# GYM MANAGEMENT SYSTEM

**CS23333 – Object Oriented Programming using Java Project Report**

*Submitted by*

# SRIMATHI B.S -231001209

**SRINITHYA A -231001212**

**SHREEKUMARAN S -231001195**

*Of*

# BACHELOR OF TECHNOLOGY

*In*

# INFORMATION TECHNOLOGY



**DEPARTMENT OF INFORMATION TECHNOLOGY**

# RAJALAKSHMI ENGINEERING COLLEGE NOVEMBER-2024

BONAFIDE CERTIFICATE

Certified that this project titled “GYM MANAGEMENT SYSTEM” is the bonafide work of “**SRIMATHI B.S(231001209), SRINITHYA A (231001212),**

**SHREEKUMARAN S (231001195)**”who carried out the project work under my supervision.

# SIGNATURE

**Dr.P.Valarmathie**

# HEAD OF THE DEPARTMENT

**SIGNATURE**

# Mr.K.E NARAYANAN

**COURSE INCHARGE**

Department of Information Technology Rajalakshmi Engineering College

Department of Information Technology Rajalakshmi Engineering College

This project is submitted for CS23333 – Object Oriented Programming using Java held on

# INTERNAL EXAMINAR EXTERNAL EXAMINAR

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | List of Figures | 4 |
|  | List of Tables | 5 |
| **1** | 1.1 Abstract | 6 |
|  | 1.1 Introduction | 6 |
|  | 1.3 Purpose | 6 |
|  | 1.4 Scope of Project | 7 |
|  | 1.5 Software Requirement  Specification | 7 |
| **2** | **System flow Diagrams** | 12 |
|  | 2.1 Use Case Diagram | 12 |
|  | 2.2 Entity-relationship Diagrams | 13 |
|  | 2.3 Data Flow Diagram | 13 |
| **3** | **Module Description** | 14 |
| **4** | 4.1 Design | 15 |
|  | 4.2 Database Design | 18 |
|  | 4.3 Code | 19 |
| **5** | **Conclusion** | 30 |
| **6** | **Reference** | 30 |

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Figure Numbers** | **Figure Captions** | **Pg.no** |
| 2.1 | Use Case Diagram | 12 |
| 2.2 | Entity Relation diagram | 13 |
| 2.3 | Date Flow Diagram | 13 |
| 4.1.1 | Login Page | 15 |
| 4.1.2 | Registration Page | 15 |
| 4.1.3 | Home Page | 16 |
| 4.1.4 | Cart Selection | 16 |
| 4.1.5 | Payment Page | 17 |
| 4.1.6 | Order Message | 17 |

**LIST OF Table**

|  |  |  |
| --- | --- | --- |
| **Figure Numbers** | **Figure Captions** | **Pg.no** |
| 4.2.1 | Login Table | 18 |

# Abstract

# The Gym Management System is a Java-based desktop application developed to manage gym activities efficiently, such as member registration, payments, and trainer assignments. It uses Java Swing for the user interface and JDBC to connect with a MySQL database. By applying object-oriented principles like encapsulation and modularity, the system ensures scalability and maintainability. Features like date input using jdatepicker demonstrate Java's extensibility, making this project an effective solution for automating gym operations. It exemplifies Java's practicality in building small-scale, database-driven applications.

# Introduction

# The Gym Management System is a Java-based application designed to automate gym operations, such as managing member data, tracking payments, and assigning trainers. Built with Java Swing for the user interface and JDBC for MySQL database integration, it allows efficient handling of gym-related information. The system follows object-oriented principles, ensuring modularity and scalability. Key features include a date selection component using the jdatepicker library. This solution simplifies manual tasks, improves data accuracy, and is suitable for small to medium-sized gyms.

# Purpose

# The primary goal of the Gym Management System is to create an efficient and user-friendly platform that simplifies gym operations for administrators. The objectives of the project include:

# Automating Gym Operations: Streamlining tasks like member registration, payment tracking, and trainer assignments to reduce administrative workload.

# Providing a User-Friendly Interface: Creating an intuitive interface with Java Swing for easy use by gym staff, regardless of technical expertise.

# Ensuring Data Accuracy: Using JDBC for secure database access, ensuring consistent and reliable member, payment, and trainer data.

# Improving Efficiency: Minimizing errors and time spent on manual tasks like payment tracking and scheduling.

# Providing Scalability: Designing the system with object-oriented principles for easy expansion and maintenance as the gym grows.

# Scope of the Project:

# The Gym Management System aims to automate and streamline various operations within a gym, providing a comprehensive solution for member management, payment tracking, and trainer assignments. The system includes features such as membership registration, payment tracking, scheduling, and trainer assignments, ensuring efficient data handling and reducing administrative workload. By utilizing Java Swing for the user interface and JDBC for secure database connectivity, the system ensures a smooth and user-friendly experience for gym staff. The database integration with MySQL ensures secure and consistent data management. The system is scalable, allowing for future expansion, such as adding features like online booking or enhanced reporting tools, to meet evolving gym needs.

# Software Requirement Specification:

# Introduction

# This section defines the hardware and software requirements for the Gym Management System, ensuring smooth operation, scalability, and efficient user interaction.

# Product Scope

# The system automates key gym operations such as member registration, payment tracking, and trainer assignments. Developed with Java for logic and MySQL for database management, it offers a seamless user experience for both gym staff and members. The system is designed to be scalable, supporting future features like online booking and advanced reporting.

# References and Acknowledgements

# Java Development Kit (JDK) for application development.

# MySQL documentation for database integration.

# Java Swing for the user interface.

# JDBC API for secure database connectivity.

## Overall Description

The Gym Management System is a software solution developed using Java and MySQL to automate and streamline gym operations. It manages tasks such as member registration, payment tracking, trainer assignments, and scheduling. The system integrates a secure MySQL database via JDBC for reliable data storage, and the Java Swing interface ensures ease of use for gym staff. Scalable and maintainable, the system is designed to handle the evolving needs of small to medium-sized gyms, with the potential for future enhancements like online booking and advanced reporting.

## Product Perspective

## The Gym Management System is a Java and MySQL-based solution that automates and streamlines gym operations. It allows administrators to manage member registrations, payments, trainer assignments, and class schedules efficiently. The system uses JDBC for secure database connectivity and features a Java Swing interface for ease of use. Scalable and cost-effective, it is designed for small to medium-sized gyms and can be expanded with future features like online booking or advanced reporting. This system can be easily integrated into existing gym management processes, offering a modern solution for operational efficiency.

## Product Functionality

## Member Registration: Allows members to register and update their profiles.

## Payment Tracking: Tracks member payments and dues in real-time.

## Trainer Assignment: Assigns trainers to members and manages schedules.

## Scheduling: Enables booking and managing gym classes or sessions.

## Database Integration: Uses MySQL for secure data storage and access.

## User Interface: Provides an intuitive Java Swing interface for easy navigation.

## Scalability: Supports future features like online booking and reporting.

## User and Characteristics

* **Qualification**: Users should have at least basic educational qualifications, such as matriculation, and be comfortable with English for understanding system instructions and managing data effectively.
* **Experience**: Familiarity with gym operations or membership management is beneficial, though not required. Administrators will benefit from prior experience in managing schedules, payments, and memberships.
* **Technical Experience**: Users are expected to have elementary knowledge of computers and should be comfortable with operating basic software tools (like navigating windows, entering data into forms, etc.) to interact with the system efficiently.

8

## Operating Environment

### Hardware Requirements

* Processor: Intel i3 or higher (or equivalent AMD processor)
* Operating System: Windows 8,10, 11
* Processor Speed: 2.0 GHz
* RAM: 4GB
* Hard Disk: 500GB

### Software Requirements

### System Requirements

### Software Requirements

### Database Requirements

### Development Tools

### Libraries and Frameworks

### Network Requirements

### Constraints

### System Access: Access is limited to authorized personnel only, ensuring that sensitive data is protected and that the system is used by designated staff members only.

### Data Volume: The system is designed to handle a moderate volume of data, such as member details, payments, and schedules. However, for very large datasets (e.g., large gym chains), additional optimizations may be required to maintain system performance.

### Internet Connectivity: The system requires a stable internet connection for updates, remote access, and future integration of online features, which could be a limitation in regions with unreliable internet service.

### User Knowledge: Users should have basic computer literacy and some familiarity with gym operations. Those with limited technical expertise might require some training to navigate the system effectively.

### Hardware Limitations: The system is optimized for computers with at least 4 GB RAM and 2 GB disk space. Performance may degrade on systems with lower specifications.

## User Interface

The Medical E-Commerce Store provides user-friendly, menu-driven interfaces for:

* **Register**: Allows new users to create an account by providing essential information like name, contact details, and membership preferences.
* **Login**: Enables existing users (administrators, trainers, or members) to log in using their credentials (username and password) for personalized access.
* **Member Dashboard**: Displays personalized details like membership status, upcoming sessions, payment history, and profile settings.
* **Trainer Assignment**: Admins can assign trainers to specific members and track their schedules.
* **Payment Tracking**: Admins can view and manage member payments, ensuring that dues are up to date.
* **Scheduling**: Allows users (administrators and members) to view and manage gym sessions, including booking and canceling classes or training sessions.
* **Order/Transaction History**: Members can view and track previous transactions, payments, and appointments, providing transparency and easy access to their gym-related activities.

## Hardware Interface

* Screen resolution of at least 640 x 480 or above.
* Compatible with any version of Windows 8, 10, 11.

## Software Interface

## Login and Registration Screens

## Admin Dashboard

## Member Dashboard

## Trainer Dashboard

## Scheduling Interface

## Payment Interface

## Functional Requirements

## User Registration and Authentication:

## Allow users (administrators, trainers, members) to register by providing details like name, email, password, and contact information.

## Provide secure login/logout functionality with encrypted password storage to ensure safe access.

## Gym Member Management:

## Enable administrators to view, update, and manage gym member profiles, including membership status, payment history, and session details.

## Trainer Assignment:

## Allow administrators to assign trainers to members and manage trainer schedules.

## Trainers should be able to view and update their assigned sessions and manage progress tracking.

## Session Scheduling:

## Allow members to book or cancel gym classes and personal training sessions through a scheduling interface.

## Admins can manage and update the class schedule.

## Payment Tracking:

## Enable tracking of member payments for membership and training services.

## Allow members to view their payment status and history.

## Admin Functions:

## Provide admins with tools to manage member, trainer, and session data.Allow for reports on payments, member activity, and session attendance.

## Non-functional Requirements

## Performance:

## The system should load within 2-3 seconds and handle a high volume of simultaneous users without delays or crashes.

## Scalability:

## The system must be able to accommodate increasing user data, transactions, and gym schedules without significant performance issues.

## Security:

## Ensure that all user data is securely stored and transmitted, using encryption methods like SSL for login and payment processes.

## Availability:

## The system should be available 99.9% of the time, with minimal downtime for scheduled maintenance.

## Usability:

## The system should offer a user-friendly interface that is intuitive for administrators, trainers, and members, ensuring ease of navigation and efficient task completion.

## Reliability:

## The system should be stable and consistently perform as expected with minimal downtime or errors.

## Compatibility:

## The system should be compatible with major browsers (Chrome, Firefox, Safari) and devices (desktops, tablets, smartphones) to ensure broad accessibility.

# SYSTEM FLOW DIAGRAMS

# New Gym Management Program with Attributes described on ER Diagram Template | Relationship diagram, Class diagram, Gym management software

## Figure 2.1 Use Case Diagrams

A diagram of a company

Description automatically generated

**Figure 2.2 Entity Relationship Diagram**

A diagram of a gym management system

Description automatically generated

**Figure 2.3 Data-flow diagram**

# MODULE DESCRIPTION

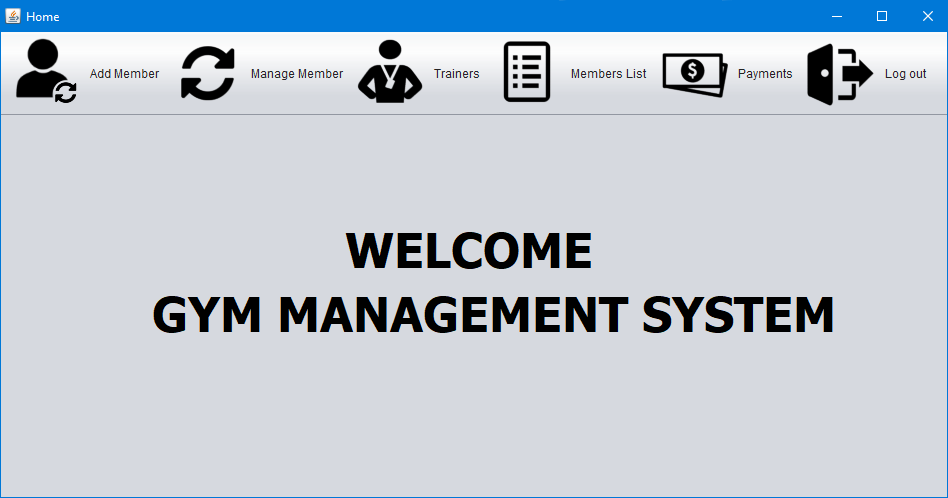
* **User Management**:  
  Handles user registration, login, and profile management for both gym members and administrators. Ensures secure authentication and role-based access.
* **Member Management**:  
  Allows admins to manage member profiles, including personal details, membership status, payment history, and session bookings.
* **Session Scheduling and Trainer Assignment**:  
  Admins can assign trainers to members, manage schedules, and track session attendance. Members can book and view sessions.
* **Payment and Checkout**:  
  Integrates secure payment gateways for processing transactions, including membership and session fees, using various payment methods.
* **Admin Dashboard**:  
  Provides admins with an interface to manage sales, track inventory, monitor member activity, and generate reports.
* **Inventory and Equipment Management**:  
  Tracks gym equipment and resources, allowing admins to manage stock levels and ensure necessary supplies are available for members.

# 

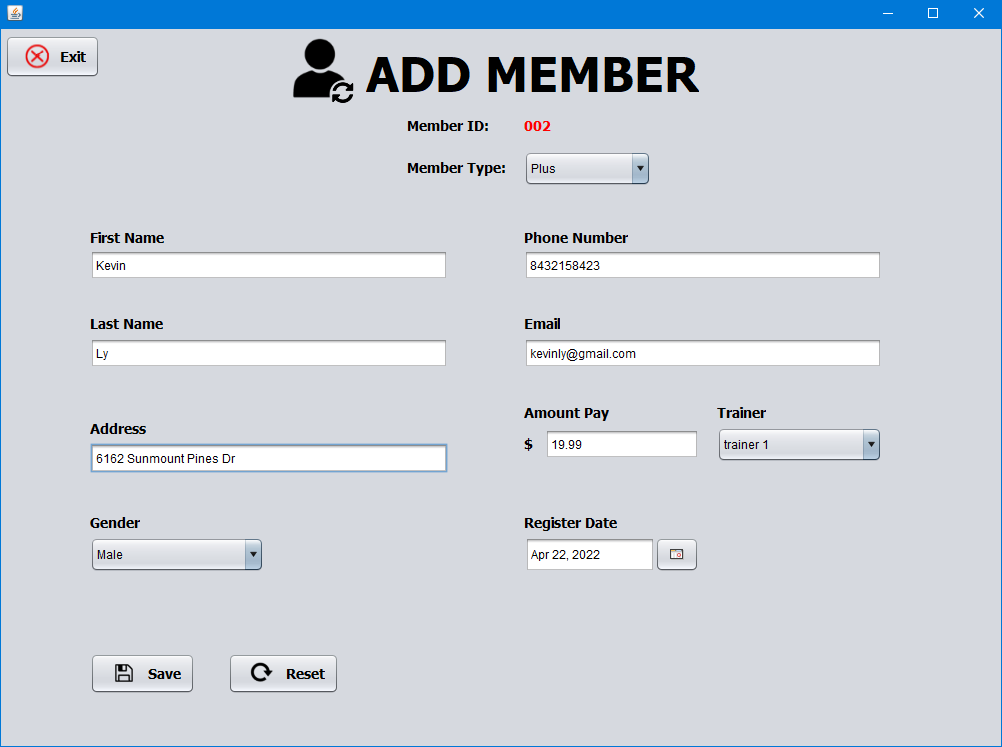
# DESIGN:

# A screenshot of a login screen Description automatically generated

## Figure 4.1.1 Login Page



## Figure 4.1.3 Home page



**Figure 4.1.4 Add member page**

A screenshot of a computer

Description automatically generated

## Figure 4.1.5 Payment page

# Database Design

The database design for the **Gym Management System** includes several key tables that store and manage data across different modules:

* **Users Table**: Stores user details for members, trainers, and administrators, including personal information, roles, and login credentials.
* **Sessions Table**: Manages details about gym sessions, such as time, trainer assignments, and member bookings.
* **Payments Table**: Tracks member payments for memberships, sessions, and other services, ensuring financial data is accurately stored.
* **Inventory Table**: Tracks gym equipment and resources, including stock levels and updates, to ensure availability for members.
  1. **IMPLEMENTATIONS (CODE)**

***/\****

***\* To change this license header, choose License Headers in Project Properties.***

***\* To change this template file, choose Tools | Templates***

***\* and open the template in the editor.***

***\*/***

***package com.mycompany.gymmanagementsystem;***

***import database.ConnectionProvider;***

***import java.awt.Color;***

***import java.sql.Connection;***

***import java.sql.ResultSet;***

***import java.sql.Statement;***

***import javax.swing.ImageIcon;***

***import javax.swing.JLabel;***

***import javax.swing.JOptionPane;***

***/\*\****

***\****

***\* @author quach***

***\*/***

***public class benefitsPanel extends javax.swing.JPanel {***

***/\*\****

***\* Creates new form NewJPanel***

***\*/***

***public benefitsPanel() {***

***initComponents();***

***basic1.setSelected(false);***

***basic2.setSelected(false);***

***plus1.setSelected(false);***

***plus2.setSelected(false);***

***plus3.setSelected(false);***

***premium1.setSelected(false);***

***premium2.setSelected(false);***

***premium3.setSelected(false);***

***premium4.setSelected(false);***

***}***

***/\*\****

***\* This method is called from within the constructor to initialize the form.***

***\* WARNING: Do NOT modify this code. The content of this method is always***

***\* regenerated by the Form Editor.***

***\*/***

***@SuppressWarnings("unchecked")***

***// <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents***

***private void initComponents() {***

***jLabel1 = new javax.swing.JLabel();***

***plus1 = new javax.swing.JCheckBox();***

***basic1 = new javax.swing.JCheckBox();***

***basic2 = new javax.swing.JCheckBox();***

***plus2 = new javax.swing.JCheckBox();***

***plus3 = new javax.swing.JCheckBox();***

***premium1 = new javax.swing.JCheckBox();***

***premium2 = new javax.swing.JCheckBox();***

***premium3 = new javax.swing.JCheckBox();***

***premium4 = new javax.swing.JCheckBox();***

***jComboBox1 = new javax.swing.JComboBox<>();***

***setPreferredSize(new java.awt.Dimension(451, 571));***

***jLabel1.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N***

***jLabel1.setIcon(new javax.swing.ImageIcon(getClass().getResource("/icons/benefiticon.png"))); // NOI18N***

***jLabel1.setText("BENEFITS");***

***plus1.setBackground(new java.awt.Color(204, 204, 204));***

***plus1.setText("ALL GROUP EXERCISE CLASSES");***

***basic1.setBackground(new java.awt.Color(204, 204, 204));***

***basic1.setText("USE OF ALL STRENGTH EQUIPMENT");***

***basic2.setBackground(new java.awt.Color(204, 204, 204));***

***basic2.setText("USE OF ALL CARDIO EQUIPMENT");***

***plus2.setBackground(new java.awt.Color(204, 204, 204));***

***plus2.setText("1 PERSONAL TRAINING SESSION");***

***plus3.setBackground(new java.awt.Color(204, 204, 204));***

***plus3.setText("1 GROUP TRAINING SESSION");***

***premium1.setBackground(new java.awt.Color(204, 204, 204));***

***premium1.setText("USE OF BASKETBALL COURTS");***

***premium2.setBackground(new java.awt.Color(204, 204, 204));***

***premium2.setText("USE OF RACQUETBALL COURTS");***

***premium3.setBackground(new java.awt.Color(204, 204, 204));***

***premium3.setText("UNLIMITED STUDIO CYCLING");***

***premium4.setBackground(new java.awt.Color(204, 204, 204));***

***premium4.setText("UP TO TWO GUESTS PER VISIT");***

***jComboBox1.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] "Basic", "Plus", "Premium" }));***

***jComboBox1.addItemListener(new java.awt.event.ItemListener() {***

***public void itemStateChanged(java.awt.event.ItemEvent evt) {***

***jComboBox1ItemStateChanged(evt);***

***}***

***});***

***javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);***

***this.setLayout(layout);***

***layout.setHorizontalGroup(***

***layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)***

***.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()***

***.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)***

***.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)***

***.addComponent(basic1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)***

***.addComponent(plus1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)***

***.addComponent(basic2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)***

***.addComponent(plus2, javax.swing.GroupLayout.DEFAULT\_SIZE,***

***.addComponent(jComboBox1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)***

***.addGap(29, 29, 29)***

***.addComponent(basic1)***

***.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)***

***.addComponent(basic2)***

***.addGap(18, 18, 18)***

***.addComponent(plus1)***

***.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)***

***.addComponent(plus2)***

***.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)***

***.addComponent(plus3)***

***.addGap(18, 18, 18)***

***.addComponent(premium1)***

***.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)***

***.addComponent(premium2)***

***.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)***

***.addComponent(premium3)***

***.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)***

***.addComponent(premium4)***

***.addContainerGap(151, Short.MAX\_VALUE))***

***);***

***}// </editor-fold>//GEN-END:initComponents***

# CONCLUSION

The Gym Management System is an efficient and comprehensive solution designed to automate and streamline the day-to-day operations of a gym. It simplifies tasks such as user management, session scheduling, trainer assignments, payment tracking, and inventory management. By leveraging Java and MySQL, the system ensures data integrity, secure transactions, and easy scalability to accommodate future growth.

With its user-friendly interface and powerful backend, this system improves operational efficiency, reduces administrative workload, and enhances the overall gym experience for both members and staff. Additionally, its flexible design allows for easy expansion and integration of new features, making it a reliable solution for managing small to medium-sized gyms. The system not only meets the current needs of gym operations but also provides a foundation for future advancements in the fitness industry.

# REFERENCES

# Books:

# Head First Java by Kathy Sierra and Bert Bates – Beginner-friendly Java guide.

# Java: The Complete Reference by Herbert Schildt – Comprehensive Java reference.

# Database System Concepts by Abraham Silberschatz et al. – Covers database design.

# User Interface Design and Evaluation by Debbie Stone et al. – Focuses on UI/UX principles.

# Websites:

# Oracle Java Tutorials: <https://docs.oracle.com/javase/tutorial/>

# SQL Basics (W3Schools): <https://www.w3schools.com/sql/>

# Java Swing (Javatpoint): <https://www.javatpoint.com/java-swing>

# Software Development Life Cycle (GeeksforGeeks): <https://www.geeksforgeeks.org/software-development-life-cycle-sdlc/>