



Automatic Time Table Generator - For Faculty

Software Requirement
Specifications Report

Team 1

SUBMITTED

BY: -

P. Tharun Chowdary -22N31A66E2

S. Vennela-22N31A66J4

V. Sreeja-22N31A66J0

R. Koushik Maharushi -22N31A66F3

Under the Guidance of: -

Dr. P. Hari Krishna M.Tech, PhD

Associate Professor

Department of Computational Intelligence

TABLE OF CONTENTS.....	2
INTRODUCTION	3
1. LITERATURE SURVEY	3-7
1.1 Existing System	3
1.2 Drawbacks of Existing System	4
1.3 Proposed System.....	4
1.4 Advantages.....	5
1.5 System Architecture.....	5
1.6 UML Diagrams.....	6-7
1.6.1 Use Case Diagram.....	6
1.6.2 Class Diagram.....	6
1.6.3 Sequential Diagram	7
1.6.4 Activity Diagram	7
2.OVERALL DESCRIPTION.....	8
2.1 Feasibility Study.....	8
2.1.1 Operational Feasibility.....	8
2.1.2 Economic Feasibility.....	8
2.1.3 Technical Feasibility.....	8
2.1.4 Legal and Ethical Feasibility	8
3.SYSTEM ANALYSIS.....	9
3.1 Software Requirement Specifications	9
3.2 Hardware and Software Requirements	9
3.2.1 Hardware Requirements.....	9
3.2.2 Software Requirements	9
4.FUNCTIONAL REQUIREMENTS.....	10
4.1 User Login	10
4.2 View Class Schedule.....	10
4.3 View Department Schedule	10
4.4 Manage Faculty Information.....	10
4.5 Notifications.....	10
4.6 User-Friendly Interface	10
5.NONFUNCTIONAL REQUIREMENTS.....	11-12
5.1 Performance.....	11
5.2 Reliability.....	11
5.3 Usability.....	11
5.4 Compatibility.....	11
5.5 Maintainability.....	11
6.REFERENCE.....	12

INTRODUCTION

The **Automatic Time Table Generator - For Faculty** is a software solution designed to streamline scheduling in academic institutions by automating faculty timetables based on set constraints and preferences. This system reduces manual effort and minimizes errors, creating balanced, conflict-free schedules. Its user-friendly interface allows faculty to view schedules, request swaps, and adapt to changes easily. Department heads (HODs) can oversee schedules, manage assignments, and make data-driven decisions using historical reports. By providing structured, efficient timetabling, this tool enhances institutional efficiency, reduces administrative workload, and improves the academic experience for both faculty and students.

1. LITERATURE SURVEY

1.1 Existing System:

The current system for managing academic schedules in many institutions relies heavily on manual processes or simple digital tools, such as spreadsheets, to create and maintain timetables. This approach is labor-intensive, requiring significant administrative time and effort to coordinate schedules, prevent conflicts, and ensure classes are adequately staffed. Errors are common due to the complexity of managing various constraints, including classroom availability, faculty preferences, and course requirements. Adjustments for last-minute changes, such as faculty absences or class swaps, are difficult to manage, often leading to confusion or unintentional overlaps.

Faculty members usually access static schedules, which lack real-time updates, making it challenging for them to stay informed of schedule modifications. Heads of Departments (HODs) also face difficulties with limited oversight, as there is no centralized system to track schedules or analyze data on scheduling trends. Consequently, this system is inefficient, lacks adaptability, and falls short of supporting the dynamic needs of academic institutions today.

1.2 Drawbacks of Existing System:

- Time-Consuming Process.
- Prone to Errors.
- Limited Flexibility.
- Lack of Real-Time Updates.
- Inefficient Resource Utilization.
- Lack of Integration.

1.3 Proposed System:

The proposed system, titled "Automatic Time Table Generator," is designed to streamline the scheduling process within academic institutions by automating timetable creation. This solution addresses the challenges of manual scheduling, where overlaps, conflicts, and inefficient use of faculty resources are common. By incorporating predefined constraints and preferences, the system ensures that each timetable is balanced, optimizing both faculty and classroom availability.

With a user-friendly interface, faculty can easily view their schedules and, if needed, request swaps. The system includes an approval workflow that enables quick adaptations to accommodate last-minute changes without disrupting the overall schedule. This flexibility not only enhances faculty satisfaction but also provides stability for students by reducing unplanned schedule changes.

Additional features allow Head of Departments (HODs) and administrators to access department-wide schedules and historical data for better decision-making. This hierarchical access supports effective resource allocation and long-term planning. Overall, the Automatic Time Table Generator offers a reliable, adaptable, and efficient solution for modernizing scheduling processes, benefiting both faculty and students.

1.4 Advantages:

- Automated Scheduling.
- User Friendly Interface.
- Flexibility and Accessibility.
- Real Time Communication.
- Streamline Data Entry and Management.

1.5 System Architecture:

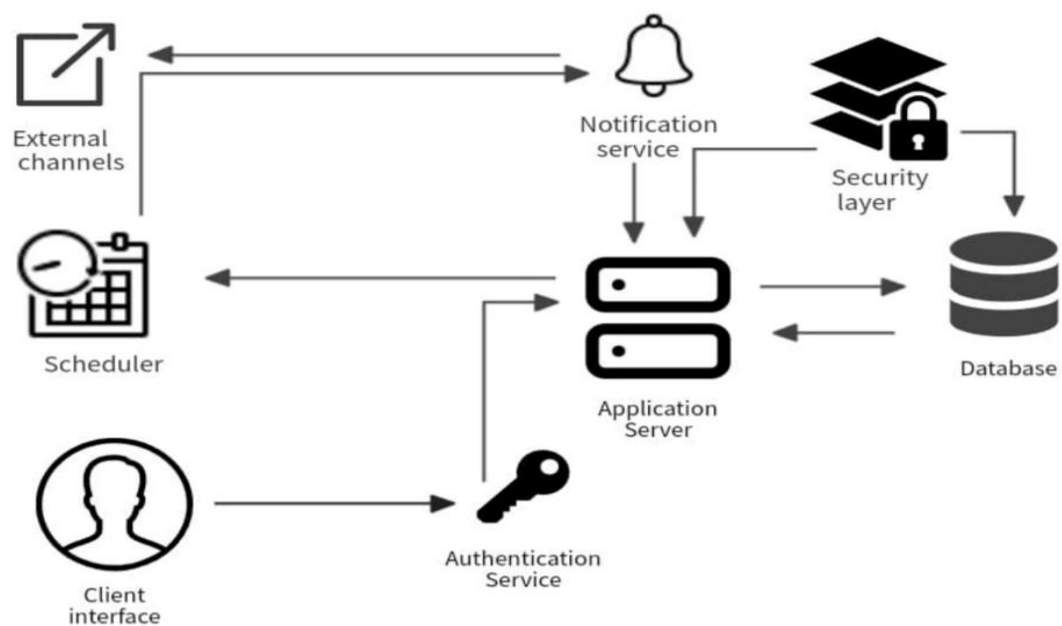


Fig 1.1 System Architecture of Automatic Time Table Generator - For Faculty

1.6 UML Diagrams:

1.6.1 Use Case Diagram:

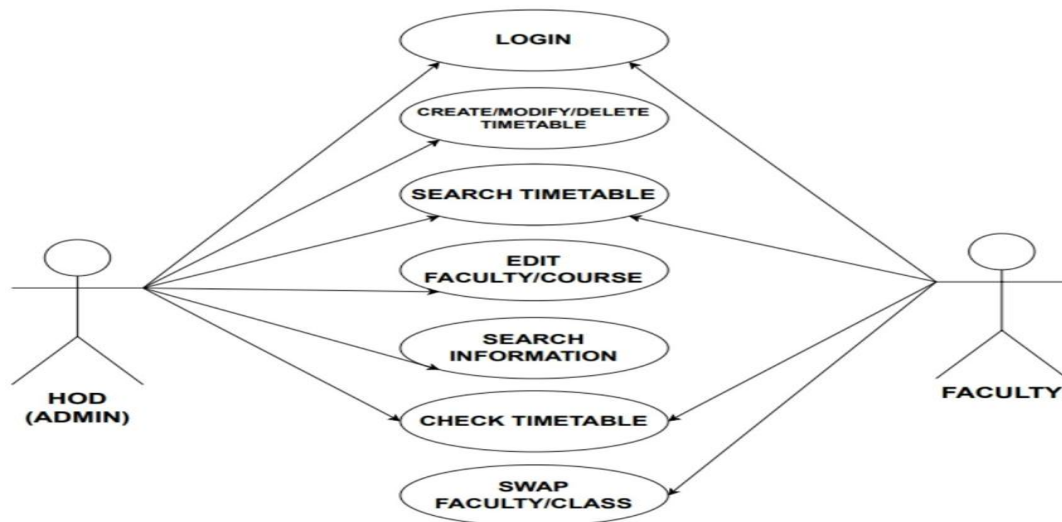


Fig 1.2 Use Case Diagram of Automatic Time Table Generator - For Faculty

1.6.2 Class Diagram:

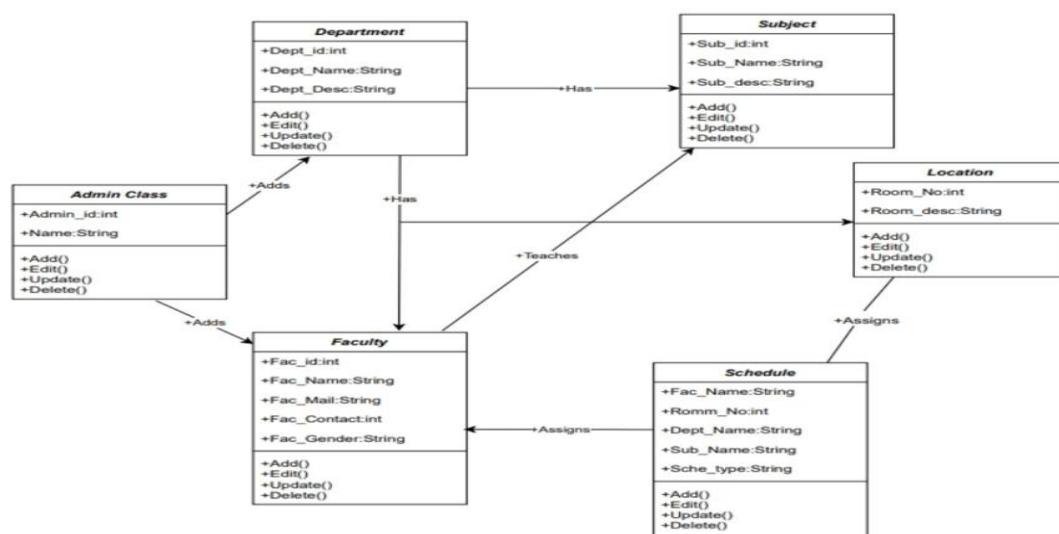


Fig 1.3 Class Diagram of Automatic Time Table Generator - For Faculty

1.6.3 Sequence Diagram:

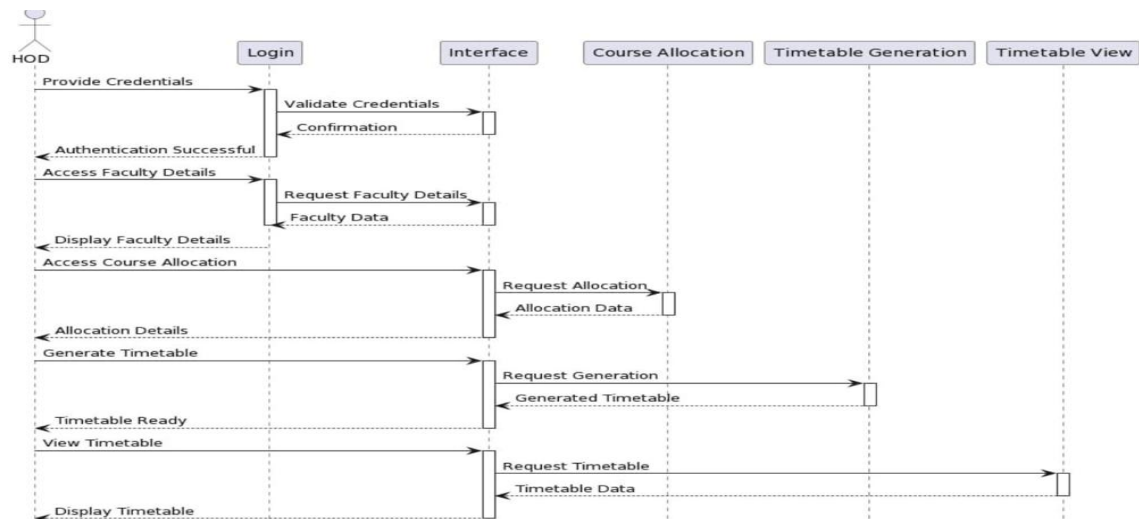


Fig 1.4 Sequence Diagram of Automatic Time Table Generator - For Faculty

1.6.4 Activity Diagram:

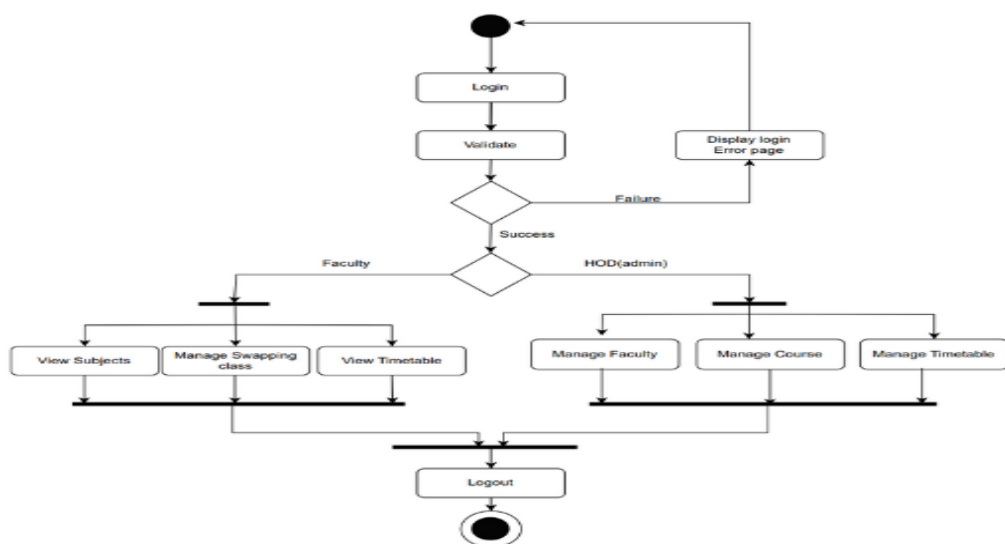


Fig 1.5 Activity Diagram of Automatic Time Table Generator - For Faculty

2.OVERALL DESCRIPTION

2.1 Feasibility Study:

The feasibility study for the "Automatic Time Table Generator" evaluates various factors to ensure successful implementation and long-term sustainability:

2.1.1 Operational Feasibility:

This system aims to simplify the scheduling process, making it more efficient and accessible for faculty, HODs, and administrators. Its user-friendly interface requires minimal training, and the automated approach aligns well with the day-to-day needs of academic institutions, ensuring smooth integration into existing workflows.

2.1.2 Economical Feasibility:

The system reduces manual scheduling labor and associated errors, leading to significant cost savings over time. By optimizing faculty and classroom usage, institutions can potentially reduce overhead costs. The system's initial investment in software and training is balanced by its long-term benefits, making it a financially sound choice.

2.1.3 Technical Feasibility:

Built on a scalable architecture, the system is compatible with commonly used hardware and software in educational institutions. It can be hosted on existing servers or cloud infrastructure, ensuring seamless integration without requiring substantial technical upgrades.

2.14 Legal and Ethical Feasibility:

The system complies with data privacy and security regulations, ensuring that faculty schedules and personal information are protected. It also maintains transparency in schedule management, supporting ethical standards and fostering trust among users.

3. SYSTEM ANALYSIS

3.1 Software Requirement Specifications

A software requirements specification (SRS) is a comprehensive description of the intended purpose and environment for software under development. The SRS fully describes what the software will do and how it will be expected to perform. An SRS minimizes the time and effort required by developers to achieve desired goals and minimizes the development cost. A good SRS defines how an application will interact with system hardware, other programs, and human users in a wide variety of real-world situations.

3.2 Hardware and Software Requirements

3.2.1 Hardware Requirements

- HDD 512GB
- RAM 4GB
- PROCESSOR i5
- KEY BOARD
- MOUSE

3.2.2 Software Requirements

- Frontend: HTML5, CSS3, Javascript ES6, php 7.4
- Backend: MySQL 8.0
- Operating System: Windows 11

4. FUNCTIONAL REQUIREMENTS

4.1 User Login:

A secure login system allows different access levels for faculty, HODs, and administrators, ensuring data privacy and role-based access to functions

4.2 View Class Schedule:

Faculty members can view their individual schedules, allowing them to check assigned classes and make arrangements as needed.

4.3 View Department Schedules:

HODs and administrators have access to department-wide schedules, enabling them to oversee class distribution and faculty assignments efficiently.

4.4 Manage Faculty Information:

The system includes a module to update faculty details, preferences, and availability, supporting accurate and personalized timetable generation.

4.5 Notifications:

Automatic notifications inform faculty of schedule updates, swap approvals, and changes, keeping everyone informed in real time.

4.6 User-Friendly Interface:

The interface is intuitive and easy to navigate, designed to streamline schedule management with minimal training needed for users.

5. NONFUNCTIONAL REQUIREMENTS

Non-functional requirements describe how the system should perform, rather than what it should do. Here are simplified non-functional requirements for the Faculty Time Table Management System (FTTMS):

5.1 Performance: The system should generate schedules quickly, with minimal processing time, even during high-demand periods. It should handle multiple users simultaneously without lag.

5.2 Reliability: The system must be stable, maintaining uptime for consistent access, and ensuring schedule accuracy. Regular backups prevent data loss in case of system failure.

5.3 Usability: Designed with a user-friendly interface, the system should be easy for all users to navigate with minimal training, supporting smooth interaction for faculty and administrators.

5.4 Compatibility: The system should work on various devices and operating systems, ensuring access on desktops, tablets, and mobile devices for flexibility and convenience.

5.5 Maintainability: The system should be easy to update and maintain, allowing for quick bug fixes and feature enhancements to adapt to evolving institutional needs.

6. REFERENCES:

- Krithi, Dr M Ramakrishna, student management system a survey Computer Science and Engineering, Vemana Institute of Technology, Bangalore-34 International Research Journal of Computer Science (IRJCS) Issue 05, Volume 4 (May 2017).
- Lalit Mohan Joshi M.tech schola BTKIT Dwarahat, Almora, Uttarakhand "A Research Paper on College Management System" International Journal of Computer Applications (0975-8887) Volume 122-No.11, July 2015.
- S. W. Liew, N. F. M. Sani, M. T. Abdullah, R. Yaakob, and M. Y. Sharum, “A Research Paper on Faculty Management System in Educational Institutions”, vol. 83, pp. 201–207, Jun. 2019.