SOFTWARE DEVELOPMENT LIFE CYCLE FOR CASE STUDY

Student Management System - SDLC Plan

1. **Planning Phase (Day 1-2)**

🚀 Requirements Gathering:

- Analyze scope of work document
- Identify stakeholders (Admin users, Students)
- Define system requirements
- Document users stories

🚀 Project Planning:

- Create project timeline
- Allocate resources
- Define milestones
- Risk assessment
- Technology stack selection

🚀 Deliverables:

- Software Requirements Specification (SRS)
- Project Plan
- Risk Register
- Technology Stack Documentation

2. **Analysis Phase (Day 2-3)**

🚀 System Analysis

- Use case analysis
- Data flow diagrams
- Entity relationship diagrams
- Security requirements analysis

🚀 Design Planning

- UI/UX wireframes
- Database schema
- System architecture
- API specifications

🚀 Deliverables:

- System Design Document
- UI/UX Mockups
- Database Schema
- API Documentation

3. **Design Phase (Day 3-4)**

🚀 Technical Design

- Class diagrams
- Sequence diagrams
- Components diagrams
- Security design patterns

🚀 UI/UX Design

- Login interface design
- Admin dashboard design
- Student view design
- QR code integration design (tentative)

🚀 Deliverables:

- Detailed Technical Specifications
- UI/UX Design Documents
- Security Design Document

- Test Strategy Document

4. **Development Phase (Day 4-5)**

🚀 Sprint 1: Core Framework
- Project setup with Maven
- Basic class implementation
- Database connectivity
- Unit test framework setup

🚀 Sprint 2: Authentication

- QR code generation
- PIN verification system
- Security implementation
- User management implementation

🚀 Sprint 3: GUI Development

- Login interface
- Admin dashboard
- Student view
- QR code scanning interface

🚀 Sprint 4: Features & Integration

- CRUD operations
- Data persistence
- Search functionality
- Integration testing

🚀 Deliverables:

- Source Code
- Unit Tests
- Integration Tests
- Build Script

5. **Testing Phase (Day 6-7)**

   🚀 Unit Testing

   - Model classes testing
   - Service layer testing
   - UI component testing

   🚀 Integration Testing

   - End-to-end testing
   - Security testing
   - Performance testing
   - User acceptance testing

   🚀 Deliverables:

   - Test Cases
   - Test Results
   - Bug Reports
   - Performance Reports

6. **Deployment Phase (Day 8-9)**

   🚀 Preparation

   - Documentation finalization
   - User manual creation
   - System deployment plan
   - Backup and recovery plan

   🚀 Deployment

   - System installation
   - Database setup
   - User training
   - Go-live support

🚀 Deliverables:

- Deployment Guide
- User Manual
- Training Materials
- Support Documentation

7. **Maintenance Phase (Ongoing)**

🚀 Support

- Bug fixing
- Performance monitoring
- Security updates
- User support

🚀 Enhancement

- Features requests
- System optimization
- Regular updates
- Security patches

🚀 Quality Control

- Code reviews
- Testing procedures
- Documentation review
- Security audits

🚀 Standards Compliance

- Coding standards
- Documentation standards
- Security standards
- UI/UX standards

🚀 Timeline Management

- Sprint planning
- Daily standups
- Sprint reviews
- Milestone tracking

🚀 Resource Management

- Risk management

Tools & Technologies

🚀Development Tools

- IntelliJ IDEA Ultimate
- IntelliJ IDEA Community Edition (Free)
- Maven
- Git for version control
- Java 17 or latest 21 to 22

🚀 Testing Tools

- JUnit for unit testing
- Built-in unit test in IntelliJ IDEA Ultimate

🚀Project Management Tools

- Google docs for task tracking
- GitHub for documentation
- GitHub for code repository
- Microsoft Teams and Google Meet for communication

This SDLC plan follows an Agile methodology while incorporating traditional software development phases. The timeline is estimated for a small team (3-4 developers) and can be adjusted based on actual team size and requirements.

GitHub Link: https://github.com/flexycode/CCOBJPGL_FINAL_PROJECT