SOFTWARE REQUIREMENT SPECIFICATION

NAME	DHIWAKAR GK
ROLL NO	7376221CD109
SEAT NO	204
PROJECT ID	4
PROBLEM STATEMENT	BITHACK[EVENT MANAGEMENT]

1.INTRODUCTION :

The purpose of the project is to create a user-friendly online platform for organizing and participating in college hackathon events. It aims to streamline event registration, team formation, and communication among participants. By providing a centralized hub for event management, the project seeks to enhance the overall experience for students involved in hackathons.

2. SCOPE :

The project scope includes developing a platform where college students can easily register for hackathon events using their college email IDs. They can create teams, invite members, and collaborate on projects. Admin features enable event management and student oversight. Security measures include password hashing for user protection. The project aims to simplify event organization, foster collaboration, and ensure a secure and user-friendly experience for all participants.

2. System Overview:

2.1 STUDENT :

Students start this process by using the platform to sign up for the selected event. After registering, students can form teams with other players who have comparable abilities or interests. Students can also extend an invitation to friends or classmates to join their teams, which promotes teamwork and friendship. This simplified process improves the overall collegiate hackathon experience by making it simpler for students to organize into teams and collaborate on creative ideas.

2.2 ADMINS :

The platform allows the administrator to execute a number of functions. This makes it possible to manage the event listings effectively by adding, removing, reading, and changing events. The administrator can also examine comprehensive student data, which gives them the ability to manage participant information and guarantee efficient event planning. This procedure makes it possible for the administrator to effectively oversee the hackathon platform by giving them complete control over student management and event planning.

3. FEATURES :

3.1 LOGIN AND REGISTRATION :

Students must use their college email IDs only to check in to the platform. They go to the website, click the login link, and input their password and college email address. The system then authenticates these credentials by cross-referencing them with the college email database. Access to features like event listings and team formations is granted upon successful login. Upon registering, new users are required to submit personal information. With their password and college email address, users can access their account after registering. By requiring a login, users may verify their college connections and maintain platform security.

3.2 EVENT REGISTRATION :

Students only need their college email addresses to log in. They go to the website, click the login link, and input their password and college email address. The system then authenticates these credentials by cross-referencing them with the college email database. Access to features like event listings and team formations is granted upon successful login. Upon registering, new users are required to submit personal information. With their password and college email address, users can access their account after registering.

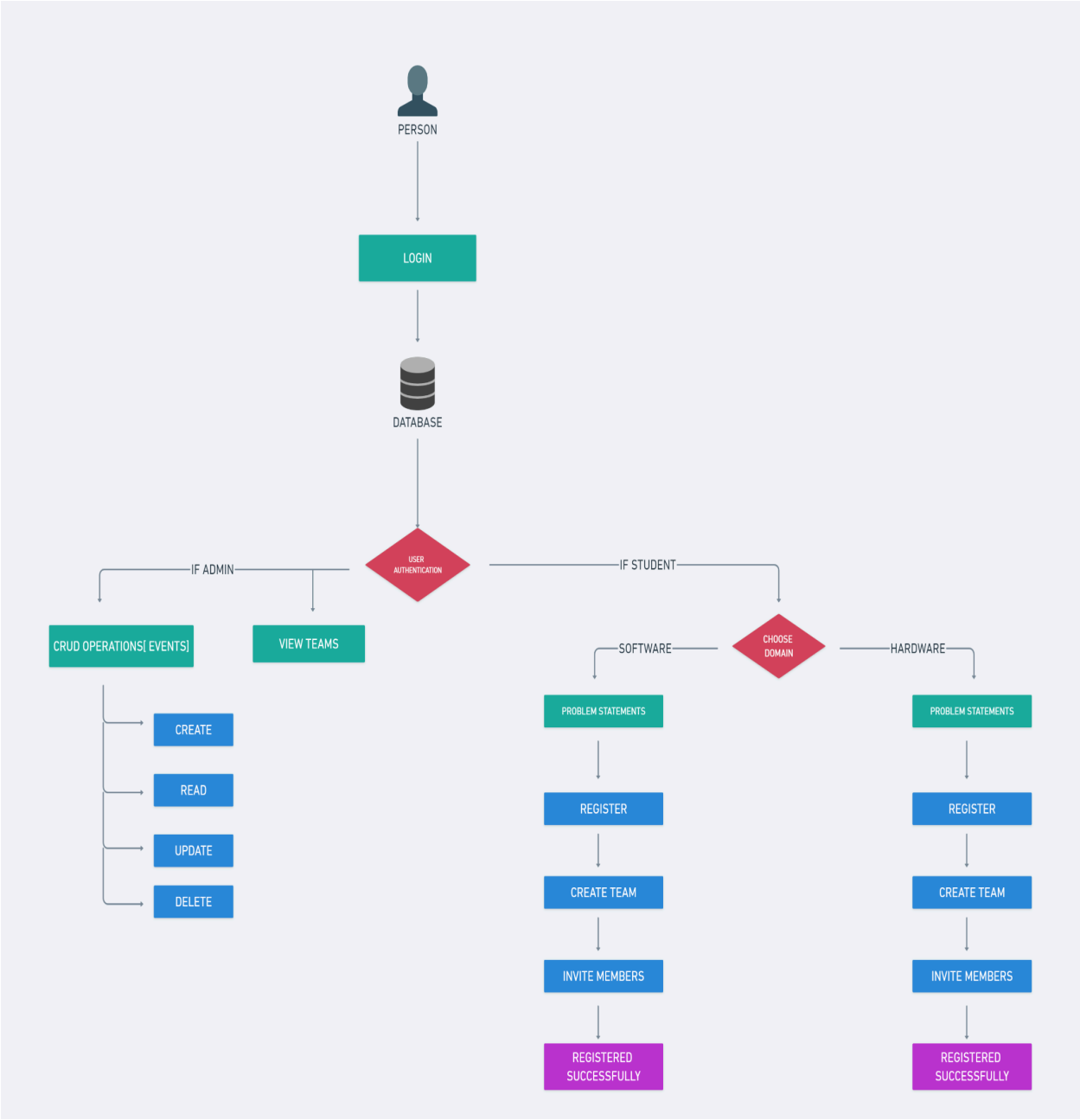
3.3 ADMIN ACCESS :

The student team details are reviewed and confirmed by the administration. After gaining access to the platform, they go to the team management area. They can see things like the names and email addresses of team members as well as possibly project information there. Teams must adhere to any rules or specifications specified for the tournament, and the administrator makes sure this happens. They might confirm team makeups, ensure teams follow tournament rules, and assess the accuracy of the information.

3.4 ADMIN'S ANALYTICS DASHBOARD :

The administrator can access the platform and navigate to the application overview section. There, they can view the number of applications sorted by category. This feature provides valuable insights into the distribution of applications across different event categories. By monitoring the number of applications in each category, the administrator can assess the level of interest and plan accordingly for event logistics and resources.

PROJECT FLOW :



4 FUNCTIONAL REQUIREMENTS :

4.1 USER MANAGEMENT :

Students use their login credentials to register and log in. They can form teams for events and extend invitations to other users once they're logged in. Students can more readily form cooperative teams thanks to this technique, which increases involvement in events.

4.2 ADMIN MANAGEMENT :

The admin has control over events, including creating, updating, and deleting them. They can also read event information. Additionally, the admin can view details of both students and teams. This functionality allows the admin to manage events effectively and stay informed about participant and team information.

4.3 EVENT REGISTRATION :

Students give their name, email address, and domain while registering for activities. They also include the name and size of the team. They can also include team members in their group. Students may easily register for events with this method, which makes it simple for them to enter their information, choose their preferred team, and work together with classmates.All things considered, this registration procedure puts an emphasis on ease of use and simplicity while encouraging successful team creation and involvement in college hackathons.

4.4 ANALYTICS DASHBOARD :

The admin can see how many people applied for each category of event. They can also check how many appointments were requested for each category. This helps the admin understand which types of events are popular and how many appointments are needed. With this information, they can plan events better and make sure there are enough resources for each category.

5 NON FUNCTIONAL REQUIREMENT :

5.1 PERFORMANCE:

It takes a few seconds for students to be able to log in. In order to provide seamless platform access, the system handles their login request during this time. In order to execute authentication and security checks and make sure that only authorized users may access the platform, a waiting period is required. Students can log in and use the platform's features and services after the waiting period has ended.

5.2 SCALABILITY :

We use Vue.js for the front end and Express for the back end to scale our platform to accommodate more users and data efficiently. The website is made user-friendly by Vue.js, and Express handles tasks like information management and user request processing. By combining these two solutions, we can make sure that our platform continues to function quickly and efficiently even when more users and data are added.

5.3 SECURITY :

To keep data safe, we use MongoDB for storing it securely. We also use a hash algorithm to protect passwords. This means that passwords are converted into a special code that's hard to decode. By doing this, even if someone gets access to the database, they can't easily see the passwords. This helps to keep user accounts and their information safe from unauthorized access.

5. STACK :

FRONTEND	Vue.js , Quasar [For UI]
BACKEND	Node.js & Express.js
DATABASE	MongoDB
RESTFUL API	AXIOS