

SOFTWARE REQUIREMENT SPECIFICATION FOR BITHACK 24- HACKATHON EVENTS FORUM

Name:Nanditha N R

Reg no:7376222CT135

Seat no:84

Project ID:4

Problem statement:BITHACKS'24 Hackathon Events Forum

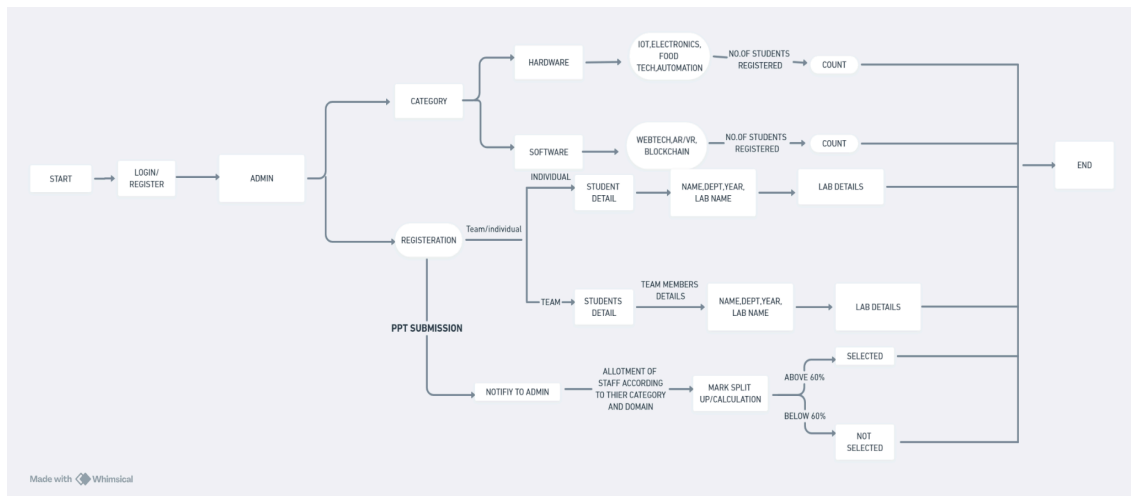
1.INTRODUCTION:

1.1. Purpose:

The purpose of this document is to present a detailed description of the BITHACK 24-Hackathon events forum. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli.

1.2. Scope of Project:

- The BITHACK 24 Hackathon Forum will serve as a central platform for hackathon organizers and participants, facilitating event management, Organizers can create, manage, and promote hackathon events through a comprehensive dashboard.Participants can register for event
- Organizers can create detailed event pages with customizable templates, including schedules, themes, rules, and judging criteria. Integrated tools to track preparation tasks, deadlines, and milestones, ensuring all event components are on schedule.



2. SYSTEM OVERVIEW:

2.1.Organizers:

Provide organizers with comprehensive event management tools to plan, organize, and oversee the hackathon efficiently. Offer features to facilitate team formation and Enable organizers to share valuable resources, guidelines, and tips with participants to enhance their hackathon experience.

2.2.Participants:

Simplify the registration process for participants to easily sign up for the hackathon. Having a user friendly interface for having a smooth experience for participants. Providing the necessary details for registration and participants.

3. FEATURES:

1. Login and registration:

Offers an easy registration process with a user-friendly interfaces. Students can register for an account or login if they are already registered.

2.Event Management:

Creation of detailed event description with their proper theme, tracks, challenges, prizes and the judging criteria. Setting up the reminder system for the event updates for both organizers and participants.

3.Profile Creation:

Creating a profile by highlighting their skills, interests, preferences and past hackathon experiences. Participants are allowed to choose their teammates according their preferences and they all have a communication access through the hackathon website itself.

4. Organizer's Analytical Dashboard:

Organizer can view the number of registrations by the event category, number of teams formed, participants detail, submissions and mainly to generate the reports for post events analysis.

4.FUNCTIONAL REQUIREMENTS:

1.Login and registration:

- Admins validate the user's credentials are matching or not.
- Organizers/Admins are responsible for managing the platform, overseeing the hackathon events and maintaining the proper databases

2.Event management:

- Organizers can create new hackathon events, specifying details such as event name, date, location (if applicable), description, and registration deadline.
- Organizers can edit event details before and after the event is published.Events are visible to all users, allowing participants to browse and register for upcoming hackathons.

3.Communication:

- Organizers can send announcements to all participants regarding event updates, deadlines, or important information.
- Participants receive email notifications for important events such as project submissions, team invitations, and event updates.

4.Admin Features:

- Admins can manage user accounts, including registration approval, account suspension, and password reset. They can moderate content posted on the platform, including forum discussions, project submissions, and event details.

NON-FUNCTIONAL REQUIREMENTS:

1.Performance:

The system must respond to the user's action immediately for the better user experience and should handle a simultaneous load without significant slowdowns.

2.Security:

User data must be encrypted during transmission and storage. Access to sensitive features must be restricted to authorized administrative users using secure authentication methods to ensure data integrity and user privacy.

3. Ease of Use:

The user interface must prioritize simplicity and clarity by providing intuitive navigation and clear error messages to assist users with input errors or system errors, promoting a positive user experience.

4. Reliability:

The system must be available throughout all days with minimal downtime. A robust backup and recovery system must be in place to prevent data loss in the event of a system failure or crash and to ensure uninterrupted service to users.

5. Scalability:

The system architecture should be designed to adapt to a growing user base and increasing amount of data over time. It should be easily scalable to add additional features and functions to meet changing requirements, ensuring adaptability and longevity.

STACK: Spring Boot stack(Java)

Front End - ReactJs,Angular

Back End - Java with Spring Boot

Database - MySql,Postgre Sql

API - OpenAPI,SOAP APIs,REST Ful API

