

Software Requirement Specification

STUDENT NAME: NAVYA M J

ROLL NO: 7376221EC241

SEAT NO:136

PROJECT ID: 23

Project Details:

Project Name	Time Table Generation
Domain	camps
Stack	Python stack (AI)

1. Problem statement

Develop an automated system to generate optimized class timetables in a camps, ensuring no scheduling conflicts, balanced teacher workloads, and optimal classroom utilization. The system should also accommodate special requirements such as lab sessions, elective courses etc.. Additional features include attendance tracking, event management, and support for feedback and continuous improvement. This system aims to streamline timetable creation, enhance efficiency, and ensure a seamless experience for all users.

2. Introduction:

2.1 Purpose:

The purpose of this system is to streamline the timetable creation process, reduce administrative workload, and enhance the overall efficiency and satisfaction of students and staff by providing well-structured and conflict-free schedules.

2.2 Product Scope:

The product scope includes the automated generation of class timetables based on various constraints such as teacher availability, classroom capacity, and subject requirements. It ensures conflict resolution by identifying and resolving scheduling overlaps, allowing administrators to edit timetables manually if any conflicts are found. The system optimizes resource allocation for classrooms, labs, and other facilities, automatically creating schedules once resources are fed into the system. It is scalable to handle multiple departments and courses, integrates with existing campus management systems, and ensures data privacy and security.

2.3 Product Value:

Faculty and Administrators will have a better overall project management experience as a result of the process automation, which saves time. The product offers significant value by streamlining the timetable creation process, drastically reducing administrative workload, and minimizing human errors. It ensures efficient utilization of campus resources such as classrooms and labs, leading to optimal space management. By resolving scheduling conflicts and accommodating special requirements, it enhances the satisfaction of both students and teachers. The system's scalability and integration capabilities ensure it can grow with the institution and seamlessly work with existing systems.

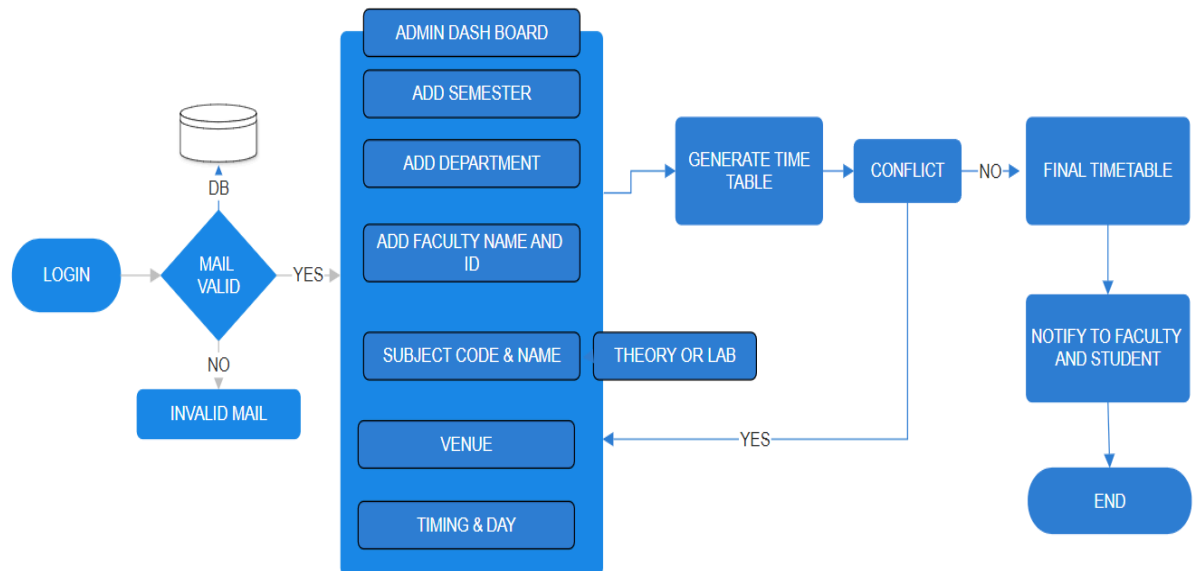
3. System overview:

3.1 User's:

1.Administrator

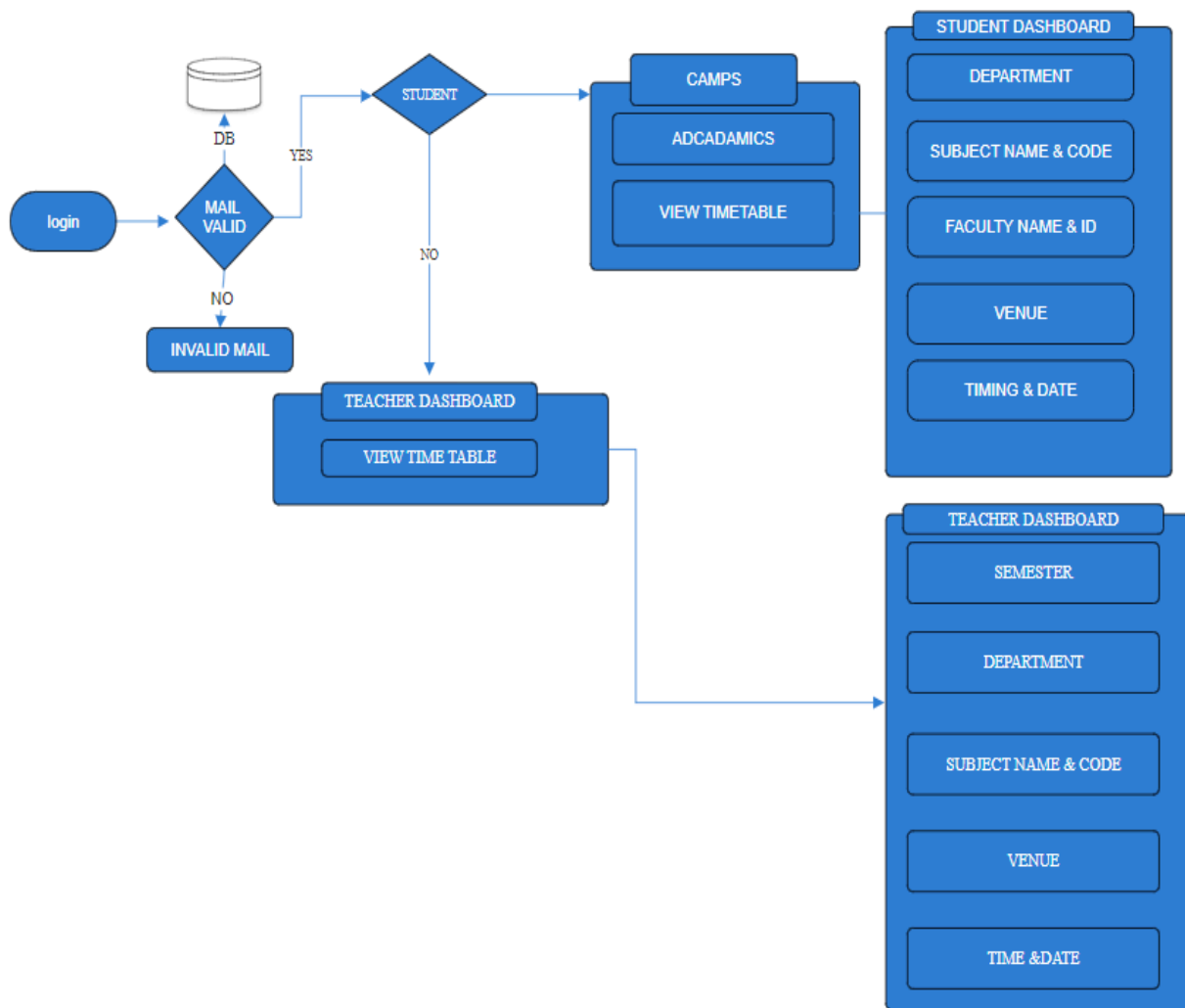
Users accounts can be managed by administrators, who can also assign roles and access levels. They enter and maintain information on classes, rooms, and faculty availability, ensuring that records are up to date. They manage conflict detection and resolution, manually adjusting schedules as needed. Administrators handle real-time updates, communicate scheduling changes, ensure integration with college systems, provide technical support, troubleshoot problems, enforce data security, and ensure

compliance with rules.



2.student and teacher:

Students and teachers can view the time table based on the data given by the administrator.



4. Stack Architecture and Infrastructure:

Front-end	HTML, CSS, JavaScript
Back-end	DIGANGO
Database	MySQL
API	RESTful services

5. Dependencies

5.1 Google Authentication

This dependency includes ensuring seamless integration with Google's authentication services, such as OAuth, to allow users to securely log in using their

Google accounts.

5.2 Proper Working and Performance of the Database:

Retrieval, storage, and management of participant, project, and Effective evaluation metric data are ensured via a high-performing database. Create a well organized, normalized database design that can include all the necessary data pieces, including project IDs, reward points, and work logs.

5.3 Proper Hosting:

This involves selecting a hosting company that can accommodate the expected demand and usage rates of the platform with enough resources, scalability, and reliability. Additionally, sufficient security measures should be provided by the hosting environment to safeguard user data and stop illegal access.

6.Functional requirements:

□ User Management:

- Admin can log in with an account and have access for generating time tables .
- Students and Teachers can log in and have access to see the time table.

7. Admin Dashboard:

Constraint Management:

- Ability to define constraints such as teacher availability, classroom availability, and subject periods.
- Option to set priority levels for different constraints.
- Feature to preview and modify constraints before generating the final timetable.

Timetable Management:

- Automated timetable generation based on the provided constraints.

- Manual adjustments to the generated timetable for any specific needs.
- Save, publish, and share the timetable with students and faculty.

8.Faculty and Student Dashboard:

User Profiles:

- Personal profiles for students and faculty with basic information and schedule.
- Option to update personal information and preferences.

Timetable Viewing:

- Access to personal timetable with class details and timings.
- Calendar view to see the timetable for the week or month.
- Downloadable timetable in PDF or Excel format.