PHARMACY SYSTEM

Medicine

Contents

Pharmacy System Project

A graduation project dissertation by:

[Abdulhadi Mustafa]

[**Maryam Mohammed**]

[Amna Ali]

[Amna Ali]

[Mohamed Hussien]

Supervised by: ***Eng. Ahmed El gazar***

SOFTWARE DEVELOPMENT LIFE CYCLE

Introduction

The Pharmacy System Project aims to simplify and secure the online purchasing process, providing users with a reliable platform to access essential medications without visiting a pharmacy. By leveraging a structured software development approach, this project enhances the functionality of online medicine browsing, secure payment processing, and comprehensive inventory management for pharmacy administrators. The system is designed to accommodate user and admin roles, ensuring efficient management of orders, medicine categories, and user accounts while prioritizing data security, scalability, and ease of use for all stakeholders.

➢ Business Need

The pharmacy system is designed to address the need for a more convenient and secure method of purchasing medicines online. In today's world, customers are looking for faster ways to buy medicine without physically visiting a pharmacy. Furthermore, pharmacy administrators require a system that allows them to manage inventories, categorize medicines, track orders, and handle payments.

Key Business Objectives:

- Online Medicine Purchase: Enable users to browse, search, and buy medicines from the comfort of their homes.
- Secure Payments: Integrate secure payment options for transactions.
- Inventory Management: Provide administrators with tools to manage medicines, including categories, prices, and stock.
- User Roles: Implement different roles such as admin and non-admin users to enhance system security and control.
- Authentication & Authorization: Ensure a secure login system that protects user data and prevents unauthorized access.

System Requirements

➢ Functional Requirements:

- User Registration & Login: Users can register, log in, and maintain their profile. By default, new users are non-admin and need activation by an admin to gain admin privileges.
- Medicine Browsing: Users should be able to browse and filter medicines by category, name, or price.
- Order Management: Users can add medicines to their shopping bag, adjust quantities, and place orders.
- Admin Functionality: Admins can manage medicines and categories, view orders, and promote regular users to admin roles.
- Payment Integration: The system should support secure payment gateways like Paymob for user transactions.
- Validation and Error Handling: Forms should have built-in validation for user input (e.g., password confirmation, email, and age validation).

➢ Non-Functional Requirements:

- Performance: The system should handle concurrent users, ensuring smooth browsing and checkout experiences.
- Security: Secure personal and payment data through encrypted connections and authentication methods.
- Scalability: The system should be scalable, able to handle an increasing number of users, medicines, and orders as the pharmacy grows.
- Usability: The system should be intuitive, making it easy for both admins and customers to navigate and use its features.

Analysis

➢ Use Case Analysis:

- **Customer Use Case:**
  - Register and log in to the system.
  - Browse medicines and view details.
  - Add medicines to the shopping bag.
  - Place an order and make secure payments.
  - View order history.
- **Admin Use Case:**
  - Manage medicine categories (add, update, or remove).
  - Manage medicine details (price, stock, expiry dates, etc.).
  - Promote users from non-admin to admin.
  - Track customer orders and oversee system usage.

➢ Data Analysis:

The key data entities in the project include:

- User Data: Information about registered users, including their role (admin or non-admin), login credentials, and age.
- Medicine Data: Each medicine has details such as its name, price, production date, expiry date, and category.
- Order Data: Captures the details of user orders, including medicine quantity, total cost, and associated user information.
- Category Data: Medicines are grouped into categories, which help in organizing and filtering medicines.

➢ Class Analysis:

The system includes several classes that represent core functionalities:

- ApplicationUser: Inherits from IdentityUser to manage authentication and user-specific data such as age and shopping bag.
- Bag: Stores all user orders and links them to the respective users.
- Category: Holds information about medicine categories such as name, budget, and associated medicines.
- Medicine: Represents a medicine entity with attributes like name, price, production date, expiry date, and category.
- Order: Stores order details like name, cost, quantity, and associated bag.
- DTOs: Used to manage data transfer between models and the user interface, ensuring the correct data structure for registration, login, and user information.

Design

➢ Architecture Design:

Layered Architecture: The project follows a multi-layered architecture with separation of concerns:

- Presentation Layer: Handles UI, forms, and user interactions. This layer ensures that users can browse and place orders.
- Business Logic Layer: Contains logic for user authentication, registration, and managing orders and medicines.
- Data Access Layer: Manages communication with the SQL Server database, ensuring the correct storage of user, medicine, and order data.

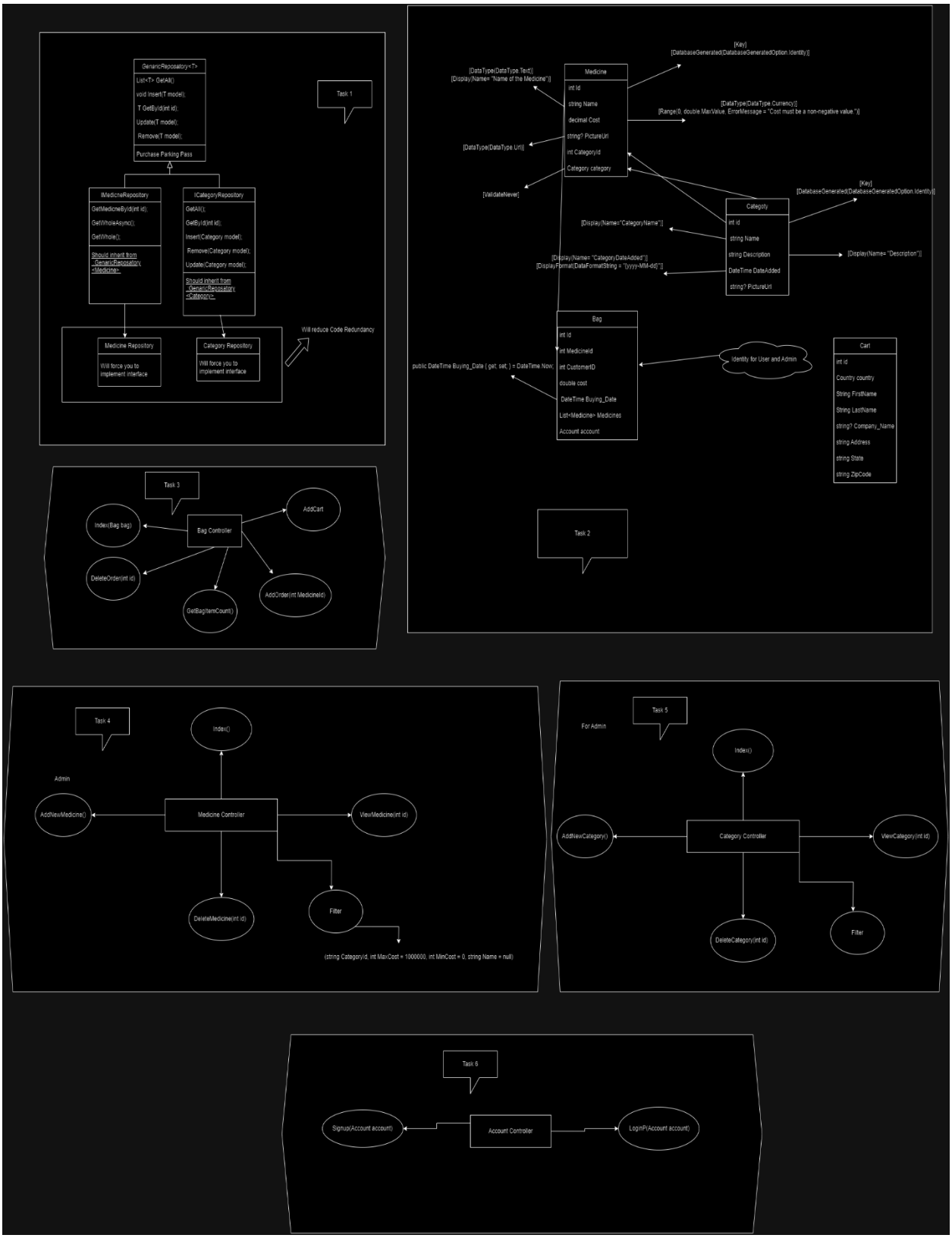
➢ Class Design:

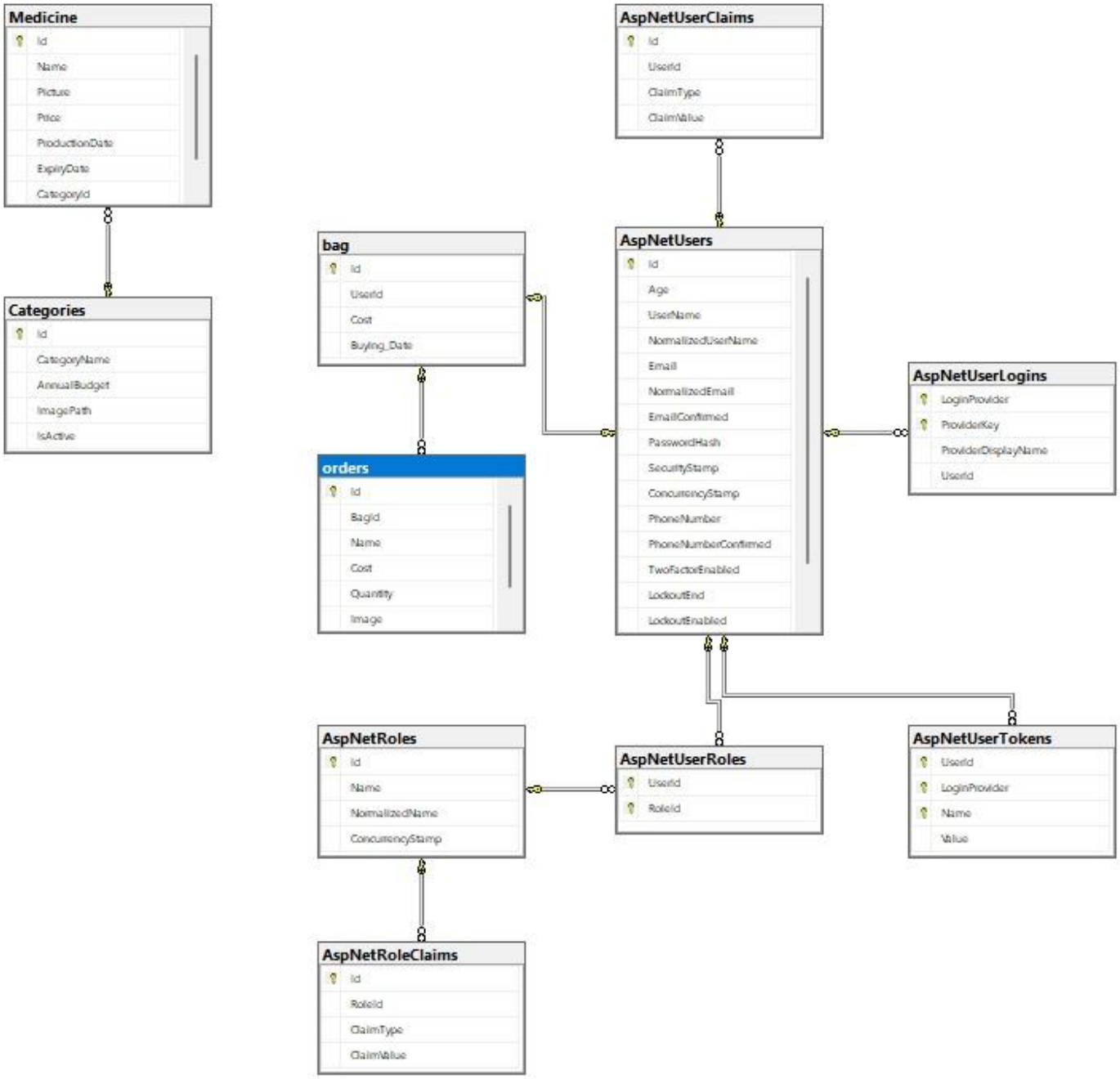
- **Application User:** Extends the Identity User class to add age and bag details for a user.
- **Bag Class:** Contains a list of Order objects and is linked to the Application User. It also tracks the cost and purchase date of the order.
- **Category Class:** Used to organize medicines into manageable categories with validation to ensure valid data entry.
- **Medicine Class:** Ensures medicines are properly categorized, with validations on production and expiry dates.
- **Order Class:** Tracks the medicines, quantity, and cost within each order.

➢ Database Design

Relational Database (SQL Server):

- Tables for Users, Medicines, Orders, Categories, and Bags.
- Relationships between entities such as User and Bag, Bag and Order, Category and Medicine.


GenaricReposatory<T>
List<T> GetAll();
void Insert(T model);
T GetById(int id);
Update(T model);
Remove(T model);
Purchase Parking Pass
Task 1
IMedicineRepository
GetMedicineById(int id);
GetWholeAsync();
GetWhole();
Should inherit from :GenaricReposatory <Medicine>
ICategoryRepository
GetAll();
GetById(int id);
Insert(Category model);
Remove(Category model);
Update(Category model);
Should inherit from :GenaricReposatory <Category>
Medicine Repository
Will force you to implement interface
Category Repository
Will force you to implement interface
Will reduce Code Redundancy
Medicine
int Id
string Name
decimal Cost
string? PictureUrl
int CategoryId
Category category
[Key]
[DatabaseGenerated(DatabaseGeneratedOption.Identity)]
[DataType(DataType.Text)]
[Display(Name= "Name of the Medicine")]
[DataType(DataType.Currency)]
[Range(0, double.MaxValue, ErrorMessage = "Cost must be a non-negative value.")]
[DataType(DataType.Url)]
[ValidateNever]
Categoty
int id
string Name
string Description
DateTime DateAdded
string? PictureUrl
[Display(Name="CategoryName")]
[Display(Name= "Description")]
[Display(Name= "CategoryDateAdded")]
[DisplayFormat(DataFormatString = "{yyyy-MM-dd}")]
Bag
int id
int MedicineId
int CustomerID
double cost
DateTime Buying_Date
List<Medicine> Medicines
Account account
public DateTime Buying_Data { get; set; } = DateTime.Now;
Identity for User and Admin
Cart
int id
Country country
String FirstName
String LastName
string? Company_Name
string Address
string State
string ZipCode
Task 2
Task 3
Index(Bag bag)
Bag Controller
AddCart
DeleteOrder(int id)
GetBagItemCount()
AddOrder(int MedicineId)
Task 4
Admin
Index()
AddNewMedicine()
Medicine Controller
ViewMedicine(int id)
DeleteMedicine(int id)
Filter
(string CategoryId, int MaxCost = 1000000, int MinCost = 0, string Name = null)
Task 5
For Admin
Index()
AddNewCategory()
Category Controller
ViewCategory(int id)
DeleteCategory(int id)
Filter
Task 6
Signup(Account account)
Account Controller
LoginP(Account account)


Medicine
Id
Name
Picture
Price
ProductionDate
ExpiryDate
CategoryId
Categories
Id
CategoryName
AnnualBudget
ImagePath
IsActive
AspNetUserClaims
Id
UserId
ClaimType
ClaimValue
bag
Id
UserId
Cost
Buying_Date
AspNetUsers
Id
Age
UserName
NormalizedUserName
Email
NormalizedEmail
EmailConfirmed
PasswordHash
SecurityStamp
ConcurrencyStamp
PhoneNumber
PhoneNumberConfirmed
TwoFactorEnabled
LockoutEnd
LockoutEnabled
AspNetUserLogins
LoginProvider
ProviderKey
ProviderDisplayName
UserId
orders
Id
BagId
Name
Cost
Quantity
Image
AspNetRoles
Id
Name
NormalizedName
ConcurrencyStamp
AspNetUserRoles
UserId
RoleId
AspNetUserTokens
UserId
LoginProvider
Name
Value
AspNetRoleClaims
Id
RoleId
ClaimType
ClaimValue

Implementation

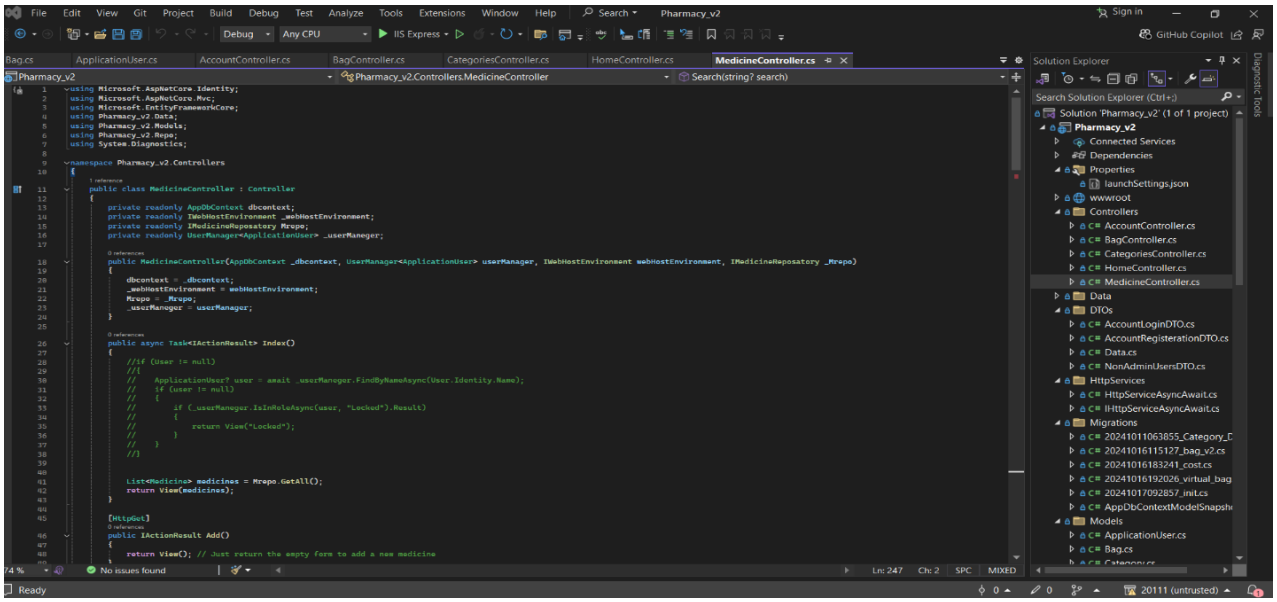
➢ Technology Stack:

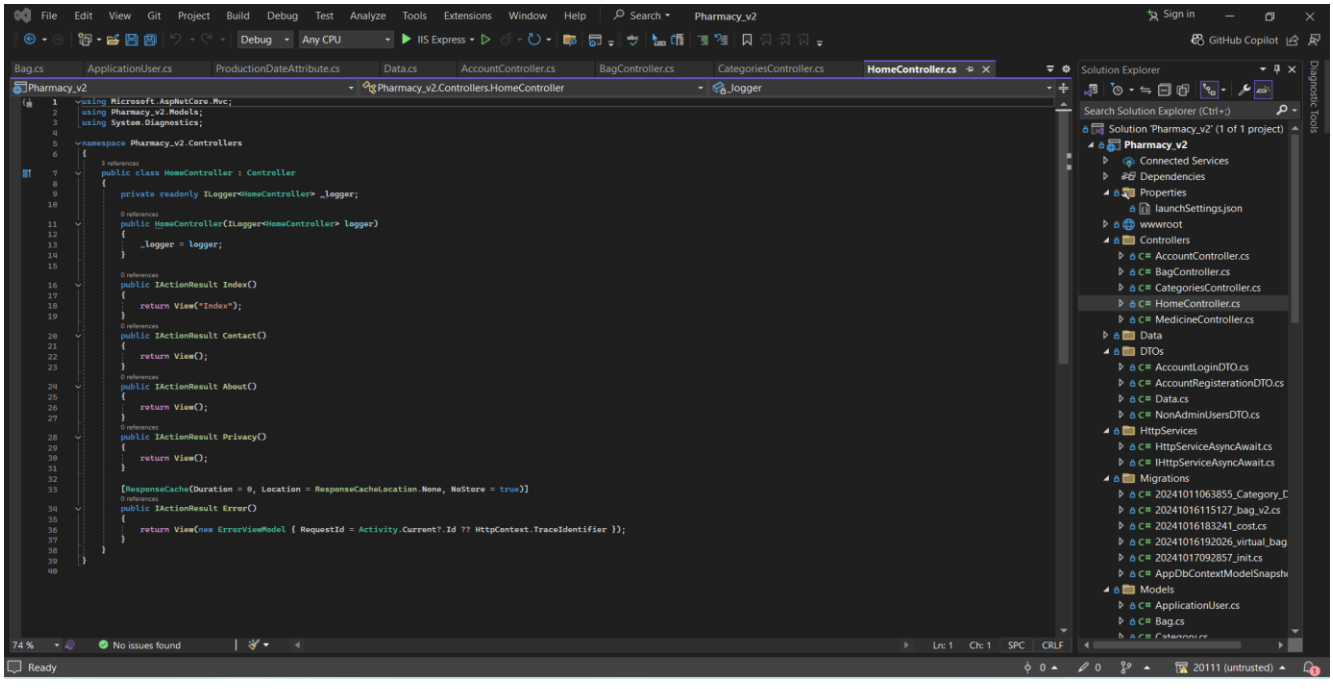
- **Backend:**
  - C# ASP.NET Core 8 MVC: The core framework for building the web application. ASP.NET Core provides the necessary tools to build a secure, scalable, and maintainable application.
  - Identity Framework: Used to handle authentication and authorization, enabling role-based access control (e.g., distinguishing between admins and users).
  - Entity Framework Core: Used for database access, handling object-relational mapping (ORM) to the SQL Server.
- **Frontend:**
  - HTML/CSS: Used to design user interfaces for the pharmacy system.
  - JavaScript: Provides dynamic content loading, and form validation, and enhances user interactions.
- **Database:**
  - SQL Server: Used to store user, medicine, category, and order data. It provides structured, relational data storage and querying capabilities.
- **Payment Gateway:**
  - Paymob: Integrated as a secure payment platform to allow users to make online payments for their orders.

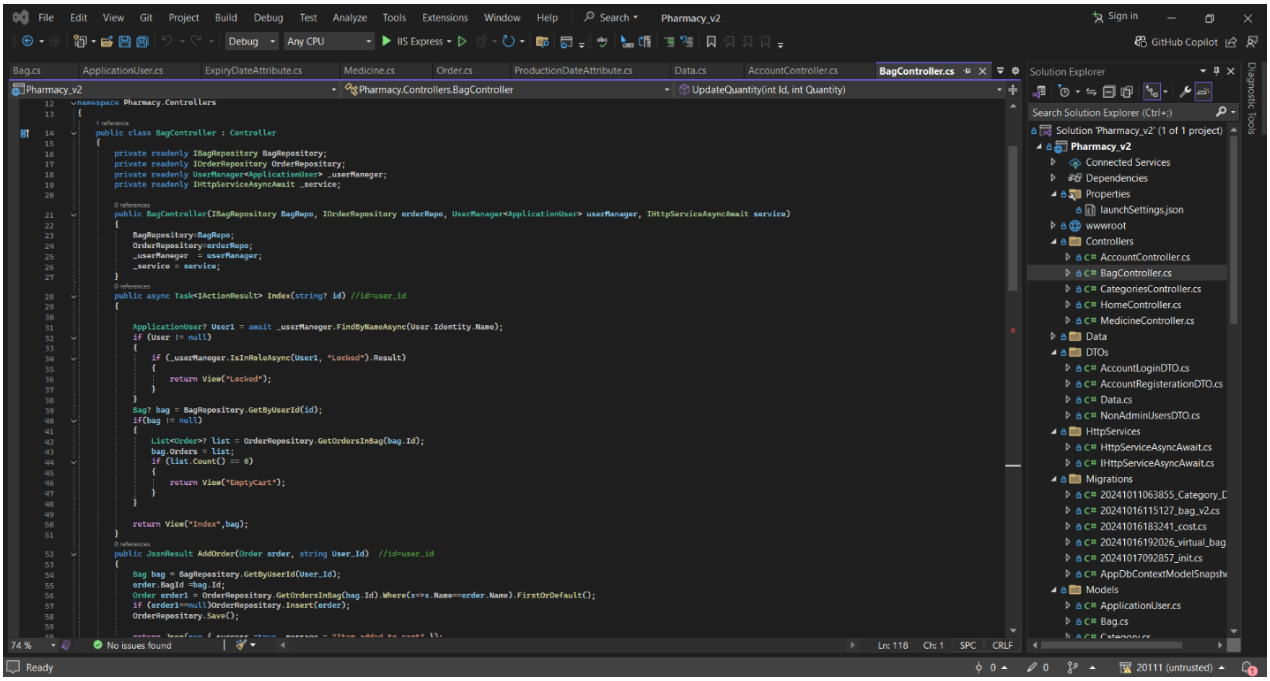
➢ Code Implementation:

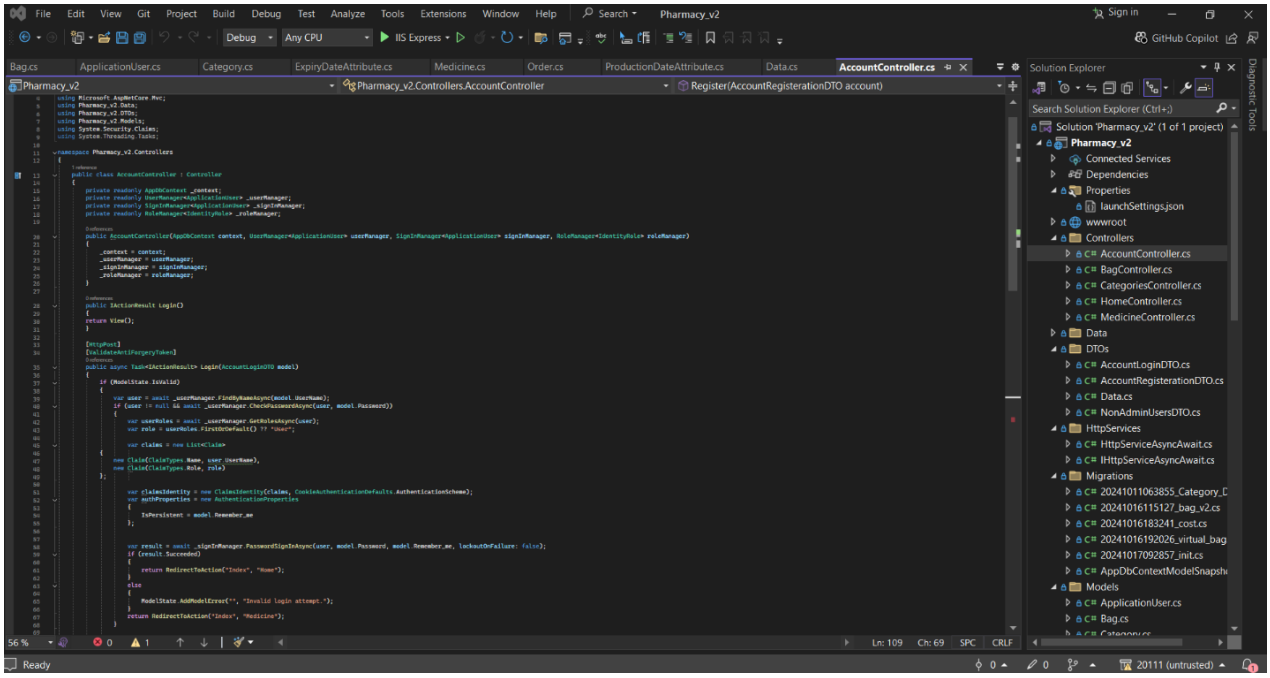
- Controllers: Implement the logic for handling user requests (e.g., logging in, registering, placing an order).
- DTOs: Manage the transfer of data between the user interface and models (e.g., registration details, login information).
- Custom Validation Attributes: Ensure that critical fields, such as production and expiry dates, are correctly validated according to business rules.

Some Code of the Project





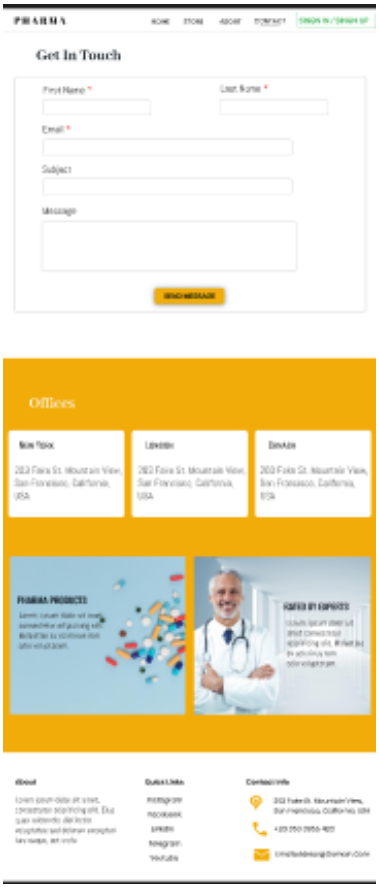


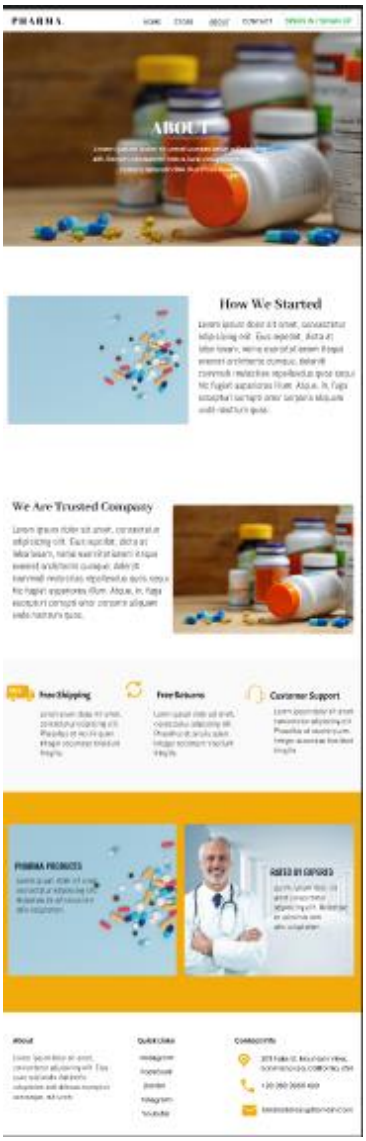


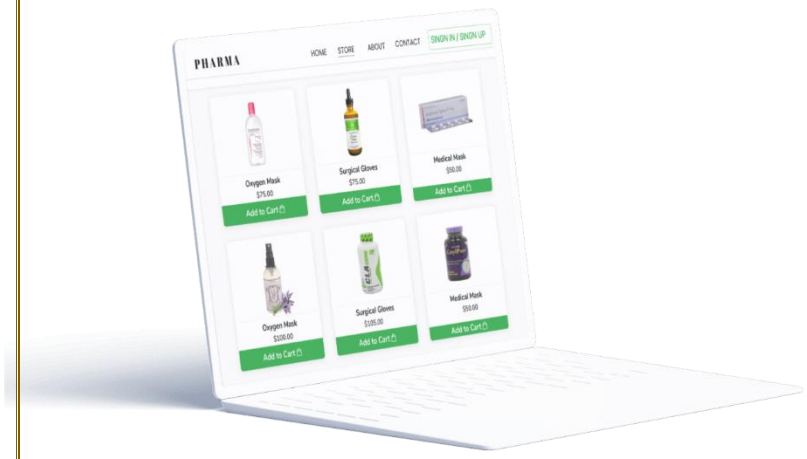
Testing and Quality Assurance

- Unit Testing: Testing individual components such as login, registration, and ordering processes to ensure they function as expected.
- Integration Testing: Testing the interactions between the frontend, backend, and database, making sure all components work together seamlessly.
- User Acceptance Testing (UAT): Ensuring that the system meets user expectations and that both customers and admins can interact with the system easily.

User interface (UI) design






PHARMA
HOME
STORE
ABOUT
CONTACT
SINGN IN / SINGN UP
Oxygen Mask
$75.00
Add to Cart
Surgical Gloves
$75.00
Add to Cart
Medical Mask
$50.00
Add to Cart
Oxygen Mask
$100.00
Add to Cart
Surgical Gloves
$105.00
Add to Cart
Medical Mask
$50.00
Add to Cart


Welcome In
PHARMA
Username
Password
Remember_me
Sign In
Don't have an account ? Sign Up here


Join With Us
Username
Email
Age
Password
Confirm Password
Sign Up
Already have an account ? Sign In here

SOFTWARE USER MANUAL

Introduction

The Pharmacy System is a web-based application developed using C# ASP.NET Core 8 MVC framework. It provides a user-friendly interface for both admins and users to manage and purchase medicines efficiently. Admins can add, update, or delete categories of medicines, while users can browse and purchase medicines. The system integrates secure login and registration features using authentication and authorization mechanisms.

System Requirements

➢ Device Requirements

- Desktop/Laptop: Any modern computer running Windows, macOS, or Linux.
- Mobile Devices: Smartphones or tablets running iOS or Android.

➢ Web Browser Requirements

The website works best on the latest versions of popular web browsers:

- Google Chrome (recommended for best performance).
- Mozilla Firefox.
- Microsoft Edge.
- Safari (for macOS and iOS users).

Make sure your browser is up to date to avoid compatibility issues.

➢ Internet Connection

A stable internet connection is necessary for accessing the website.
Minimum Speed: 1 Mbps (for basic browsing).
Recommended Speed: 5 Mbps or higher (for a faster and smoother experience).

➢ Screen Resolution

For the best viewing experience, your screen resolution should be at least 1024x768 pixels.

➢ Getting Started

In your web browser, enter the website address (https:// www.pharmacysystem.com) and press Enter.
When you arrive at the homepage, you will have two options depending on whether you have an account:

Option 1: Login (If You Already Have an Account)

1. Click the "Login" Button on the homepage or in the top navigation bar.
   - This will take you to the Login Page.

2. Enter Your Credentials:
   - Email/Username: Type in the email or username associated with your account.
   - Password: Enter your password.

3. Choose Your Role:
   - If you are an Admin, you will be taken to the Admin Dashboard where you can manage the website, products, categories, and users.
   - If you are a User, you will be directed to your User Dashboard where you can browse, shop, and manage your personal account settings.

4. Click "Log In" to access your account.
   - If the credentials are correct, you will be redirected to the appropriate dashboard based on your role.

5. Forgot Your Password?
   - If you can't remember your password, click the "Forgot Password" link on the login page.
   - Follow the instructions to reset your password via email.

Option 2: Register (If You Don't Have an Account Yet)

1. **Click the "Register" Button on the homepage or in the top navigation bar.**
   - This will take you to the Registration Page.

2. **Fill in Your Details:**
   - Full Name: Enter your full name.
   - Email Address: Provide a valid email address.
   - Create a Password: Choose a strong password for your account.
   - Confirm Password: Re-enter your password to confirm it.

3. **Choose Your Role:**
   - Select whether you're registering as a **User** or an **Admin**.
     - Admins will need to enter additional authorization or admin code (if applicable).

4. **Click "Register":**
   - Once all the information is filled in correctly, click the "**Register**" button.
   - You may receive a confirmation email. Follow the instructions in the email to verify your account (if applicable).

5. **Log In After Registration:**
   - After registering, you will be automatically logged in or directed to log in using your new credentials.

Next Steps After Logging In

- **For Admins:** You will be able to access the Admin Dashboard to manage website content such as adding or editing products, categories, and users. You'll also have control over other administrative features like managing orders and viewing reports.
- **For Users**: You can now browse the website, shop for products, manage your orders, and update your personal account details.

Features Overview

Admin Role:

- Add, edit, or delete categories of medicines.
- Manage user accounts and view purchase history.
- Promote a user to the admin role, giving them administrative privileges.

User Role:

- Browse available medicines.
- Add medicines to the cart and purchase.
- View order history.

Login & Registration:

- Secure login for both admins and users with role-based access control.
- Validation of registration with email confirmation

User Interface Guide

- Login Page: Users and admins can log in using the same interface.
- Registration Page: For new users to create an account.
- Admin Dashboard: Contains sections for managing medicine categories, users, and orders.
- User Dashboard: Users can view and purchase medicines, and view order history.

How to Perform Tasks

Task 1: Adding Medicine Categories (Admin)

1. Log in as an admin.
2. Navigate to the "Categories" section.
3. Click "Add Category" and fill in the details (e.g., category name, description).
4. Save changes to add the new category.

Task 2: Promoting a User to Admin (Admin)

1. Log in as an admin.
2. Go to the "User Management" section.
3. Locate the user you wish to promote.
4. Click the "Promote to Admin" button.
5. Confirm the action and the user will be granted admin privileges.

Task 3: Purchasing Medicines (User)

1. Log in as a user.
2. Browse the medicines available under different categories.
3. Add medicines to the cart.
4. Proceed to check out and complete the purchase using the payment methods provided.

Task 4: Registering an Account

1. Click on "Register" from the login page.
2. Fill out the form with the required information (username, email, password).
3. Validate the email and log in to access the system.

Troubleshooting

- **Issue:** Login Not Working
  Solution: Ensure you are entering the correct credentials. If forgotten, use the "Forgot Password" feature to reset your password.

- **Issue:** Unable to Add Medicine Categories (Admin)
  Solution: Check if you are logged in as an admin and have the necessary permissions.

FAQ

- Q: What is the Payment method on the website?
- A: Paymod is our payment method, you will go to your bag and click
- Q: Can users and admins use the same login page?
- A: Yes, both roles use the same login page but are redirected to different dashboards based on their role.

Contact Support

For any technical issues or assistance:

- Email: support@pharmacysystem.com
- Phone: +1 (800) 987-6543
- Website: www.pharmacysystem.com/support