

Department of Computer

Engineering Academic Term: First

Term 2023-24

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

Practical No:	3
Title:	Implementing Project Using Kanban Method on JIRA Tool in Software Engineering
Date of Performance:	10-08-23
Roll No:	9595
Team Members:	Atharva Dalvi

Rubrics for Evaluation:

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:

Department of Computer

Engineering Academic Term: First

Term 2022-23

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

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:

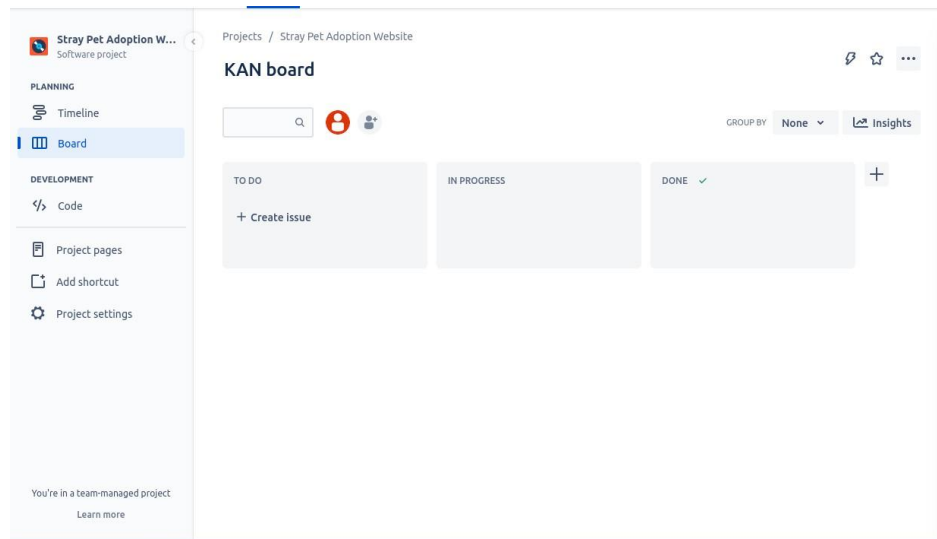
Computers with internet access for accessing the JIRA tool

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.


1. Create a KANBAN board






2. Define the workflow and assign tasks and issues

 **Stray Pet Adoption W...**
Software project

PLANNING
 Timeline
 **Board**

DEVELOPMENT
 Code




 Project pages
 Add shortcut
 Project settings


You're in a team-managed project


[Learn more](#)

Projects / Stray Pet Adoption Website



KAN board







GROUP BY: **None** 


 Insights

TO DO 3



Design the user interface
 SPA-1 



Create a landing page layout
 SPA-3 


Design and implement pet profile pages
 SPA-6 



 Create issue


IN PROGRESS 2

Develop the user login and registration
 SPA-4 

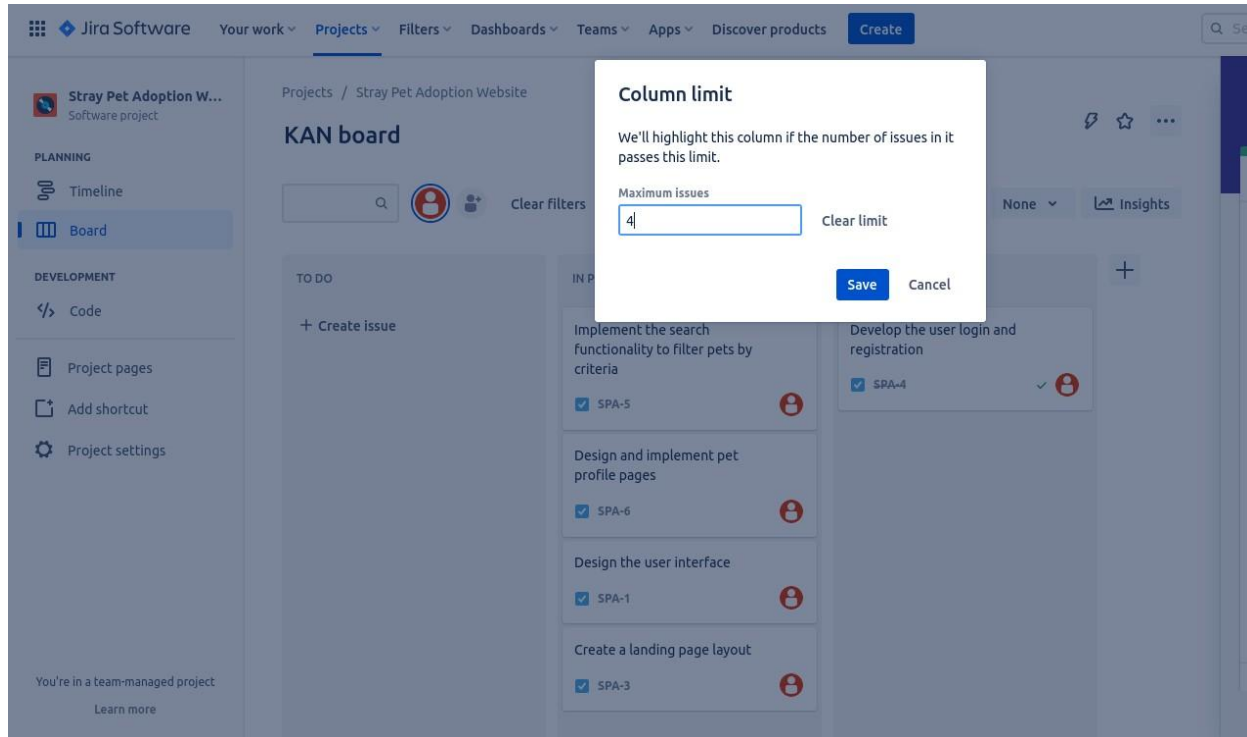
Implement the search functionality to filter pets by criteria
 SPA-5 

DONE 1 

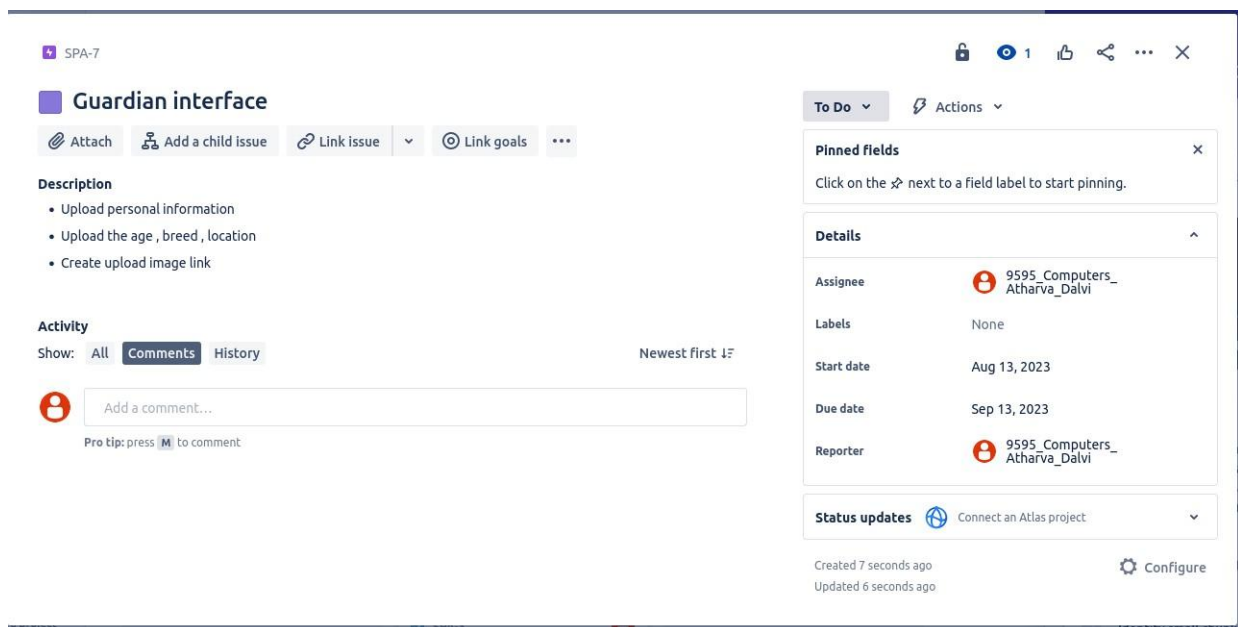
Set up the database schema for the pet information
 SPA-2 



3. Work in progress



4. Create Issues



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

1. Flexibility and Adaptability:

- Kanban: Kanban is highly flexible and adaptable. It doesn't prescribe fixed iterations or roles, allowing teams to make continuous adjustments based on changing priorities. Teams can easily accommodate new tasks or change the order of work items as needed.
- Scrum: Scrum is more structured and less flexible in comparison. It operates in fixed timeboxed iterations (sprints), and any changes to the sprint scope are typically discouraged during the sprint. Adaptability is limited to the end of each sprint during the sprint review and planning for the next sprint.

2. Workflow Management:

- Kanban: Kanban visualizes the entire workflow on a board and focuses on optimizing the flow of work items through various stages. It provides real-time visibility into work in progress (WIP) and emphasizes limiting WIP to optimize throughput and cycle time.
- Scrum: Scrum organizes work into fixed-length sprints with predefined roles (Scrum Master, Product Owner, Development Team) and ceremonies (Sprint Planning, Daily Standup, Sprint Review, Sprint Retrospective). It concentrates on delivering a potentially shippable product increment at the end of each sprint.

3. Different Project Scenarios:

- Kanban: Kanban is well-suited for projects with constantly changing priorities, support or maintenance tasks, or when the development process is already established and incremental improvements are needed. It's also suitable for teams that prefer a more fluid and less prescriptive approach.
- Scrum: Scrum is ideal for projects with clear requirements, a stable scope, and a need for regular inspection and adaptation. It is often used in product development where there is a desire for fixed-length iterations and a focus on delivering a potentially releasable product increment at the end of each sprint.

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

- **Visualize Workflow:**
Ensure that the Kanban board clearly represents the team's workflow stages, from backlog to completed items. Use swimlanes to differentiate between different types of work or teams if necessary.
- **WIP Limits:**
Verify that appropriate WIP limits are set for each stage. Ensure that these limits are based on the team's capacity and help prevent bottlenecks or overloading of tasks in any particular stage.
- **Priority and Dependencies:**
Use color-coding or tags to indicate task priorities or dependencies. This helps team members quickly identify critical items or relationships between tasks.
- **Definition of Done (DoD):**
Clearly define the criteria for when a task is considered "done." This ensures consistency and helps prevent incomplete work from progressing through the board.
- **Blocked Items:**
Implement a system for flagging and addressing blocked items. This could include a separate section on the board or a dedicated column for blocked tasks, along with a process for resolving blockages efficiently.

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

- Visualize Workflow: Ensure that the Kanban board clearly represents the team's workflow stages, from backlog to completed items. Use swimlanes to differentiate between different types of work or teams if necessary.
- WIP Limits: Verify that appropriate WIP limits are set for each stage. Ensure that these limits are based on the team's capacity and help prevent bottlenecks or overloading of tasks in any particular stage.
- Priority and Dependencies: Use color-coding or tags to indicate task priorities or dependencies. This helps team members quickly identify critical items or relationships between tasks.
- Definition of Done (DoD): Clearly define the criteria for when a task is considered "done." This ensures consistency and helps prevent incomplete work from progressing through the board.
- Blocked Items: Implement a system for flagging and addressing blocked items. This could include a separate section on the board or a dedicated column for blocked tasks, along with a process for resolving blockages efficiently.
- Cycle Time Tracking: Utilize JIRA's reporting features to track cycle times for tasks. Analyze historical data to identify areas where cycle times can be improved and adjust processes accordingly.
- Regular Review: Schedule regular board review meetings to discuss the board's effectiveness, WIP limits, and any process bottlenecks. Encourage team members to provide feedback and suggest improvements.
- Automation: Leverage automation features in JIRA to streamline repetitive tasks, such as moving items to the next stage when criteria are met or sending notifications when WIP limits are approached.

