SOFTWARE REQUIREMENT SPECIFICATION FOR APPROVAL BETWEEN ASSIGNER AND ASSIGNEE

NAME	MADHUMITHA J
ROLL NO	7376221CD122
DEPARTMENT	COMPUTER SCIENCE AND DESIGN
PROJECT ID	5
PROBLEM STATEMENT	Task creation and approval between assigner and assignee

1. INTRODUCTION

1.1. Purpose

The goal of this project is to create a more efficient method for assigners and assignees to create and approve tasks. The objective of this system is to improve task management efficiency, accountability, and communication inside an organisation.

1.2. Scope of the Project

Task Creation: Putting in place a simple procedure that assigners can use to define tasks, include deadlines and priorities, and describe them. Assigning duties to the right people and making sure they are informed and have access to all the information they need is known as task assignment. Task Approval: Creating a well-defined approval process that allows assigners to approve finished work and assignees to accept or request changes to assignments. Monitoring Progress: Ensuring accountability and transparency by offering tools to monitor the state of work from inception to completion.

2. SYSTEM OVERVIEW

2.1. Users

1.FACULTY

Regular users are the main participants in the task management process. They can be assigners, who create and delegate tasks, or assignees, who are responsible for completing tasks.

2.Admin

Admins have the highest level of access and control within the system. They are responsible for overseeing the overall task management process and ensuring that the system operates smoothly.

2.2. Features

1. Better Management and Organisation of Tasks

Clearly defined responsibilities and deadlines are ensured by the structured work assignment that faculty members are able to develop and assign.
Centralised Task Repository: Since all tasks are kept in one place, it is simple to keep track of progress and oversee several projects at once.

2. Improved Cooperation and Exchange of Information

Real-Time information: To keep everyone informed about progress and any changes, faculty members receive real-time information on task statuses.

3. Effective Time Management and Workflow

Automated Notifications & Reminders: Faculty members can minimise the likelihood of missing crucial assignments and deadlines by using automated reminders.

4. Enhanced Transparency and Accountability

Clear Accountability: Task assignments and completions are clearly documented for assigners and assignees, which improves accountability.

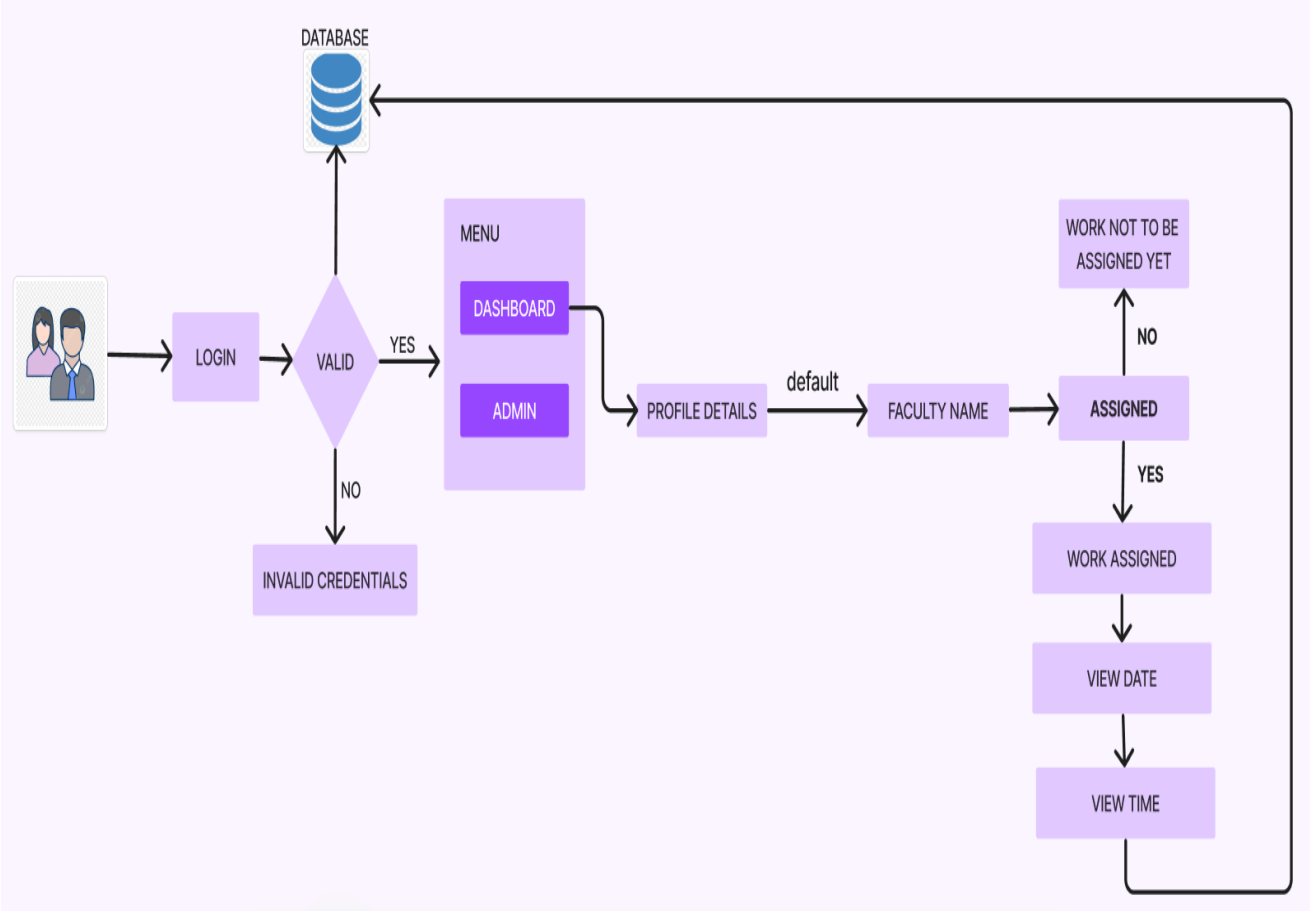
5. A Lower Burden of Administration

- Simplified Task Management: By automating repetitive processes and offering an intuitive task management interface, the system lessens the administrative load on faculty members.
- Emphasis on Core Responsibilities: Faculty members can concentrate more on their primary duties, such teaching and research, by simplifying work management.
-

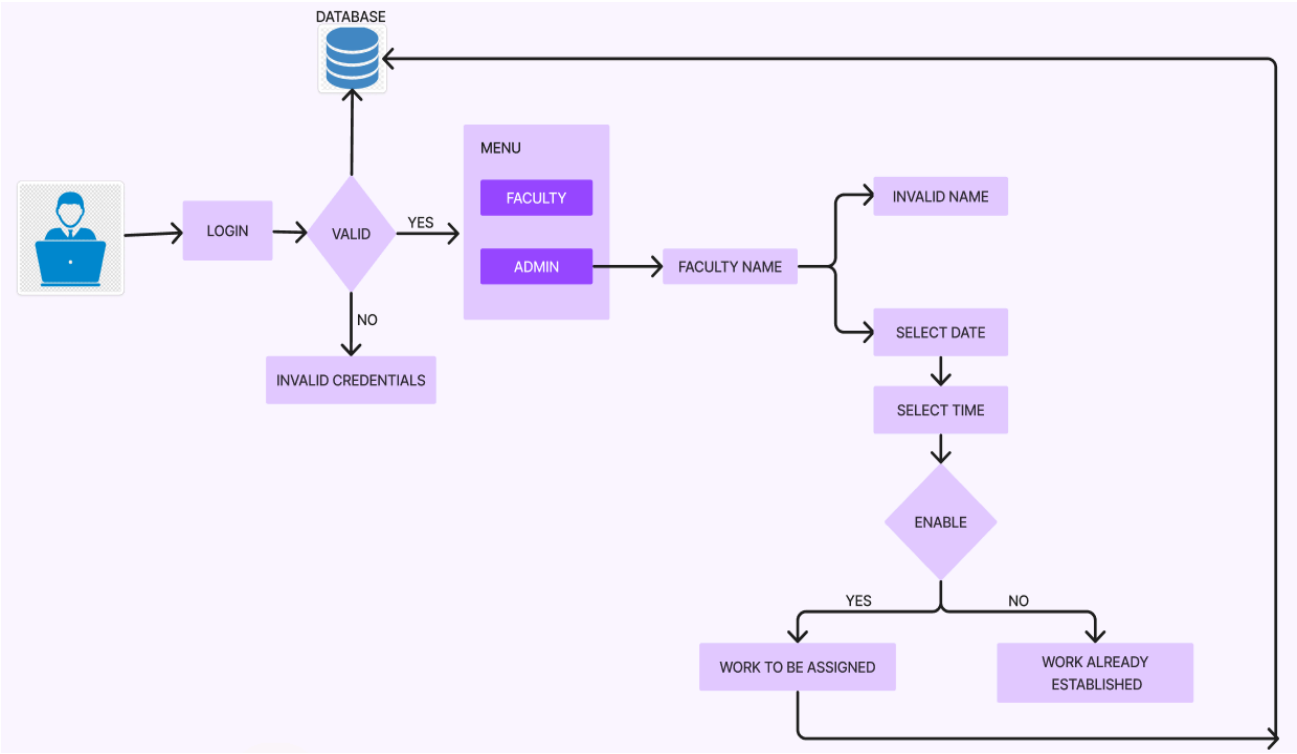
6. Constant Enhancement and Flexibility

- User Feedback Integration: Faculty members can offer input on the system's operation, allowing for ongoing development and modification to meet their changing requirements.
- Scalability: The system may develop to meet the institution's expanding needs and adapt to evolving task management specifications.

USER INTERFACE [FACULTY]



ADMIN INTERFACE



FUNCTIONAL REQUIREMENTS

User Administration

- Admins have the ability to add, modify, and remove user accounts.
- Roles (Admin or Regular User) can be assigned by admins to individual users.
- Users with regular accounts can edit their profile details.

Task Establishment

- New tasks can be created by assigners and include information such a title, description, priority, deadline, and connected files.
- Prior to being assigned, tasks can be saved as drafts.
- Templates are available for assigning recurring tasks.

Assignment of Tasks

- One or more assignees may receive tasks from assigners.
- Depending on availability, tasks can be redistributed either automatically by the system or manually the assigner.
- Task dependencies are set by assigners.

Access Control and Security

- controlling user permissions using role-based access management.
- safe methods for logging in and authenticating.
- Data encryption for communication and task information.

Messages and Reminders

- Automated alerts for approvals, deadlines, updates, and work assignments.
- Notification settings that users can customise.
- Notifications of impending deadlines.

NON-FUNCTIONAL REQUIREMENTS

Performance

Tasks and user interfaces ought to load on the system in two seconds.
Up to 1,000 concurrent users should not cause performance issues for the system.

Scalability

In order to handle more users and jobs, the system must be scalable.
The architecture need to make it simple to add new features and minimise downtime.

Reliability

Data loss should be avoided by performing regular backups of your data.

Usability

Navigating the user interface should be simple and intuitive.
It should take very little training for users to begin utilising the system efficiently.
A well-documented codebase will make upgrades and maintenance easier.

Security

While in transit and at rest, all data ought to be encrypted.
The system must abide by all applicable data protection laws (such as the CCPA and GDPR).

STACK:

FRONT END	React JS
BACK END	Java with Spring Boot
DATABASE	My SQL and Postgre SQL