PAK-AUSTRIA FACHHOCHSCHULE:
INSTITUTE OF APPLIED SCIENCES AND TECHNOLOGY

Team Members:

Safi Ullah ***(B22F0549SE031)***

Sharyar Naveed ***(B22F0782SE014)***

Muhammad Moin ***(B22F1629SE148)***

Course:

Software Construction and Development
(COMP – 370)

Program:

Software Engineering – 22 – RED

Instructor:

Dr. Nabeel Ahmed

Submitted Date:

29 – January – 2024

Title:

Project Proposal

Project Proposal

---

1. Title:

GitHub Integration for Agile Team Release Management

---

2. Problem Statement:

As Agile is the word which is used everywhere now but for us in modern software development, Agile methodology is used to tackle the problem of change and faster development of the software. Agile teams often face challenges in tracking and managing releases, tasks, and CI/CD processes across different platforms. The proposed project aims to solve this challenge by developing a tool that integrates with GitHub to provide a centralized dashboard for Agile teams, offering real-time updates on releases, task progress, and CI/CD pipeline statuses, improving overall productivity and collaboration.

---

3. Objectives:

The goal of this project is to build a system that helps Agile teams manage their software releases by integrating GitHub with a dashboard that tracks release status, tasks, and CI/CD processes. The system will provide a **backend** to manage releases and tasks, and a **frontend** to visually display data for easy management.

---

4. Features:

a. Backend Features:

- **Release Tracking:**
    - We will pull data from GitHub to track software releases. This includes information about release names, descriptions, and creation dates.
    - We will store this data in a database and create an API to manage and display it.
- **Task Management:**
    - Each release will have tasks associated with it. Tasks will have statuses like "pending," "in progress," and "completed."
    - We will build an API to manage tasks and allow updates based on progress.

- **CI/CD Status:**
    - The system will fetch information about CI/CD workflows, like whether the build of software passed or failed.
    - We will show the CI/CD status on the dashboard, helping teams monitor their workflow.
- **Authentication & Security:**
    - Only authorized users will be able to access the system. We will use secure login methods, such as GitHub authentication, to keep data safe.

b. Frontend Features:

- **Dashboard:**
    - The front-end will display a user-friendly dashboard where teams can view release status, tasks, and CI/CD progress.
    - The dashboard will update in real time to show the latest changes.
- **Task Tracker:**
    - A section will be dedicated to showing the progress of tasks linked to each release.
    - Users can easily see which tasks are pending or completed.
- **CI/CD Status:**
    - The system will show whether the CI/CD pipeline is working correctly. If there is a failure, users will be notified immediately.
- **Notifications:**
    - We will send notifications to users about important events, such as task completion or build failures.
    - Notifications can be delivered through Slack or in-app alerts or email.

---

5. Functional and Non-Functional Requirements:

a. Functional Requirements:

- **Backend API:**
  - RESTful API for release tracking, task management, and CI/CD status.
  - Secure user authentication using GitHub accounts.
- **Frontend Dashboard:**
  - A responsive dashboard to display real-time data.
  - Ability to filter and view tasks based on their status.
- **Notifications:**
  - Real-time notifications for important changes, such as CI/CD failures or task completions.

b. Non-Functional Requirements:

- **Performance:**
  - The system will be able to handle multiple teams tracking different releases and tasks at the same time.
- **Security:**
  - The system will ensure that only authorized users can access or modify data.
- **Usability:**
  - The frontend will be easy to use with minimal training.
- **Scalability:**
  - The system should be able to scale and support multiple teams using it.

---

6. Technology Stack:

- **Backend**: Spring Boot (Java) and Python
- **Frontend**: Java or Html/CSS or Python (Streamlit)
- **Database**: PostgreSQL or MongoDB
- **Authentication**: GitHub OAuth or Personal Access Tokens
- **Notifications**: Slack API or Web Notifications or SMTP for the emails

---

7. Project Timeline:

- **Phase 1**: **Backend Setup and API Development**
- **Phase 2**: **Frontend Development**
- **Phase 3**: **Integration and Testing**
- **Phase 4**: **Deployment and Review**

---

8. Outcome:

This project will help Agile teams streamline their release management process by integrating GitHub with a user-friendly dashboard. By tracking releases, tasks, and CI/CD status in one place, teams will be able to work more efficiently and stay updated on the progress of their projects. The system will be secure, easy to use, and scalable for teams of all sizes.

---

9. Note:

Some of the requirements for this project may change or evolve over time as new insights or needs arise during the development process. We are committed to adapting to those changes and ensuring that the end product will work without any problem.