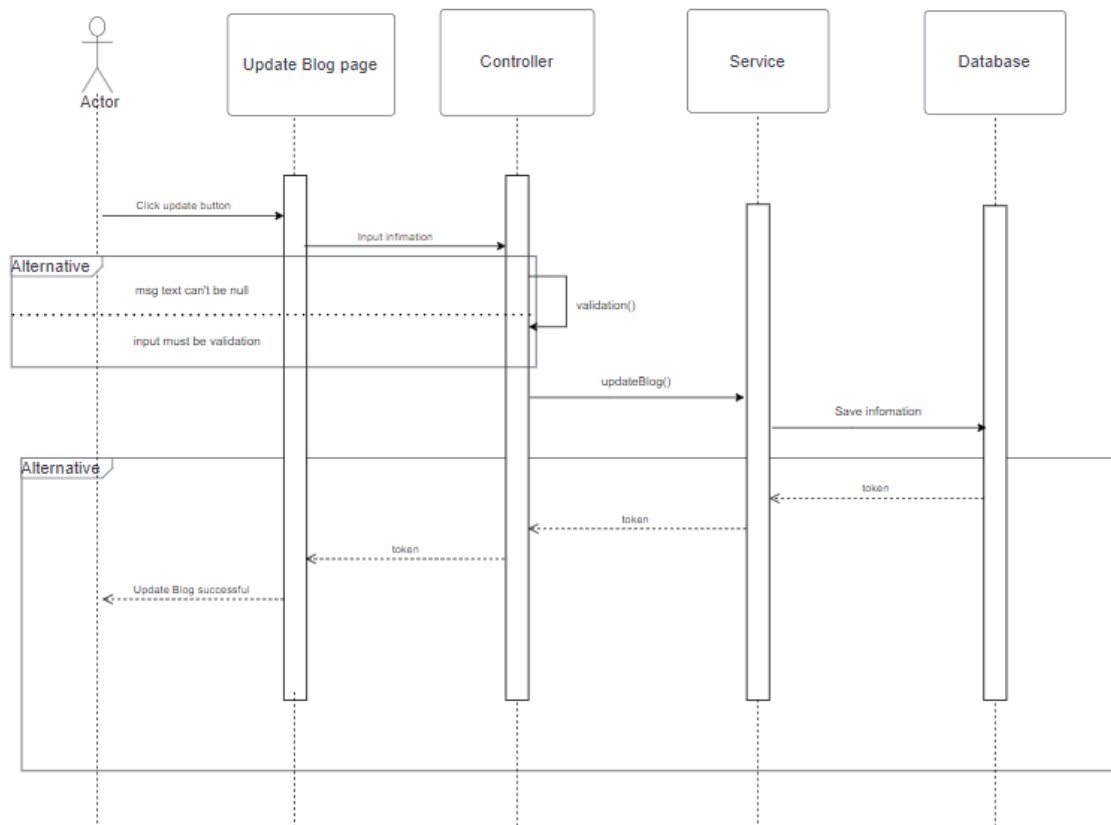
Delete Blog

No	Method	Description
01	Delete Blog	This method allows the admin or marketing to delete a blog. Admin or marketing click to View Blog and click on Delete to remove Blog out of webpage

Change status of Blog

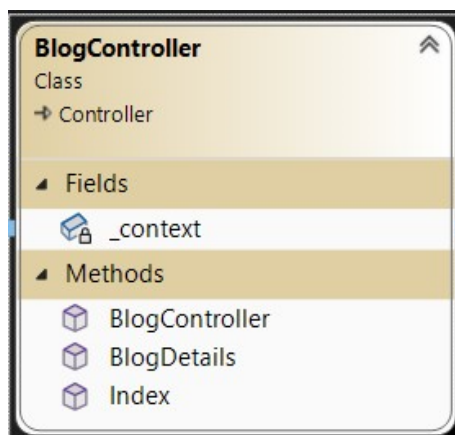
No	Method	Description
01	Change status of Blog	This method allows the admin or marketing to change the active status of the Blog. Admin or marketing click to View Blog and click to the status of a Blog they wanna change status, if the status is Enable it will change to Disable and vice versa.

a. Sequence Diagram(s)



6. Delete Blog

a. Class Diagram



b. Class Specifications

Create new course

No	Method	Description
01	Create new Blog	This method allows the admin or marketing expert to create a new course in the system

Edit Blog

No	Method	Description
01	Edit Blog	This method allows the admin or marketing to edit details of a blog with name, content, second content, and image. Admin or marketing click to View Blog and click update button and navigate to edit detail page, enter new information want to edit and click update to transfer to update image, after update image click update to finish.

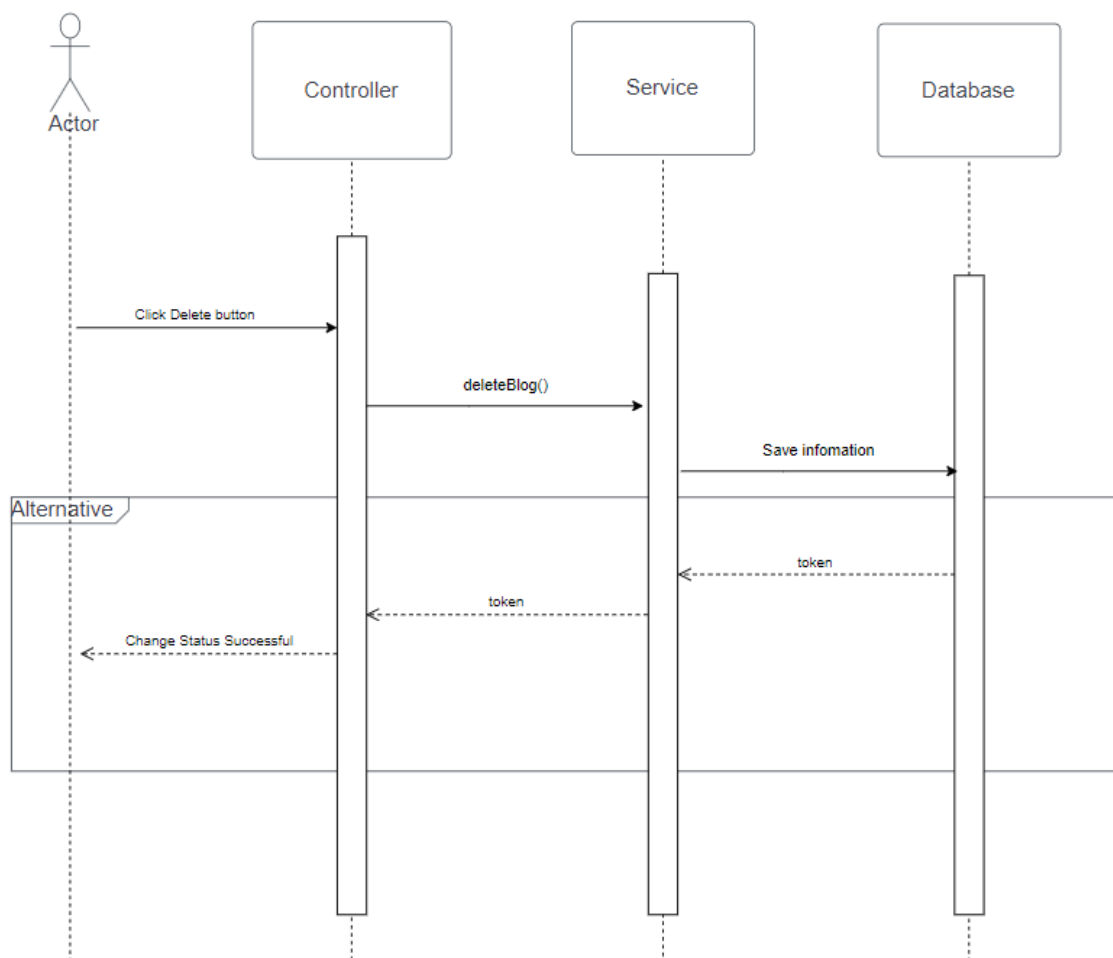
Delete Blog

No	Method	Description
01	Delete Blog	This method allows the admin or marketing to delete a blog. Admin or marketing click to View Blog and click on Delete to remove Blog out of webpage

Change status of Blog

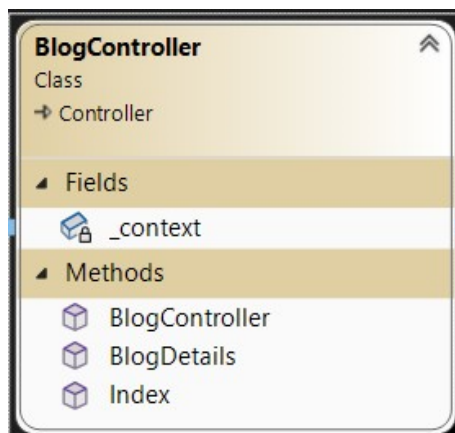
No	Method	Description
01	Change status of Blog	This method allows the admin or marketing to change the active status of the Blog. Admin or marketing click to View Blog and click to the status of a Blog they wanna change status, if the status is Enable it will change to Disable and vice versa.

a. Sequence Diagram(s)



7. Change status of Blog

a. Class Diagram



b. Class Specifications

Create new course

No	Method	Description
01	Create new Blog	This method allows the admin or marketing expert to create a new course in the system

Edit Blog

No	Method	Description
01	Edit Blog	This method allows the admin or marketing to edit details of a blog with name, content, second content, and image. Admin or marketing click to View Blog and click update button and navigate to edit detail page, enter new information want to edit and click update to transfer to update image, after update image click update to finish.

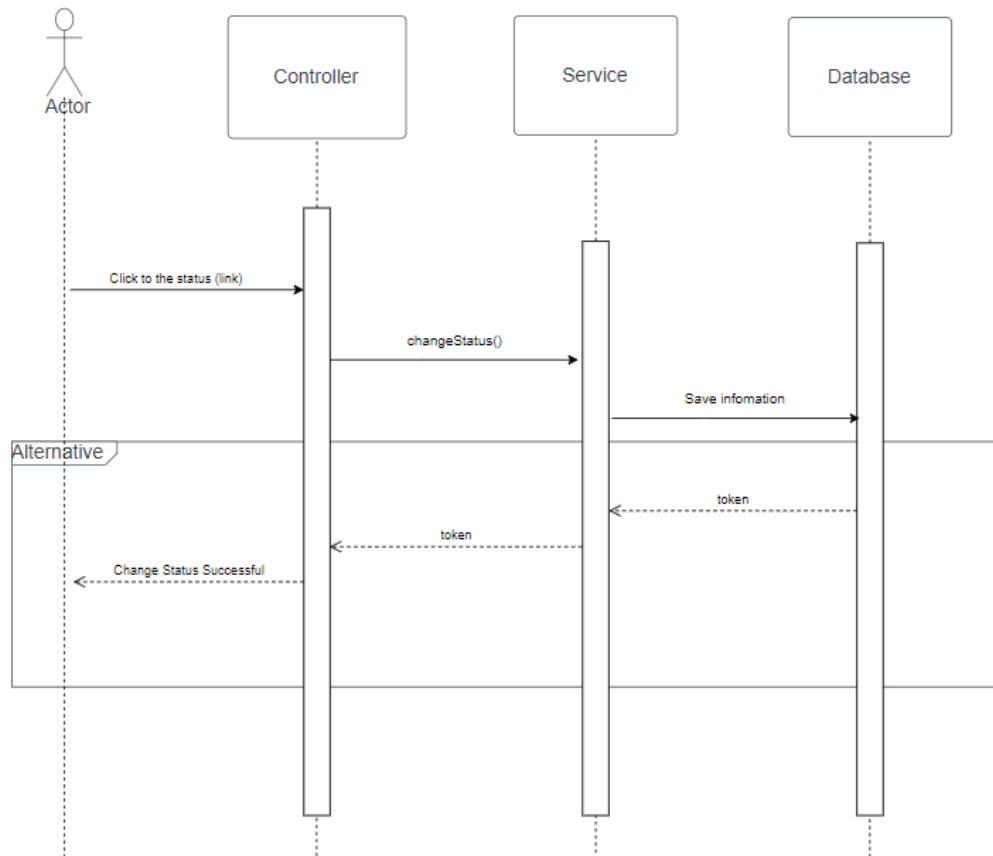
Delete Blog

No	Method	Description
01	Delete Blog	This method allows the admin or marketing to delete a blog. Admin or marketing click to View Blog and click on Delete to remove Blog out of webpage

Change status of Blog

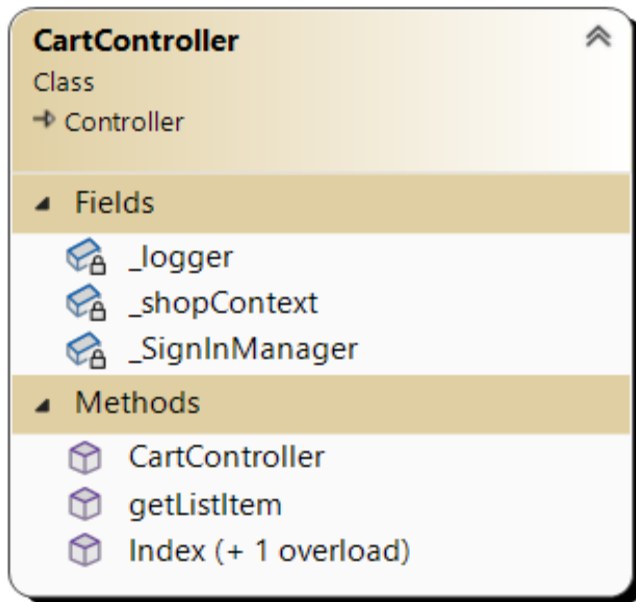
No	Method	Description
01	Change status of Blog	This method allows the admin or marketing to change the active status of the Blog. Admin or marketing click to View Blog and click to the status of a Blog they wanna change status, if the status is Enable it will change to Disable and vice versa.

a. Sequence Diagram(s)



8. Cart Contact

a. Class Diagram



b. Class Specifications

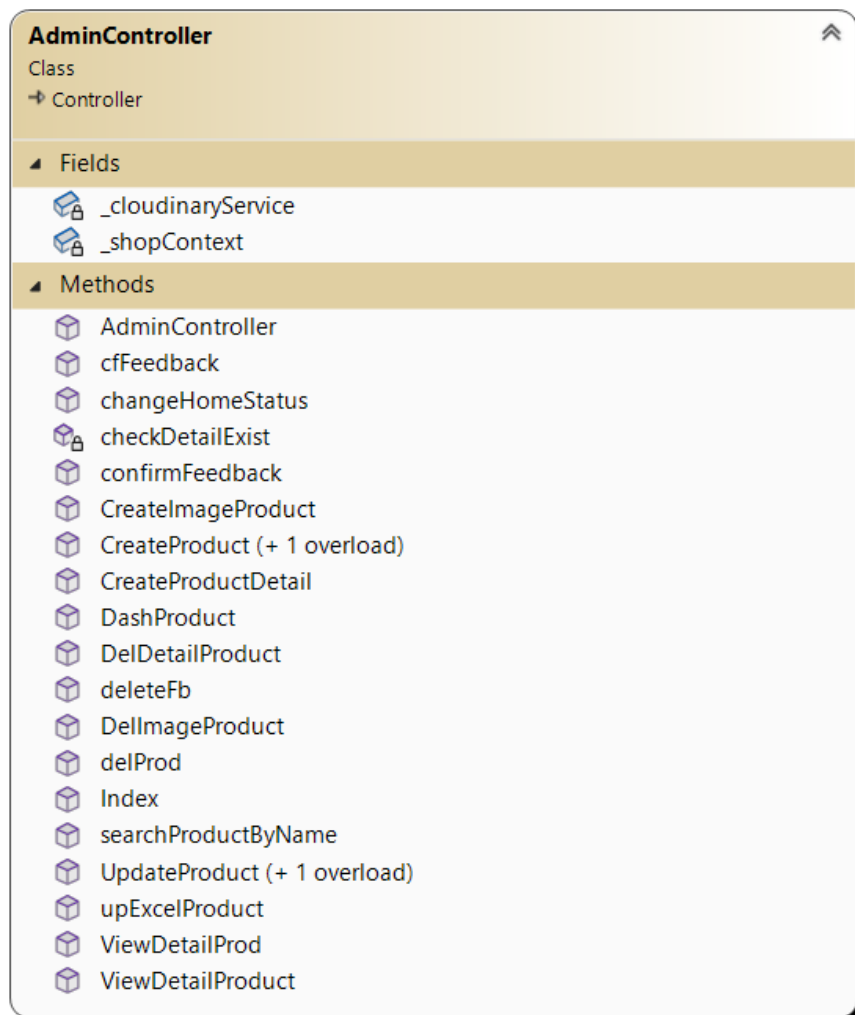
No	Method	Description
01	getListItem()	The user is logged in, the information will be retrieved from the database based on the UserId. If you are not logged in or the information is not in the database, the shopping cart information is obtained from the cookie. The list of line items from the database and cookies is then concatenated and sorted by CartItemId before returning.

No	Method	Description
01	CartController	Allows users to manage their cart, including adding and removing line items. When the user logs in, the cart information will be saved to the database. In

		case the user is not logged in or does not have an account, the shopping cart information will be saved in a cookie to maintain the cart information for the next visit.
--	--	--

9. Admin manager

a. Class Diagram



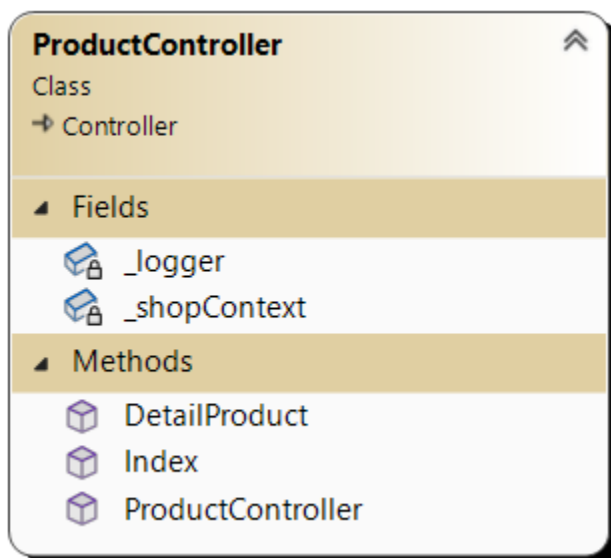
b. Class Specifications

confirm feedback

No	Method	Description
01	cfFeedback()	The user is logged in, the information will be retrieved from the database based on the UserId. If you are not logged in or the information is not in the database, the shopping cart information is obtained from the cookie. The list of line items from the database and cookies is then concatenated and sorted by CartItemId before returning.

10. Product

a. Class diagram



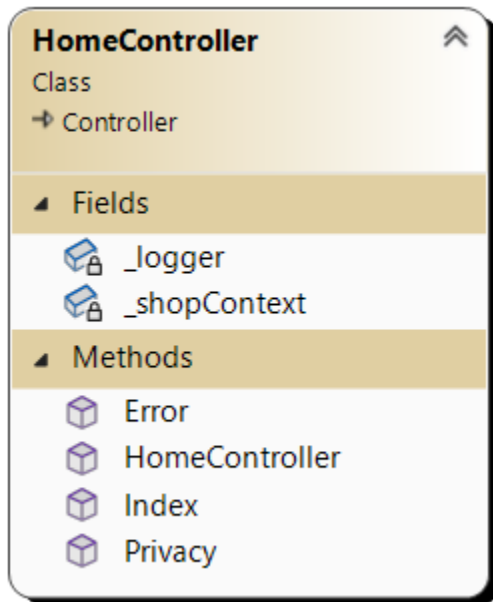
b. Class Specifications

No	Method	Description
----	--------	-------------

01	detailsProduct()	Help the application display a product detail page to the user, with complete product information including related images
----	------------------	--

11.Home

a. Class Diagram



b. Class Specifications

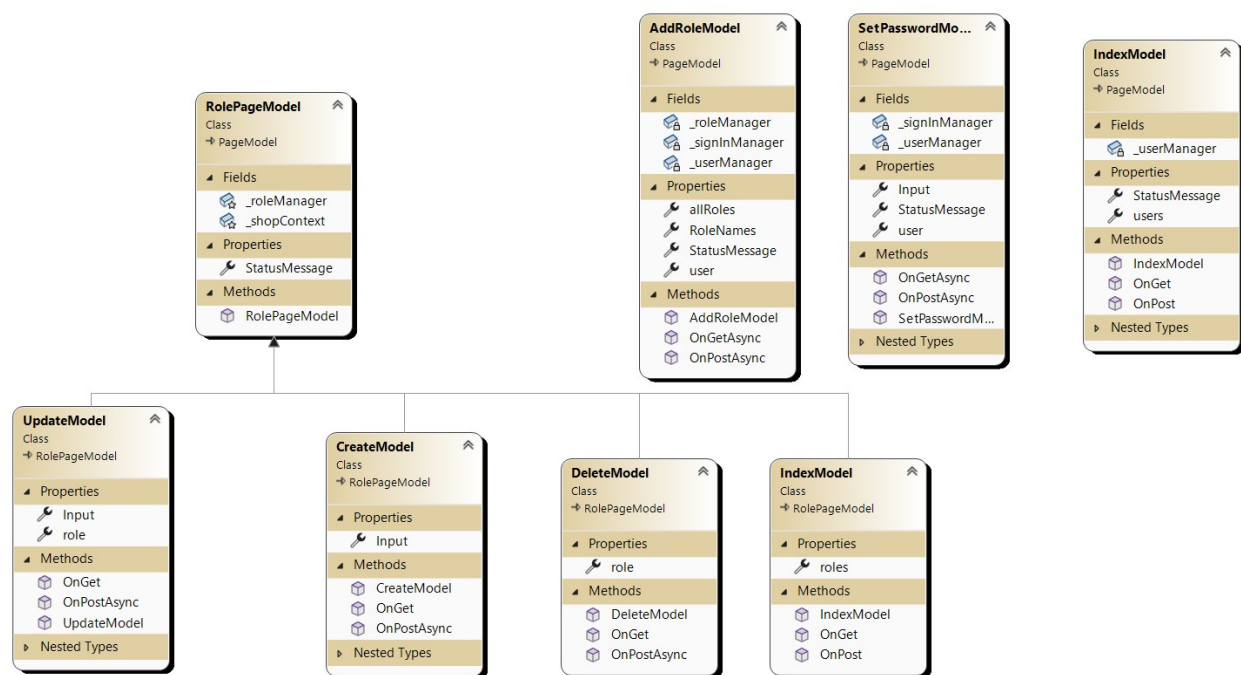
No	Method	Description
01	Privacy()	This method is used when the application has an error, and it returns a View named "Error". This view may contain error information, error codes, debugging information, or any other information useful for identifying and handling errors in the application.

HomeController

No	Method	Description
01	HomeController()	Gets a list of products from the database based on a condition and then returns this list to a page view.

12.Delete Role

a. Class diagram



b. Class Specifications

Update

No	Method	Description
01	UpdateModel()	Allow admin to update roles of customers according to selected accounts, admin, seller, marketing

Create

No	Method	Description
01	CreateModel()	Allow admin to create new Role

Delete

No	Method	Description
01	DeleteModel()	Allow admin to create delete Role from role page

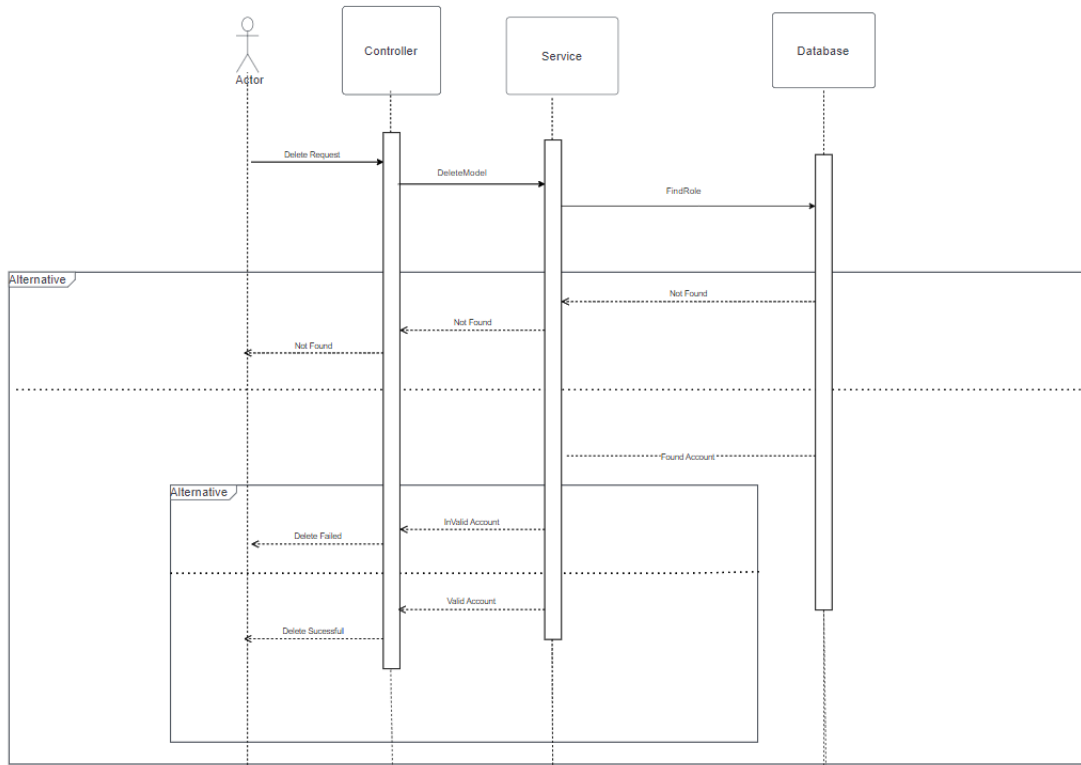
AddRole

No	Method	Description
01	AddRoleModel()	Allow admin to new role for a lot of respective function

Set password

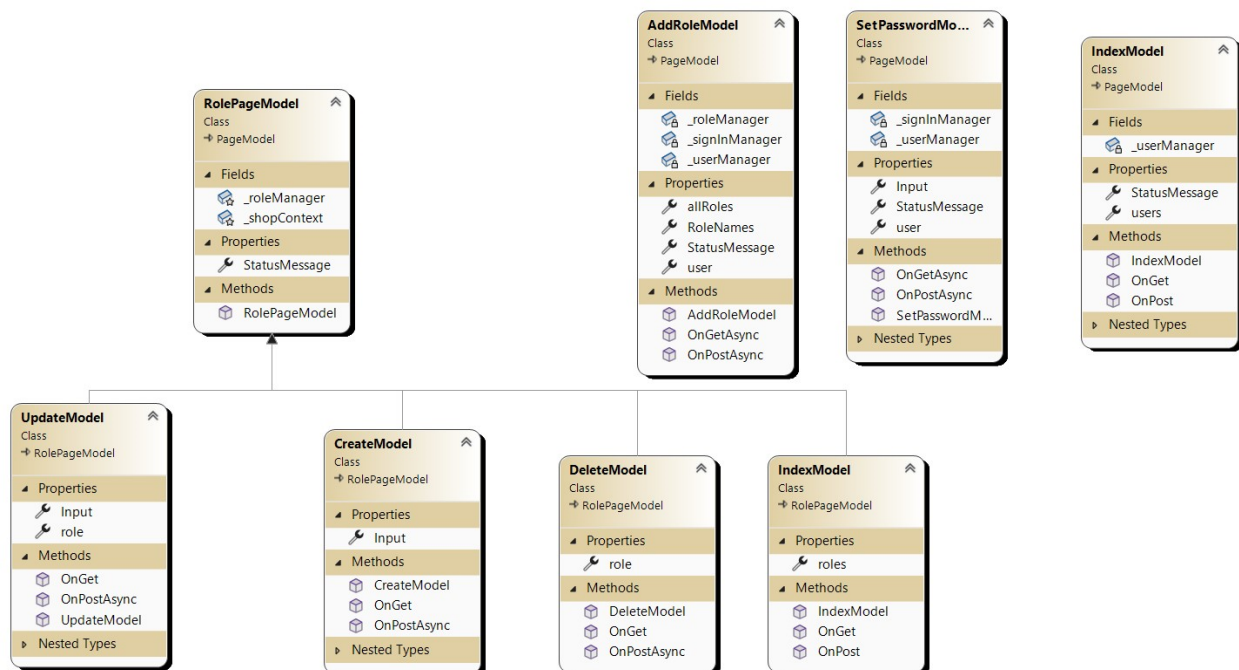
No	Method	Description
01	SerPasswordModel()	Allow admin to set password for customer,user.

c. Sequences diagram



13.Update Role

a. Class diagram



b. Class Specifications

Update

No	Method	Description
01	UpdateModel()	Allow admin to update roles of customers according to selected accounts, admin, seller, marketing

Create

No	Method	Description
01	CreateModel()	Allow admin to create new Role

Delete

No	Method	Description
01	DeleteModel()	Allow admin to create delete Role from role page

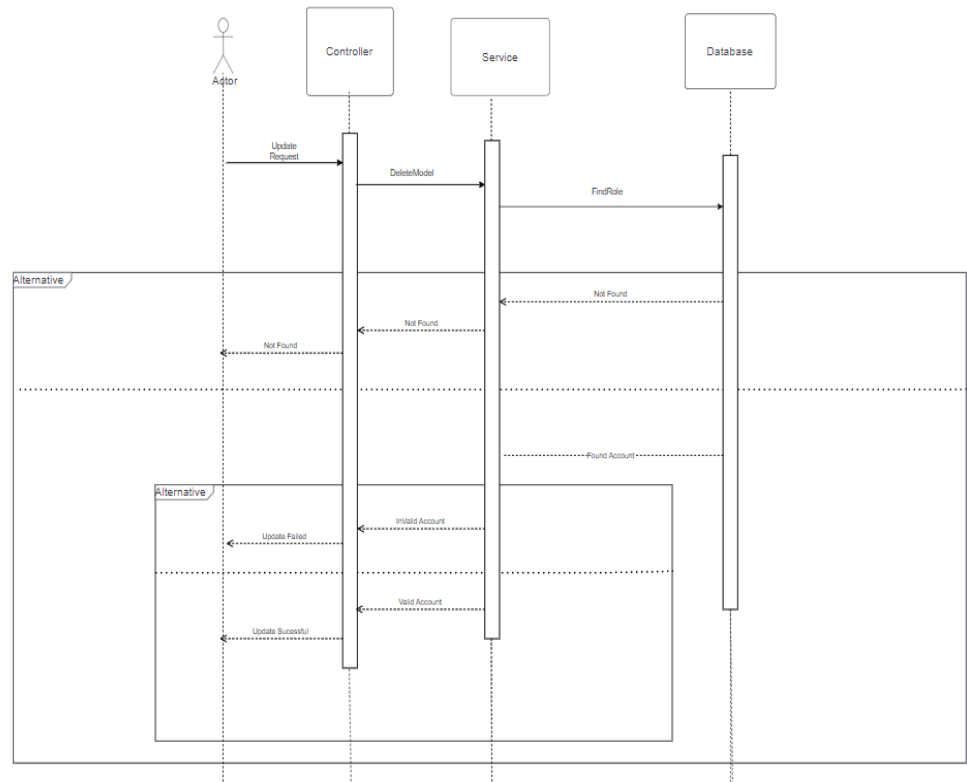
AddRole

No	Method	Description
01	AddRoleModel()	Allow admin to new role for a lot of respective function

Set password

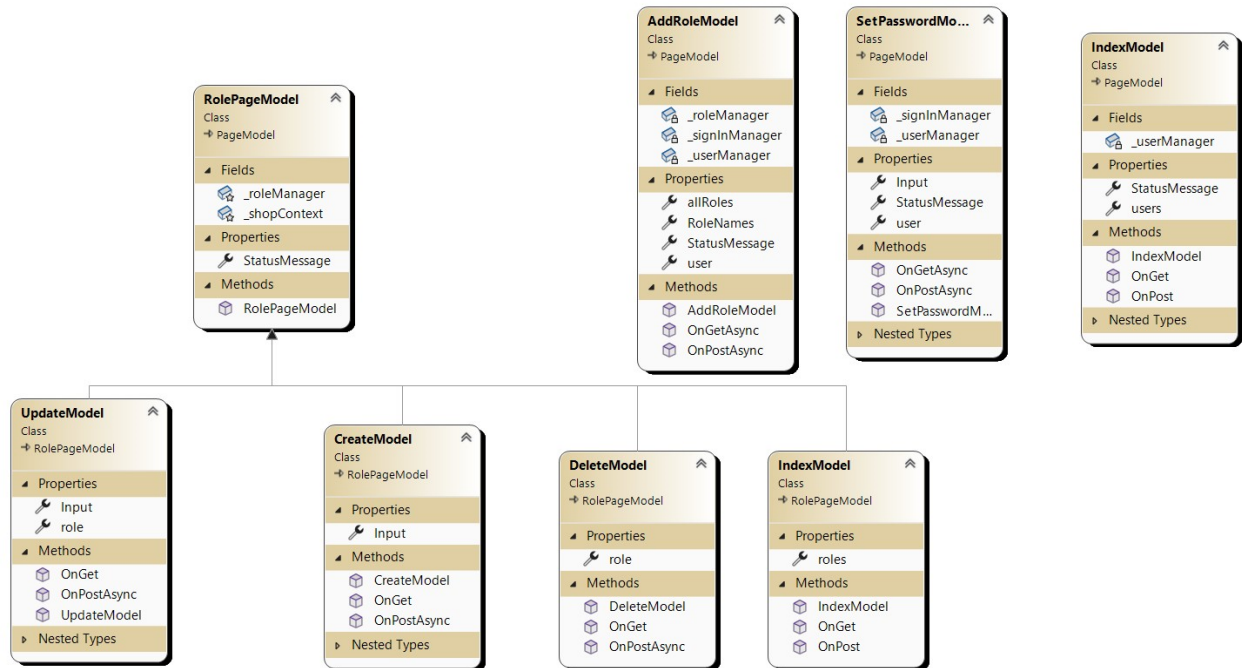
No	Method	Description
01	SerPasswordModel()	Allow admin to set password for customer,user.

c. Sequences diagram



14.Add Role

a. Class diagram



b. Class Specifications

Update

No	Method	Description
01	UpdateModel()	Allow admin to update roles of customers according to selected accounts, admin, seller, marketing

Create

No	Method	Description
01	CreateModel()	Allow admin to create new Role

Delete

No	Method	Description
01	DeleteModel()	Allow admin to create delete Role from role page

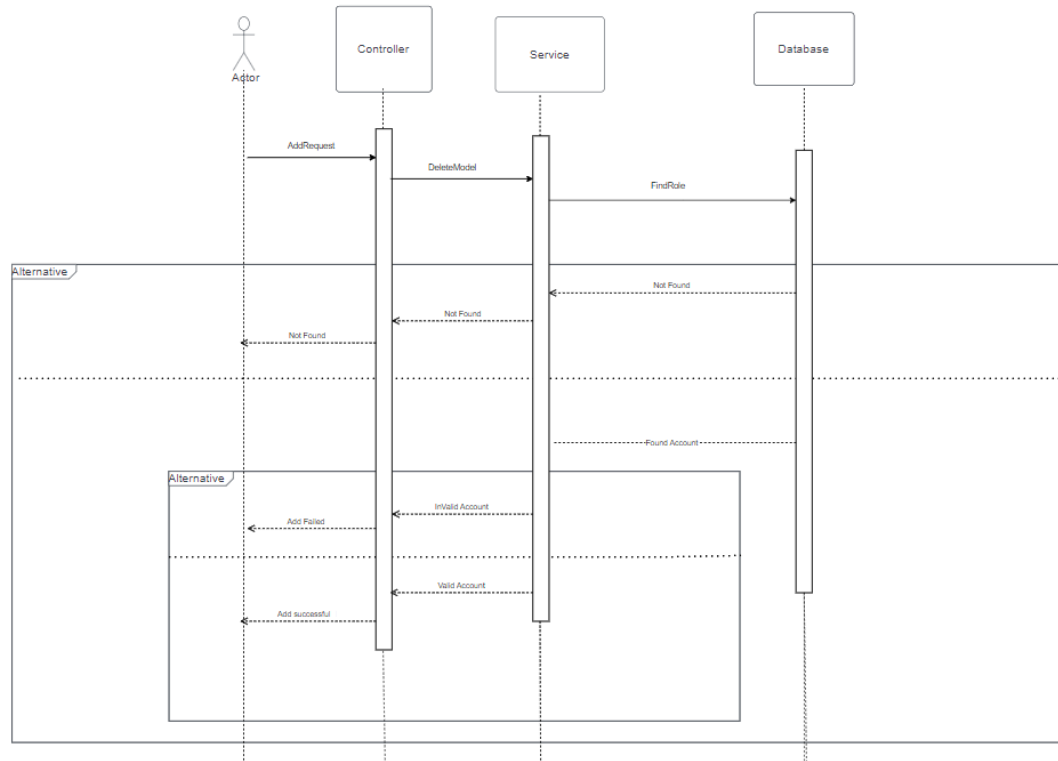
AddRole

No	Method	Description
01	AddRoleModel()	Allow admin to new role for a lot of respective function

Set password

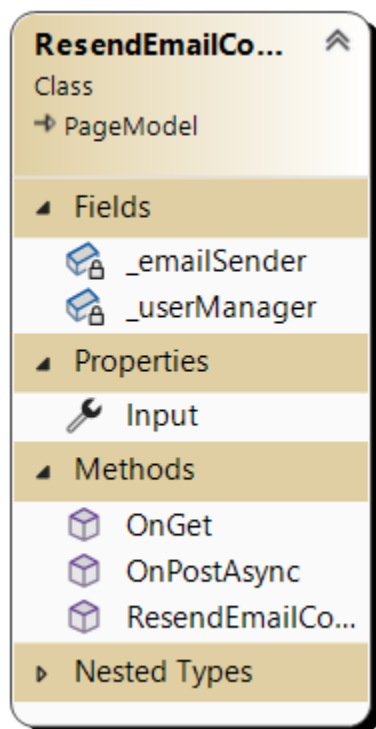
No	Method	Description
01	SerPasswordModel()	Allow admin to set password for customer,user.

c. Sequences diagram



15. ResentEmail

a. Class diagram



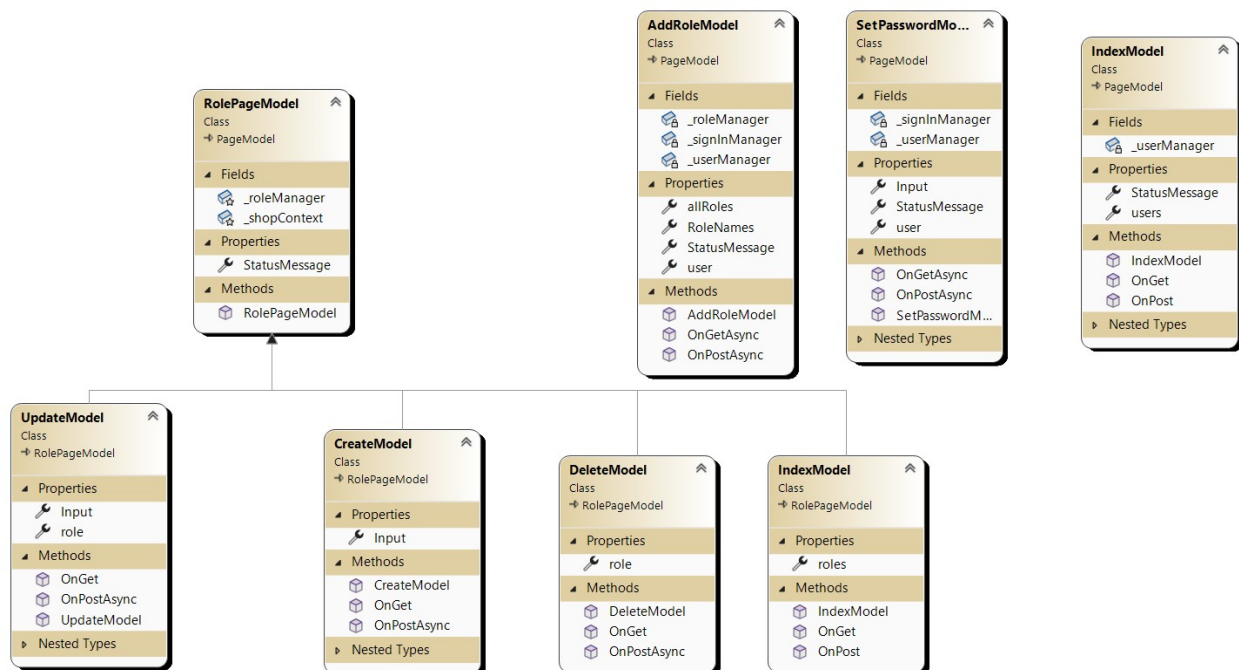
b. Class Specifications

Resent Email

No	Method	Description
01	ResentEmailModel()	The system will send a confirmation email to the user when using the functions related to creating and resetting the account password.

16. SetPassword

a. Class diagram



b. Class Specifications

Update

No	Method	Description
01	UpdateModel()	Allow admin to update roles of customers according to selected accounts, admin, seller, marketing

Create

No	Method	Description
01	CreateModel()	Allow admin to create new Role

Delete

No	Method	Description
01	DeleteModel()	Allow admin to create delete Role from role page

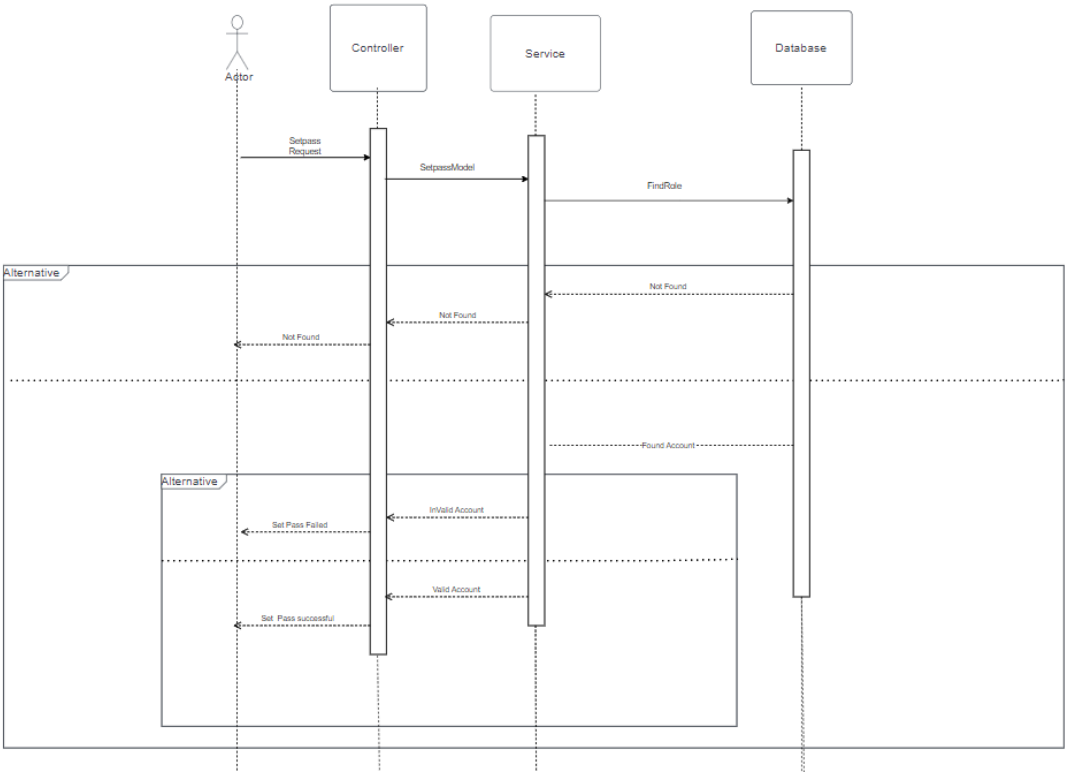
AddRole

No	Method	Description
01	AddRoleModel()	Allow admin to new role for a lot of respective function

Set password

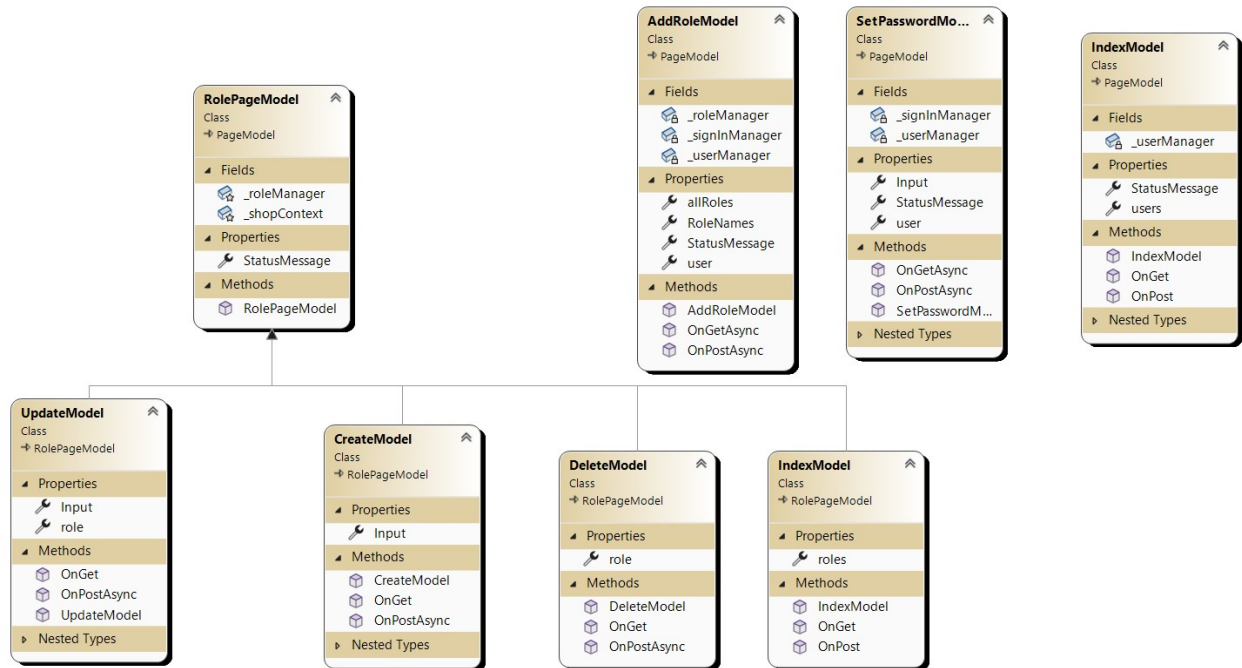
No	Method	Description
01	SerPasswordModel()	Allow admin to set password for customer,user.

c. Sequences diagram



17. Set Password

a. Class diagram



b. Class Specifications

Update

No	Method	Description
01	UpdateModel()	Allow admin to update roles of customers according to selected accounts, admin, seller, marketing

Create

No	Method	Description
01	CreateModel()	Allow admin to create new Role

Delete

No	Method	Description
01	DeleteModel()	Allow admin to create delete Role from role page

AddRole

No	Method	Description
01	AddRoleModel()	Allow admin to new role for a lot of respective function

Set password

No	Method	Description
01	SerPasswordModel()	Allow admin to set password for customer,user.

c. Sequences diagram

