PROJECT MARK CONSOLIDATION SYSTEM

NAME : NIKIL PRASANTH K S
ROLL NUMBER : 7376221SE132
SEAT NUMBER : 287
PROJECT ID : 7

1.1 Purpose

The purpose of this document is to provide a system to effectively manage project marks and update them to individual student dashboards.

1.2 Scope

The Project work mark consolidation system is intended to facilitate the entering of the students' marks through csv format and display the marks to students if there is no moderation.

1.3 Definitions, Acronyms, and Abbreviations

- **SRS**: Software Requirements Specification
- **UI**: User Interface

2. Overall Description

2.1 Product Perspective

The Project work mark consolidation system will be a standalone web application that interfaces with a database to manage marks of students.

2.2 User Classes and Characteristics

Students: Users who are students at the college.
Faculty: Users who are faculty members at the college.
Admin: To manage project marks and update them to individual student dashboards.

2.3 Operating Environment

The application will be accessed through a web browser on desktop and mobile devices.

2.4 Design and Implementation Constraints

The application will be developed using the SpringBoot stack (MySQL,, React.js, SpringBoot).

3. Specific Requirements

3.1 Functional Requirements

1. **User Registration**: Users should be able to register for an account.
2. **User Login**: Registered users should be able to log in to the application.
3. **Admin Panel**: Admin users should be able to enter student marks and edit them.
4. **Notifications**: Users should receive notifications via email or text message once their booking is approved or rejected.

3.2 Non-functional Requirements

1. **Performance**: The application should be able to handle multiple concurrent users without significant slowdowns.
2. **Security**: User data should be stored securely and protected against unauthorized access.
3. **Usability**: The user interface should be intuitive and easy to use.
4. **Reliability**: The application should be reliable and available whenever users need to access it.

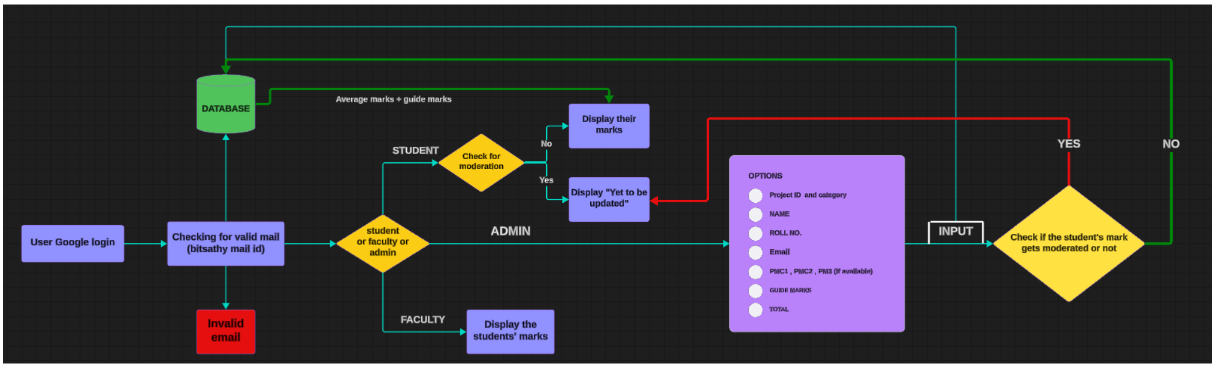
3.3 User Interfaces

- **Login Page**: Allows users to log in to their accounts.
- **Admin Panel**: Allows admin users to enter student marks and edit them

3.4 System Interfaces

- **Database**: The application will interface with a MongoDB database to store user and booking information.

4.FLOW CHART :


DATABASE
Average marks + guide marks
Display their marks
STUDENT
Check for moderation
No
Yes
Display "Yet to be updated"
User Google login
Checking for valid mail (bitsathy mail id)
student or faculty or admin
ADMIN
Invalid email
FACULTY
Display the students' marks
OPTIONS
Project ID and category
NAME
ROLL NO.
Email
PMC1 , PMC2 , PM3 (if available)
GUIDE MARKS
TOTAL
INPUT
YES
NO
Check if the student's mark gets moderated or not

5. Technology Stack

5.1 Backend

Spring boot: For building the backend of the project marks management system, providing streamlined development, auto-configuration, and easy integration with databases and web services. It enables rapid development of RESTful APIs

5.2 Frontend

- **React.js**: Used for building the user interface, providing a dynamic and responsive experience for users.

5.3 Database

- **MySQL**: Serves as the relational database management system (RDBMS) for storing and managing the project marks data within the system

6. Role of Each Component

6.1 SpringBoot

1. Simplifies the development of robust, production-ready backend applications by providing a convention-over-configuration approach and embedded server capabilities.
2. **API Development**: Create RESTful APIs for communication between the frontend and backend.

6.2 React.js

- **User Interface**: Build a dynamic and interactive user interface for booking resources and managing bookings.
- **State Management**: Manage application state and data flow using React's state and context APIs.

6.3 MySQL

- **Database Storage**: Store resource information, booking history, and user data in a flexible and scalable manner.