PHOTO LAB MANAGEMENT SYSTEM

21CS503 - MINI PROJECT

A PROJECT REPORT

Submitted by

RADHAKRISHNAN B (727721EUCS112)

MAHA PRABHU S G (727721EUCS067)

PRANAV K S (727721EUCS103)

in partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution, Affiliated to Anna University Chennai - 600 025)

ANNA UNIVERSITY: CHENNAI

NOVEMBER 2023

SRI KRISHNA COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution)

(Approved by AICTE and Affiliated to Anna University, Chennai)

ACCREDITED BY NAAC WITH "A" GRADE

BONAFIDE CERTIFICATE

Certified that this mini project report titled **"PHOTO LAB MANAGEMENT SYSTEM"** is the bonafide work of **"RADHAKRISHNAN B (727721EUCS112), MAHA PRABHU S G (727721EUCS067), PRANAV K S (727721EUCS103)"** who carried out the mini project work under my supervision.

SIGNATURE

Ms. N. POORANAM

SUPERVISOR,

ASSISTANT PROFESSOR

SIGNATURE

Dr. K. SASI KALA RANI

HEAD OF THE DEPARTMENT

Department of Computer Science and Engineering

Sri Krishna College of Engineering and Technology

Kuniamuthur,

Coimbatore.

This Mini Project report is submitted for Autonomous Mini Project Viva-Voce examination held on ……...

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our sincere thanks to the management and **Dr. J. JANET M.E., Ph.D.,** Principal, Sri Krishna College of Engineering and Technology, Coimbatore for providing us the facilities to carry out this mini project work.

We are thankful to **Dr. K. SASI KALA RANI M.E., Ph.D.,** Professor and Head, Department of Computer Science and Engineering, for her continuous evaluation and comments given during the course of the mini project work.

We express our deep sense of gratitude to our supervisor **Ms. N. POORANAM M.E., (Ph.D)**, Assistant Professor, Department of Computer science and Engineering for her valuableadvice, guidance and support during the course of our mini project work.

We express our heartfelt sense of gratitude and thanks to our beloved parents, family and friends who have helped during the mini project course.

ABSTRACT

The Photo Lab Management System is an advanced software solution tailored to revolutionize the management and operations of modern photo processing laboratories and studios. The primary objective of this system is to offer a seamless, integrated platform that caters to the multifaceted requirements of the photo laboratory, ranging from customer bookings, inventory management, to digital bill processing and print orders.

The core ambition of this system lies in elevating the efficiency of photo lab services while providing an unparalleled customer experience. By digitizing and automating conventional tasks, it ensures precision and speed in every lab process. For instance, the customer management feature allows studios to maintain comprehensive customer profiles, storing their preferences and past orders, thus facilitating personalized service and marketing strategies.

There are two systems to run in the project: one for the admin, i.e., the photo lab manager to create, update, view or delete products, to view customer details and orders placed by the customers. The other one is for the customers to select products and place orders, so that the system will generate bill for the total amount for the purchased items.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
DB	DATABASE
JDBC	JAVA DATABASE CONNECTIVITY
CSR	CUSTOMER SERVICE REPRESENTATIVE
CRM	CUSTOMER RELATIONSHIP MANAGEMENT

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF PHOTO LAB MANAGEMENT SYSTEM

The Photo Lab Management System is an all-encompassing software solution tailored to rejuvenate and streamline the functions of contemporary photo processing laboratories and studios. Serving as the central pivot for supervising various segments of photography processing and management, the system covers customer management, digital photo processing, print orders, billing and payment management.

Central to this system is the enhancement of the customer's experience in photography services. It provides customers the autonomy to create profiles, catalog their preferences, and efficiently browse through various photo processing options. The user-friendly interface further augments the process of placing orders for photo prints or digital enhancements, allowing customers to detail their specifications, view the final version, and securely complete payments. Efficient photo order processing is embedded at the heart of the system, ensuring that customer demands are swiftly communicated to the labs. An integrated payment gateway provides a variety of methods, facilitating customers to settle their bills effortlessly online.

For administrators and studio managers, the system offers tools to update photo processing options, manage promotions, and ensure customers are presented with the latest offerings. With the inclusion of robust analytics and reporting capabilities, studios can gain profound insights into customer preferences, seasonal trends, and operational efficiencies, driving informed decision-making and proactive strategy formulation. Given the sensitivity of photographs, advanced data protection measures, fortified user authentication protocols, and encryption methodologies are employed. Moreover, regular backups safeguard against potential data losses.

Adaptable by nature, the Photo Lab Management System can be molded to cater to various photography environments, be it a standalone photo studio or a chain of photo labs. Upholding compliance with industry best practices and data protection norms remains pivotal, guaranteeing the safeguarding of customers' cherished memories. Comprehensive tutorials and dedicated support ensure seamless navigation for both photo lab technicians and customers.

1.2 EXPLANATION OF PHOTO LAB MANAGEMENT SYSTEM

The Photo Lab Management System is a refined software solution conceptualized to reinforce and optimize operations of modern photo labs and studios. This digital platform stands as the pillar for the photography industry, harmonizing a myriad of intricate functions and procedures. Its cardinal aim revolves around bolstering efficiency, enriching client engagement, and maximizing profitability in the fiercely competitive world of photography.

The field of managing photos involves many aspects, including customer bookings, inventory management, print orders, and payments. The Photo Lab Management System adeptly maneuvers through these domains, integrating automation and a structured approach, thereby simplifying an otherwise intricate workflow.

Integral features of such a system span across customer management, studio coordination, inventory tracking, photo processing, order management, payment processing, and analytics. Moreover, cutting-edge systems are now integrating predictive analytics, granting studio owners insights for effective demand forecasting, inventory planning, and targeted marketing campaigns.

The significance of the Photo Lab Management System in today's digital age is monumental. It refines the customer journey through seamless order placements, transparent communication, real-time order updates, and efficient service delivery. Additionally, the system amplifies operational efficiencies, paving the way for innovative marketing endeavors and enhancing revenue streams.

Effective data input modules are crucial, collecting pivotal data about customers and their photographic preferences. This ensures precision in service delivery and fosters a personalized experience. The integration of different photography services and offerings should be seamless, with provisions for easy updates and customizations.

In essence, the Photo Lab Management System stands as a pivotal software asset, indispensable in the modern photo studio landscape, orchestrating customer delight and operational synchronization.

CHAPTER 2

LITERATURE STUDY

2.1 FUNCTIONS AND STRENGTH OF THE SYSTEM

The Photo Lab Management System heralds a transformative shift from the traditional approaches employed in photo processing laboratories and studios. Infused with an array of modern features, it aims to redefine the realm of photo processing, enhancing customer experience and elevating operational efficiencies. Moving away from conventional methods where studio clerks manually track and process orders, this digital system introduces a seamless and automated experience. Customers, through an intuitive digital portal, can effortlessly register, upload photos, choose processing options, and place orders. Integrated payment solutions negate the need for physical monetary transactions, bolstering the convenience and security quotient for both the labs and customers.

Among the paramount strengths of the system is its robust client management capability. It is adept at managing an expansive client database, markedly reducing manual data entry. Customers can establish personalized profiles, delineate their preferences, and peruse their order archives with ease. The platform simplifies tasks like client search, data modification, and profile removal, tasks which were traditionally labor-intensive. Furthermore, the system offers an advanced inventory management feature. This eliminates manual tracking of photo processing materials, ensuring real-time monitoring of inventory levels. Should there be a dip in essential items, the system can facilitate automated reorder mechanisms with vendors.

From a customer standpoint, the Photo Lab Management System drastically uplifts the service experience. Automated notifications, encompassing order confirmations and processing updates, foster transparency and escalate customer satisfaction levels. On the operational front, the system significantly trims down human intervention, especially in realms like data entry and payment handling, translating to tangible cost efficiencies. Its ability to judiciously manage inventory further curbs overheads and minimizes wastage.

2.2 PROBLEM IN THE CURRENT SYSTEM

2.2.1 Manual Processes:

The existing system heavily relies on manual methods for order taking, inventory management, and client communication. This leads to inefficiencies, longer processing times, and a greater chance of errors.

2.2.2 Transactions that are inefficient:

Customers regularly express unhappiness with the manual system's cumbersome and error-prone transaction procedure. Order placing and payment processing are frequently slow, which degrades customer service and frustrates customers.

2.2.3 High maintenance costs:

The existing system requires a lot of work and is expensive to maintain. Manual record-keeping and data input involve a significant amount of labor, which raises labor expenses. Higher operating costs are also a result of the requirement for tangible record-keeping resources like paper and storage space.

2.2.4 Limited Scalability:

With an increase in customer demands and order volumes, the traditional system struggles to scale up, often resulting in delayed deliveries and unsatisfied customers.

2.2.5 Data Management Issues:

Maintaining physical records and databases can lead to data loss, misplacements, and inconsistency in record-keeping.

2.2.6 Lack of Integration:

Different facets of operations, from order taking to delivery, work in silos. This lack of integration can lead to inefficiencies and communication breakdowns between departments.

2.3 STUDY ABOUT PHOTO LAB MANAGEMENT SYSTEM

A comprehensive review of the literature related to Photo Lab Management Systems showcases a transformative shift in this field. The convenience, efficiency, and integrated capabilities offered by modern photo lab management systems are reshaping how customers and businesses approach photographic services.

Several dominant themes can be discerned from the extant literature. Firstly, the infusion of technology in transforming photographic processes is evident. Digital platforms, software, and applications have fundamentally altered how photos are ordered, processed, and delivered. The capability to digitize, archive, and even restore photographs has presented avenues for businesses to provide enhanced services to customers. The influence of technology is not only seen in improving service quality but also in streamlining internal lab operations.

Moreover, a customer-centric approach is increasingly taking center stage. Systems equipped with Customer Relationship Management (CRM) tools have been highlighted as crucial for a personalized customer experience. These tools can archive customer preferences, order histories, and feedback, allowing businesses to tailor their offerings and enhance customer engagement.

Challenges facing the modern photo lab industry also find mention in the literature. Issues like maintaining photo quality, ensuring efficient and safe storage of digital archives, adapting to rapidly changing technologies, and navigating the competitive landscape are recurrent topics. Solutions often emphasize the importance of continuous training, investment in high-quality equipment, and partnerships with technology providers.

In conclusion, the current literature on Photo Lab Management Systems underscores the critical role of technology, the imperative for a customer-oriented approach, the importance of sustainability, and the necessity for strategic adaptability in the ever-evolving photo processing industry. As the domain continues to progress, future studies are expected to delve deeper into emerging technologies, the changing dynamics of customer expectations, and innovative solutions to industry-specific challenges.

CHAPTER 3
REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

The Photo Lab Management System is a sophisticated software solution tailored to refine the operations of photo processing laboratories. Fundamental to the system is its user registration and authentication mechanism, ensuring both staff and customers have secure, personalized access. Central to its design is the photo order management module, allowing customers to upload, customize, and track their photo processing orders while empowering staff to manage these efficiently. The system integrates an exhaustive database detailing photo specifications, such as dimensions, resolution, and print type, ensuring precision in the photo development process.

A robust inventory management component is paramount, overseeing the stock levels of photo processing materials and sending alerts when replenishments are necessary. The billing module streamlines the financial interactions, seamlessly generating invoices for services availed and logging payment details. Enhanced by analytics and reporting capabilities, the system offers invaluable insights into customer preferences and business performance, guiding strategic decision-making.

Furthermore, communication utilities embedded within ensure customers are always apprised of their order status, and can easily liaise with staff for any clarifications. Employee management tools further ensure optimal workforce distribution and task assignments. Altogether, the Photo Lab Management System amalgamates these functional requirements to revolutionize photo lab operations, heightening efficiency, customer contentment, and business growth.

3.2 NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements of the Photo Lab Management System are indispensable in guaranteeing its dependability, efficiency, and user-friendliness. Foremost, the system should demonstrate exemplary performance, proficiently managing substantial

user traffic and vast data sets, while swiftly responding to user inputs. Considering the growth prospects of photo labs and evolving customer requirements, scalability becomes non-negotiable, ensuring that the system can seamlessly accommodate future expansions.

Security is of utmost concern in this digital age. The system must incorporate rigorous security protocols to protect sensitive photographic content and personal customer data. Features such as encryption, multifactor authentication, and adherence to pertinent data protection standards are essential to maintain the sanctity of the stored information.

Consistent reliability is crucial, requiring the system to function optimally around the clock with negligible downtime. Measures like frequent data backups and a fail-safe disaster recovery plan reinforce this reliability. A user-centric design ensures that the interface is intuitive, with comprehensive training modules and documentation facilitating a smooth user experience.

Cross-device and browser compatibility are essential, ensuring customers and lab staff can access the system from any platform. Adherence to photographic industry standards and applicable legislation underscores the system's credibility. Smooth integration capabilities, particularly with external printing or storage solutions, would streamline the photographic processing workflow.

Monitoring tools to gauge system health and responsiveness ensure proactive issue detection and resolution, while the system's overall design should remain cost-effective, delivering value within budgetary limits. Inclusivity is paramount, with accessibility features catering to differently-abled users. Lastly, the system should be built for the long haul, ensuring easy updates and modular inclusions as the photo lab industry evolves. In essence, these non-functional requirements collectively define a resilient, secure, and efficient Photo Lab Management System, enhancing the overall functionality and trustworthiness of photographic processing operations.

3.3 HARDWARE REQUIREMENTS

- Processor Type: Intel Core i5 or equivalent
- Processor Speed: 3.40 GHz or higher
- RAM (Memory): 8 GB or higher
- Hard Disk: 500 GB SSD (Solid State Drive) or higher
- Keyboard: Standard 101/102 keys keyboard
- Mouse: Optical mouse or equivalent pointing device
- Display: Monitor with a minimum resolution of 1920x1080 pixels

3.4 SOFTWARE REQUIREMENTS

- Operating System: Windows 10 or Windows 11
- Web Server: Apache Tomcat or equivalent
- Programming Language: Java
- Database Management System: MySQL or equivalent
- Integrated Development Environment (IDE): Eclipse, IntelliJ IDEA, or any Java IDE of choice
- Front-end Technologies: HTML, CSS, JavaScript
- Back-end Framework: Spring Framework (Java)
- Version Control: Git
- Security: SSL/TLS certificate for secure communication

- Payment Gateway Integration: PayPal, Stripe, or other preferred payment gateways
- Mobile App (Optional): Android Studio for Android app development, Xcode for iOS app development

CHAPTER 4

SYSTEM ANALYSIS

4.1 PROBLEM STATEMENT

The problem statement for photo lab management system can be articulated as follows: “Photo Lab Management System is a computerized system for complete and effective management of photo processing labs. With this system, the billing and inventory management in photo studios can be done efficiently to ensure the optimum utilization of available resources, thereby strengthening the productivity levels.

4.2 EXISTING APPROACH

When looking at the landscape of photo lab management, various traditional and contemporary approaches emerge, each with its strengths and limitations. Historically, photo labs have relied heavily on manual methods. Customers would physically visit the studio or lab, drop off their films or photos, fill out order forms detailing their requirements, and then return to pick up their finished products. While this approach has the benefit of face-to-face interaction and customer trust, it's time-consuming and limits the lab's reach to local clientele.

Many photo labs then transitioned to a digital approach by setting up basic websites. Customers could view services, prices, and perhaps upload photos for processing. However, these early websites often lacked the dynamic interactivity customers sought, like real-time order tracking, image previews, or sophisticated editing tools. Plus, they still required manual backend processing by lab staff.

To reach a broader audience, some photo labs partnered with third-party online platforms that aggregated multiple service providers. While this expanded their reach and offered a more polished online interface, it often came at the cost of individual branding and direct customer engagement, and sometimes, significant commission fees.

In a push for brand distinction, several photo labs ventured into developing dedicated mobile apps. Customers could order services, upload photos, and track progress right from their smartphones. But, as with any app-based approach, the challenges include the cost of development, regular updates, compatibility issues, and the necessity for customers to download the app.

Emerging technologies like cloud storage integrations, AI-driven image enhancements, and virtual reality previews have begun to infuse into the sector, offering opportunities for innovation. However, despite the multitude of approaches available, photo labs often grapple with issues like data privacy, image quality consistency, delivery logistics, and integrating the latest tech seamlessly. As the digital world continues to evolve, the demand is increasing for more cohesive, user-friendly, and technologically advanced solutions that can address the multifaceted challenges of photo lab management.

4.3 DISADVANTAGE OF EXISTING APPROACH

The existing approach to photo lab management exhibits numerous drawbacks that hinder its overall efficiency and customer satisfaction. Starting with traditional brick-and-mortar operations, customers often faced challenges with accessibility. Requiring a physical visit to the lab, waiting for processing, and then returning for pickups made it time-consuming and inconvenient.

Early digital transitions, like basic websites, certainly provided an online presence, but often lacked interactive features. Customers couldn’t see real-time status updates of their orders, request modifications, or get immediate feedback on their requirements. These static websites also didn’t always integrate with modern payment gateways, limiting the convenience of online transactions.

For photo labs collaborating with third-party online aggregators, while it expanded their digital footprint, there were challenges of diluted brand identity. Such platforms might prioritize their branding over individual labs, making it hard for a specific lab to distinguish itself. The commission fees and lack of direct customer data access further compounded the challenges.

The labs that ventured into app-based services encountered barriers too. App development and maintenance costs were significant. Furthermore, convincing customers to download yet another app on their devices, especially if they're occasional users, posed a challenge.

The integration of various technologies, like cloud storage or AI enhancements, into the photo lab operations, is resource-intensive and requires continual updates to keep up with evolving tech trends. Moreover, with the increasing importance of data privacy, many existing systems might not be equipped with the necessary security protocols, risking breaches of sensitive customer data.

Lastly, even with digital transformations, many photo labs faced challenges in delivering physical prints promptly and safely, especially in geographies with complex logistics. Consequently, while the existing approaches provided certain benefits, they also came with a host of challenges, underscoring the need for a more integrated and modernized photo lab management system.

4.4 ADVANTAGES OF THE PROPOSED SYSTEM

The proposed photo lab management system seeks to address the shortcomings of the existing approaches and integrate the benefits of modern technology, ultimately offering a holistic solution for photo lab operations. Here are the significant advantages of this new system:

4.4.1 Efficient Operations:

The system centralizes all operations, from order placements to delivery, ensuring streamlined processes and reduced manual interventions. This efficiency translates to quicker order processing and increased customer satisfaction.

4.4.2 User-Friendly Interface:

A well-designed, intuitive interface ensures that both customers and staff can navigate the system with ease. This makes placing orders, tracking progress, and managing inventory much more straightforward.

4.4.3 Data Security:

Recognizing the importance of data privacy, the proposed system incorporates robust security protocols to protect customer data, ensuring compliance with prevailing regulations and building trust with users.

4.4.4 Cost Savings:

While the initial setup might have associated costs, in the long run, the automation and efficient management lead to significant cost savings in terms of manpower, resource utilization, and operational expenses.

4.4.5 Scalability:

The system is designed to grow with the business. As the photo lab expands its offerings or scales its operations, the system can be easily upgraded to accommodate these changes.

4.4.6 Real-time Communication:

With integrated communication tools, customers can receive real-time updates about their orders, ask questions, or request modifications, ensuring transparency throughout the process.

In essence, the proposed photo lab management system offers a balanced mix of advanced technology, user-centric design, and efficient operations. Its implementation promises to revolutionize the way photo labs operate, ensuring they remain competitive in the digital age while delivering unparalleled value to their customers.

CHAPTER 5

SYSTEM DESIGN

5.1 INPUT DESIGN

The input design for a Photo Lab Management System is paramount in ensuring an efficient, user-friendly, and error-free operation. Given the nuanced requirements of photo processing, several aspects must be prioritized:

Starting with user-centric design, it's essential to ensure that customers can easily upload their photos. The system should support various file formats like JPEG, PNG, RAW, etc., and offer guidelines on the preferred resolution and file size. Additionally, the interface should provide real-time feedback, indicating successful uploads or errors.

For tasks that require detailed specifications, such as photo edits or custom print options, user-friendly forms are essential. Drop-down menus for common edit requests, sliders for adjustments like brightness or contrast, and visual icons for features like cropping or filters can streamline user input.

Address and contact information should have clear fields. Features like address auto-complete and saving addresses for future use can enhance user convenience. For those opting for in-store pickups, a simple date and time selector should be available.

For the lab staff and administrators, a sophisticated input design would allow them to update the status of orders, input inventory details for photo papers, chemicals, or other supplies, and adjust pricing or promotional details. Batch processing tools can enable them to input actions or status updates for multiple orders at once.

Given that photo processing can sometimes be subjective, an option for customers to provide notes or special instructions is essential. Similarly, a feedback mechanism to capture user inputs on the quality of prints or edits will help the lab maintain high standards and address any concerns promptly.

A crucial aspect of the Photo Lab Management System would be the order tracking and status updates. Input fields to update order stages, like 'Processing', 'Ready for Pickup', or 'Dispatched', can keep the customer informed.

Incorporating regular testing sessions and gathering user feedback will fine-tune the input design, ensuring it remains intuitive and meets the dynamic needs of both the customers and the photo lab staff. The end goal is a seamless blend of functionality and user experience, making photo processing a hassle-free task for all involved parties.

5.2 OUTPUT DESIGN

The output design of the Photo Lab Management System is integral in conveying processed information to the users effectively, making it a pivotal component of the user experience. Given the visual nature of photography, the output should be designed to be visually appealing, intuitive, and informative.

Customer Dashboard: Once logged in, customers should be greeted with a dashboard that displays their order history, order statuses (e.g., "In Process", "Completed", "Shipped"), and estimated delivery or pick-up times. Each order can be clickable to provide more detailed information.

Invoice and Billing Details: After placing an order, customers should be provided with a detailed, itemized invoice. This invoice should be downloadable in PDF format and also sent to the customer's registered email.

In essence, the output design of the Photo Lab Management System should be centered around user needs, providing clear, visually engaging, and timely information. Regular user feedback will further help refine and optimize these output interfaces, ensuring they align perfectly with user expectations and business objectives.

5.3 MODULE DESCRIPTION

A Photo Lab Management System is an intricate software solution tailored to streamline and refine the operations of photo processing labs. This system is compartmentalized into several key modules, each honed to perform a distinct function:

5.3.1 User Management Module:

Oversees customer registration, authentication, and profile management. This ensures that both staff and customers have secure access to relevant parts of the system.

5.3.2 Order and Scheduling Module:

Customers can place orders for photo prints or digital enhancements, and staff can track and manage these requests, integrating them with real-time work schedules.

5.3.3 Inventory Management Module:

Monitors photo paper stocks, printing supplies, and other essentials. It maintains optimal inventory levels and sends alerts or automates orders when restocking is needed.

5.3.4 Billing and Payment Module:

Generates quotes and invoices based on services availed, tracks payment status, and offers multiple payment gateway integrations for customer convenience.

5.3.5 Security and Data Protection Module:

Implements stringent measures like encryption and access controls to protect sensitive customer photos and personal data.

5.3.6 Scalability and Performance Module:

Ensures that as the lab's customer base grows, the system can adapt without performance hitches, and maintains smooth operations irrespective of the workload.

In essence, these interwoven modules form the robust infrastructure of the Photo Lab Management System, equipping photo labs to offer top-notch services, optimize workflow, enhance customer satisfaction, and ultimately, thrive in the competitive realm of photo processing.

5.4 PHOTO LAB MANAGEMENT SYSTEM DIAGRAMS

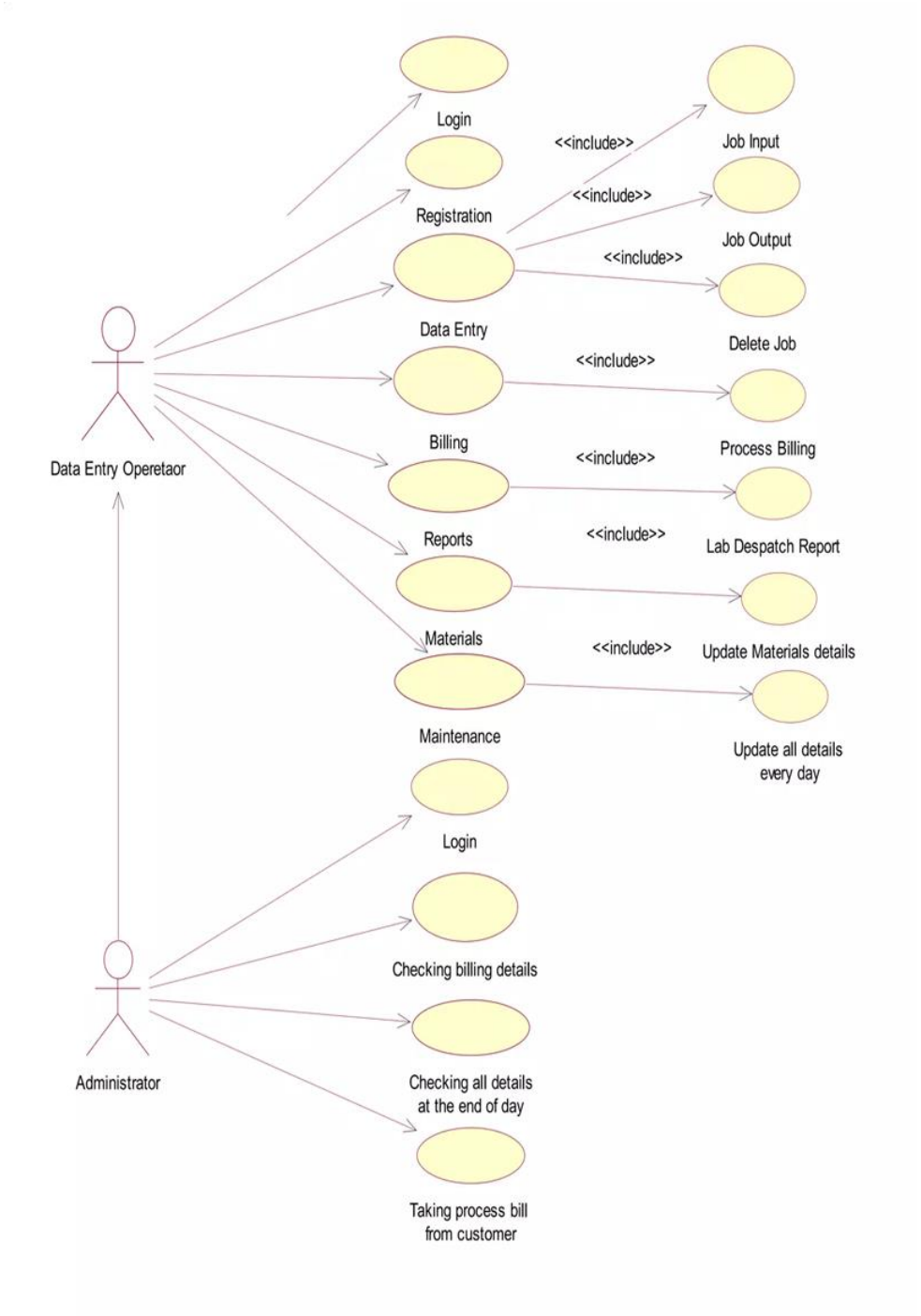

Fig. 5.4.1 USE CASE DIAGRAM

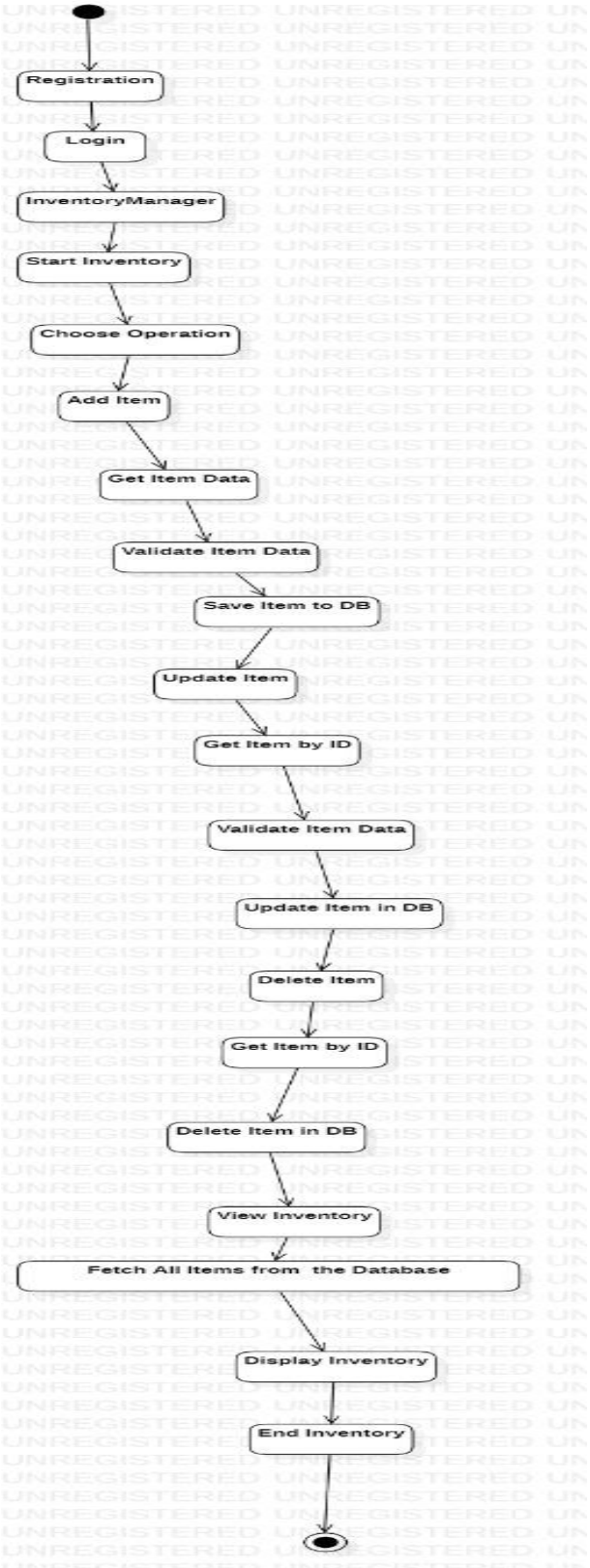

Fig. 5.4.2 FLOW DIAGRAM

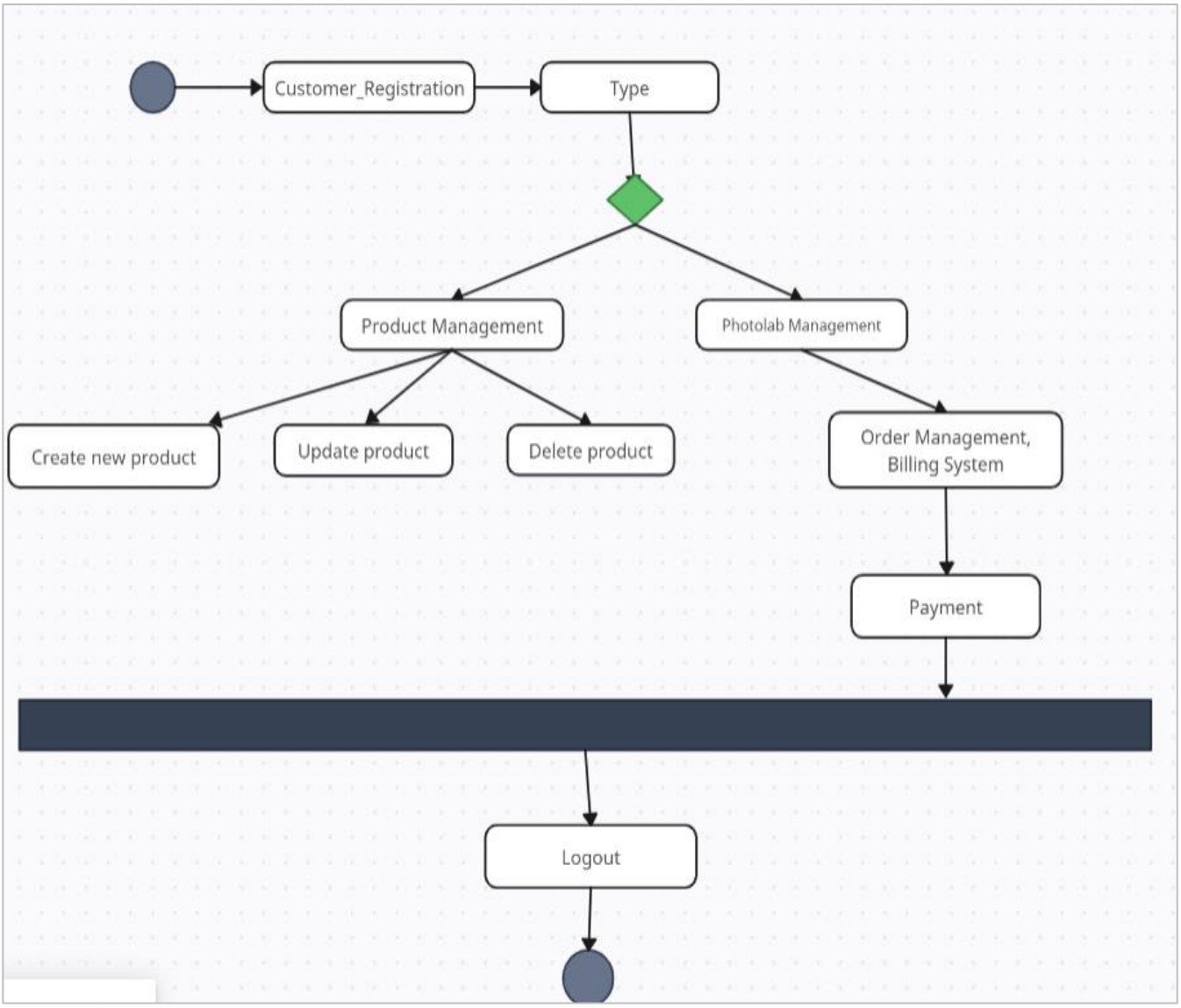

Fig. 5.4.3 STATE DIAGRAM

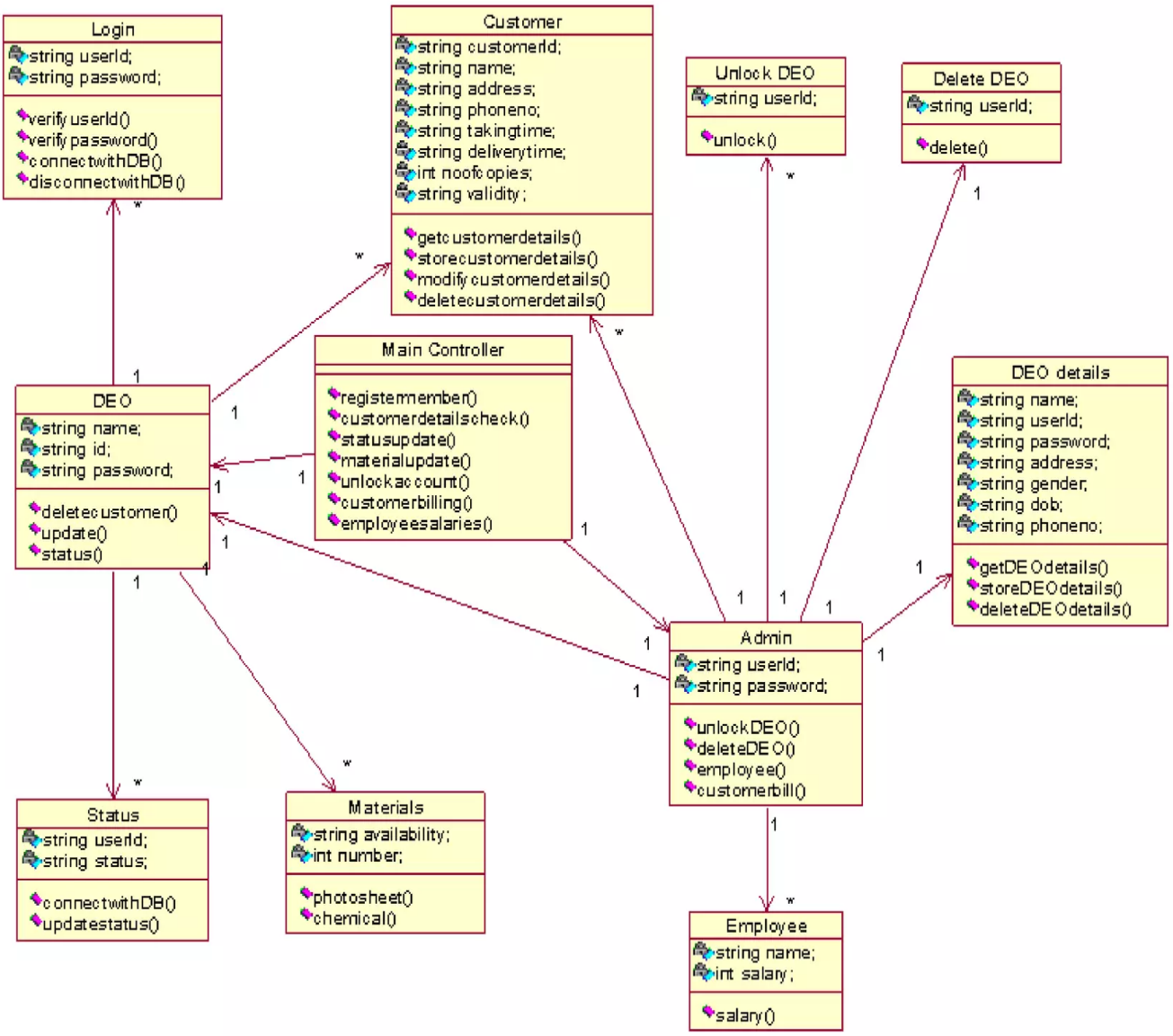

Fig. 5.4.4 CLASS DIAGRAM

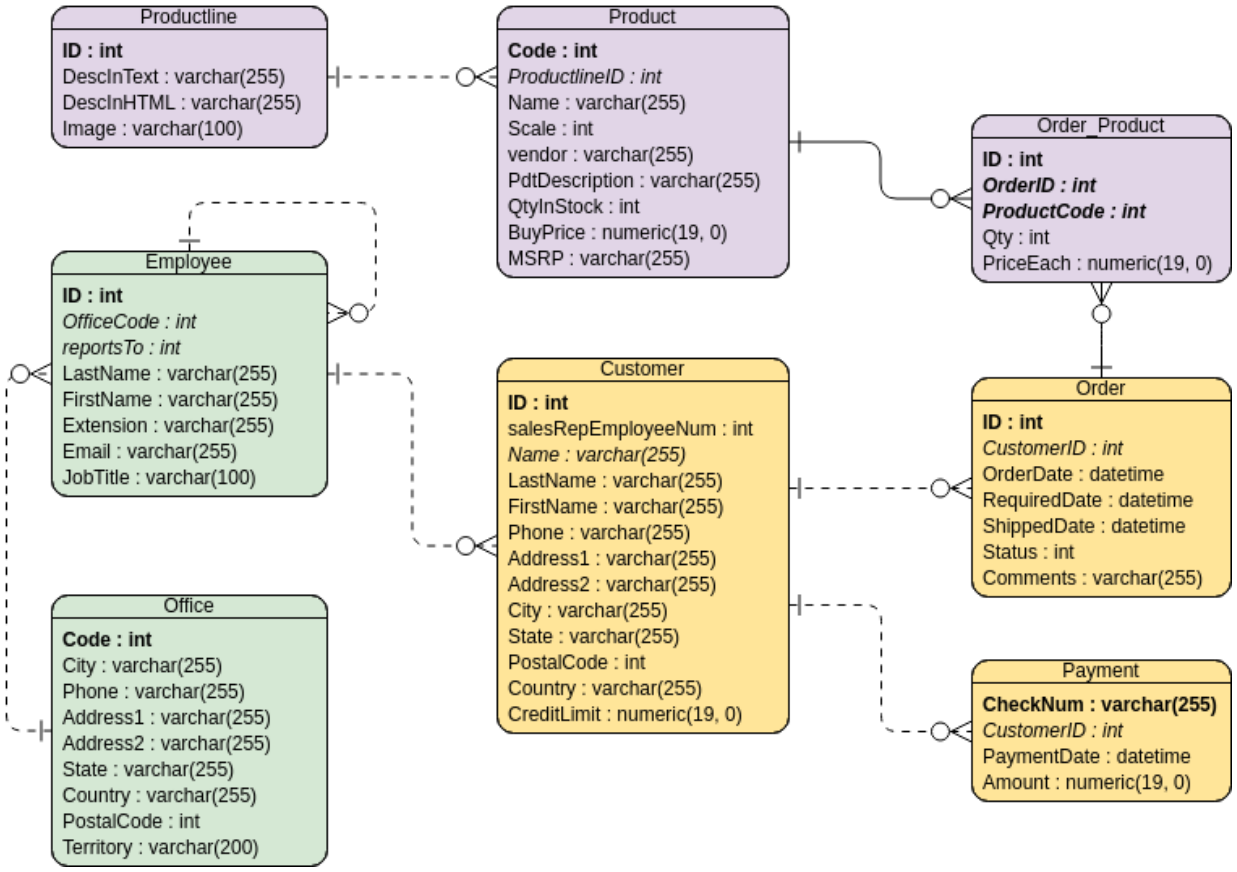

Fig. 5.4.5 ENHANCED E R DIAGRAM

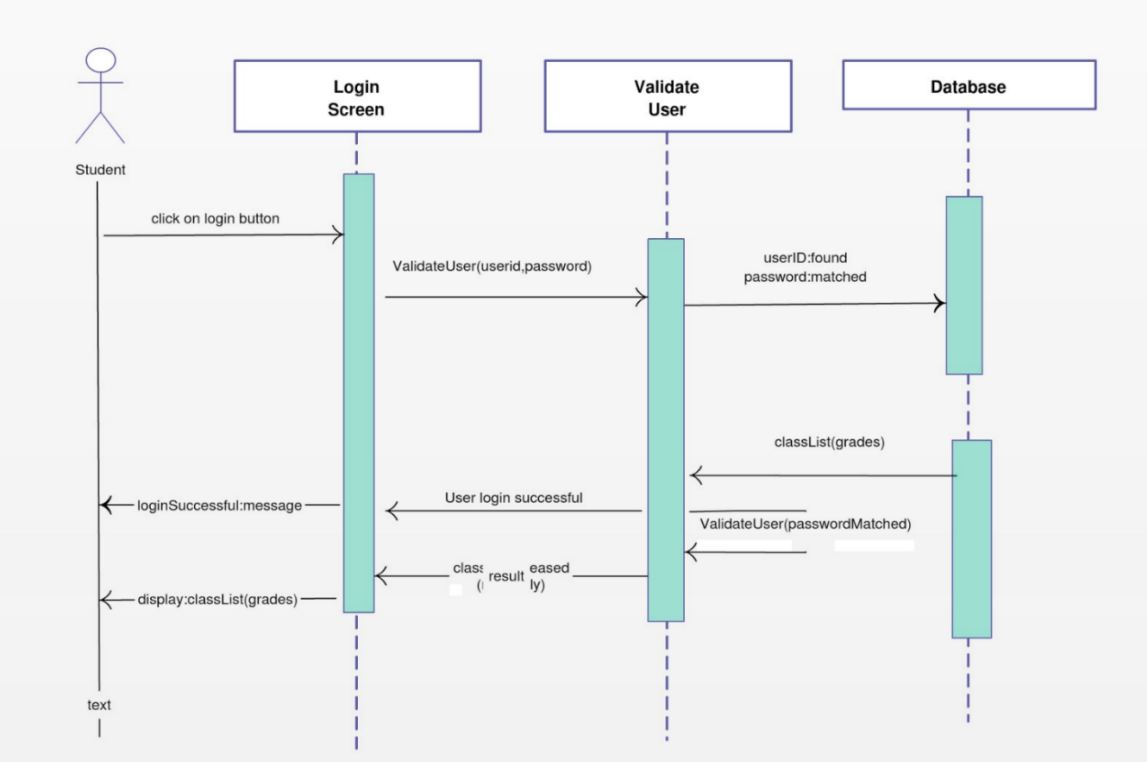

Fig. 5.4.6 SEQUENCE DIAGRAM

CHAPTER 6

IMPLEMENTATION

6.1 CODING

Main.java:

```
package miniProjPackage;
import java.util.Scanner;
import java.sql.SQLException;

public class Main {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);

      while(true) {
         System.out.println("Select the system to run:");
         System.out.println("1. Product Management");
         System.out.println("2. Photo Lab Management");
         System.out.println("3. Exit");
         int choice = scanner.nextInt();
         try {
            switch (choice) {
               case 1:                                          // Product Management
                  product product = new product();
                  product.run();                                // Polymorphism
                  break;
               case 2:                                          // Photo Lab Management
                  PhotoLabManagement photoLabManagement = new PhotoLabManagement();
                  photoLabManagement.run(); // Polymorphism
                  break;
```

```
                case 3:                                                    // Exit
                    System.out.println("Exiting...");
                    scanner.close();
                    System.exit(0);

                default:
                  System.out.println("Invalid choice! Please choose a number between 1 and 3.");
                   break;
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

DatabaseOperation.java:

```
package miniProjPackage;

import java.sql.*;

public abstract class DatabaseOperation {
    protected static final String url = "jdbc:mysql://localhost:3306/miniproj";
    protected static final String user = "root";
    protected static final String password = "root";
    protected Connection conn;

    public DatabaseOperation() throws SQLException {
        conn = DriverManager.getConnection(url, user, password);
    }
    public abstract void run();                                            // Abstraction
}
```

product.java:

```
package miniProjPackage;

import java.sql.*;
import java.util.Scanner;

public class product extends DatabaseOperation {                    // Inheritance
    public product() throws SQLException {
        super();
    }

    public static void main(String[] args) {
        try {
            product product = new product();
            product.run();                                           // Polymorphism
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
    @Override                // Polymorphism: Product provides its own implementation of run()
    public void run()  {                                             // Declare SQLException
        Scanner scanner = new Scanner(System.in);
        PreparedStatement pstmt;

        try {                                        // Include a try-catch block for SQLException
            while(true) {
                System.out.println("Please select an operation:");
                System.out.println("1. Create a new product");
                System.out.println("2. View all products");
                System.out.println("3. Update a product's cost");
                System.out.println("4. Delete a product");
                System.out.println("5. View Customer details");
                System.out.println("6. View Orders");
```

```
            System.out.println("7. Exit");
            int choice = scanner.nextInt();

            switch (choice) {

               case 1:                                                    // CREATE
                  System.out.println("Enter new product ID:");
                  int prodId = scanner.nextInt();
                  System.out.println("Enter new product name:");
                  scanner.nextLine();
                  String prodName = scanner.nextLine();
                  System.out.println("Enter new product cost:");
                  while (!scanner.hasNextDouble()) {
                     System.out.println("That's not a valid number! Try again:");
                     scanner.next();                                      // this is important!
                  }
                  double prodCost = scanner.nextDouble();

                  String insertProduct = "INSERT INTO product (prodId, prodName, cost)
                  VALUES (?, ?, ?)";
                  pstmt = conn.prepareStatement(insertProduct);
                  pstmt.setInt(1, prodId);
                  pstmt.setString(2, prodName);
                  pstmt.setDouble(3, prodCost);
                  pstmt.executeUpdate();
                  System.out.println("Product has been added successfully.");
                  break;

               case 2:                                                     // READ
                  Statement stmt = conn.createStatement();
                  ResultSet rs = stmt.executeQuery("SELECT * FROM product");
                  while (rs.next()) {
                     System.out.println("Product ID: " + rs.getInt("prodId") + ", Product Name:
" + rs.getString("prodName") + ", Cost: " + rs.getDouble("cost"));
```

```
    }
    break;

case 3:                                                    // UPDATE
    System.out.println("Enter product ID to update:");
    prodId = scanner.nextInt();
    System.out.println("Enter new product cost:");
    prodCost = scanner.nextDouble();

    String updateProduct = "UPDATE product SET cost = ? WHERE prodId = ?";
    pstmt = conn.prepareStatement(updateProduct);
    pstmt.setDouble(1, prodCost);
    pstmt.setInt(2, prodId);
    pstmt.executeUpdate();
    System.out.println("Product price has been updated successfully.");
    break;

case 4:                                                    // DELETE
    System.out.println("Enter product ID to delete:");
    prodId = scanner.nextInt();

    String deleteProduct = "DELETE FROM product WHERE prodId = ?";
    pstmt = conn.prepareStatement(deleteProduct);
    pstmt.setInt(1, prodId);
    pstmt.executeUpdate();
    System.out.println("Product has been deleted successfully.");
    break;
case 5: // PRINT CUSTOMER TABLE
    Statement customerStmt = conn.createStatement();
    ResultSet customerRs =
    customerStmt.executeQuery("SELECT * FROM customer");
    System.out.println("Customer details:");
    while (customerRs.next()) {
        System.out.println("Customer ID: " + customerRs.getInt("cId") + ",
```

```
    Customer Name: " + customerRs.getString("cName") + ", Mobile Number: " +
    customerRs.getString("mobileNo") + ", Address: " + customerRs.getString("address"));
                    }
                    break;
                case 6: // PRINT ORDER TABLE
                    Statement orderStmt = conn.createStatement();
                    ResultSet orderRs = orderStmt.executeQuery("SELECT * FROM orders");
                    System.out.println("Order details:");
                    while (orderRs.next()) {
                        System.out.println("Order ID: " + orderRs.getInt("ordId") + ", Order Name:
" + orderRs.getString("ordName") + ", Cost: " + orderRs.getDouble("cost") + ", Product ID: "
+ orderRs.getInt("prodId") + ", Customer ID: " + orderRs.getInt("cId"));
                    }
                    break;
                case 7:                                                    // EXIT
                    System.out.println("Exiting...");
                    conn.close();
                    scanner.close();
                    System.exit(0);

                default:
                  System.out.println("Invalid choice! Please choose a number between 1 and 5.");
                  break;
              }
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

PhotoLabManagement.java:

```
package miniProjPackage;

import java.sql.*;
import java.util.*;

public class PhotoLabManagement {
    private static final String url = "jdbc:mysql://localhost:3306/miniproj";
    private static final String user = "root";
    private static final String password = "root";

    public static void main(String[] args) {
        PhotoLabManagement photoLabManagement = new PhotoLabManagement();
        photoLabManagement.run();
    }

    public void run() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Customer ID:");
        int cId = scanner.nextInt();



         try {
            Connection conn = DriverManager.getConnection(url, user, password);
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM product");

            Map<Integer, String> products = new HashMap<>();
            Map<Integer, Double> productPrices = new HashMap<>();

            int i = 1;
            while (rs.next()) {
                products.put(i, rs.getString("prodName"));
```

```
            productPrices.put(i, rs.getDouble("cost"));
            i++;
        }

        for (Map.Entry<Integer, String> entry : products.entrySet()) {
            double productCost = productPrices.get(entry.getKey());
            System.out.println(entry.getKey() + ". " + entry.getValue() + " - Cost: " +
        productCost);
        }
        List<Integer> selectedProducts = new ArrayList<>();
        System.out.println("Select product(s) by entering the corresponding number (enter
        0 when done):");

        while (true) {
            int prodNumber = scanner.nextInt();
            if (prodNumber == 0) {
                break;
            }
            selectedProducts.add(prodNumber);
        }

        double totalPrice = 0;
        for (int prodNumber : selectedProducts) {

        totalPrice += productPrices.get(prodNumber);
        }

        System.out.println("Total Price: " + totalPrice);

        String insertOrder = "INSERT INTO orders (ordName, cost, prodId, cId) VALUES
    (?, ?, ?, ?)";
        PreparedStatement pstmt = conn.prepareStatement(insertOrder);

        for (int prodNumber : selectedProducts) {
```

```
                pstmt.setString(1, products.get(prodNumber));
                pstmt.setDouble(2, productPrices.get(prodNumber));
                pstmt.setInt(3, prodNumber);
                pstmt.setInt(4, cId);
                pstmt.executeUpdate();
            }

            System.out.println("Order has been saved successfully.");

            } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

6.2 OUTPUT

Main page:

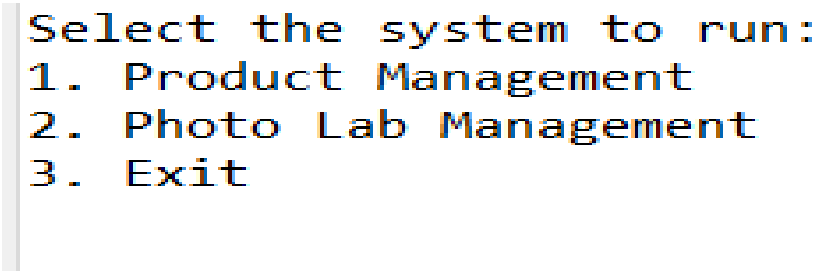

Fig. 6.2.2 Main Page

Product page:

```
2. Photo Lab Management
3. Exit
1
Please select an operation:
1. Create a new product
2. View all products
3. Update a product's cost
4. Delete a product
5. Exit
2
Product ID: 1, Product Name: Passport Size, Cost: 100.0
Product ID: 2, Product Name: Stamp Size, Cost: 90.0
Product ID: 3, Product Name: Photo Frame, Cost: 400.0
Product ID: 4, Product Name: Wedding Photoshoot, Cost: 10000.0
Product ID: 5, Product Name: Lamination, Cost: 210.0
```

Fig. 6.2.3 Products

Order and Billing page:

```
Select the system to run:
1. Product Management
2. Photo Lab Management
3. Exit
2
Enter Customer ID:
1
1. Passport Size - Cost: 100.0
2. Stamp Size - Cost: 90.0
3. Photo Frame - Cost: 425.0
4. Wedding Photoshoot - Cost: 10000.0
5. Lamination - Cost: 210.0
Select product(s) by entering the corresponding number
1
3
5
0
Total Price: 735.0
Order has been saved successfully.
```

Fig. 6.2.4 Order and Billing

Customer Details:

```
Please select an operation:
1. Create a new product
2. View all products
3. Update a product's cost
4. Delete a product
5. View Customer details
6. View Orders
7. Exit
5
Customer details:
Customer ID: 1, Customer Name: John Doe, Mobile Number: 1234567890, Address: 123 Main St
Customer ID: 2, Customer Name: David Beckham, Mobile Number: 0987654321, Address: 456 Maple St
Customer ID: 3, Customer Name: Cristiano Waldo, Mobile Number: 8787654321, Address: 116 Kaiwal St
Customer ID: 4, Customer Name: Leo Wessi, Mobile Number: 9080123456, Address: 109 NTA Street
Customer ID: 5, Customer Name: Whyrat Kohli, Mobile Number: 9123456710, Address: 134 Wardo Colony
```

Fig. 6.2.5 Customer Details

Order Details:

```
6
Order details:
Order ID: 1, Order Name: Wedding Photoshoot, Cost: 10000.0, Product ID: 4, Customer ID: 3
Order ID: 2, Order Name: Passport Size, Cost: 100.0, Product ID: 1, Customer ID: 3
Order ID: 3, Order Name: Passport Size, Cost: 100.0, Product ID: 1, Customer ID: 3
Order ID: 4, Order Name: Photo Frame, Cost: 400.0, Product ID: 3, Customer ID: 5
Order ID: 5, Order Name: Lamination, Cost: 210.0, Product ID: 5, Customer ID: 5
Order ID: 6, Order Name: Passport Size, Cost: 100.0, Product ID: 1, Customer ID: 4
Order ID: 7, Order Name: Passport Size, Cost: 100.0, Product ID: 1, Customer ID: 4
Order ID: 8, Order Name: Lamination, Cost: 210.0, Product ID: 5, Customer ID: 4
Order ID: 9, Order Name: Photo Frame, Cost: 400.0, Product ID: 3, Customer ID: 5
Order ID: 10, Order Name: Passport Size, Cost: 100.0, Product ID: 1, Customer ID: 3
Order ID: 11, Order Name: Stamp Size, Cost: 90.0, Product ID: 2, Customer ID: 3
Order ID: 12, Order Name: Stamp Size, Cost: 90.0, Product ID: 2, Customer ID: 3
Order ID: 13, Order Name: Lamination, Cost: 210.0, Product ID: 5, Customer ID: 3
```

Fig. 6.2.6 Order Details

CHAPTER 7

TESTING

7.1 INTRODUCTION

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive. A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

7.2 STRATEGIC APPROACH TO SOFTWARE TESTING

The software engineering process can be viewed as a spiral.Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behaviour, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn. A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally, we arrive at system testing, where the software and other system elements are tested as a whole.

7.3 UNIT TESTING

Unit testing focuses verification effort on the smallest unit of software design, the module. The unit testing we have is white box oriented and some modules the steps are conducted in parallel.

7.3.1 WHITE BOX TESTING

To follow the concept of white box testing we have tested each form. We have created independently to verify that Data flow is correct and all conditions are exercised to check theirvalidity.

7.3.1.1.1 All loops are executed on their boundaries. This type of testing ensures that all independent paths have been exercised at least once

7.3.1.1.2 All logical decisions have been exercised on their true and false sides

7.3.1.1.3 All loops are executed at their boundaries and within their operational bounds

7.3.1.1.4 All internal data structures have been exercised to assure their validity paths.

7.3.2 CONDITIONAL TESTING

In this part of the testing each of the conditions were tested to both true and false aspects. And all the resulting paths were tested so that each path that may generate on a particular condition is traced to uncover any possible errors.

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing.

7.3.3 DATA FLOW TESTING

This type of testing selects the path of the program according to the location of definition and use of variables. This kind of testing was used only when some local variable were declared. The definition-use chain method was used in this type of testing.

7.3.4 LOOP TESTING

In this type of testing all the loops are tested to all the limits possible. The following exercise was adopted for all loops:

7.3.4.1 All the loops were tested at their limits, just above them and just below them. All theloops were skipped at least once.

7.3.4.2 For nested loops test the inner most loop first and then work outwards. For concatenatedloops the values of dependent loops were set with the help of connected loop.

7.3.4.3 Unstructured loops were resolved into nested loops or concatenated loops and tested asabove.

7.3.4.4 Each unit has been separately tested by the development team itself and all the inputhave been validated.

7.4 TEST CASE

A test case, in software engineering, is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do. The mechanism for determining whether a software program or system has passed or failed such a test is known as a test oracle. In some settings, an oracle could be a requirement or use case, while in others it could be a heuristic. It may take many test cases to determine that a software program or system is considered sufficiently scrutinized to be released. Test cases are often referred to as test scripts, particularly when written - when they are usually collected into test suites.

7.4.1 TEST CASE I

```
Select the system to run:
1. Product Management
2. Photo Lab Management
3. Exit
1
Please select an operation:
1. Create a new product
2. View all products
3. Update a product's cost
4. Delete a product
5. View Customer details
6. View Orders
7. Exit
3
Enter product ID to update:
3
Enter new product cost:
425
Product price has been updated successfully.
```

EXPECTED OUTPUT: New Price: 425. Product price has been updated successfully.

ACTUAL OUTPUT: New Price: 425. Product price has been updated successfully.

```
mysql> select * from product;
+--------+--------------------+-------+
| prodId | prodName           | cost  |
+--------+--------------------+-------+
|      1 | Passport Size      |   100 |
|      2 | Stamp Size         |    90 |
|      3 | Photo Frame        |   400 |
|      4 | Wedding Photoshoot | 10000 |
|      5 | Lamination         |   210 |
+--------+--------------------+-------+
```

```
mysql> select * from product;
+--------+--------------------+-------+
| prodId | prodName           | cost  |
+--------+--------------------+-------+
|      1 | Passport Size      |   100 |
|      2 | Stamp Size         |    90 |
|      3 | Photo Frame        |   425 |
|      4 | Wedding Photoshoot | 10000 |
|      5 | Lamination         |   210 |
+--------+--------------------+-------+
```

Fig. 7.4.1 – The DB has been updated successfully as we can see the Price of "Photo Frame" has been updated.

7.4.2 TEST CASE II

```
Select the system to run:
1. Product Management
2. Photo Lab Management
3. Exit
2
Enter Customer ID:
1
1. Passport Size - Cost: 100.0
2. Stamp Size - Cost: 90.0
3. Photo Frame - Cost: 425.0
4. Wedding Photoshoot - Cost: 10000.0
5. Lamination - Cost: 210.0
Select product(s) by entering the corresponding number
1
3
5
0
Total Price: 735.0
Order has been saved successfully.
```

EXPECTED OUTPUT: Order has been saved successfully.

ACTUAL OUTPUT: Order has been saved successfully.

```
mysql> select * from orders;
+-------+--------------------+-------+--------+------+
| ordId | ordName            | cost  | prodId | cId  |
+-------+--------------------+-------+--------+------+
|     1 | Wedding Photoshoot | 10000 |      4 |    3 |
|     2 | Passport Size      |   100 |      1 |    3 |
|     3 | Passport Size      |   100 |      1 |    3 |
|     4 | Photo Frame        |   400 |      3 |    5 |
|     5 | Lamination         |   210 |      5 |    5 |
|     6 | Passport Size      |   100 |      1 |    4 |
|     7 | Passport Size      |   100 |      1 |    4 |
|     8 | Lamination         |   210 |      5 |    4 |
|     9 | Photo Frame        |   400 |      3 |    5 |
|    10 | Passport Size      |   100 |      1 |    3 |
|    11 | Stamp Size         |    90 |      2 |    3 |
|    12 | Stamp Size         |    90 |      2 |    3 |
|    13 | Lamination         |   210 |      5 |    3 |
|    14 | Stamp Size         |    90 |      2 |    2 |
|    15 | Photo Frame        |   425 |      3 |    2 |
|    16 | Lamination         |   210 |      5 |    2 |
|    17 | Passport Size      |   100 |      1 |    1 |
|    18 | Photo Frame        |   425 |      3 |    1 |
|    19 | Lamination         |   210 |      5 |    1 |
+-------+--------------------+-------+--------+------+
```

Fig. 7.4.2.1 – Orders Table in Database

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 CONCLUSION

In conclusion, deploying a Photo Lab Management System has the potential to revolutionize the operations of photo labs and studios significantly. Such a system addresses the intricate demands of photo processing businesses, offering structured solutions for billing, inventory management, and customer interactions. With this digital transformation, photo studios can ensure the optimum utilization of resources, enhancing productivity and client satisfaction.

Modern technology, like this system, not only streamlines processes but also offers the opportunity to expand client relations and services. Clients appreciate the ease of tracking their orders, understanding pricing structures, and the transparency in services offered. Furthermore, with the ability to manage resources and inventory digitally, studios can avoid overstocking, reduce wastage, and effectively manage operational costs.

In essence, a well-implemented Photo Lab Management System can redefine the way photo studios operate. It not only brings efficiency to daily tasks but also helps in carving out a competitive edge in a rapidly evolving industry. As the digital era progresses, tools like the Photo Lab Management System will undoubtedly be instrumental in shaping the future of the photo processing business.

8.2 FUTURE WORK:

The Photo Lab Management System can evolve and refine its services in the coming times to deliver a more enhanced and user-friendly photo processing experience. Recommendation engines could be fortified by harnessing artificial intelligence (AI) and machine learning, which can suggest photo enhancement or framing options based on user preferences and trending styles. The processing and print preparation times could be optimized using Internet of Things (IoT) devices, ensuring that customer orders are processed in the most efficient manner. Moreover, integrating augmented reality (AR) and virtual reality (VR) might allow users to virtually "experience" how their framed photos or albums would look in real spaces before finalizing orders.

The future of photo lab systems will also be shaped by sustainability and eco-friendly measures. As the world tilts towards eco-conscious choices, the system can introduce sustainable photo paper, reduce chemical waste, and collaborate with green suppliers for materials.

To cater to the broad spectrum of photography enthusiasts and professionals, the system should be adaptive to include newer photography techniques and trends, such as infrared photography, lomography, or even 3D photo prints. The continuous innovations in camera technology, especially in smartphones, will introduce photos in diverse formats and resolutions. The system must be adept to process these with precision.

Further, direct integration with popular cloud storage solutions or photo-sharing platforms can provide users the comfort of sending their digital photos for processing without the need for manual transfers. This seamless connectivity can significantly enhance user convenience.

In conclusion, as the world of photography and tech continues to intersect and evolve, the Photo Lab Management System must prioritize innovation, eco-friendliness, and adaptability to changing user preferences. Its sustained success will hinge on its ability to match strides with technological advancements and emerging photography trends.

REFERENCES

[1] Amey Thakur and Karan Dhiman, “Chat Room Using HTML, PHP, CSS, JS, AJAX.”, International Research Journal of Engineering and Technology (IRJET), 1948–1951, 08 June 2021. https://doi.org/10.6084/m9.figshare.14869167.

[2] Amey Thakur and Karan Dhiman, “Chat Room Using HTML, PHP, CSS, JS, AJAX.”, ArXiv, abs/2106.14704 (2021): n. Pag.

[3] Amey Thakur."Car Rental System", Volume 9, Issue VII, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: 402-412, ISSN: 2321-9653, https://doi.org/10.22214/ijraset.2021.36339.

[4] Amey Thakur, Mega Satish."Digital Bookstore", Volume 9, Issue VII, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: 1302-1313, ISSN: 2321-9653, https://doi.org/10.22214/ijraset.2021.36609

[5] Kofler, Michael. "phpMyAdmin." The Definitive Guide to MySQL5 (2005): 87-116.

[6] Musciano, Chuck, and Bill Kennedy. HTML & XHTML: The Definitive Guide: The Definitive Guide. " O'Reilly Media, Inc.", 2002.

[7] Raggett, Dave, Arnaud Le Hors, and Ian Jacobs. "HTML 4.01 Specification." W3C recommendation 24 (1999).

[8] Blansit, B. Douglas. "An Introduction to Cascading Style Sheets (CSS)." Journal of Electronic Resources in Medical Libraries 5, no. 4 (2008): 395-409.

[9] W3Schools. "W3Schools." (2013).

[10] Chavan, Varsha, Priya Jadhav, Snehal Korade, and Priyanka Teli. "Implementing customizable online food ordering system using web-based application." International Journal of Innovative Science, Engineering & Technology 2, no. 4 (2015): 722-727

APPENDICES

APPENDIX – I

SOURCE CODE

Main.java:

```
package miniProjPackage;
import java.util.Scanner;
import java.sql.SQLException;

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while(true) {
            System.out.println("Select the system to run:");
            System.out.println("1. Product Management");
            System.out.println("2. Photo Lab Management");
            System.out.println("3. Exit");

            int choice = scanner.nextInt();
            try {
                switch (choice) {
                    case 1:                                          // Product Management
                        product product = new product();
                        product.run();                               // Polymorphism
                        break;

                    case 2:                                          // Photo Lab Management
                     PhotoLabManagement photoLabManagement = new PhotoLabManagement();
                        photoLabManagement.run(); // Polymorphism
                        break;
```

```
                case 3:                                          // Exit
                    System.out.println("Exiting...");
                    scanner.close();
                    System.exit(0);

                default:
                System.out.println("Invalid choice! Please choose a number between 1 and 3.");
                    break;
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

DatabaseOperation.java:

```
package miniProjPackage;

import java.sql.*;

public abstract class DatabaseOperation {
    protected static final String url = "jdbc:mysql://localhost:3306/miniproj";
    protected static final String user = "root";
    protected static final String password = "root";
    protected Connection conn;

    public DatabaseOperation() throws SQLException {
        conn = DriverManager.getConnection(url, user, password);
    }
    public abstract void run();                                  // Abstraction
}
```

product.java:

```
package miniProjPackage;

import java.sql.*;
import java.util.Scanner;

public class product extends DatabaseOperation {                    // Inheritance
    public product() throws SQLException {
        super();
    }

    public static void main(String[] args) {
        try {
            product product = new product();
            product.run();                                           // Polymorphism
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
    @Override              // Polymorphism: Product provides its own implementation of run()
    public void run()  {                                                    //          Declare
SQLException
        Scanner scanner = new Scanner(System.in);
        PreparedStatement pstmt;

        try {                                    // Include a try-catch block for SQLException
            while(true) {
                System.out.println("Please select an operation:");
                System.out.println("1. Create a new product");
                System.out.println("2. View all products");
                System.out.println("3. Update a product's cost");
                System.out.println("4. Delete a product");
```

```
System.out.println("5. View Customer details");
System.out.println("6. View Orders");
System.out.println("7. Exit");
int choice = scanner.nextInt();

switch (choice) {

  case 1:                                                    // CREATE
    System.out.println("Enter new product ID:");
    int prodId = scanner.nextInt();
    System.out.println("Enter new product name:");
    scanner.nextLine();
    String prodName = scanner.nextLine();
    System.out.println("Enter new product cost:");
    while (!scanner.hasNextDouble()) {
      System.out.println("That's not a valid number! Try again:");
      scanner.next();
    }
    double prodCost = scanner.nextDouble();

String insertProduct = "INSERT INTO product (prodId, prodName, cost) VALUES (?, ?,
?)";
    pstmt = conn.prepareStatement(insertProduct);
    pstmt.setInt(1, prodId);
    pstmt.setString(2, prodName);
    pstmt.setDouble(3, prodCost);
    pstmt.executeUpdate();
    System.out.println("Product has been added successfully.");
    break;

  case 2:                                                    // READ
    Statement stmt = conn.createStatement();
    ResultSet rs = stmt.executeQuery("SELECT * FROM product");
    while (rs.next()) {
```

```
                System.out.println("Product ID: " + rs.getInt("prodId") + ", Product
Name: " + rs.getString("prodName") + ", Cost: " + rs.getDouble("cost"));
                }
                break;

            case 3:                                                    // UPDATE
                System.out.println("Enter product ID to update:");
                prodId = scanner.nextInt();
                System.out.println("Enter new product cost:");
                prodCost = scanner.nextDouble();

                String updateProduct = "UPDATE product SET cost = ? WHERE prodId =
?";
                pstmt = conn.prepareStatement(updateProduct);
                pstmt.setDouble(1, prodCost);
                pstmt.setInt(2, prodId);
                pstmt.executeUpdate();
                System.out.println("Product price has been updated successfully.");
                break;

            case 4:                                                    // DELETE
                System.out.println("Enter product ID to delete:");
                prodId = scanner.nextInt();

                String deleteProduct = "DELETE FROM product WHERE prodId = ?";
                pstmt = conn.prepareStatement(deleteProduct);
                pstmt.setInt(1, prodId);
                pstmt.executeUpdate();
                System.out.println("Product has been deleted successfully.");
                break;
                  case 5: // PRINT CUSTOMER TABLE
                Statement customerStmt = conn.createStatement();
                ResultSet customerRs =
customerStmt.executeQuery("SELECT * FROM customer");
```

```
                System.out.println("Customer details:");
                while (customerRs.next()) {
                    System.out.println("Customer ID: " + customerRs.getInt("cId") + ", Customer Name: " + customerRs.getString("cName") + ", Mobile Number: " + customerRs.getString("mobileNo") + ", Address: " + customerRs.getString("address"));
                }
                break;
            case 6: // PRINT ORDER TABLE
                Statement orderStmt = conn.createStatement();
                ResultSet orderRs = orderStmt.executeQuery("SELECT * FROM orders");
                System.out.println("Order details:");
                while (orderRs.next()) {
                    System.out.println("Order ID: " + orderRs.getInt("ordId") + ", Order Name: " + orderRs.getString("ordName") + ", Cost: " + orderRs.getDouble("cost") + ", Product ID: " + orderRs.getInt("prodId") + ", Customer ID: " + orderRs.getInt("cId"));
                }
                break;
            case 7:                                                    // EXIT
                System.out.println("Exiting...");
                conn.close();
                scanner.close();
                System.exit(0);

            default:
              System.out.println("Invalid choice! Please choose a number between 1 and 5.");
               break;
          }
        }
      } catch (SQLException ex) {
        ex.printStackTrace();
      }
   }
}
```

PhotoLabManagement.java:

```
package miniProjPackage;

import java.sql.*;
import java.util.*;

public class PhotoLabManagement {
    private static final String url = "jdbc:mysql://localhost:3306/miniproj";
    private static final String user = "root";
    private static final String password = "root";

    public static void main(String[] args) {
        PhotoLabManagement photoLabManagement = new PhotoLabManagement();
        photoLabManagement.run();
    }

    public void run() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter Customer ID:");
        int cId = scanner.nextInt();



         try {
            Connection conn = DriverManager.getConnection(url, user, password);
            Statement stmt = conn.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT * FROM product");

            Map<Integer, String> products = new HashMap<>();
            Map<Integer, Double> productPrices = new HashMap<>();

            int i = 1;
            while (rs.next()) {
```

```
            products.put(i, rs.getString("prodName"));
            productPrices.put(i, rs.getDouble("cost"));
            i++;
        }

        for (Map.Entry<Integer, String> entry : products.entrySet()) {
            double productCost = productPrices.get(entry.getKey());
  System.out.println(entry.getKey() + ". " + entry.getValue() + " - Cost: " + productCost);
        }
        List<Integer> selectedProducts = new ArrayList<>();
System.out.println("Select product(s) by entering the corresponding number (enter 0 when done):");

        while (true) {
            int prodNumber = scanner.nextInt();
            if (prodNumber == 0) {
                break;
            }
            selectedProducts.add(prodNumber);
        }

        double totalPrice = 0;
        for (int prodNumber : selectedProducts) {
                totalPrice += productPrices.get(prodNumber);
        }
        System.out.println("Total Price: " + totalPrice);

        String insertOrder = "INSERT INTO orders (ordName, cost, prodId, cId) VALUES (?, ?, ?, ?)";
        PreparedStatement pstmt = conn.prepareStatement(insertOrder);
```

```
            for (int prodNumber : selectedProducts) {
                pstmt.setString(1, products.get(prodNumber));
                pstmt.setDouble(2, productPrices.get(prodNumber));
                pstmt.setInt(3, prodNumber);
                pstmt.setInt(4, cId);
                pstmt.executeUpdate();
            }

            System.out.println("Order has been saved successfully.");

        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

APPENDIX – II

SCREENSHOT

Home page:

```
Select the system to run:
1. Product Management
2. Photo Lab Management
3. Exit
```

Product page:

```
2. Photo Lab Management
3. Exit
1
Please select an operation:
1. Create a new product
2. View all products
3. Update a product's cost
4. Delete a product
5. Exit
2
Product ID: 1, Product Name: Passport Size, Cost: 100.0
Product ID: 2, Product Name: Stamp Size, Cost: 90.0
Product ID: 3, Product Name: Photo Frame, Cost: 400.0
Product ID: 4, Product Name: Wedding Photoshoot, Cost: 10000.0
Product ID: 5, Product Name: Lamination, Cost: 210.0
```

Order and Billing page:

```
Select the system to run:
1. Product Management
2. Photo Lab Management
3. Exit
2
Enter Customer ID:
1
1. Passport Size - Cost: 100.0
2. Stamp Size - Cost: 90.0
3. Photo Frame - Cost: 425.0
4. Wedding Photoshoot - Cost: 10000.0
5. Lamination - Cost: 210.0
Select product(s) by entering the corresponding number
1
3
5
0
Total Price: 735.0
Order has been saved successfully.
```

Customer Details:

```
Please select an operation:
1. Create a new product
2. View all products
3. Update a product's cost
4. Delete a product
5. View Customer details
6. View Orders
7. Exit
5
Customer details:
Customer ID: 1, Customer Name: John Doe, Mobile Number: 1234567890, Address: 123 Main St
Customer ID: 2, Customer Name: David Beckham, Mobile Number: 0987654321, Address: 456 Maple St
Customer ID: 3, Customer Name: Cristiano Waldo, Mobile Number: 8787654321, Address: 116 Kaiwal St
Customer ID: 4, Customer Name: Leo Wessi, Mobile Number: 9080123456, Address: 109 NTA Street
Customer ID: 5, Customer Name: Whyrat Kohli, Mobile Number: 9123456710, Address: 134 Wardo Colony
```

Order Details:

```
6
Order details:
Order ID: 1, Order Name: Wedding Photoshoot, Cost: 10000.0, Product ID: 4, Customer ID: 3
Order ID: 2, Order Name: Passport Size, Cost: 100.0, Product ID: 1, Customer ID: 3
Order ID: 3, Order Name: Passport Size, Cost: 100.0, Product ID: 1, Customer ID: 3
Order ID: 4, Order Name: Photo Frame, Cost: 400.0, Product ID: 3, Customer ID: 5
Order ID: 5, Order Name: Lamination, Cost: 210.0, Product ID: 5, Customer ID: 5
Order ID: 6, Order Name: Passport Size, Cost: 100.0, Product ID: 1, Customer ID: 4
Order ID: 7, Order Name: Passport Size, Cost: 100.0, Product ID: 1, Customer ID: 4
Order ID: 8, Order Name: Lamination, Cost: 210.0, Product ID: 5, Customer ID: 4
Order ID: 9, Order Name: Photo Frame, Cost: 400.0, Product ID: 3, Customer ID: 5
Order ID: 10, Order Name: Passport Size, Cost: 100.0, Product ID: 1, Customer ID: 3
Order ID: 11, Order Name: Stamp Size, Cost: 90.0, Product ID: 2, Customer ID: 3
Order ID: 12, Order Name: Stamp Size, Cost: 90.0, Product ID: 2, Customer ID: 3
Order ID: 13, Order Name: Lamination, Cost: 210.0, Product ID: 5, Customer ID: 3
```