

Kathmandu University
Department of Computer Science and Engineering
Dhulikhel, Kavre



A Project Proposal
on
“Kakshya”

[Code No: ENGG 102]

(For partial fulfillment of I Year/ II Semester in Computer Engineering)

Submitted by

Bibesh Basnet (Roll No. 06)
Achyut Gautam (Roll No. 15)
Aman Karn (Roll No. 22)
Sandesh Khadka (Roll No. 24)
Prajwal Yadav (Roll No. 56)

Submitted to

Suman Shrestha
Department of Computer Science and Engineering

Submission Date: 11th November, 2024

Abstract

This project proposes the development of **Kakhsya**, a Classroom Management Application, using C++, MySQL, and the Qt framework to enhance classroom scheduling and resource allocation within educational institutions. **Kakhsya** aims to streamline the scheduling process by identifying and displaying vacant classrooms in real-time, allowing users to make efficient, data-driven decisions regarding classroom usage.

The system is designed to provide distinct role-based access for Coordinators, Teachers, and Class Representatives (CRs), each with tailored permissions that ensure secure and efficient functionality. Coordinators will have comprehensive control over room assignments, scheduling, and conflict resolution. Teachers can view and request room allocations specific to their classes, while CRs can access classroom schedules to stay updated on room changes and communicate this information to their peers.

By minimizing scheduling conflicts and providing timely updates on room availability, **Kakhsya** seeks to improve classroom utilization, reduce administrative overhead, and facilitate smoother campus operations. Ultimately, the system will contribute to a more organized campus environment, fostering better communication and ensuring optimal use of classroom resources.

Keywords: *Kakhsya, Classroom management, Campus scheduling, Object-Oriented Programming, Procedural Programming, C++, MySQL, Qt.*

Table of Contents

Abstract	i
List of Figures	iv
Acronyms/Abbreviations	v
Chapter 1: Introduction	1
1.1 Background	1
1.2 Objectives	2
1.3 Motivation and Significance	2
1.4 Expected Outcomes	2
Chapter 2: Related Works/Existing Works.....	4
2.1 Roomzilla.....	4
2.2 Ad Astra	5
2.3 25Live (by collegeNET)	5
2.4 Addressing Challenges through our project.....	6
Chapter 3: Procedure and Methods.....	8
3.1 Procedure and Methods.....	8
3.1.1 Research and Requirements Analysis	8
3.1.2 Requirement Definition	8
3.1.3 Programming Language Selection	8
3.1.4 User Interface Design	8
3.1.5 Database Setup.....	9
3.1.6 Implementation of Scheduling and Conflict Resolution Logic	9
3.1.7 Role-Based Access Control	9
3.1.8 Notification System	10

3.1.9 Testing and Debugging	10
3.1.10 Documentation	10
3.2 System Design	10
3.2.1 Use Case Diagram.....	11
3.2.2 Flow Chart	11
Chapter 4: System Requirement Specifications.....	14
4.1 Software Specifications	14
4.1.1 Front-end Tools:.....	14
4.1.2 Back-end Tools:	15
4.2 Hardware Specifications	15
Chapter 5: Project Planning and Scheduling	16
APPENDIX.....	17
References.....	18

List of Figures

Figure 1 Roomzilla.....	4
Figure 2 Ad Astra.....	5
Figure 3 25Live.....	6
Figure 4 User Case Diagram.....	11
Figure 5 Flow Chart.....	12
Figure 6 Gantt Chart.....	16
Figure 7 Logo.....	18

Acronyms/Abbreviations

iOS	iPhone Operating System
AI	Artificial Intelligence
macOS	Macintosh Operating System
MySQL	My Structured Query Language
RDBMS	Relational Database Management System
GUI	Graphical User Interface
HDD	Hard Disk Drive
QML	Qt Modelling Language
CR	Class Representative

Chapter 1: Introduction

Kakhsya addresses a common issue in educational institutions: classroom scheduling and real-time availability updates. By providing distinct access roles for coordinators, teachers, and CRs, the application will help optimize classroom use, reduce scheduling conflicts, and improve the organization of campus resources.

1.1 Background

Educational institutions often face challenges in effectively managing classroom schedules and ensuring the availability of rooms for various purposes. With growing student populations and limited classroom resources, the need for an efficient system to allocate and track classroom usage has become essential. Current manual or semi-automated methods can lead to overlapping bookings, last-minute schedule adjustments, and confusion among students and faculty. Recent advancements in software development, especially with frameworks like Qt for cross-platform applications, have opened doors to building more responsive, user-friendly management systems tailored to the specific needs of educational institutions.

Despite the increasing adoption of digital tools for campus management, many institutions still rely on basic scheduling software or spreadsheet-based solutions. While these solutions may handle basic scheduling tasks, they often lack real-time updates, role-based access control, and robust data handling to ensure optimal classroom usage. Some existing systems are costly, complex, and require significant customization to meet each institution's needs, which can hinder their effective use. Addressing these limitations, **Kakhsya** aims to develop a streamlined, accessible, and role-based Classroom Management Application that offers real-time vacancy updates, reduces scheduling conflicts, and enhances resource management across the campus.

1.2 Objectives

The objectives of **Kakhsya** are as follows:

- To provide real-time updates on classroom availability, minimizing scheduling conflicts.
- To streamline classroom booking and allocation with distinct role-based access for Coordinators, Teachers, and Class Representatives.
- To improve campus resource management through better classroom utilization.
- To create an intuitive interface for users, enhancing usability and reducing administrative overhead.

1.3 Motivation and Significance

The motivation for this project arises from the observed inefficiencies in existing classroom management practices. Many institutions struggle with outdated or overly simplistic scheduling tools that lack dynamic features, leading to frequent conflicts and miscommunication. By choosing this topic, we aim to develop a solution that addresses these limitations by providing a real-time, user-friendly, and role-based management system.

Kakhsya differentiates itself from existing solutions by focusing on the distinct needs of Coordinators, Teachers, and CRs, each with specific access and usage rights. By automating vacancy updates and room allocations, this project will offer a solution that is more responsive, efficient, and adaptable to changing schedules.

The application will include several key features, such as real-time availability tracking, a responsive interface designed with Qt, and secure role-based access. These features contribute to the overall robustness of the application and ensure that it is both user-friendly and highly effective.

1.4 Expected Outcomes

The expected outcomes of **Kakhsya** include:

- A real-time, role-based platform that simplifies classroom scheduling and allocation.
- A reduction in scheduling conflicts and improved efficiency in room utilization.
- Enhanced campus resource management, leading to fewer instances of underused or overcrowded classrooms.
- Improved communication and organization across the institution, benefiting students, teachers, and administrators alike.

Chapter 2: Related Works/Existing Works

This section will review similar applications, including university scheduling systems and resource management tools, discussing their strengths and areas where this project seeks to improve.

2.1 Roomzilla

Roomzilla is a room booking and scheduling app designed to simplify room and resource booking, making it easy for students and faculties to find available spaces for meetings, events, or study sessions. The platform is optimized for mobile devices, enabling students and faculty to check room availability and make bookings on the go. It offers real-time room booking with instant updates, making it easy to find and reserve available spaces. Being cloud-based, this app can scale to accommodate multiple locations or campuses with ease.

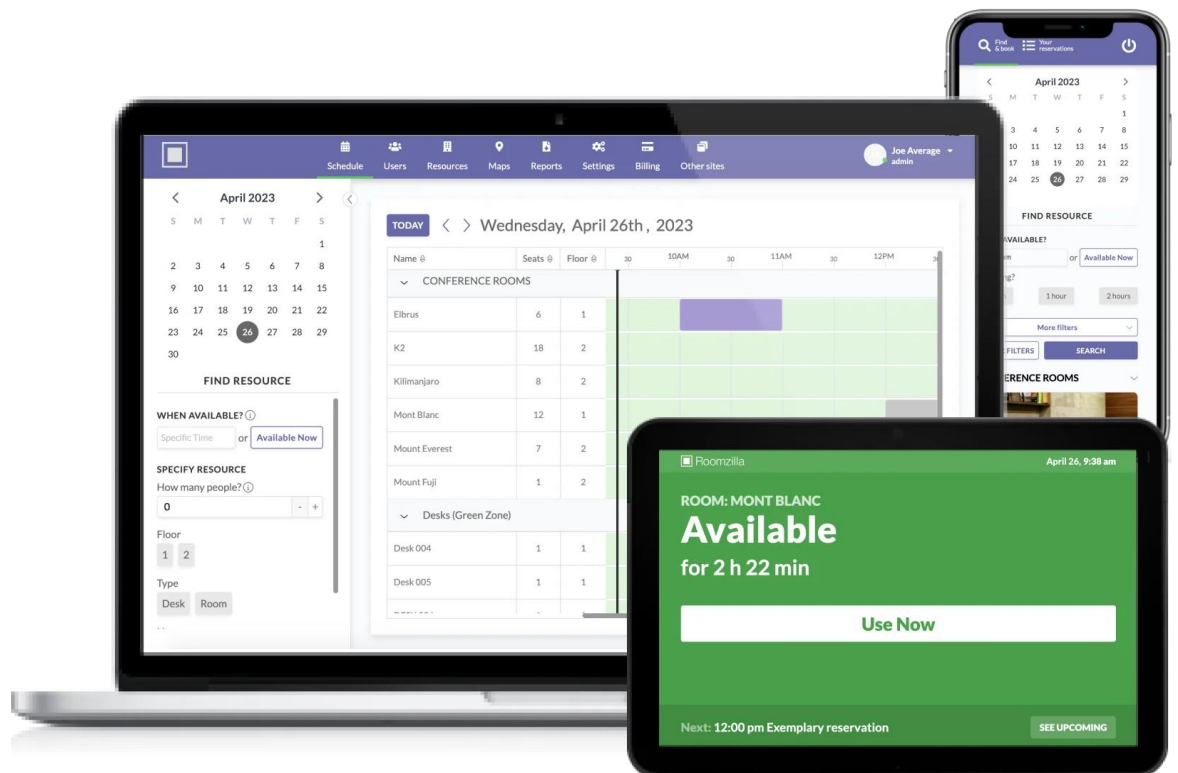


Figure 1: Roomzilla

2.2 Ad Astra

Ad Astra is a sophisticated scheduling and resource management software primarily used in higher education institutions for classroom scheduling, resource allocation, and academic planning. It is designed to optimize the utilization of physical spaces, faculty time, and other resources within an academic environment, helping institutions make data-driven decisions and improve operational efficiency.

It provides detailed reports on room usage, scheduling conflicts, and overall utilization metrics, helping institutions identify underused or overbooked spaces. Being cloud-based, it offers scalability, remote access, and automatic updates.

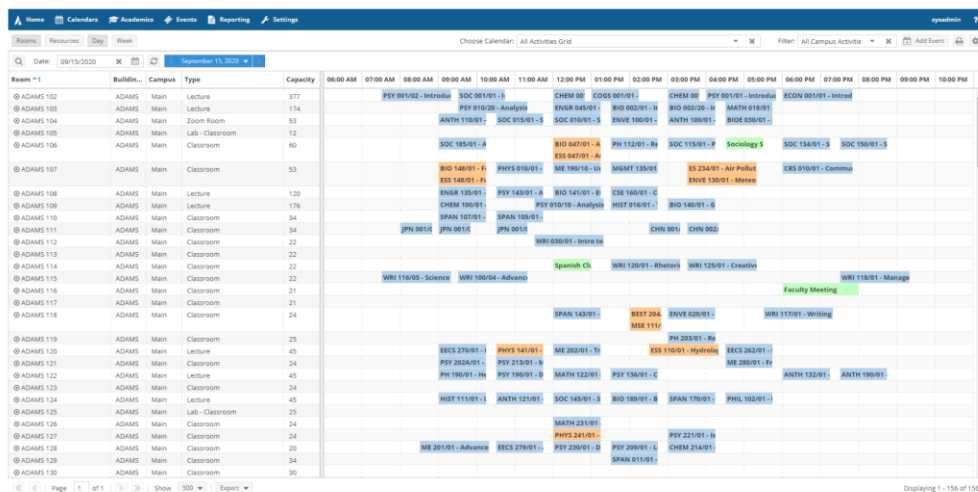


Figure 2: Ad Astra

2.3 25Live (by collegeNET)

25Live is the one education scheduling software system that simplifies every scheduling task, from quick room searches and requests to the most complex classroom scheduling and calendar publishing projects. The system is flexible, allowing institutions to customize workflows for scheduling rooms, events, and

resources based on their unique needs. This can include setting up policies for different types of events (academic, student activities, external bookings, etc.), which helps prevent misuse of spaces.

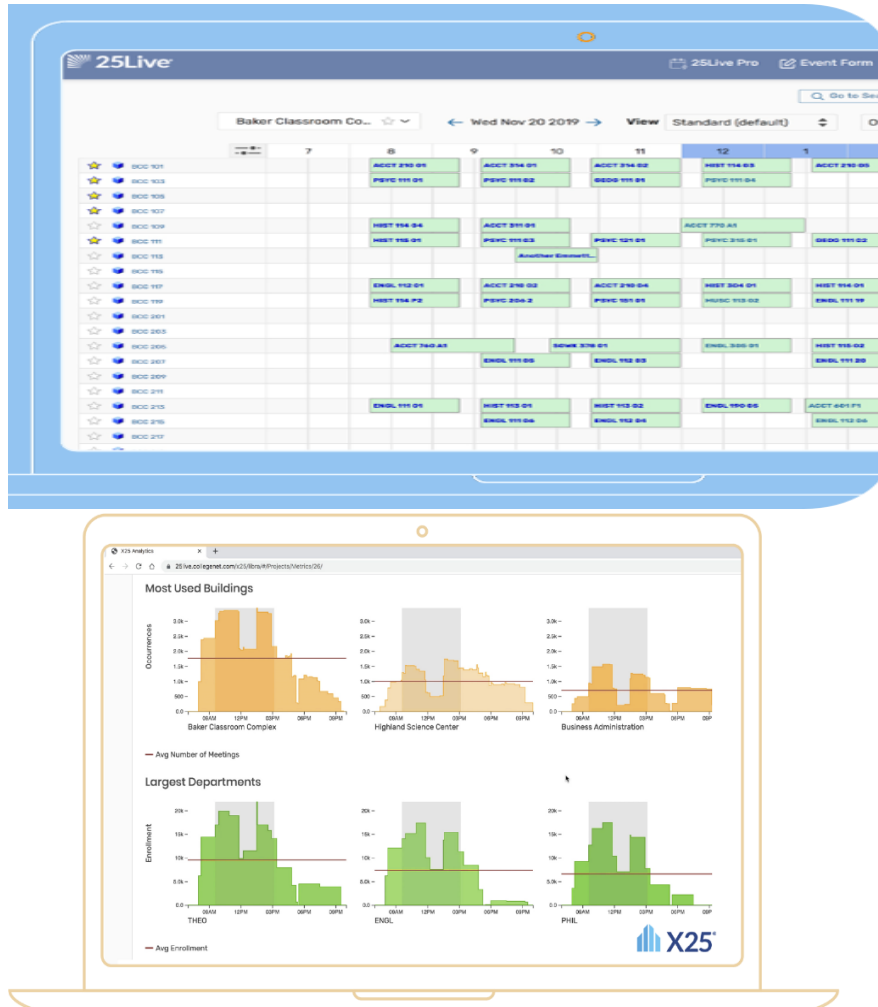


Figure 3: 25 Live

2.4 Addressing Challenges through our project

While existing solutions like **Roomzilla**, **AdAstra**, and **25Live** provide scheduling functionalities, they come with limitations such as cumbersome interfaces, limited real-time updates, and restricted customization options. Our project, **Kakhsya**, aims to address these challenges to ensure smooth and efficient system operations.

Kakhsya will feature a clean, intuitive, and responsive UI, making it accessible to both technical and non-technical users. By minimizing the number of steps required to find and schedule rooms, and offering features like tooltips, quick tutorials, and accessible support, the system will prioritize ease of use. Additionally, we will implement highly customizable scheduling rules to accommodate diverse needs, from recurring courses to multi-day events.

To enhance flexibility, **Kakhsya** will offer a mobile-friendly interface, enabling users to manage schedules and room bookings on the go. Key functionalities, such as checking room availability, scheduling resources, and receiving notifications, will be fully accessible across devices.

The system will also provide real-time reporting and analytics on room utilization, resource bookings, and scheduling trends. Dashboards will display critical metrics like space utilization, booking conflicts, and other key data points, empowering users to make informed decisions and optimize resource management.

Chapter 3: Procedure and Methods

3.1 Procedure and Methods

This chapter outlines the methodologies applied in the development of **Kakshya**, a Classroom Management Application designed to optimize classroom scheduling and resource allocation within an educational institution. It enables coordinators, teachers, and CRs to manage classroom usage efficiently in real-time. Below is a step-by-step methodology from research to deployment.

3.1.1 Research and Requirements Analysis

In the initial phase, we conducted research to identify the key requirements for a classroom management system tailored to an academic environment. We analyzed existing solutions to understand common challenges, such as scheduling conflicts and limited classroom resources. The insights gained helped us define system objectives, including efficient resource allocation, role-based access control, and real-time room availability tracking for **Kakshya**.

3.1.2 Requirement Definition

We defined specific requirements based on user roles for **Kakshya**: Coordinators require full control over room assignments and conflict resolution, Teachers need access to view and request rooms, and CRs must be able to view and share schedules. Key functionalities include room booking, availability updates, and automated notifications.

3.1.3 Programming Language Selection

To meet the technical requirements, C++ was chosen as the primary programming language. C++ offers robust data handling, efficiency, and compatibility with the MySQL database and Qt framework, making it suitable for developing a stable and secure application.

3.1.4 User Interface Design

A user-friendly interface is essential for effective interaction with the system. Using the Qt framework, we will create a graphical user interface (GUI) with a calendar view of classroom schedules, a search bar for room availability, and real-time notifications. Qt Designer will be used to design the interface, ensuring it is intuitive and easy to navigate for all user roles.

3.1.5 Database Setup

To manage classroom data securely, **Kakshya** uses **MySQL** as the database management system. The database schema includes tables for **Classrooms**, **Schedules**, **Users**, and **Requests**. Each table is designed to store relevant information with relational links, ensuring data integrity and efficient querying. The **Schedules** table will store time slots, allowing for quick retrieval of available rooms based on user requests. This structure ensures smooth data flow and accessibility, enabling the system to efficiently handle real-time room availability and booking information.

3.1.6 Implementation of Scheduling and Conflict Resolution Logic

The core functionality of **Kakshya** includes room scheduling and conflict detection algorithms. The system will check for overlaps and alert **Coordinators** if conflicts arise. **Teachers** can submit room requests, which will be verified against existing schedules to prevent double-booking. This conflict detection logic is implemented in **C++** for real-time processing, ensuring an efficient, responsive experience.

3.1.7 Role-Based Access Control

The application will provide different access levels based on user roles:

- **Coordinators:** Full access to manage classrooms, schedules, and resolve conflicts.
- **Teachers:** Limited access to view and request room allocations for their classes.

- **Class Representatives (CRs):** View-only access to check room availability and updates.

3.1.8 Notification System

We will integrate a **notification system** within **Kakshya** to keep users informed of any schedule changes, room reassignments, or requests requiring approval. Notifications will appear within the **Qt-based interface** and will be sent as alerts to relevant users, ensuring timely updates and smooth operations. This system will help all stakeholders (Coordinators, Teachers, and CRs) stay up to date with real-time changes, minimizing the risk of conflicts and delays.

3.1.9 Testing and Debugging

The application will undergo thorough testing, including:

- **Unit Testing:** Verifying each function (e.g., conflict detection, role-based permissions) for accuracy.
- **Integration Testing:** Ensuring database interactions, GUI functions, and backend logic work seamlessly.
- **User Testing:** Gathering feedback from potential users (e.g., coordinators, teachers) to refine the interface and features.

3.1.10 Documentation

The project will be documented to outline design, functionality, and implementation details, ensuring future maintenance and updates are easy to manage. User guides will also be provided for each role to maximize ease of use.

3.2 System Design

The system design section includes flowcharts, use case diagrams, and architecture diagrams for the Classroom Management Application, providing a visual representation of the workflow.

3.2.1 Use Case Diagram

The use case diagram illustrates interactions between users and the system. Coordinators can assign rooms, Teachers can request rooms, and CRs can view schedules. These interactions are mapped to their respective permissions, ensuring users only access relevant functionalities.

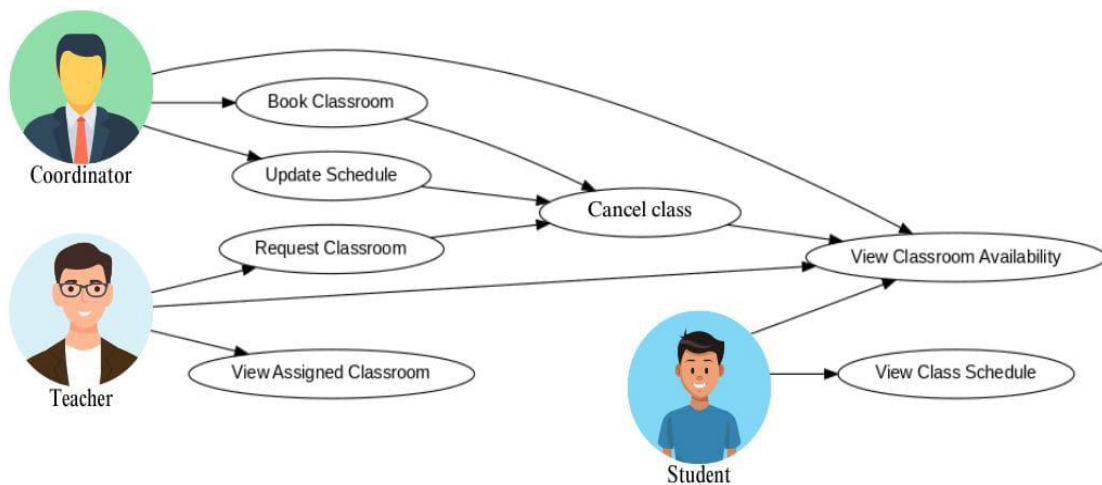


Fig 4: Use Case Diagram

This use case diagram represents the interactions within the Classroom Management Application:

1. Coordinator manages classrooms, resolves scheduling conflicts, and notifies users.
2. Teacher views room availability, requests classrooms, and receives notifications.
3. Class Representative (CR) accesses and shares the class schedule.

3.2.2 Flow Chart

A flow chart will show the sequence of steps for core processes, such as booking a classroom or resolving a conflict. Each step, from request submission to conflict detection and approval, will be represented in a clear, step-by-step format to aid understanding of the system's operations.

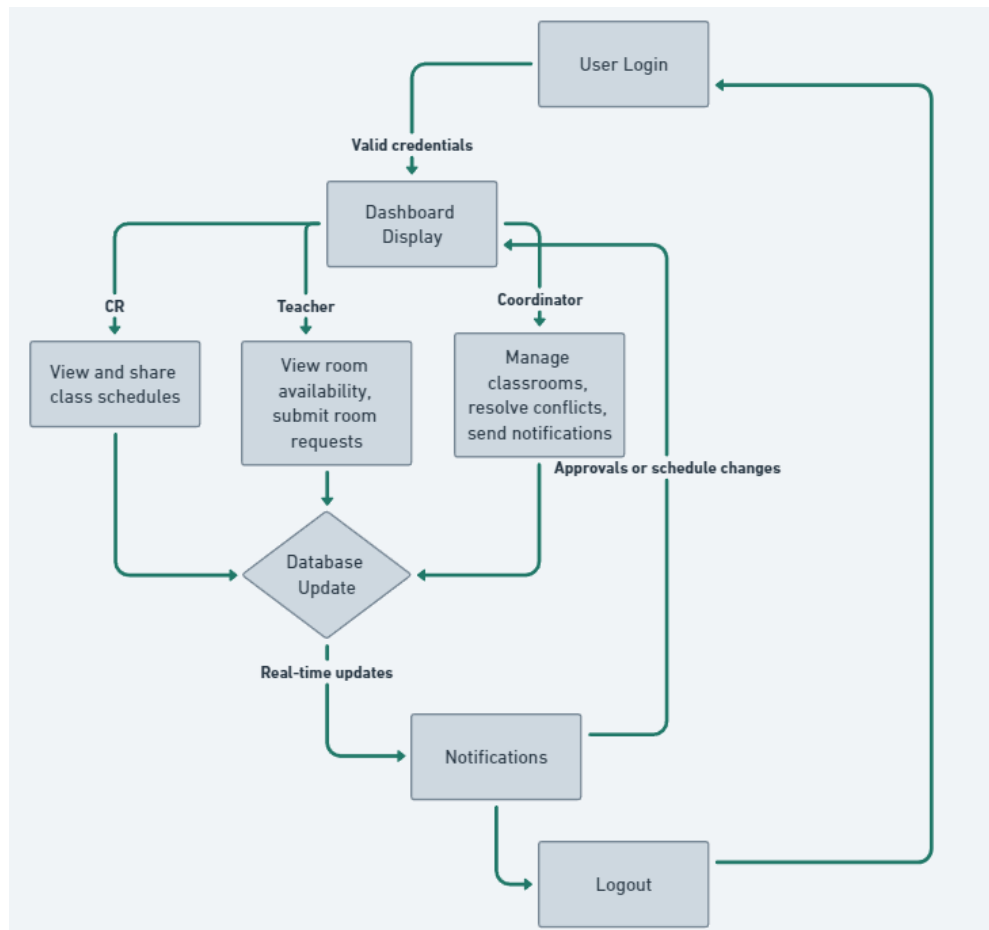


Fig 5: Flow Chart

This flowchart represents the main steps in the Classroom Management Application:

1. User Login: Each role (Coordinator, Teacher, CR) logs in with role-based access.

2. **Dashboard Display:**The main dashboard is shown based on the user's role permissions. This dashboard serves as the primary interface for interacting with **Kakshya**.

3. **Role-Based Actions:**

- Coordinator: Manages classrooms, resolves conflicts, and sends notifications.
- Teacher: Views room availability and submits room requests.
- CR: Views and shares class schedules.

4. Database Update: Real-time updates on room availability and scheduling.

5. Notifications: Alerts users about approvals or schedule changes.

6. Logout: Users log out to end their session, ensuring the security and privacy of their actions within **Kakshya**.

Chapter 4: System Requirement Specifications

To develop an effective and user-friendly Classroom Management Application, **Kakshya**, careful consideration has been given to both software and hardware specifications. The application's architecture is designed to provide a seamless, responsive user experience while ensuring reliability and efficiency in handling data. The front-end will be created using the Qt Framework, known for its cross-platform capabilities and robust graphical user interface (GUI) development tools, making it ideal for an intuitive and visually appealing interface. For the back-end, C++ will be used to handle the core application logic, offering performance efficiency, while MySQL will serve as the database to securely manage and retrieve scheduling information.

4.1 Software Specifications

The software required during the development phase of the application can be classified as:

4.1.1 Front-end Tools:

- **Qt Creator:** Qt Creator stands out as a versatile and cross-platform IDE primarily tailored for C++ and QML (Qt Modeling Language) development. Its feature-rich code editor enhances productivity with capabilities like syntax highlighting, code completion, and refactoring tools. Notably, it excels in enabling cross-platform development, allowing developers to create applications compatible with various operating systems like Windows, macOS, Linux, Android, and iOS from a unified codebase.
- **C++:** Using C++ as a front-end tool, especially within the context of the Qt framework, offers a robust approach to building visually appealing, high-performance applications. C++ is renowned for its efficiency and performance, making it well-suited for applications requiring fast computations and low-latency interactions. C++ offers developers flexibility and control over system resources and memory management, enabling optimization for specific use cases.

4.1.2 Back-end Tools:

- **C++:** C++ serves as a formidable backend tool, renowned for its exceptional performance, efficiency, and control. Its inherent speed and low-level access to system resources make it particularly well-suited for developing high-performance applications. With C++, developers can finely tune their code to optimize execution times and minimize memory usage, resulting in applications that run smoothly and efficiently even under heavy loads.
- **MySQL:** MySQL is primarily a free and open-source relational database management system (RDBMS) that allows you to store and organize large amounts of data in a structured way. Its relational model allows for the efficient organization and retrieval of data through structured queries ensuring data integrity and consistency.

4.2 Hardware Specifications

For smooth development and testing of **Kakshya**, the following hardware specifications are recommended. Since Qt Creator, Design Studio, and C++ are cross-platform tools, the project can be developed on a variety of operating systems, including Windows, macOS, and Linux. On the hardware side, the system requires a minimum of 2 GB RAM, an Intel i3 processor or higher, and at least 500 MB of HDD space, ensuring compatibility with commonly available campus computing resources. These specifications are selected to ensure smooth operation, even on basic hardware, making the application accessible and scalable across various educational institutions. A stable internet connection is required for downloading software, accessing the MySQL database (if hosted remotely), and managing version control (e.g., Git).

Chapter 5: Project Planning and Scheduling

The journey begins with coming together, progress is marked by unity, and success is achieved through collaboration. We will work jointly on our project, **Kakshya**, using GitHub, which will support our planning and time management efforts. Our goal is to complete the project within 12 weeks. The Gantt chart outlines the project's various phases.

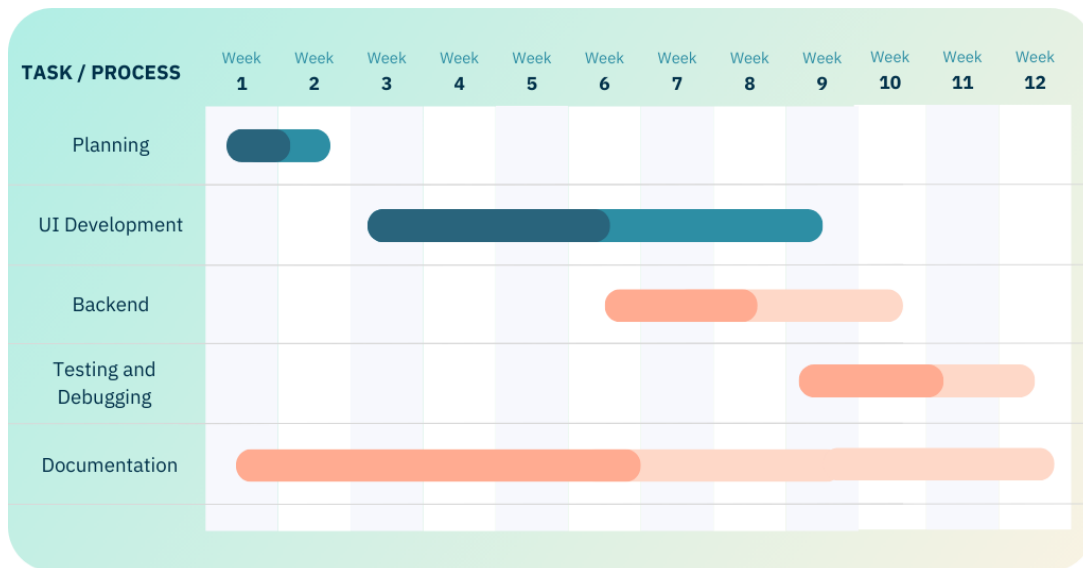


Fig 6: Gantt Chart

APPENDIX

System Architecture Diagram

The system architecture illustrates the relationship between the user interface, the backend logic, and the database. It demonstrates how data flows between the front-end (Qt-based interface), back-end (C++ logic), and the MySQL database to facilitate real-time classroom scheduling and management for **Kakshya**.

Database Schema

The database schema consists of the following tables:

- **Classrooms:** Stores details of classrooms (ID, name, capacity, equipment).
- **Schedules:** Contains room booking details (classroom ID, date, time, user ID).
- **Users:** Stores user information (ID, name, role, contact details).
- **Requests:** Tracks room reservation requests (status: pending, confirmed, or rejected).

UI Mock-ups

- **Login Page:** A simple login page to authenticate users based on roles (Coordinator, Teacher, CR).
- **Dashboard:** Displays classroom availability, current bookings, and options to request rooms.
- **Calendar View:** A calendar interface showing room availability, color-coded by booking status.

Logo of our application - 'Kakshya'



Fig 7: Logo

References

- Cross-platform software development for Embedded & Desktop. Qt. (n.d.). Retrieved November 6, 2024 , from <https://www.qt.io/>
- MySQL (n.d.). Retrieved November 6 , 2024 , from <https://www.mysql.com/>
- Roomzilla (n.d.). Retrieved November 6,2024 ,from <https://www.roomzilla.net/>
- Ad Astra (n.d.). Retrieved November 6,2024 ,from <https://www.aais.com/astra-schedule>
- 25 Live (n.d.). Retrieved November 6,2024 ,from <https://collegenet.com/scheduling/25live>
- GitHub. Available at: <https://github.com/>