CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

The "Time Table Generator" project streamlines college timetable creation using HTML, CSS, Python, and MySQL, enhancing efficiency through automation and control, connecting with various modules and processes. Information technology is widely used for communication. Administrators add students, staff, subjects, and timetable details using Python. HTML and CSS make the front-end of an online application more effective and secure. This project aims to provide students with a helpful time table generator system for easy access to timetable details. Python and SQLite have been used in the development of an online automatic timetable generator. The program, Automatic Timetable Generator, automates the creation of timetables, allowing for automatic management of periods and scheduling for instructors. It also controls tardiness and early arrivals. The system sets maximum and minimum workloads for faculty members for effective scheduling. The current system faces challenges when teachers are absent or late, making manual assignment of substitute teachers challenging. Although majority college organization work has been mechanized, the lecture timetable preparation is still commonly done by hand due to its inherent difficulties.

The physical lecture-timetable preparation demands significant time and efforts. The manual lecture-timetable scheduling is a limitation fulfillment problem in which we find a result that satisfies the given set of constraints. There have been numerals of approaches made in the earlier period to the difficulty of constructing timetables for colleges and schools. Timetabling problems may be solving by diverse methods inherited from operation study such as graph coloring, local search measures such as tab search, simulated annealing, genetic algorithms or from backtracking based constraint fulfillment handling. In our project, timetable problem is formulated as a constraint fulfillment problem and we proposed a realistic timetable algorithm which is capable of taking care of both hard and soft constraints. It is a complete timetable solution for Colleges which help to overcome the challenges in manually constructing the timetable.

Currently timetable is managed manually. It will help to manage all the Periods automatically and also will be helpful for faculty to get timetable It will also manage timetable when any teacher is absent late coming or early going. Maximum and minimum work-load for a Faculty for a day, week and month will be specified for the efficient generation of timetable. In the existing system, the problem occurs when any teacher is on the leave and he will not able to inform or inform it late than the manual assignment of substitute teacher is also a very difficult job. When the schedule tracker is generated manually, there is a case when the department head wants to makes some changes in the lectures. At this situation, the chances of the collision of the period or assignment of the teachers will increase because it is not possible for one teacher to remember all assignment done earlier. The manual maintenance of the databases of items, schedule tracker processing is a time taking process a somehow erroneous. So there is a need for the new system to resolve such problems.

1.2 OBJECTIVE

- **Automation**: Develop a system that automates the process of creating timetables, reducing the manual effort and time required for scheduling courses.
- **Efficiency**: Improve the efficiency of timetable generation by implementing algorithms that optimize resource utilization and minimize conflicts.
- **Flexibility**: Provide a flexible and customizable solution that accommodates various scheduling preferences, constraints, and course requirements.
- **User-Friendly Interface**: Design an intuitive user interface that allows users to easily interact with the system, input course details, and view generated timetables.
- **Randomization**: Implement a randomized scheduling approach to ensure diversity in timetables and prevent the creation of predictable patterns.
- **Constraint Handling**: Develop mechanisms to handle constraints such as room availability, instructor preferences, and any other specific limitations in the scheduling process.
- **Clear Presentation**: Present the generated timetable in a clear and structured format, making it easy for users to interpret and plan.
- **Optimization Algorithms**: Explore and implement optimization algorithms to enhance the overall quality of timetables, considering factors like course distribution and balancing.
- **Scalability**: Design the system to be scalable, allowing for the addition of new courses and scalability to accommodate growing educational institutions.
- **User Empowerment**: Empower users with the ability to make informed decisions by providing insights into the scheduling process, potential conflicts, and alternative options.

1.3 SCOPE

This project will be very beneficial to the university because managing numerous faculties and assigning courses to them simultaneously by hand is a very challenging task that this project will assist in managing effectively. This faculty timetable can be readily controlled while taking into account the maximum and lowest workload. The faculty data in the database can also be used to keep track of the faculty's expertise in specific fields. Attribute The accuracy of the project will allow for a more corrective approach to the creation of this schedule. This project will produce output that is mostly corrective and error-free. The project's potential future improvement is the creation of a master schedule for the departments and the entire college. Further adjustments can be made while maintaining the project's approach and methods to accomplish this improvement. Additionally, it can be utilized to assign a certain time slot that the instructor prefers. The university website may incorporate this timetable maker, making it more useful. The implementation of a time table management system can make it simpler for the schools to assign a teacher to a class in the event of an absent.

CHAPTER-2
LITERATURE REVIEW

2.1 LITERATURE REVIEW

A literature review on automatic timetable generators highlights the significance of this technology in educational institutions. Research indicates that manual timetable creation is a time-consuming and error-prone task, necessitating the development and implementation of automated solutions. Studies show that automatic timetable generators effectively consider various constraints, such as room availability, faculty preferences, and student course requirements, to create optimized and conflict- free schedules. The literature emphasizes the efficiency gains and error reduction achieved through automation, leading to improved resource utilization and overall institutional effectiveness. Additionally, researchers discuss the adaptability of these generators to dynamic scheduling environments, enabling quick adjustmentsin response to changing academic needs. Overall, the literature supports the adoption of automatic timetable generators as a valuable tool for enhancing the operational efficiency and organizational structure of educational institutions.

In some years two main approaches have been successful for implementing the timetabling problem. The first approach is based on local search procedures method such as simulated annealing, tabu search and genetic algorithms. These methods express constraints as some value of functions, which are minimized by a heuristic search of better solutions in reference of some initial feasible solution. The second approach is based on constraint programming (CP). Its main advantage is declaratively a direct statement of the constraints serves as part of the program. This makes the program easy to modify, which is critical in timetabling problems. The constraints are handled through a system of constraint propagation, which decrease domains of variables, coupled with backtracking search. The main disadvantages of these approaches are 1. Difficulties with expressing hard and soft constraints, 2. The need to determine their parameters through experimentation and 3. Possible problems with improving the initial feasible solution, which– as a rule – may be determined without problem? An attempt to overcome the drawbacks along soft constraints was discussed, successfully combined local search with constraint satisfaction to decrease their drawbacks.

2.2 NEED OF WORK

- Efficiency Improvement: Manual timetable creation is time-consuming and prone to errors. An automated system ensures efficient allocation of resources and minimizes the time required for scheduling.
- Resource Optimization: The project addresses the need for optimizing resource utilization, including classrooms, instructors, and other facilities, to enhance the overall efficiency of educational operations.
- Adaptability to Constraints: Educational institutions face various constraints such as room availability, instructor preferences, and specific institutional requirements. The project aims to handle these constraints effectively, ensuring a coherent and conflict-free timetable.
- Scalability: With the ability to adapt to institutions of varying sizes and structures, the project provides a scalable solution that can meet the scheduling needs of different educational settings.
- User-Friendly Interface: The creation of a user-friendly interface addresses the need for an accessible and intuitive system that allows users to input and modify course details with ease, fostering a positive user experience.
- Diversity in Timetables: Introducing randomized scheduling approaches adds diversity to timetables, preventing predictability and promoting adaptability. This feature is crucial for accommodating different preferences and ensuring a fair distribution of courses.
- Reduction of Manual Effort: Automation reduces the manual effort required for timetable creation, allowing educational institutions to allocate human resources more effectively in other critical areas.
- Enhanced Planning and Organization: The project contributes to better planning and organization within educational institutions, facilitating smoother operations and improving the overall learning environment.

2.3 PROBLEM STATEMENT

The problem statement for the Automatic Timetable Generator project is rooted in the inefficiencies and challenges associated with manual timetable creation in educational institutions, including time-consuming scheduling processes, difficulty in adapting to diverse constraints, suboptimal resource utilization, lack of scalability, absence of a user-friendly interface, predictable timetables, and ineffective constraint handling. The project aims to address these issues by developing an automated system that streamlines the scheduling process, optimizes resource allocation, handles constraints effectively, ensures scalability, provides a user-friendly interface, introduces diversity in timetables, reduces manual effort, and ultimately enhances the overall efficiency and adaptability of educational institutions.

CHAPTER-3
SYSTEM DEVELOPMENT

3.1 EXISTING SYSTEM

Typically, it involves gathering information about available resources, such as classrooms, teachers, and time slots, and then manually inputting constraints and preferences. This process can be time-consuming and prone to errors. Moreover, adjustments to the timetable often require manual intervention, making the system less flexible. In some cases, institutions may use spreadsheet software or basic scheduling tools, which lack sophisticated optimization algorithms. As a result, the current system may struggle to efficiently handle complex scheduling scenarios, leading to suboptimal timetables and potential scheduling conflicts. To address these limitations, there is a need for a more advanced and automated timetable generation system that can consider a wider range of constraints, preferences, and optimization criteria to produce efficient and conflict-free schedules.

3.2 PROPOSED METHODOLOGY

The proposed methodology for developing an Automatic Timetable Generator involves a systematic approach beginning with a thorough analysis of requirements, including input from stakeholders. Data collection encompasses the gathering of relevant scheduling information, followed by the selection and implementation of optimization algorithms, such as Genetic Algorithms or Simulated Annealing, to efficiently generate timetables. Constraints, including room availability and teacher preferences, are identified and addressed through strategic constraint handling. A user-friendly interface is designed for administrators, integrating preferences and allowing for manual adjustments. The algorithm is then implemented, tested, and optimized, with a focus on performance and usability. The system undergoes validation and user testing before deployment, followed by continuous monitoring, updates, and maintenance to ensure ongoing effectiveness and responsiveness to user needs. This methodology aims to create a robust and adaptable Automatic Timetable Generator tailored to the unique requirements of educational institutions.

3.3 SYSTEM ARCHITECTURE

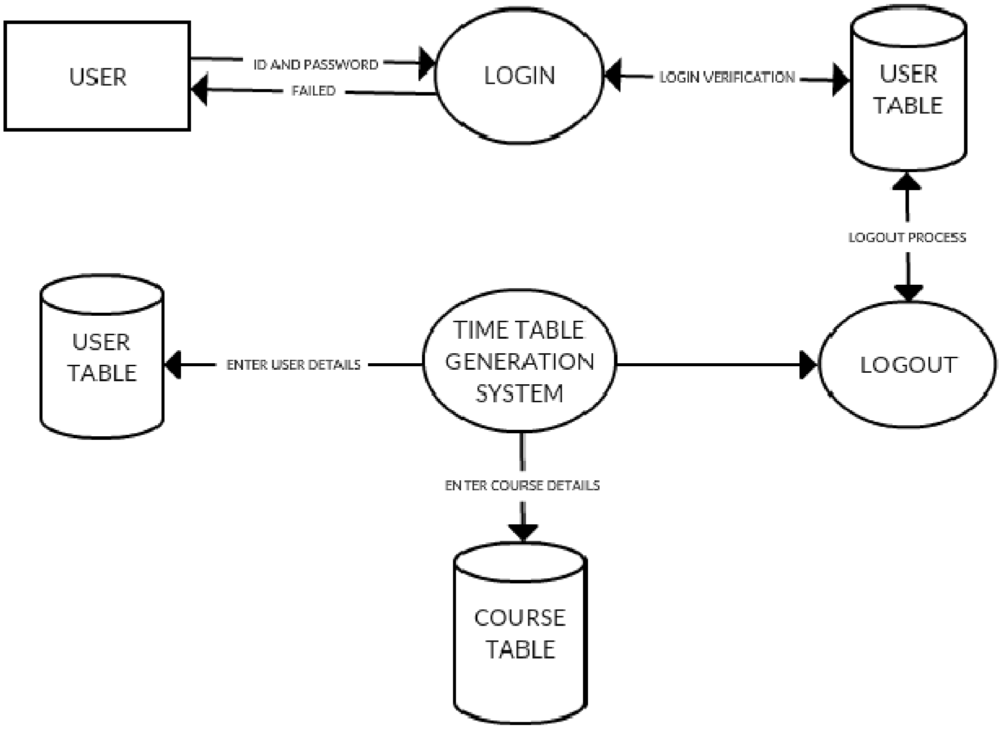

Figure 1: System Architecture

CHAPTER-4
DESIGN METHODOLOGY

4.1 DATA FLOW DIAGRAM

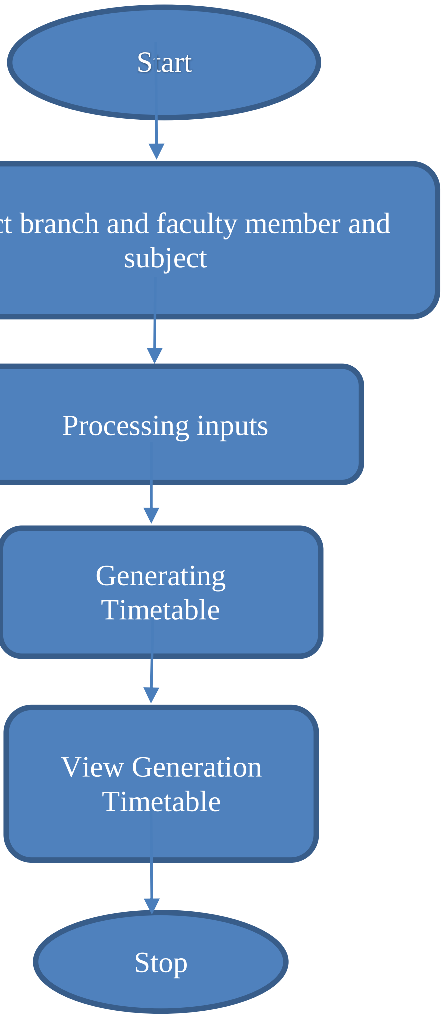
Start
Select branch and faculty member and subject
Processing inputs
Generating Timetable
View Generation Timetable
Stop

Figure 2: Data Flow Diagram

4.1.1 ER DIAGRAM

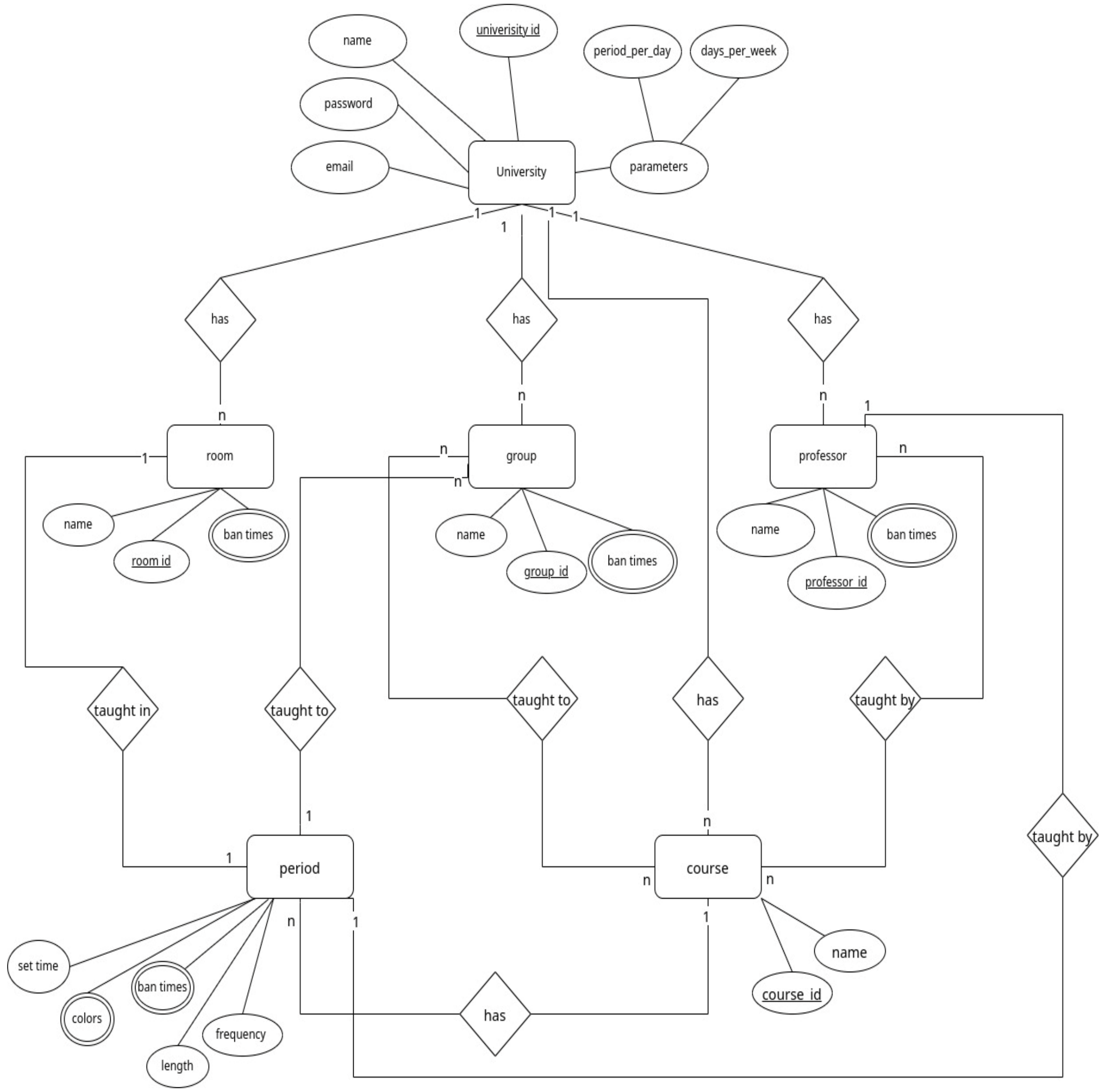

Figure 3: ER Diagram

4.1.2 UML DIAGRAM

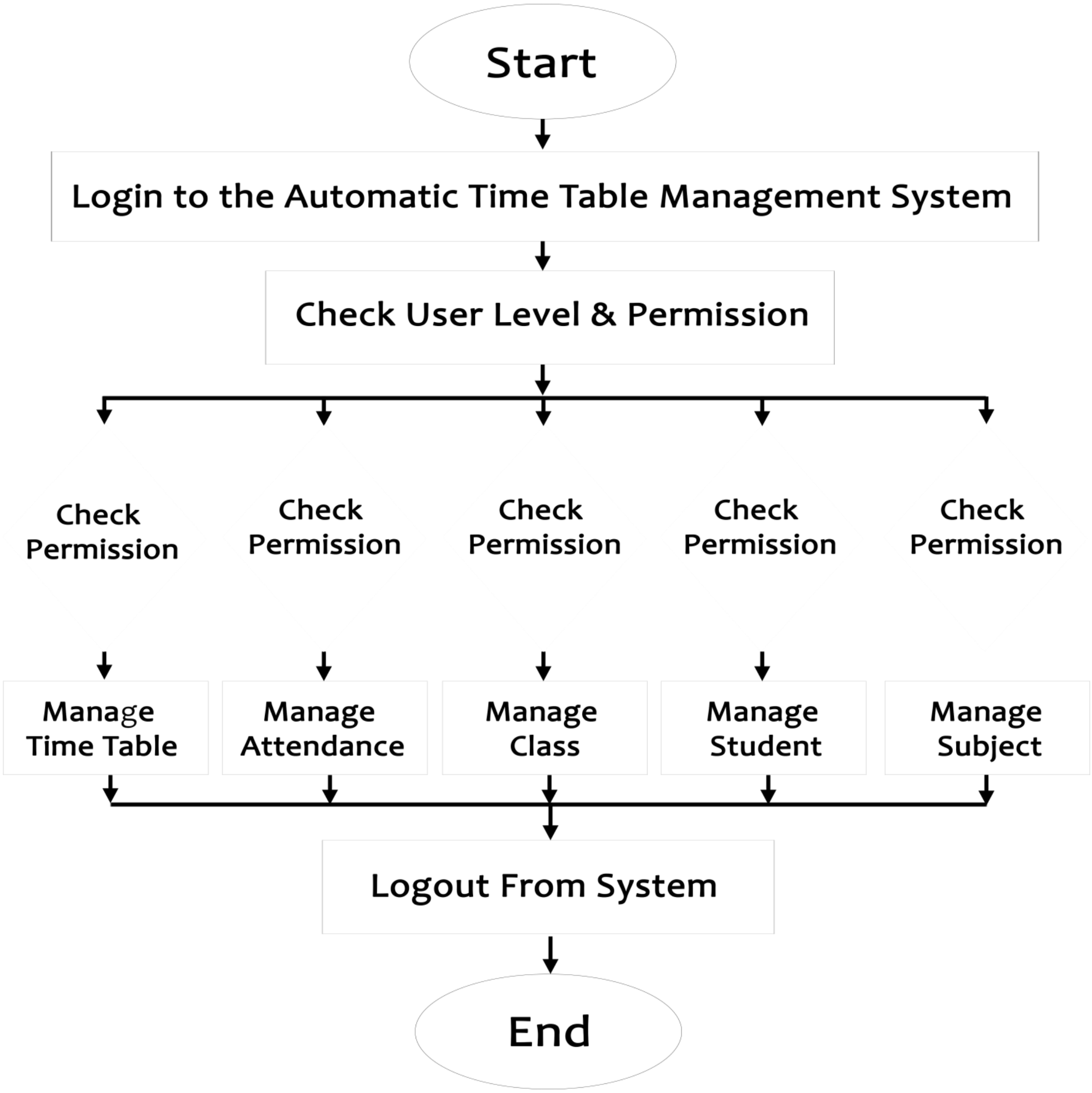

Figure 4: UML Diagram

4.1.3 CLASS DIAGRAM

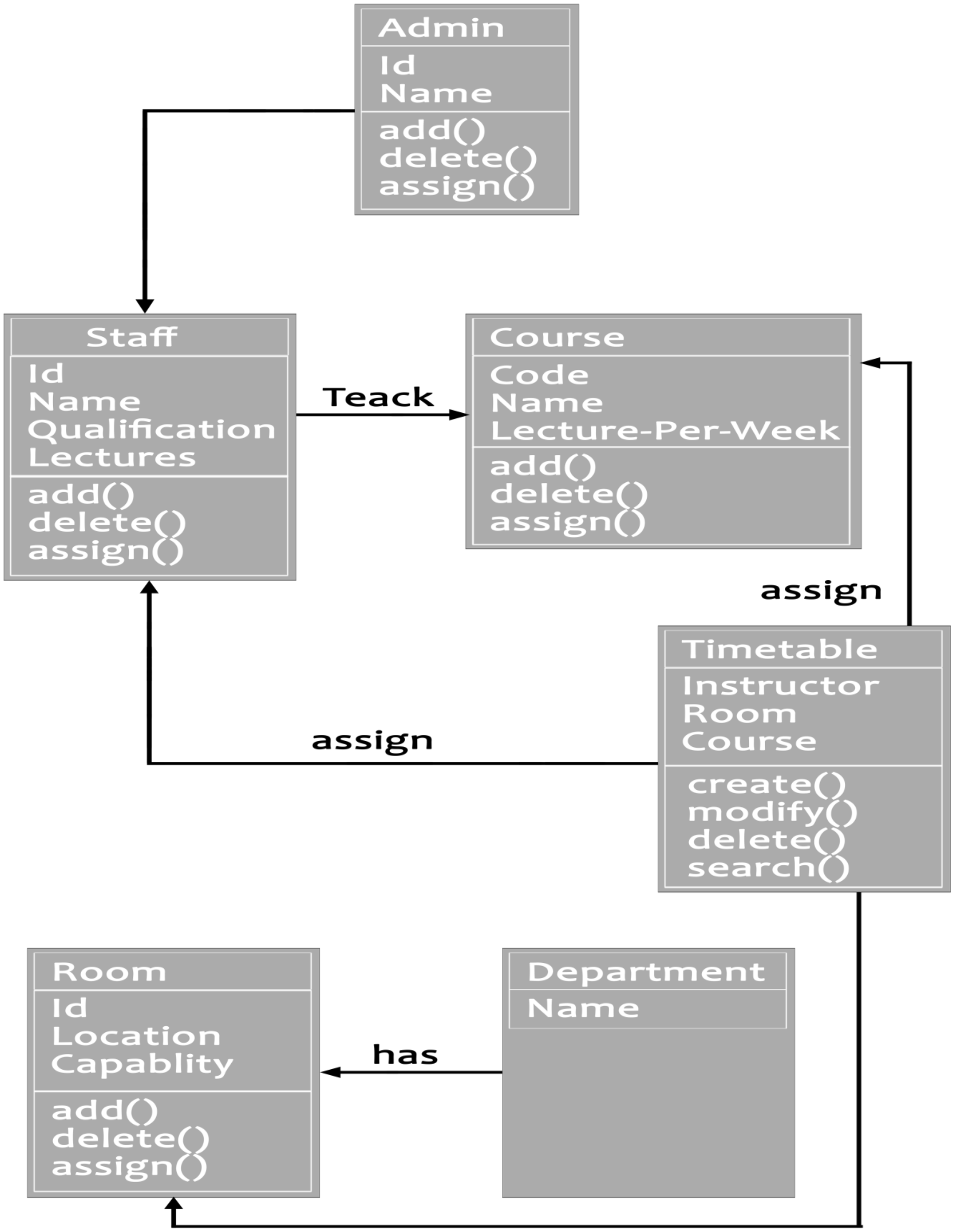

Figure 5: Class Diagram

4.1.4 USE CASE DIAGRAM

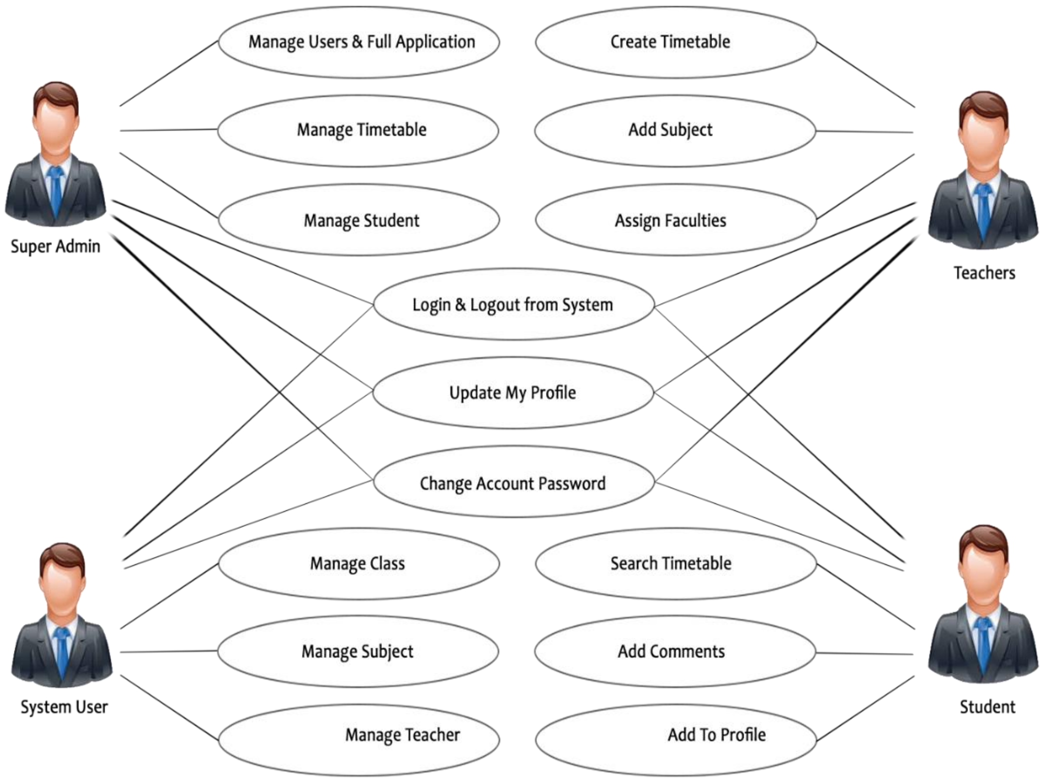

Figure 6: Use Case Diagram

4.1.5 SEQUENCE DIAGRAM

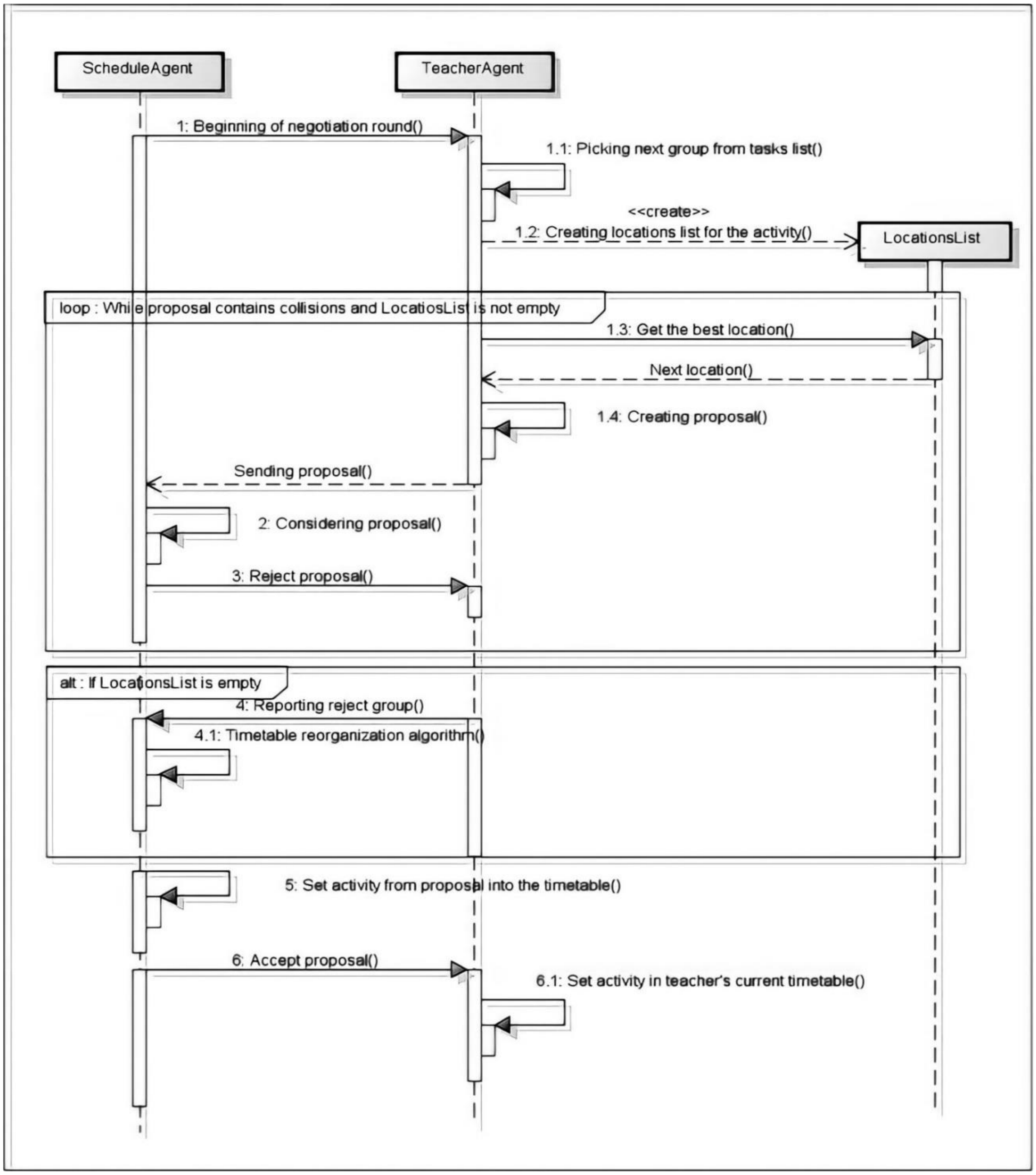
ScheduleAgent
TeacherAgent
1: Beginning of negotiation round()
1.1: Picking next group from tasks list()
<<create>>
1.2: Creating locations list for the activity()
LocationsList
loop : While proposal contains collisions and LocatiosList is not empty
1.3: Get the best location()
Next location()
1.4: Creating proposal()
Sending proposal()
2: Considering proposal()
3: Reject proposal()
alt : If LocationsList is empty
4: Reporting reject group()
4.1: Timetable reorganization algorithm()
5: Set activity from proposal into the timetable()
6: Accept proposal()
6.1: Set activity in teacher's current timetable()

Figure 7: Sequence Diagram

CHAPTER-5
IMPLEMENTATION DETAILS

5.1 FUNCTIONAL REQUIREMENTS

- **Course Management**: The system should allow the addition, modification, and deletion of courses, including details such as course name, duration, and any specific constraints.
- **Timetable Generation**: The system must automatically generate timetables based on defined courses, considering factors like course duration, instructor availability, room constraints, and user preferences.
- **User Authentication**: Implement user authentication mechanisms to ensure secure access, allowing only authorized users, such as administrators or scheduling personnel, to modify or generate timetables.
- **User Interface**: Provide an intuitive and user-friendly interface for users to input course details, view generated timetables, and make adjustments easily.
- **Constraint Handling**: Handle various constraints efficiently, including room availability, instructor preferences, and any other user-defined constraints, to ensure conflict-free timetables.
- **Randomized Scheduling**: Incorporate randomized scheduling approaches to diversify timetables, preventing predictability and ensuring fairness in course allocations.
- **Optimization Algorithms**: Implement optimization algorithms to maximize resource utilization, such as evenly distributing courses, minimizing gaps, and avoiding resource conflicts.
- **Reporting and Insights**: Provide reporting features to offer insights into the generated timetables, including potential conflicts, resource utilization statistics, and alternative scheduling options.
- **Scalability**: Design the system to be scalable, accommodating institutions of varying sizes and structures, and ensuring performance even as the volume of data increases.
- **Export and Import Functionality**: Include functionality to export generated timetables in different formats (e.g., PDF, Excel) for easy sharing and import capabilities for existing data or timetables.
- **Notifications**: Implement a notification system to alert users about important scheduling events, conflicts, or changes to the timetable.
- **Feedback Mechanism**: Provide a mechanism for users to provide feedback on generated timetables, allowing for continuous improvement and refinement of the scheduling algorithm.

5.2 NON-FUNCTIONAL REQUIREMENTS

- **Performance**: The system must handle a significant number of courses, users, and scheduling constraints efficiently, ensuring timely generation of timetables even for large datasets.
- **Scalability**: The system should scale seamlessly to accommodate the growth of educational institutions, supporting an increasing number of courses, instructors, and users.
- **Usability**: The user interface should be intuitive and user-friendly, requiring minimal training for administrators and scheduling personnel to operate the system effectively.
- **Reliability**: The system must be reliable, ensuring consistent and accurate timetable generation while minimizing errors or disruptions.
- **Security**: User authentication mechanisms should be robust to prevent unauthorized access, ensuring the confidentiality and integrity of scheduling data.
- **Flexibility**: The system should be flexible, allowing for easy adaptation to changes in course structures, institutional policies, and scheduling preferences.
- **Availability**: The system should be available for use during critical scheduling periods, and any scheduled downtime for maintenance should be minimal and communicated in advance.
- **Auditability**: The system should log user activities and changes to timetables, providing an audit trail for accountability and analysis purposes.
- **Compliance**: The system should comply with relevant data protection and privacy regulations, ensuring the secure handling of sensitive scheduling information.
- **Response Time**: The system's response time for generating timetables and handling user interactions should be optimized for a smooth and efficient user experience.

5.2.1 HARDWARE REQUIREMENTS

- Processor : i5-12500
- Speed : 3.00 GHz
- RAM : 8.00 GB
- Hard Disk : 256 GB

5.2.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 11
- Front End : HTML, CSS, Bootstrap
- Back End : PHP, MySQL
- Tools : VS Code
- Browser : Google Chrome

CHAPTER-6
CONCLUSION

6.1.1 CONCLUSION

It is complicated task that to handle many Faculty's and allocating subjects for them at a time physically. So us proposed system will help to overcome this disadvantage. Thus we can produce timetable for any number of courses and multiple semesters. This system will help to create dynamic pages so that for implementing such a system we can make use of the different tools are widely applicable and free to use also. Managing different faculties and allocating subjects to them at the same time is a daunting task. As a consequence, the method we recommend would aid in addressing this drawback. Thus we can produce timetable for any number of courses and multiple semesters. This system will help to create dynamic pages so that for implementing such a system we can make use of the different tools are widely applicable and free to use also. The main advantage of this project is that it allows users to store information in one location and view it via Monitor. Students will see the schedule with a fast turnaround instead of doing boring paper work. This method is easy to use and produces timetables quicker and more efficiently, saving time.

CHAPTER-7
FUTURE SCOPE

7.1 FUTURE SCOPE

In our system, there are some problems those are user has to format it a bit after it is prepared. The system will generate a particular class timetable at a time and once the user downloads the current timetable sheet then the next timetable can be generated. In future work, we will overcome these disadvantages by using some logical approaches.

CHAPTER-8
REFRENCES

8.1 RESEARCH JOURNALS PAPERS

- Automated Time Table Generator Using Spread Sheet for Educational Modules
- Automatic Timetable Generation Using Genetic Algorithm
- Automated Time Table Generation Using Multiple Context Reasoning for University Modules
- Automated Time Table System
- A Study On Automatic Timetable Generation
- Automatic Timetable Generation

8.2 WEBSITES

- https://www.scribd.com/doc/243446327/Project-Synopsis-for-Automatic-Timetable-Generator
- https://www.academia.edu/67485558/Automatic_Time_Table_Generator
- https://www.ijsr.net/archive/v6i1/ART20164529.pdf
- https://www.slideshare.net/bhavyag24/timetable-generator-ppt-1pptx
- https://ieeexplore.ieee.org/document/1004507
- http://www.aui.ma/sse-capstone-repository/pdf/spring-2022/Automatic%20Timetable%20Generator.pdf