**ACS 560 SOFTWARE ENGINEERING**

**GROUP PROJECT: REQUIREMENTS DOCUMENT**

**Project Title:**

DevTrack: Project Management Tool

**Version Summary:**

Version 1.0 – The first release includes core functionality to create, manage, and track multiple projects and task management capabilities.

**TABLE OF CONTENTS**

1. Introduction
2. Glossary
3. User Requirements
4. System Requirements
5. System Architecture
6. System Models
7. Sequence Diagram
8. Class Diagram
9. Use Case Diagram
10. Conclusion

**INTRODUCTION**

DevTrack is a comprehensive web-based project management tool designed to streamline project workflows. Built using modern web technologies such as React for the frontend and Spring Boot for the backend, it offers a powerful and user-friendly interface for a team to use for project management. The tool provides key features such as project creation, progress tracking, and real-time updates. By integrating these functionalities into one platform, DevTrack eliminates the need for multiple tools and manual tracking, improving overall efficiency.

The primary goal of DevTrack is to provide a centralized platform for managing multiple projects, tracking progress, and ensuring accountability. It is especially useful for organizations that handle complex projects with multiple teams or departments. The tool is structured to benefit project managers by offering them the ability to assign tasks, track completion, and view project overviews in real-time. Team members can stay updated on their assigned tasks and deadlines, making it easier for them to complete their work on time. DevTrack uses a database to store critical project-related data, including user details, project descriptions, and task statuses. The system also employs JWT-based authentication to provide secure access to the platform, ensuring that sensitive information is protected. By integrating these modern technologies, DevTrack offers a scalable, secure, and efficient platform for managing any type of project, from small teams to large organizations.

**GLOSSARY**

1. **DevTrack**: The name of the project management tool.
2. **Task**: A unit of work assigned to a user within a project.
3. **Project**: A collection of related tasks with a specific objective or goal.
4. **JWT**: JSON Web Token, used for secure user authentication and authorization.
5. **React**: The frontend JavaScript library used for building the user interface.
6. **Spring Boot**: The backend framework used for creating the REST API.
7. **REST API**: An interface that allows the frontend to interact with the backend through HTTP requests.
8. **MySQL**: A relational database used to store application data such as projects, tasks, and users.
9. **Redux**: A state management tool used in the frontend to handle application state.
10. **Backlog**: A prioritized list of tasks that need to be completed within a project.
11. **Controller**: A backend component responsible for handling HTTP requests and responses.

**USER REQUIREMENTS**

1. User Registration and Authentication

* As a user, I want to register for an account and securely log in using my credentials so that I can access the system’s features and manage my projects and tasks.
* As a user, I want to receive a secure token after logging in so that I can remain authenticated during my session without re-entering my credentials.

1. Create, Update, and Delete Projects

* As a project manager, I want to create a new project by entering project details like name, description, start date, and end date so that I can organize my work and set timelines for project completion.
* As a project manager, I want to update the details of a project (such as the deadline or description) so that I can keep the project information up to date.
* As a project manager, I want to delete a project when it's no longer relevant so that I can clean up and organize my project list.

1. Task Management

* As a project manager, I want to add tasks to a project so that I can break down the project into smaller, manageable pieces of work for my team.
* As a project manager, I want to assign tasks to specific team members so that everyone knows their responsibilities and deadlines.
* As a team member, I want to update the status of my tasks (e.g., "In Progress", "Completed") so that my project manager can track my progress.
* As a project manager, I want to view the overall progress of the tasks within a project so that I can monitor the project’s health and completion rate.

1. Task Prioritization

* As a team member, I want to prioritize tasks by assigning a priority level (e.g., High, Medium, Low) so thatI can focus on the most urgent and important tasks first.
* As a project manager, I want to filter tasks by priority so that I can easily identify high-priority tasks that need immediate attention.

1. Dashboard Overview

* As a project manager, I want to view a dashboard showing all my projects and the status of tasks within those projects so that I can get a quick overview of project progress and upcoming deadlines.
* As a team member, I want to view a dashboard showing the tasks assigned to me so that I can manage my workload and stay on top of my assignments.

1. User Role Management

* As an admin, I want to assign roles such as Project Manager or Team Member to different users so that I can control who has access to certain features within the system.
* As a project manager, I want to have the ability to manage projects, assign tasks, and view reports so that I can effectively lead the project.
* As a team member, I want to only see the tasks assigned to me so that I can focus on completing my tasks without distraction.

1. Project Assignment

* As a project manager, I want to assign tasks to specific team members so that everyone on my team knows who is responsible for each task.
* As a team member, I want to receive a notification when a task is assigned to me so that I can immediately start working on it.

1. Real-Time Updates

* As a project manager, I want to see real-time updates when tasks are created, updated, or completed so that I am always aware of the current project status without needing to refresh the page.
* As a team member, I want to see real-time changes in task statuses so that I can collaborate more effectively with my team and keep track of project progress.

1. Search and Filter

* As a user, I want to search for tasks or projects based on keywords so that I can quickly find specific tasks or projects in the system.
* As a user, I want to filter tasks by due date, priority, or status so that I can focus on the most relevant tasks.

1. Notifications

* As a team member, I want to receive notifications when a new task is assigned to me or when the task’s status is updated so that I stay informed about my responsibilities.
* As a project manager, I want to receive notifications when a project’s key milestones are achieved so that I can track progress more effectively.

1. Task Comments

* As a team member, I want to add comments to tasks so that I can ask questions, provide updates, or share important details with my project manager or other team members.
* As a project manager, I want to review comments on tasks so that I can stay informed of ongoing discussions and progress without needing to ask for updates directly.

1. Secure Access

* As a user, I want to access the system over a secure HTTPS connection so that all my data and communications are encrypted and protected from unauthorized access.
* As an admin, I want to ensure that sensitive user data, such as passwords, is stored securely using encryption so that the system complies with security best practices and prevents data breaches.

**SYSTEM REQUIREMENTS**

1. Backend Framework

* As a developer, I want to use **Spring Boot 2.0** to implement the backend API so that I can build scalable, maintainable RESTful services that handle business logic, user authentication, and database operations.

1. Frontend Technology

* As a frontend developer, I want to use **ReactJS** to build a responsive and dynamic user interface so that users can interact with the system in real-time and have a smooth user experience.
* As a frontend developer, I want to use **Redux and Thunk** for state management and handling asynchronous actions so that the frontend remains performant and well-organized, even with multiple API calls.

1. Authentication and Authorization

* As a backend developer, I want to implement JWT tokens for authentication so that user sessions are securely managed, and users can access the system without needing to re-enter their credentials repeatedly.
* As a security-conscious developer, I want to ensure role-based access control is implemented with JWT tokens so that different types of users (Admin, Project Manager, Team Member) can only access the features and data they are authorized to use.

1. Database Management

* As a backend developer, I want to use a relational database such as MySQL or PostgreSQL for storing user, project, task, and comment data so that I can easily query and update data while ensuring relationships between tables are maintained.
* As a developer, I want to ensure that the database is optimized for performance and supports complex queries so that users can retrieve data quickly, even in large projects.

**SYSTEM ARCHITECTURE**

A diagram of a software application

Description automatically generated

The full-stack application architecture is split into server-side and client-side components. On the server side, the Spring Boot application handles the logic and communicates with a database through the repository layer. The application uses REST controllers to expose APIs, which are called by the service layer to interact with the repository. Authentication is handled using JWT (JSON Web Token), where tokens are validated for secure communication between the client and server. On the client side, a web browser runs React components that interact with a Redux store to manage the application's state. API calls are made to the Spring Boot backend, and the JWT token is validated during these interactions. The data flows between the backend and frontend securely through API calls, ensuring smooth communication between the two sides.

**SYSTEM MODELS**

A diagram of a project

Description automatically generatedSEQUENCE DIAGRAM

The sequence diagram above depicts a sequence of actions in a project management system where a user interacts with the frontend to log in, create projects, and manage tasks. After the user inputs login or registration details, the frontend sends the credentials to a JWT authentication service, which validates them and returns a token for further interactions. The user can create projects, assign tasks, update tasks, and comment, with these actions sent as requests to the backend API, which stores the information in the database and confirms success. Real-time task updates are handled through a WebSocket server, ensuring instant updates are displayed on the user's interface, providing a smooth and responsive experience.

A diagram of a computer

Description automatically generatedCLASS DIAGRAM

The class diagram above represents various entities and their relationships in a project management system. The AuthenticationManager manages user authentication through an HTTPSecureConnection, ensuring secure access. A User entity contains login and personal information, and is involved in tasks and projects. Users are assigned specific Roles that determine their permissions. The Project aggregates tasks, with each task having a title, description, and due date. Tasks can be filtered by a SearchFilter and are associated with TaskStatus and Priority levels. Users can leave Comments on tasks, which helps in collaboration. Notifications are handled by the WebSocketManager, which ensures real-time communication updates for the user. Each of these components works together to manage projects, assign roles, track tasks, and provide feedback and notifications.

A diagram of a project manager

Description automatically generatedUSE CASE DIAGRAM

The use case diagram for the DevTrack System above, depicts the interactions between two types of users: Developer and Project Manager, and the various actions they can perform within the system. Both users can perform tasks such as Create Project, Create Tasks, Update Tasks and Login. Additionally, both users have access to the Prioritize Tasks use case, indicating shared responsibility in managing task priorities. But only Project Manager will have the ability to delete the task from the dashboard. The diagram highlights that while developers and project managers have overlapping capabilities, they likely contribute in different roles to managing projects and tasks within the system.

In conclusion, DevTrack is a comprehensive project management platform that simplifies team collaboration, task tracking, and project execution. With features like project creation, task assignment, real-time updates, and role-based access control, it ensures that users can effectively manage their responsibilities. By leveraging ReactJS for the frontend, Spring Boot for the backend, and JWT for secure authentication, DevTrack provides a seamless and secure user experience. Its ability to scale and maintain data integrity through MySQL makes it an ideal solution for teams of any size, offering a reliable platform to monitor progress, meet deadlines, and achieve project success.

GITHUB REPOSITORY LINK –

<https://github.com/jahnavib26/DevTrack>

TEAM MEMBERS –

1. ADITHYA SEESANABILU NAGARAJ
2. JAHNAVI BOLLINENI
3. SINDUJA KUNA