Vivekanand Education Society's Institute of Technology



Department of Computer Engineering

Group No.: 15 Date:- 01/08/2024

Project Synopsis Template (2024-25) - Sem V

AgileFlow

### Dr. Rohini Temkar

Internship Incharge — CMPN

Harshita Bodwani

V.E.S. I.T [2022.harshita.bodwani@yes.ac.in](mailto:2022.harshita.bodwani@yes.ac.in)

Gazal Keshwani

V.E.S. I.T [2022.gazal.keshwani@ves.ac.in](mailto:2022.gazal.keshwani@ves.ac.in)

Khushi Jeswani

V.E.S.I.T

[2022.khushi.jeswani@ves.ac.in](mailto:2022.khushi.jeswani@ves.ac.in)

Yash Sainani

V.E.S.I.T

[2022.yash.sainani@yes.ac.in](mailto:2022.yash.sainani@yes.ac.in)

**Abstract**

AgileFlow is designed to be a versatile project management and issue-tracking system tailored for Agile methodologies. This tool will encompass key features such as issue tracking, customizable workflows, agile boards, and comprehensive reporting to streamline project management and enhance team collaboration.

**Introduction**

In the realm of project management, Agile methodologies have become essential for dynamic and iterative development processes. AgileFlow aims to replicate and enhance the functionalities of leading project management tools like Jira, focusing on Agile principles to optimize project tracking and team coordination.

# **Problem Statement**

Many existing project management tools are either too generic or come with high costs and limited customization options. There is a need for a more accessible, customizable tool that effectively supports Agile practices and provides a comprehensive solution for managing tasks and tracking progress.

# **Proposed Solution**

AgileFlow will provide a tailored solution that mirrors key functionalities of established tools while offering customization and flexibility. The system will feature:

* Issue tracking with custom statuses
* Agile boards (Scrum and Kanban)
* Workflow customization
* Real-time reporting and analytics

# **Methodology / Block Diagram**

**Registration** and **Login**

* Step 1: Users visit the Jira clone's landing page.
* Step 2: New users click on the "Sign Up" button and fill out the registration form with their details (name, email, password).
* Step 3: Registered users click on the "Login" button and enter their credentials to access their account.

**Dashboard Overview**

* Step 1: Upon successfullogin, users are redirected to the dashboard.
* Step 2: The dashboard provides an overview of active projects, assigned tasks, and recent

acdvity

**Creating a New Project**

* Step 1: Users click on the "Create Project" button.
* Step 2: A form appears where users can enter project details such as name, description, and team members.
* Step 3: Users submit the form, and the new project is created and added to their

dashboard.

**Managing Projects**

* Step 1: Users select a project from the dashboard to view detailed information.
* Step 2: Within the project, users can see the backlog, active sprints, and completed tasks.
* Step 3: Users can edit project details, add or remove team members, and manage project settings.

**Creating Tasks**

* Step 1: Users navigate to the project's backlog or sprint board.
* Step 2: Users click on the "Create Task" button.
* Step 3: A form appears where users enter task details such as title, description, assignee, priority, and due date.
* Step 4: Users submit the form, and the task is added to the selected project.

**Updating Tasks**

* Step 1: Users click on a task to view its details.
* Step 2: Users can update the task status (e.g., To Do, In Progress, Done), edit task details, add comments, and attach files.
* Step 3: Changes are saved automatically or by clicking the "Save" button.

**Commenting on Tasks**

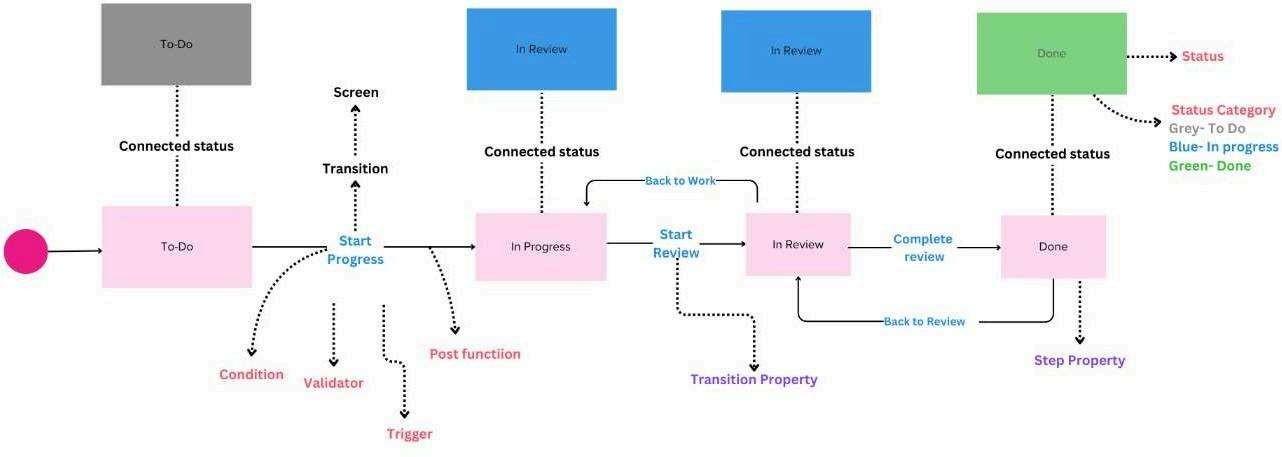
* Step 1: Users open a task to view its details.
* Step 2: In the comments section, users can add new comments or reply to existing ones.
* Step 3: Comments are saved and displayed in real-time.

**Notifications**

* Step 1: Users receive notifications for task updates, comments, and mentions.
* Step 2: Notifications are accessible from the dashboard or via email, depending on user preferences.

**Generating Reports**

* Step 1: Users navigate to the reporting section from the dashboard.
* Step 2: Users select the type of report (e.g., project status, team performance).
* Step 3: Users customize the report parameters and generate the report.



## Hardware, Software and Tools Requirements

### **Hardware Requirements**

**Development Machines:**

* + Processor: Intel i5 or higher, or equivalent AMD processor.
  + RAM: 16GB or more for smooth multitasking.
  + Storage: SSD with at least 256GB available space.
  + Display: Full HD monitor or higher for better UI development.

**Server:**

* + **Processor:** Intel Xeon or equivalent, multiple cores.
  + RAM: 32GB or more for handling multiple requests.
  + **Storage:** SSD with RAID configuration for redundancy, at least 500GB.
  + Network: High-speed internet connection, gigabit Ethernet.

### **Software Requirements**

**Operating System:**

* + **Development Machines:** Windows 10/11, macos, or a Linux distribution (Ubuntu, Fedora, etc.).
  + **Server:** Ubuntu Server LTS, CentoS, or similar.

#### Development Tools:

* + **Node.js:** Version 14 or later.
  + Next.js: Version 13 or later.
  + React: Latest stable version.
  + **React Query:** Latest stable version.
  + **Code Editor:** VS Code, WebStorm, or similar.
  + **Package** Manager: npm or yarn.
  + Database: PostgreSQL, MySQL, MongoDB, or another suitable database.
  + **Containerization:** Docker for consistent development and deployment environments.
  + **Version Control:** Git, with repositories hosted on GitHub, GitLab, or Bitbucket.

#### Libraries and Frameworks:

* + **CSS-ill-JS:** Styled-components or Emotion.
  + State Management: Redux or React Context API.
  + **Form Handling:** Formik or React Hook Form.
  + Authentication: NextAuth.js or Auth0.
  + API **Routes:** Next.js API routes or Express.js.
  + **Testing:** Jest, React Testing Library, Cypress.

#### Build and Deployment Tools:

* + **CI/CD:** GitHub Actions, GitLab CI, Jenkins.
  + **Hosting:** Vercel for Next.js (preferred), AWS, Digitalocean, or similar cloud providers.
  + **Monitoring:** Sentry for error tracking, Prometheus, and Grafana for performance monitoring.
  + Logging: Winston, Loggly, or ELK stack (Elasticsearch, Logstash, Kibana).

.

**Proposed Evaluation Measures**

O **Functionality:** Verify that all core features are correctly implemented and operational.

O **Usability:** Collect user feedback to evaluate the system’s ease of use and interface design.

O **Performance:** Evaluate the application’s performance under various load conditions.

O **Scalability:** Ensure the system **can** handle increasing numbers of users and projects.

### **Conclusion**

AgileFlow will provide a flexible and cost-effective solution for Agile project management and issue tracking. The project will contribute to more efficient project management and enhanced team productivity by delivering an adaptable tool that supports Agile methodologies.

## References

1. F. Calefato, A. Giove, F. Lanubile and M. Losavio, "A Case Study on Tool Support for Collaboration in Agile Development," *2020 ACM/IEEE 15th Internatio nal Conference on Global Software Engineering (ICGSE),* Seoul, Korea, Republic of, 2020, pp. 11-21.
2. keywords:(Automation;Co11aboration;Companies;Software;Software engineering;Slack;computer-mediated communication;Jira;collaboration too1s;agile deve1opment;action research;case study}.
3. Samridhi Sachdeva, Priyanka Paygude ,Snehal Chaudhary ,Sonali Idate 2018 International Conference On Advances in Communication and Computing Technology (ICACCT).
4. Implementing the Lean Concepts of Continuous Improvement and Flow on an Agile Software Development Project: An Industrial Case Study Balachander Swaminathan, Karuna Jain 2012 Agile India.