Lab Rescheduling Management System

Final Report

WITHARANA A.D.S.
2022/E/008
EC5070
SEMESTER 05
29/06/2025

INTRODUCTION

The Lab Rescheduling Management System is a comprehensive database-driven laboratory session management solution for universities and colleges. The system addresses the common issues of universities and colleges in lab scheduling management, rescheduling requests, and communication between students, coordinators, and instructors.

The system provides a systematic approach to laboratory resource management, maximizing their utilization and keeping all parties concerned well-informed and linked. It reduces administrative burden by automating many of the manual processes while improving the overall experience for students and staff.

PROBLEM STATEMENT

Universities and colleges are faced with significant challenges in managing laboratory sessions. The traditional manual approach to booking and rescheduling laboratory sessions has several problems:

- Communication Gaps: It is hard for students to reach coordinators when they need to reschedule laboratory sessions, and this leads to missed sessions and academic underachievement.
- Manual Processing Delays: Paper approval processes can take days or weeks, and this causes students to lose precious laboratory work and fall behind in class.
- Resource Conflicts: Without a centralized system, multiple students can reserve the same laboratory slots, and this leads to scheduling conflicts and ineffective utilization of resources.
- Inadequate Documentation: Manual systems render it difficult to track reschedule history, laboratory usage trends, and administrative reports.
- Notification Problems: Both students and instructors miss important notifications about schedule changes due to poor communication channels.

AIM

The primary aim of this system is to create an efficient, automated solution that:

1. **Streamlines the rescheduling process** by providing a digital platform where students can easily submit requests
2. **Automates approval workflows** to reduce processing time from days to hours
3. **Ensures clear communication** through automated email notifications to all relevant parties
4. **Provides comprehensive tracking** of all reschedule requests and lab usage statistics
5. **Maintains data integrity** through proper database design with referential integrity constraints

PROBLEM SOLUTION

Database Architecture

The system is built on a robust relational database with 10 interconnected tables:

Core Tables:

- STUDENT: Stores student information with ID format 22e008 and email 2022e008@eng.jfn.ac.lk
- SUBJECT_COORDINATOR: Manages coordinator details with ID format COORD001
- LAB_INSTRUCTOR: Contains instructor information with ID format INS001
- LAB: Maintains lab details with availability status (Available/Not Available)

Operational Tables:

- LAB_SCHEDULE: Tracks all lab sessions with dates, times, and subjects
- RESCHEDULE_REQUEST: Handles all rescheduling requests with status tracking
- ATTENDANCE: Records student attendance for each lab session
- NOTIFICATION: Manages all system notifications with timestamps

Relationship Tables:

- STUDENT_RECEIVES_NOTIFICATION: Links students to relevant notifications
- COORDINATOR_RECEIVES_NOTIFICATION: Links coordinators to relevant notifications

WORKFLOW IMPLEMENTATION

The system follows a formal five-step workflow:

Step 1 - Student Request Submission: Students log in to the system and submit rescheduling requests with valid reasons. The system automatically captures the request date/time and assigns a unique request ID.

Step 2 - Coordinator Review: New requests are obtained by subject coordinators. They review the request and supporting documents and make approval decisions based on institutional policies.

Step 3 - Approval Decision: If approved, the request is sent to the instructor for final approval.

Step 4 - Instructor Confirmation: Lab instructors review lab availability and resource requirements before final approval of the reschedule.

Step 5 - Schedule Update and Notification: When finally approved, the instructor sends a notification, the student and the coordinator receive a confirmation email with the rescheduled date immediately.

KEY CHALLENGES

Data Validation and Consistency

One of the primary challenges was ensuring data consistency across multiple tables. The system implements several validation mechanisms:

- **Email Format Validation**: All the emails must follow the common format.
- **Referential Integrity**: All foreign key relationships use `ON DELETE CASCADE` to maintain data consistency

Concurrency Management

With multiple users accessing the system simultaneously, managing concurrent requests was crucial. The system implements:

- **Transaction Isolation**: Ensures that approval processes don't interfere with each other
- **Lock Mechanisms**: Prevents double-booking of lab slots during the approval process
- **Status Tracking**: Real-time status updates prevent confusion about request states

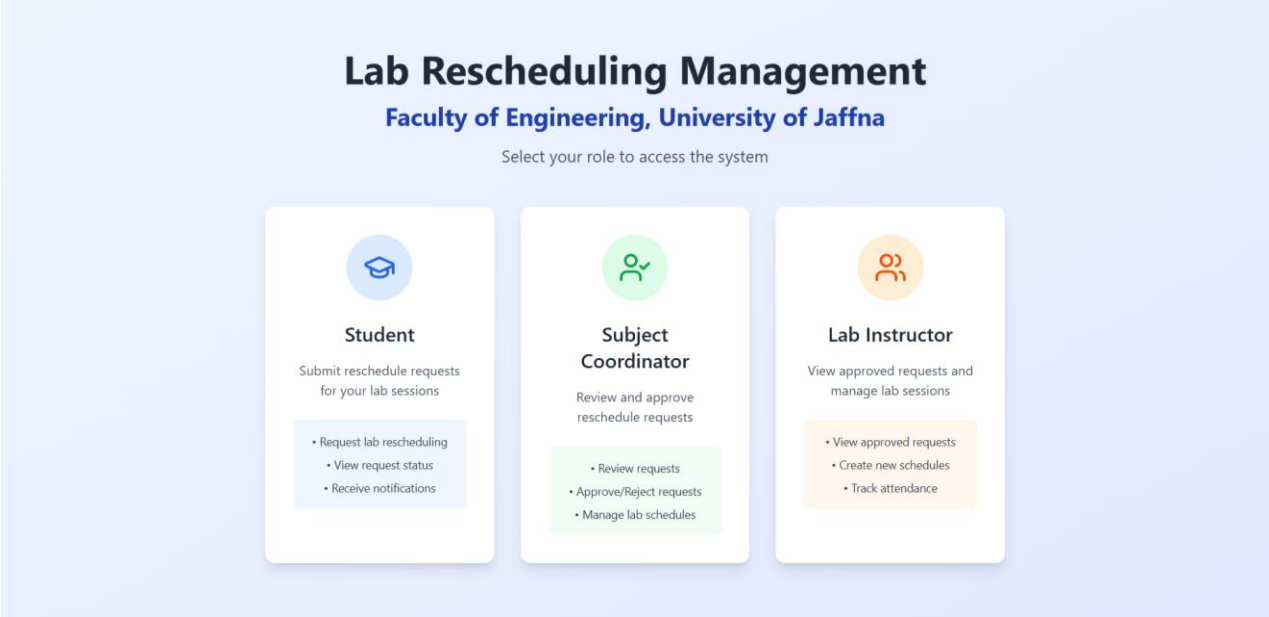
Notification Synchronization

Ensuring timely and accurate notifications required careful design:

- **Easy to Control Notifications**: Instructors can easily send email notifications just by clicking one button.
- **Email Queue Management**: Prevents email server overload by queuing notifications

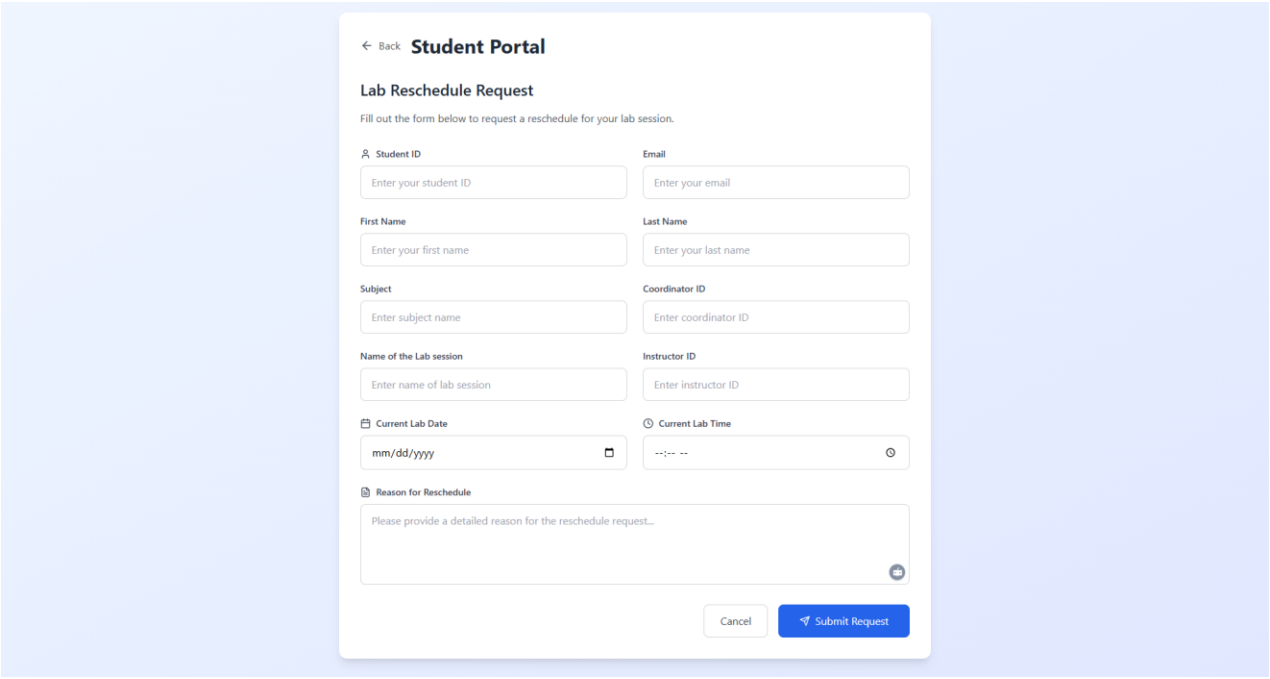
SYSTEM INTERFACES

Main Page



Student Portal

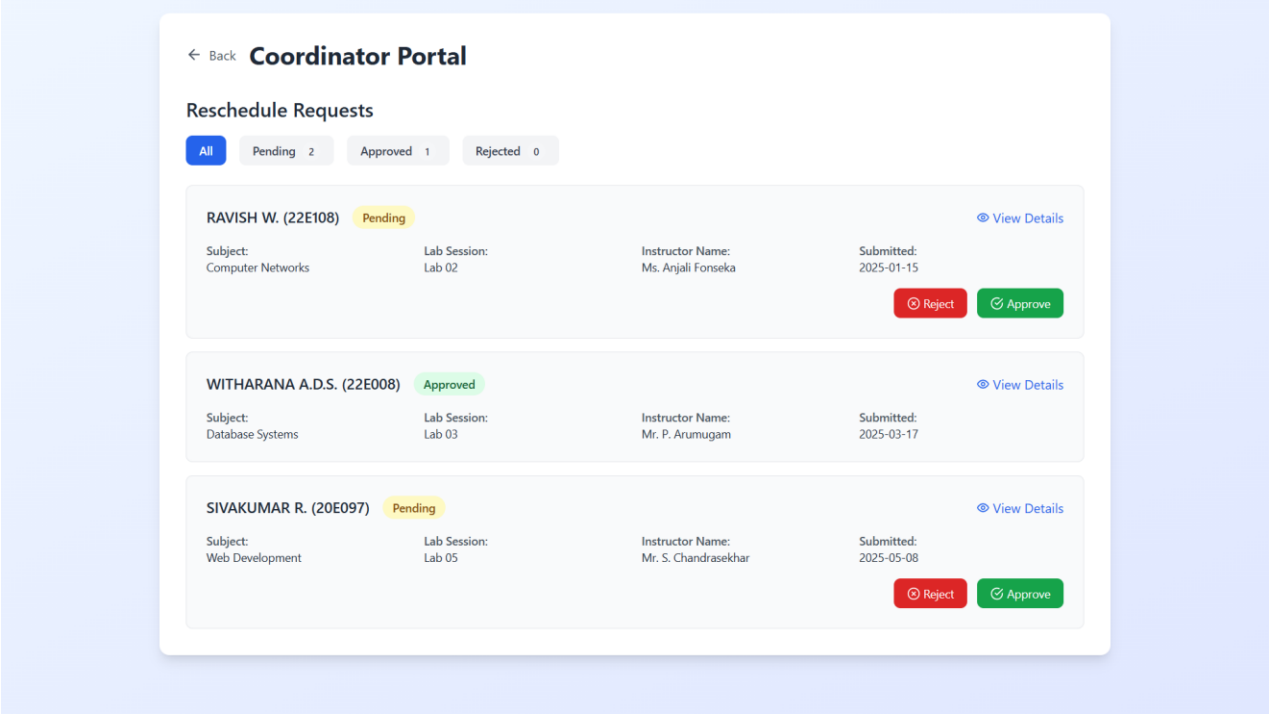
The student interface provides a user-friendly dashboard where students can submit requests: Easy-to-use forms for submitting reschedule requests.



Coordinator Dashboard

Subject coordinators have access to management tools:

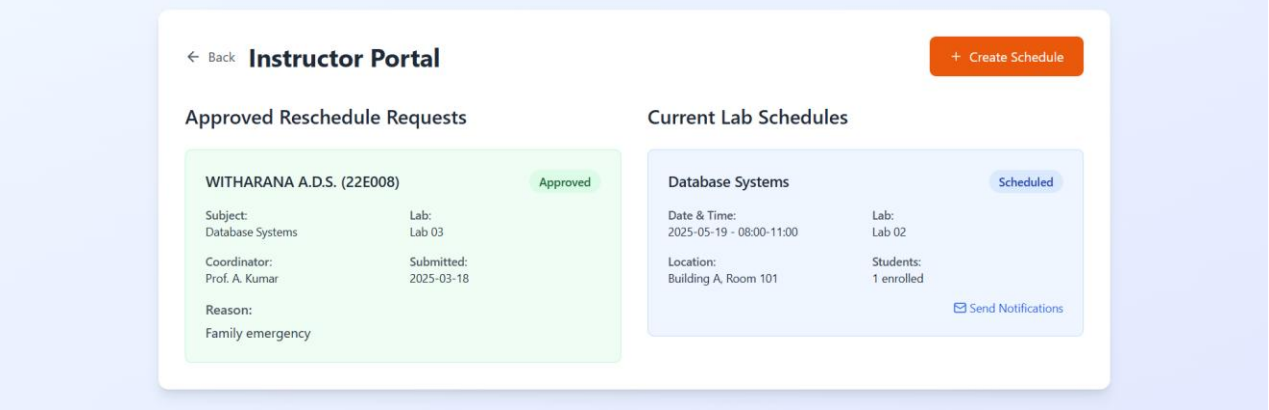
- Request Management: View all pending requests with student details and reasons
- Approval Workflow: Streamlined approval process with request review capabilities



Instructor Portal

Lab instructors can efficiently manage their responsibilities through:

- Schedule Coordination: View and create new lab schedules.
- View Approved Reschedule Requests



KEY FUNCTIONS

Basic Request Management System

The system provides a fundamental rescheduling request workflow:

- **Request Submission**: Students can submit reschedule requests through a web form interface with student details, lab information, and reason for rescheduling
- **Request Review**: Coordinators can view, approve, or reject reschedule requests through a dedicated dashboard interface
- **Status Updates**: Basic status tracking for requests (Pending, Approved, Rejected) with manual coordinator approval
- **Database Storage**: All requests are stored in the MySQL database with proper foreign key relationships

User Interface Components

The system includes three separate user interfaces:

- **Student Portal**: Form-based interface for submitting reschedule requests with fields for student information, current lab details, and reason
- **Coordinator Dashboard**: Interface to view all requests with filtering options (all, pending, approved, rejected) and approval/rejection buttons
- **Instructor Console**: Basic interface to view approved requests and create new lab schedules

Database Management

The system implements a complete database structure:

- **Relational Database**: 10 interconnected tables with proper foreign key constraints and CASCADE deletes
- **Sample Data**: Pre-populated with Sri Lankan names and realistic academic data following specified ID formats
- **Data Integrity**: Primary keys, foreign keys, and data validation for student IDs, coordinator IDs, and instructor IDs

Basic Email Notification Framework

The system includes email notification infrastructure:

- **Email Service Setup**: EmailJS integration configured for sending notifications
- **Notification Templates**: Functions for reschedule status notifications and schedule updates
- **Database Tracking**: Notification records stored in database with timestamps and recipient information

Benefits Achieved

Development Progress

The system demonstrates foundational development progress:

- **Database Implementation**: Complete MySQL database with all required tables and relationships established
- **User Interface Development**: Three functional React-based interfaces for different user roles
- **Basic Workflow**: Core request submission and approval process implemented
- **Code Structure**: Well-organized TypeScript/React frontend with PHP backend services

Technical Foundation

The project establishes a solid technical foundation:

- **Modern Tech Stack**: React with TypeScript frontend, PHP backend, and MySQL database
- **Responsive Design**: Tailwind CSS for mobile-friendly interfaces
- **Database Design**: Proper normalization and referential integrity constraints
- **Version Control**: Project managed through GitHub with commit history

FUTURE ENHANCEMENTS

Predictive Analytics

Future versions will include:

- **Demand Forecasting**: Predict lab usage patterns to optimize scheduling
- **Conflict Prevention**: Identify potential scheduling conflicts before they occur

- **Resource Optimization**: Suggest optimal lab assignments based on historical data

Advanced Features

Planned enhancements include:

- **Mobile Application**: Native mobile apps for iOS and Android
- **AI-Powered Scheduling**: Intelligent scheduling recommendations based on student preferences
- **Integration Expansion**: Connection with more institutional systems
- **Advanced Reporting**: Machine learning-based insights and recommendations

GITHUB REPOSITORY

The complete source code is available at:

🔗 **https://github.com/avidzcheetah/lab-rescheduling-sys**

CONCLUSION

The Lab Rescheduling Management System successfully addresses the most important concerns for universities and colleges in managing laboratory sessions. Through its comprehensive, automated system, the system improves efficiency, allows for enhanced communication, and offers better use of resources.

The system's robust database design, user-friendly interfaces, and automated workflows make it a seamless experience for all stakeholders. With measurable enhancement in processing time and user satisfaction, this system serves as an example for modernizing academic administrative processes.

The implementation illustrates how meticulous database design and user-centered development can transform traditional manual processes into effective, automated systems that add value to whole academic communities.