

CSE 2104 - Object Oriented Design and Programming Lab

Project Report



MediMart

Team Aurora

Submitted By

MD.Jahidul Islam Sarker (Roll : 01)

Soumik Deb (Roll : 11)

Arin Saha Turja (Roll : 37)

Tukabbir Hossain Sadi (Roll : 48)

30th Batch

Department of Computer Science & Engineering
University of Dhaka

Submitted On

27 December 2025

Table of Contents

1. Introduction	3
1.1 About the App	3
1.2 Motivation	3
1.3 Features (Bullet Points)	3
1.4 Tools, Technologies and Frameworks Used	3
1.5 Individual Responsibilities	3
1.MD.Jahidul Islam Sarker (Roll:01)	3
2.Soumik Deb (Roll 11)	
3.Arin Saha Turja(Roll 37)	
4.MD.Tukabbir Hossain Sadi(Roll 48)	
2. Design and Implementation	4
2.1 UI Developed (Screenshots)	4
2.2 Database Schema	4
2.3 Code Repository Link	4
2.4 Video Demo Link	4
3. Conclusion	5
3.1 Challenges and Solutions	5
3.2 Lessons Learned	5
3.3 Future Plan	5

1. Introduction

1.1 About the Project

MediMart is a modern JavaFX-based desktop application designed to simplify the management of medicines, customers, inventory, orders, payments, and billing inside a pharmacy or medical store environment.

1.2 Motivation

Pharmacies often struggle with:

- Manual stock management
- Tracking expiry dates
- Billing delays
- Maintaining customer order history
- Handling multiple dashboards
- Maintaining secure user sessions

Our motivation was to create a **complete yet lightweight pharmacy management solution** that:

- Improves efficiency
- Reduces human error
- Automates billing
- Provides real-time stock updates
- Ensures secure customer & admin access

1.3 Features:

- JavaFX modern UI with responsive layouts
- User authentication system (Admin & Customer)
- OTP-based password recovery
- Medicine management (Add, Edit, Delete, Update stock)
- Real-time medicine synchronization between dashboards
- Customer cart management
- Card payment interface
- Auto-generated bills & invoice export
- Order history tracking
- Search, filter, and sort medicines
- Image storage for medicines
- User profile management
- Secure session handling
- Database-backed system using SQLite
- Exception logging and crash reporting
- Analytical Reports
- Order Cancellations & Notification

1.4 Tools, Technologies and Frameworks Used

<u>Category</u>	<u>Tools / Technologies</u>
Programming Language	Java (JDK 25)
UI Framework	JavaFX 25, FXML
Architecture	MVC (Model–View–Controller)
Database	SQLite, JDBC
Styling	CSS (<code>style.css</code> , <code>theme.css</code>)
Utility Modules	DBHelper, FileLogger, ConfigManager, Session, ImageStorage, ReportExporter
Real-Time Sync Engine	MedicineSync Server & Client (Multithreading + DB Change Listener)
Security	Session Management, Prepared Statements (<i>OTP module planned but not implemented</i>)

Report Export	PDF/CSV exporting using ReportExporter (PDF via Apache PDFBox)
Threading & Performance	Java Threads for real-time synchronization
File Handling	File I/O for logs, configuration, and image storage
IDE / Build Tools	IntelliJ IDEA, JavaFX SDK

1.5 Individual Responsibilities

- **MD.Jahidul Islam Sarker (Roll : 01)**
- Designed and implemented the JavaFX user interface, including layouts for Login, Signup, Admin Dashboard, Customer Dashboard, Cart View, Bill View, and Order History.
- Developed CSS themes to ensure consistent, modern, and responsive styling across all screens.
- Implemented UI controllers for handling user interactions, navigation flows, and form validation.
- Ensured all FXML layouts were responsive, structured, and user-friendly.
- Developed backend logic for medicines, users, orders, inventory, cart management, and billing operations.
- Implemented core service modules such as `MedicineService`, `OrderService`, and `UserService`.
- Designed and maintained the SQLite database schema with secure, optimized CRUD operations.
- Implemented billing calculations, connected cart totals with orders, and optimized database queries for performance.

- Added basic validation rules for dummy card payments and connected successful payments to bill generation and order saving.
 - Implemented invoice creation through **BillView** and integrated a complete checkout workflow.
 - Implemented the PDF and CSV export feature using **ReportExporter**, enabling users to export bills and order history into downloadable reports.
 - Ensured exported files included proper formatting, calculated totals, and essential customer/order details.
 - Added secure session handling using a dedicated **Session** utility to track active users safely.
 - Created centralized error handling using **AppException** and maintained runtime logs with **FileLogger**.
 - Developed **ConfigManager** for managing configuration settings and implemented **ImageStorage** for storing and loading image assets consistently across systems.
 - Ensured all forms and inputs underwent validation and that database operations were fully SQL-injection-safe.
 - Designed the real-time stock synchronization engine using multithreading and database change listeners for live inventory updates.
 - Managed the Git workflow, resolved merge conflicts, and maintained a clean repository across team members.
-
- Prepared complete project documentation, UML diagrams, screenshots, and the final project report.
 - Assisted with debugging, refactoring, and improving overall performance and code quality.
 - Implemented Profile View with Proper Validation

- **Soumik Deb (Roll : 11)**
- Designed and implemented the JavaFX user interface, Customer Dashboard, Cart View, Bill View, and Order History.
- Developed backend logic for users, orders, cart management
- Implemented core service modules such as MedicineService, OrderService, and UserService.
- Implemented billing calculations, connected cart totals with orders.
- Organized the overall project folder structure, naming conventions, and resource management for maintainability.
- Helped implementing invoice creation through **BillView** and integrated a complete checkout workflow.
- Assisted with debugging, refactoring, and improving overall performance and code quality.

3. Arin Saha Turja (Roll 37)

- Adding medicine pictures ,changing path to display those in Customer Dashboard as those paths are different for different devices.
- Adding Details such as price,quantity ,expiration date ,medicine name, and Category for each 100 medicines .
- Giving Unique medicine Id for each medicine which is only accessible to admin .
- Implemented **MedicineSync**, **MedicineSyncServer**, and **MedicineSyncClient** to broadcast changes instantly across the application.

- Fixing Instock update of medicines so that if a customer orders a medicine ,the stock of that medicine automatically updates.
- Contributed to the overall project folder structure, naming conventions, and resource management for maintainability.
- Implemented authentication features including Login, Admin Login, and Signup functionality.

4. **MD.Tukabbir Hossain Sadi (Roll : 48) :**

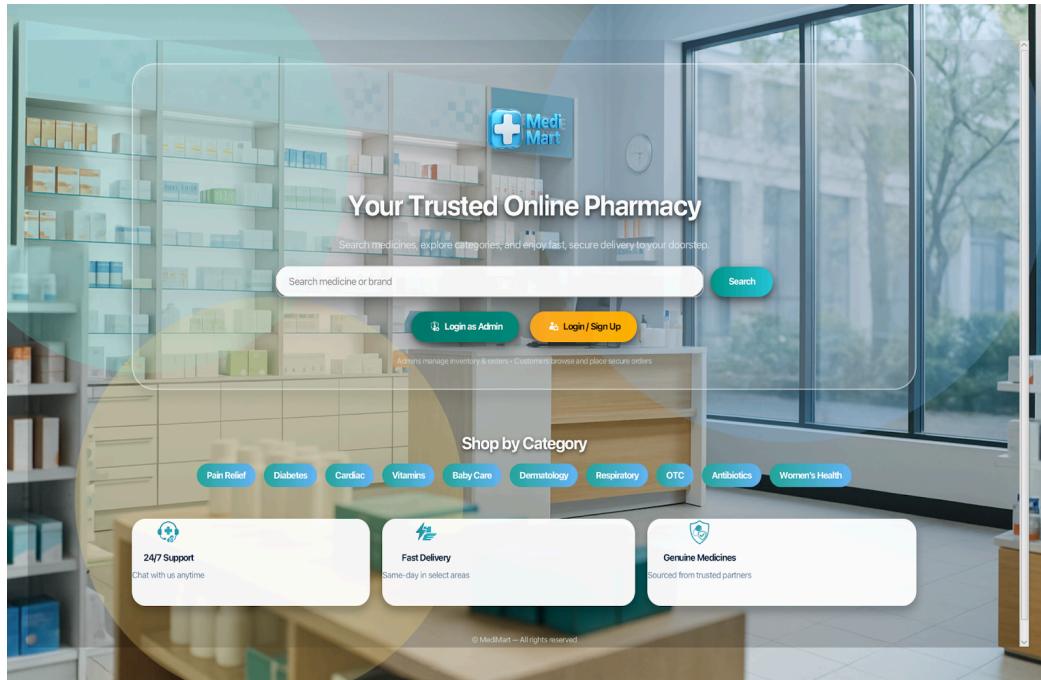
- Designed and implemented the dummy card payment system using `CardPaymentDialog` with fields for cardholder name, card number, expiry date, and CVV.
- Added basic validation rules for dummy card payments and connected successful payments to bill generation and order saving.
- Implemented centralized error messages for clean and consistent reporting.
- Added FileLogger to track login attempts, database failures, and runtime exceptions.
- Enhanced debugging by attaching human-readable error descriptions and timestamps.
- Manually tested all user flows including login, signup, ordering, cart management, inventory editing, and bill export.
- Validated edge cases such as empty carts, expired medicines, low stock warnings, repeated users, and invalid payments.
- Ensured naming conventions, folder structures, and styling rules remained consistent across the team.

- Assisted others by debugging merge conflicts and helping integrate their modules.
- Managed JavaFX dependencies and solved module-path issues for consistent builds.

2. Design and Implementation

2.1 UI Developed (Screenshots)

1) Home view



2) Admin Login

A screenshot of the Admin Login screen. The title bar says "Admin Login - MediMart". The main area is titled "Admin Login". It contains two input fields: "Email" and "Password", each with a corresponding placeholder text ("Email" and "Password") and a "Show password" checkbox below it. At the bottom of the form are two buttons: a teal "Login" button and a red "Back" button.

3) Admin Dashboard

ID	Name	Category	Price	Quantity	Expiry
1	Ace XR	Pain Relief	18.28	200	2030-02-02
2	Xyloken	Pain Relief	100.00	230	2033-03-12
3	Xpa XR	Pain Relief	20.00	290	2033-04-04
4	Ace	Pain Relief	50.00	300	2034-09-09
5	Renova XR	Pain Relief	55.00	200	2032-05-05
6	Diamicron Mr 60 mg	Diabetes	310.00	300	2033-09-06
7	Progest	Diabetes	582.00	200	2032-04-30
8	Empa 10	Diabetes	344.44	300	2031-06-09
9	Linax	Diabetes	72.72	300	2031-03-04
10	Dimerol	Diabetes	102.50	350	2033-04-21
11	Bislol 2.5	Cardiac	90.00	150	2034-08-16
12	Bislol 5	Cardiac	150.00	300	2034-09-08
13	ATV 10	Cardiac	45.45	300	2035-06-09
14	Bisoprolol 2.5	Cardiac	55.00	400	2032-03-15
15	Bisoprolol 2.5	Cardiac	55.00	200	2031-02-02
16	Vitamine B	Vitaminine	100.00	300	2035-09-07

Add Medicine Edit Selected Delete Selected Export CSV Refresh Back to Home

Refreshed Sync: Running (broadcasts on change)

4) Add Medicine Form

Add New Medicine

ID:

Name:

Category:

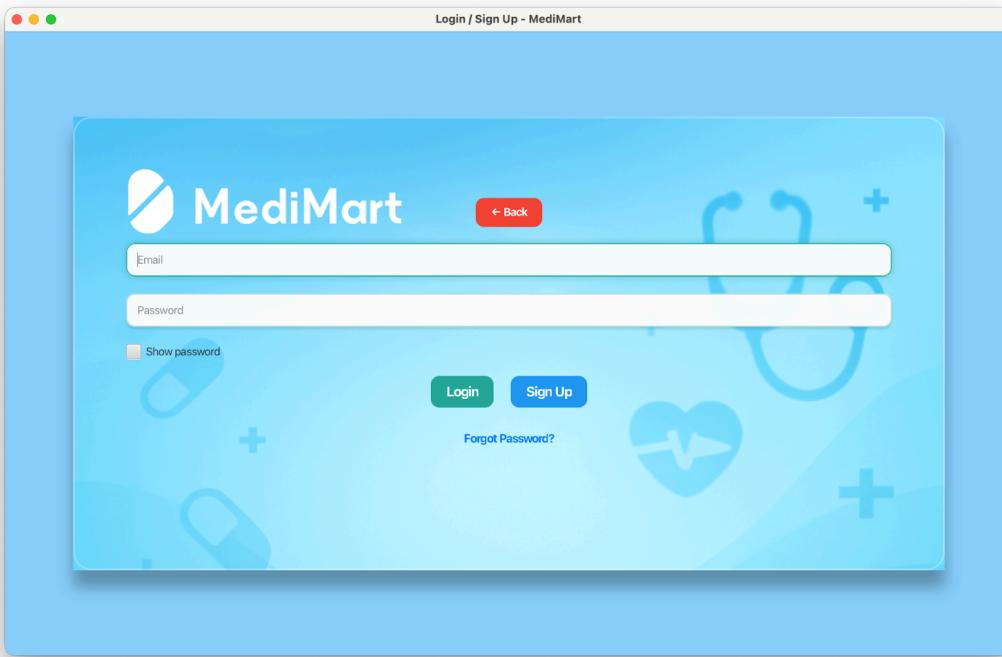
Price:

Quantity:

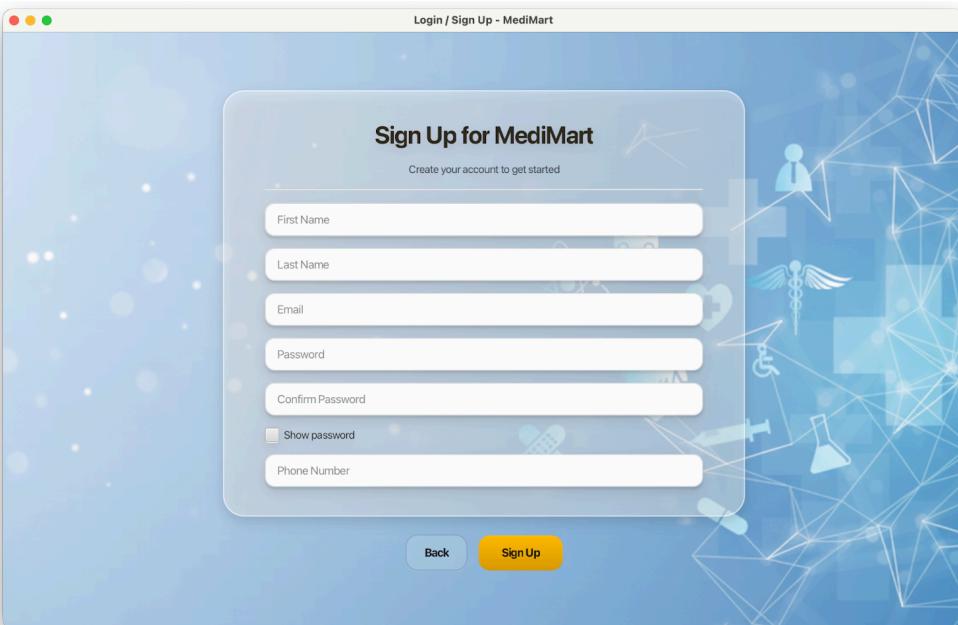
Expiry:

Image:

5) Customer Login



6) Signup Form



7) Customer Dashboard

The screenshot shows a customer dashboard with a header "Customer Dashboard" and a navigation bar with "Profile", "Order History", and "Cart" buttons. Below the header is a search bar with placeholder "Search medicine..." and a dropdown menu set to "Sort by Price: Low → High". A message "Refreshed" is displayed. The main content area features a heading "Allergy" followed by four product cards:

- Fexo 60**
Price : 80.0
Expiry: 2035-12-20
In Stock: 5
Quantity: 1
- Fenadin**
Price : 90.0
Expiry: 2031-02-02
In Stock: 20
Quantity: 1
- Delfex 120**
Price : 200.0
Expiry: 2034-05-27
In Stock: 20
Quantity: 1
- Alatrol**
Price : 350.0
Expiry: 2031-02-03
In Stock: 23
Quantity: 1

Below this section is a heading "Anemia & Other Blood Disorders" followed by five product cards:

- Rebocil**
- Zeelot-Cl**
- Zif-A**
- Oroxat 70**
- Orion**

A red "Back" button is located at the bottom center.

8) Cart View

The screenshot shows a mobile application interface for a shopping cart. At the top, there are three red, yellow, and green window control buttons. The title "Cart" is centered at the top. Below the title, there is a back button labeled "← Back". The main heading "Your Cart" is displayed in large, bold black font. A sub-instruction "Review your selected medicines before checkout." is shown below the heading. A table lists the selected items. The table has columns: Medicine, Category, Unit Price (BDT), Qty, and Line Total. One item is listed: "Fexo 60" under "Medicine", "Allergy" under "Category", "80.0" under "Unit Price (BDT)", "5" under "Qty", and "400.0" under "Line Total". At the bottom left, the total amount "Total: 400.00 BDT" is displayed in green. Below that, it says "1 item selected.". On the bottom right, there are two buttons: "Remove Selected" in a red rounded rectangle and "Proceed to Payment" in a blue rounded rectangle.

Medicine	Category	Unit Price (BDT)	Qty	Line Total
Fexo 60	Allergy	80.0	5	400.0

Total: 400.00 BDT

1 item selected.

Remove Selected **Proceed to Payment**

9) Card Payment View

The screenshot shows a 'Secure Payment' window. At the top, it says 'Secure Payment' and 'Enter your card details to complete your purchase.' Below this is a 'Billing Summary' section with 'Total Amount: 400.00 BDT'. The main area is titled 'Card Details' and contains four input fields: 'Name on Card', 'Card Number' (showing placeholder 'XXXX XXXX XXXX XXXX'), 'Expiry (MM/YY)', and 'CVV' (with placeholder '3 or 4 digits'). At the bottom are two buttons: a red 'Cancel' button and a blue 'Pay Now' button.

10) Profile View

The screenshot shows a 'My Profile' window. At the top, it says 'My Profile' and 'Update your personal details and password'. On the left is a circular profile picture placeholder with a 'Change Avatar' button below it. The main area is divided into sections: 'Profile Information' (containing fields for First Name, Last Name, Phone, Email, and Address), 'Change Password' (containing fields for Current password, New password, and Confirm new password), and a 'Save Profile' button. At the bottom right is a 'Back' button.

11) Order History

The screenshot shows a "My Order History" interface. At the top, there's a back button labeled "← Back". Below it, a section titled "Order History" with the subtitle "See your past purchases and track your spending." A table lists three orders:

Order	Date	Medicine	Qty	Total (BDT)
3	2025-12-01 00:45:14	Fexo 60	5	400.0
2	2025-12-01 00:01:26	Fexo 60	5	400.0
1	2025-12-01 00:01:26	Napa	10	150.0

At the bottom right, it says "Orders: 3 · Total spent: 950.00 BDT".

12) Invoice Preview

The screenshot shows an "Order Bill" interface. At the top, there's a back button labeled "← Back". Below it, a section titled "Order Bill / Invoice" with the subtitle "Thank you for shopping with MediMart." An order date is shown as "Order Date: 2025-12-01 11:43:06". A table lists the items:

Medicine	Qty	Unit Price	Line Total
Fexo 60	5	80.0	400.0

At the bottom left, it says "Grand Total: 400.00 BDT". There are two buttons at the bottom right: "Download PDF" and "Back to Dashboard".

13) INVOICE PDF

<https://github.com/Jahidul183019/MediMart/blob/main/screenshots/invoice.pdf>

2.2 Database Schema/File Structure

SQLite Tables Used

- users
- medicines
- inventory
- orders
- order_items
- payments

Features include:

- Auto-increment IDs
- Foreign key constraints
- Cascading deletes
- Optimized indices
- Transaction-based updates

Project Structure:

```
MediMart/
  └── src/
    └── models/
      ├── Medicine.java
      ├── Cart.java
      ├── Order.java
      ├── OrderItem.java
      ├── OrderHistoryRow.java
      ├── Inventory.java
      ├── Payment.java
      ├── Bill.java
      └── User.java

    └── services/
      ├── MedicineService.java
      ├── OrderService.java
      ├── UserService.java
      └── DBChangeListener

    └── net/
      ├── MedicineSync
      ├── MedicineSyncClient
      └── MedicineSyncServer

    └── ui/
      ├── LoginPage.java
      ├── AdminLogin.java
      ├── LoginSignup.java
      ├── AdminDashboard.java
      ├── CustomerDashboard.java
      ├── AddMedicineForm.java
      ├── EditMedicineForm.java
      ├── ForgotPasswordPage.java
      ├── ResetByEmailPage.java
      ├── ResetByPhonePage.java
      ├── ResetPasswordPage.java
      ├── VerifyOTPPage.java
      ├── UserProfileForm.java
      ├── ProfileView.java
      ├── CartView.java
      └── BillView.java
```

```
    └── OrderHistoryView.java  
    └── CardPaymentDialog.java  
    └── Main.java  
    └── MainController.java  
    └── SignupForm.java  
    └── OTPService.java  
  
    └── utils/  
        └── DBHelper.java  
        └── AppException.java  
        └── FileLogger.java  
        └── ConfigManager.java  
        └── ImageStorage.java  
        └── ReportExporter.java  
        └── Session.java  
  
    └── resources/  
        └── css/  
            └── style.css  
            └── theme.css  
            └── login.css  
        └── fxml/  
            └── LoginPage.fxml  
            └── LoginSignup.fxml  
            └── admin_dashboard.fxml  
            └── SignupForm.fxml  
            └── main.fxml  
            └── add_medicine_form.fxml  
            └── ForgotPasswordPage.fxml  
  
        └── html/  
            └── admin.html  
            └── login.html  
            └── orders.html  
  
        └── images/  
            └── admin.png  
            └── background.jpg  
            └── cart.png  
            └── customer.png
```

```
    └── fast.png  
    └── genuine.png  
    └── home_background.png  
    └── login_background.png  
    └── logo_medimart.png  
    └── order_history.png  
    └── p1.png  
    └── p2.png  
    └── p3.png  
    └── p4.png  
    └── Placeholder.png  
    └── profile.png  
    └── signup_background.png  
    └── support.png  
  
└── README.md  
└── medimart.db (SQLite Database)
```

2.3 Code Repository Link

<https://github.com/Jahidul183019/MediMart>

2.4 Video Demo Link

<https://youtu.be/5leGbDB31IU?si=d2jcZS4DyISi6IEd>

3. Conclusion

3.1 Challenges and Solutions

<u>Challenge</u>	<u>Solution</u>
Real-time medicine/stock updates were not reflected on the customer dashboard after admin changes.	Implemented a multi-threaded real-time synchronization engine using a WebSocket-like communication layer and a <code>DBChangeListener</code> to instantly broadcast updates across dashboards.
Image path issues occurred after teammates pulled the project, leading to missing or broken images.	Centralized all assets inside <code>resources/images/</code> and used the <code>ImageStorage</code> utility with unique naming and relative paths to ensure consistent loading across systems.
UI layout inconsistencies across different machines and IDE configurations.	Standardized all views using well-structured FXML layouts and unified CSS themes to maintain consistent rendering and spacing.
Difficulty managing and debugging various runtime errors across modules.	Introduced a centralized <code>AppException</code> class and implemented a <code>FileLogger</code> to capture exceptions, runtime logs, and debugging information.
Handling numerous product images while avoiding naming conflicts and invalid paths.	Developed a structured <code>ImageStorage</code> system that generates unique filenames, validates formats, and stores assets reliably within the project's resource directory.

3.2 Lessons Learned

- Importance of modular design
- How to apply MVC architecture in real applications
- How to design clean UI using FXML & CSS
- Working with SQLite and JDBC

- Real-time synchronization concepts
- Exception abstraction & system logging
- Application-level security considerations

3.3 Future Plan

The project is currently in a functional but basic version.

In future, we plan to add:

- Mobile app version
- Live AI-powered medicine suggestion
- Prescription scanning
- Multi-branch pharmacy support
- Full role-based authorization