

**Department of Computer Engineering**

**Academic Term: First Term 2023-24**

**Class: T.E /Computer Sem – V / Software Engineering**

<b>Practical No:</b>	<b>3</b>
<b>Title:</b>	<b>Implementing Project Using Kanban Method on JIRA Tool in Software Engineering</b>
<b>Date of Performance:</b>	<b>08-08-2023</b>
<b>Roll No:</b>	<b>9528</b>
<b>Team Members:</b>	<b>Aston Castelino, Chhand Chaughule, Slayde Sequiera</b>

Sr. No	Performance Indicator	Excellent	Good	Below Average	Total Score
1	On time Completion & Submission (01)	01 (On Time )	NA	00 (Not on Time)	
2	Theory Understanding(02)	02(Correct)	NA	01 (Tried)	
3	Content Quality (03)	03(All used)	02 (Partial)	01 (rarely followed)	
4	Post Lab Questions (04)	04(done well)	3 (Partially Correct)	2(submitted)	

**Signature of the Teacher:**

## Lab Experiment 03

### Experiment Name: Implementing Project Using Kanban Method on JIRA Tool in Software Engineering

**Objective:** The objective of this lab experiment is to introduce students to the Kanban method and its implementation using the JIRA tool. Students will gain practical experience in managing a software project using Kanban principles and learn how to utilize JIRA as a project management tool to visualize workflow, manage work items, and improve team productivity.

**Introduction:** Kanban is an agile project management method that emphasizes visualizing work, limiting work in progress, and continuously improving the workflow. JIRA is a popular tool that supports Kanban practices, allowing teams to manage their tasks and activities effectively.

#### Lab Experiment Overview:

1. Introduction to Kanban: The lab session begins with an overview of the Kanban method, including the principles of visualizing work, managing flow, and making incremental improvements.
2. JIRA Tool Introduction: Students are introduced to the JIRA tool and its features for implementing Kanban. They learn to create boards, swimlanes, columns, and customize workflows.
3. Defining the Project: Students are assigned a sample software project and create a Kanban board in JIRA to visualize their workflow. They set up columns to represent different stages of their development process.
4. Creating Work Items: Students create work items (tasks, user stories, or issues) on the Kanban board, representing the work that needs to be done.
5. Managing Workflow: Students move work items through the columns on the Kanban board as they progress through their development process. They monitor work in progress limits to maintain an efficient workflow.
6. Continuous Improvement: Students conduct regular team meetings to discuss the workflow, identify bottlenecks, and make improvements to enhance their efficiency.
7. Completion and Review: At the end of the lab experiment, students review their project progress on the Kanban board. They discuss their experiences with implementing the Kanban method on JIRA and share insights on its effectiveness.
8. Conclusion and Reflection: Students reflect on their experience with Kanban and JIRA, discussing the benefits and challenges they encountered during the project. They also consider how Kanban principles can be applied to future software development projects.

**Learning Outcomes:** By the end of this lab experiment, students are expected to:

- Understand the Kanban method and its application in agile project management.
- Gain practical experience in using the JIRA tool to implement Kanban boards and workflows.
- Learn to visualize work, manage flow, and limit work in progress using Kanban principles.

- Develop team collaboration skills by continuously improving the workflow through regular team meetings.
- Appreciate the importance of visualizing and managing work items for better project management.

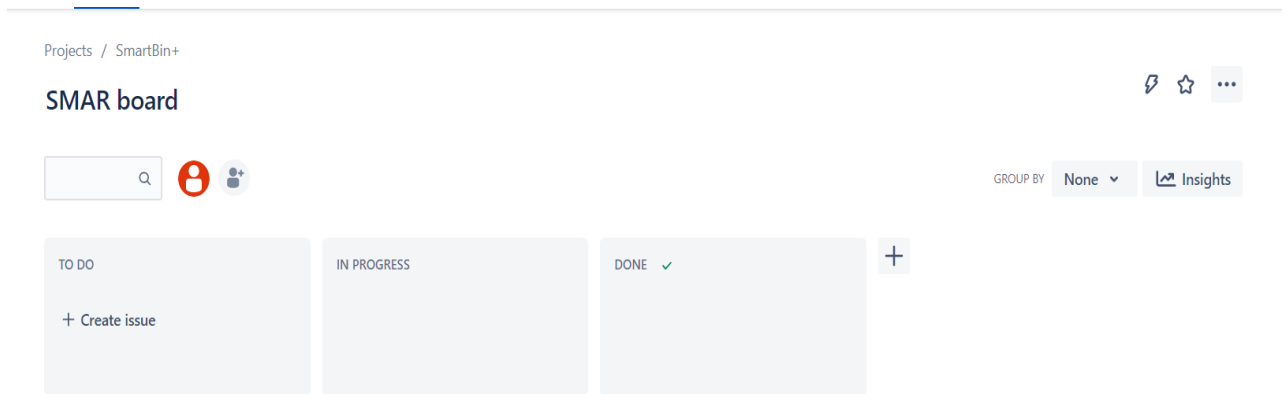
**Pre-Lab Preparations:** Before the lab session, students should familiarize themselves with the Kanban method and the basics of the JIRA tool. They should review Kanban principles, visualizing workflows, and the features of JIRA relevant to Kanban implementation.

**Materials and Resources:**

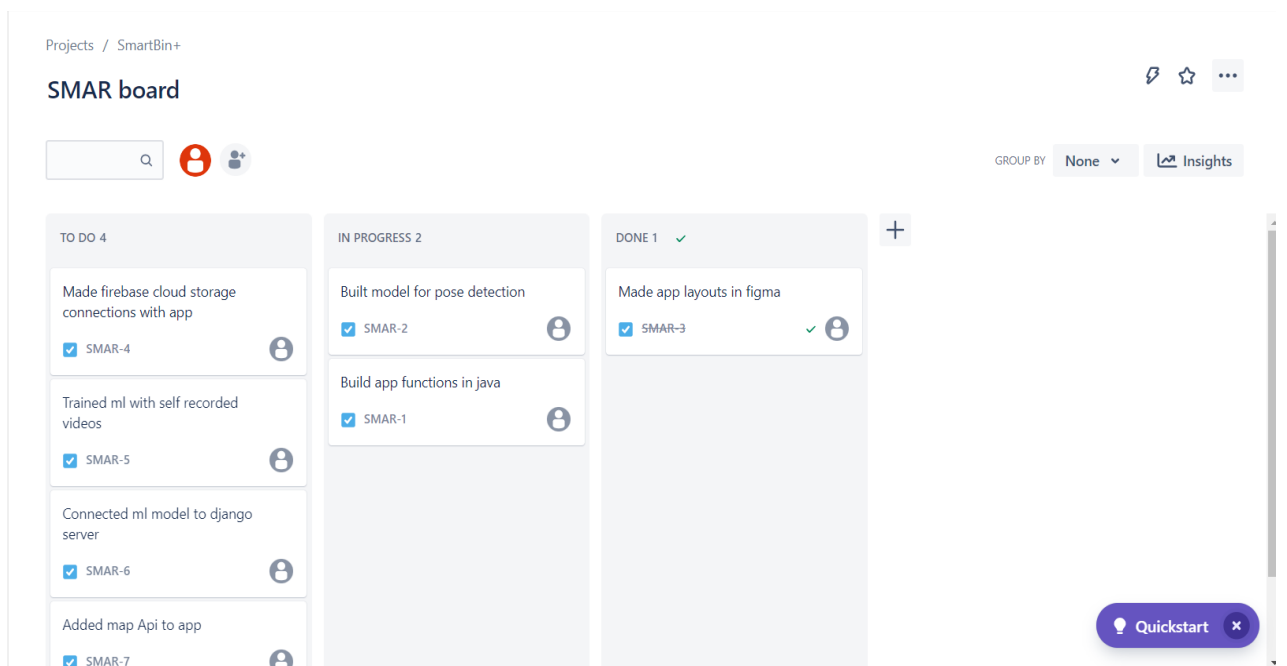
- Computers with internet access for accessing the JIRA tool
- Project brief and details for the sample software project
- Whiteboard or projector for explaining Kanban concepts

**Conclusion:** The lab experiment on implementing a project using the Kanban method on the JIRA tool provides students with practical insights into agile project management. By applying Kanban principles and utilizing JIRA's capabilities, students learn to visualize their work, manage flow efficiently, and continuously improve their development process. The hands-on experience with Kanban and JIRA fosters teamwork, collaboration, and adaptability, enabling students to effectively manage software projects with a focus on efficiency and quality. The lab experiment encourages students to adopt Kanban's lean principles, promoting a culture of continuous improvement and optimizing their workflow to deliver valuable software products.

## Creating Kanban Board



## Creating Work items



Creating Epic

SMAR-8

1

user interface

To Do

Actions

Description

create signup/login page

create rewards page

create user profile

Details

Timeline

Projects / SmartBin+

Timeline

Status category

Epic

View settings

SEP

SMAR-8 user interface

+ Create Epic

Today

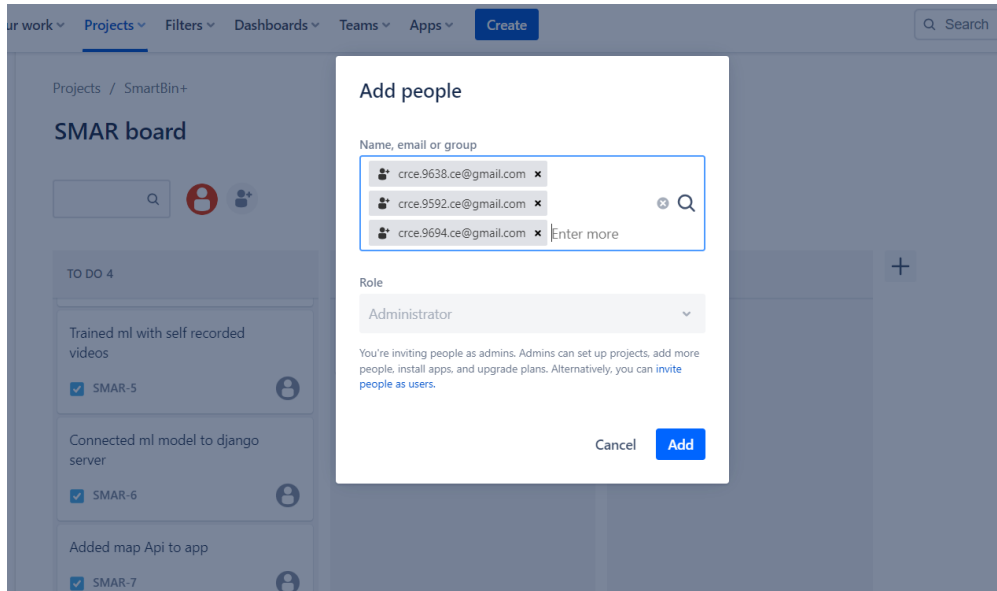
Weeks

Months

Quarters

Quickstart

## Inviting Team Members



## POSTLAB:

a) Compare and contrast the Kanban and Scrum methodologies in terms of flexibility, adaptability, and workflow management in different project scenarios.

**Ans:**

**Kanban:**

- Flexibility: Highly flexible, no fixed iterations, tasks added as capacity allows.
- Adaptability: Easily adapts to changing priorities and requirements.
- Workflow Management: Visualizes work on a Kanban board, emphasizes limiting work in progress (WIP).

**Scrum:**

- Flexibility: Less flexible due to fixed sprint durations (1-4 weeks).
- Adaptability: Adapts primarily at the end of sprints, changes during sprints discouraged.

- Workflow Management: Divides work into sprints, relies on ceremonies like sprint planning and daily stand-ups. No explicit WIP limits.

#### Project Scenarios:

##### - Kanban Scenarios:

- Support and maintenance projects
- Continuous improvement projects
- Projects with variable priorities and frequent changes

##### - Scrum Scenarios:

- New product development
- Projects with well-defined requirements
- Teams preferring structured, timeboxed work.

b) Analyse a Kanban board in JIRA and propose improvements to optimize the team's efficiency and productivity.

#### **Ans:**

Analyzing a Kanban board in JIRA and proposing improvements for team efficiency and productivity:

1. Visualize Workflow: Ensure that the board clearly represents the workflow stages from start to finish. Use custom column names that reflect the team's process and make it easier for everyone to understand the status of each task.
2. WIP Limits: Set and enforce Work In Progress (WIP) limits for each column to prevent overloading team members and maintain a steady flow of work. Adjust these limits based on the team's capacity and historical performance.
3. Prioritization: Implement a clear prioritization system, either through color-coding or tags, to help team members identify high-priority tasks easily. Ensure that the most critical work is at the top of the backlog.
4. Swimlanes: If your team handles different types of work (e.g., features, bugs, tech debt), consider using swimlanes to segregate tasks. This provides better visibility into each category of work and ensures that nothing gets lost.

5. SLAs (Service Level Agreements): Define and track Service Level Agreements for task completion times. This helps in managing customer expectations and ensuring timely delivery.

6. Automation: Integrate automation where possible, such as automatically transitioning tasks or sending notifications based on specific triggers. This reduces manual effort and ensures consistency.

7. Review and Retro Columns: Include columns for peer review and retrospectives to promote continuous improvement. After each sprint or cycle, review what went well and what needs enhancement.

8. Cycle Time Analysis: Monitor cycle times (time taken for tasks to move from one column to another) to identify bottlenecks and areas for improvement. Use this data to optimize the workflow.

9. Regular Stand-ups: Schedule daily stand-up meetings to discuss the board, identify blockers, and ensure that everyone is aligned on priorities. Encourage team members to update the board daily.

10. Feedback Loop: Encourage team members to provide feedback on the board's structure and process. Regularly review and refine the board based on this feedback to improve its effectiveness.

11. Training and Documentation: Ensure that all team members understand how to use the Kanban board effectively. Provide training if necessary, and maintain documentation on board usage.

12. Integration with Other Tools: If the team uses other tools for tasks like code reviews or testing, integrate them with JIRA to streamline the workflow further.

13. Metrics and Dashboards: Create dashboards that display key performance metrics, such as lead time, cycle time, and throughput. This helps the team track progress and make data-driven decisions.

14. Continuous Experimentation: Encourage a culture of experimentation by periodically introducing small changes to the board and workflow. Assess their impact on productivity and make adjustments accordingly.

15. Regular Board Health Checks: Periodically conduct board health checks with the team to identify issues, gather feedback, and ensure that the board remains aligned with the team's evolving needs and goals.

c) Evaluate the impact of In Progress (WIP) limits on a Kanban board and how it affects the team's throughput and cycle time.



**Ans:**

WIP (Work In Progress) limits in a Kanban board have the following impacts:

1. Improved Focus: WIP limits encourage the team to concentrate on a manageable number of tasks at a time, reducing multitasking and increasing focus.
2. Steady Flow: WIP limits maintain a steady flow of work, preventing overloading and bottlenecks, which can result in smoother and predictable delivery.
3. Reduced Cycle Time: By limiting WIP, tasks move through the workflow more efficiently, leading to reduced cycle times (time taken to complete a task).
4. Increased Throughput: Throughput (number of tasks completed in a given time) often increases as WIP limits prevent work from piling up.
5. Quick Issue Identification: WIP limits make issues and bottlenecks visible, enabling prompt issue identification and resolution.
6. Enhanced Responsiveness: Teams can respond more quickly to changes in priorities or customer demands due to reduced work in progress.
7. Better Predictability: With stabilized cycle times and throughput, teams can make more accurate predictions about when work will be completed.