

# JIRA Documentation – My learning

## Summary

This document details my knowledge of and practical experience with JIRA, a powerful and well-liked project management and issue tracking tool developed by Atlassian. Jira is frequently used for tasks like team collaboration, project planning, progress tracking, and bug tracking in Agile software development environments.

Through hands-on research, I learned how to:

- Write epics and user stories for major features.
- Assemble, oversee, and develop tasks and subtasks.
- Jira boards can be used to visually monitor the status of tasks.
- To control dependencies and barriers, use Linked Issues and Flagging.

## Introduction to JIRA

Teams use Jira, the top agile project management tool, to confidently plan, monitor, release, and support top-notch software. It serves as the only source of truth for your whole development lifecycle, giving independent teams the background knowledge they need to proceed swiftly while maintaining alignment with the overarching business objective. It makes it simple for teams to proceed, remain in sync, and communicate contextually, whether it is being used to manage straightforward projects or to support your DevOps procedures.

## Key Concepts I learned

To understand Jira workflows effectively, I implemented a real-world sample flow focused on **User login system** within a team-managed project.

1. **JIRA boards**: An agile board is a tool to help teams in organizing, visualizing, and managing their work if agile is an iterative approach to software development and project management. A selection of work items is shown in columns on the Jira board, each of which represents a step in your team's workflow for completing tasks. Every Jira project comes with a board by default, which gives the team a common view of all unfinished, ongoing, and completed work.
2. **Work Item**: Work items, formerly known as issues, are used by teams in Jira to keep track of bugs and specific tasks that need to be finished. A work item may represent a project task, helpdesk ticket, leave request form, etc., depending on how your team uses Jira. Individual work items, such as large features, user requirements, and software bugs, are usually represented by work items in Jira.

## PRI board

Q Search

Epics: 1 | Labels: | Type: 1 | Clear filters

GROUP BY: None | Insights | View settings

TO DO 2 OF 18

- Set up role-based access for logged-in users  
USER LOGIN SYSTEM  
Roles Security Access-Control  
PRI-8 KS
- Implement Forgot Password functionality.  
USER LOGIN SYSTEM  
Forgot\_Password Authentication  
Security  
PRI-5

IN PROGRESS 2 OF 18

- Handle login-related error messages.  
USER LOGIN SYSTEM  
Error-Handling Login  
PRI-6 RS
- Implement session timeout after 15 minutes of inactivity  
USER LOGIN SYSTEM  
Timeout Security Session  
PRI-7 RS

Implement login verification logic and connect to Auth API  
USER LOGIN SYSTEM  
Backend Authentication Login  
Security  
PRI-4

DONE 1 OF 13

- Design and implement the login page UI  
USER LOGIN SYSTEM  
Frontend Login  
PRI-3 KS

+ Create

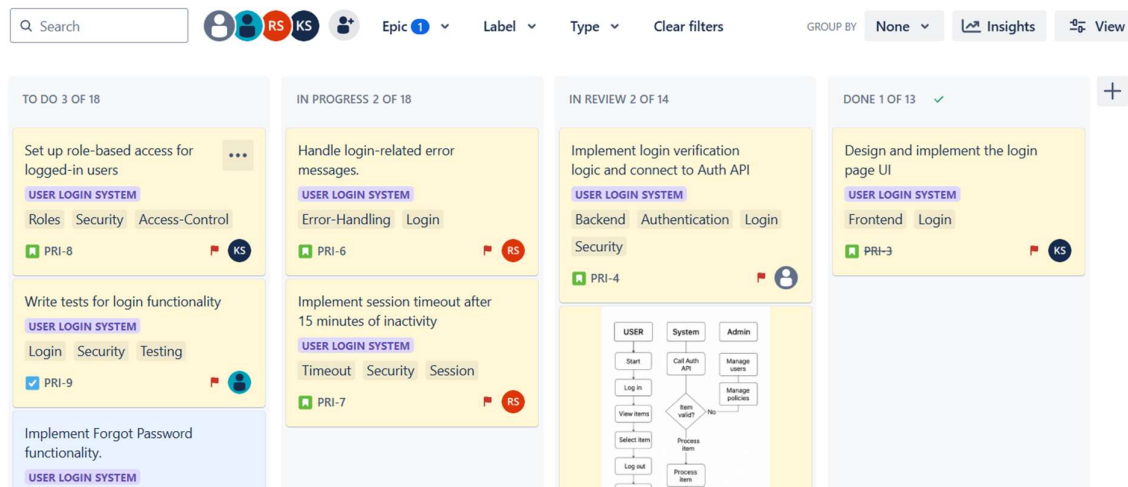
Legend: ☒ Story ☐ Task

3. **Bug**: Represents a problem that needs to be fixed.
4. **Epic**: Represents a larger body of work. Epics are often represented as a collection of multiple work items.
  - **Epic Name**: User Login System
5. **Story**: Represents a requirement expressed from the perspective of the user.

## User Stories Added Under the Epic

- **User Login**
- **User Signup**
- **Password Reset**
- **Session Management**
- **Role-Based Access Control**

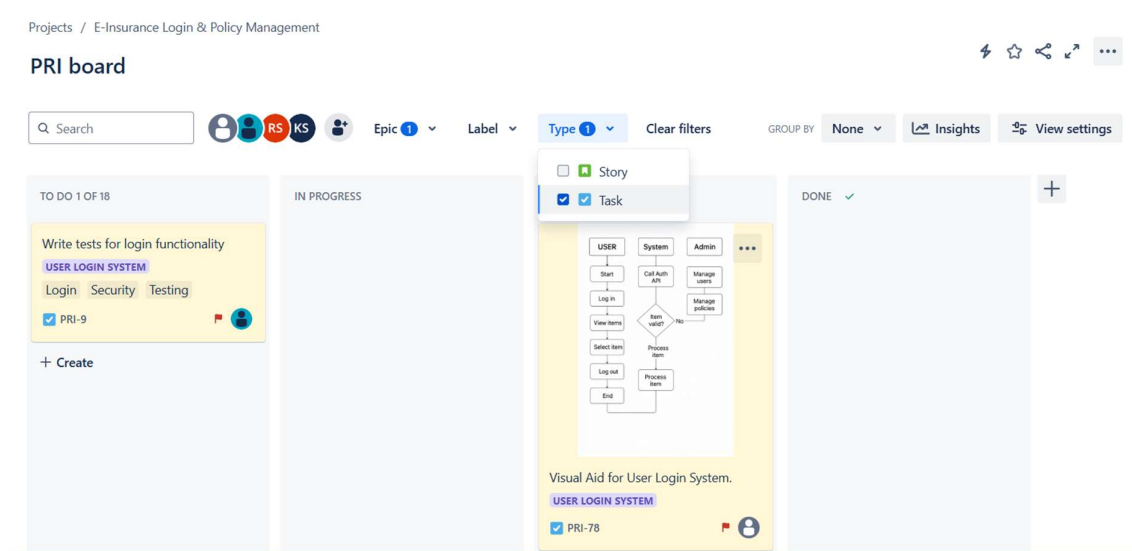
## PRI board



6. **Task:** Represents a task that needs to be done. Tasks are used as ‘catch-all’s’ and when the work cannot be accurately represented by the other work types.
7. **Sub-task:** A sub-task can be made for each type of work, representing a more detailed breakdown of the work needed to finish a typical work item.

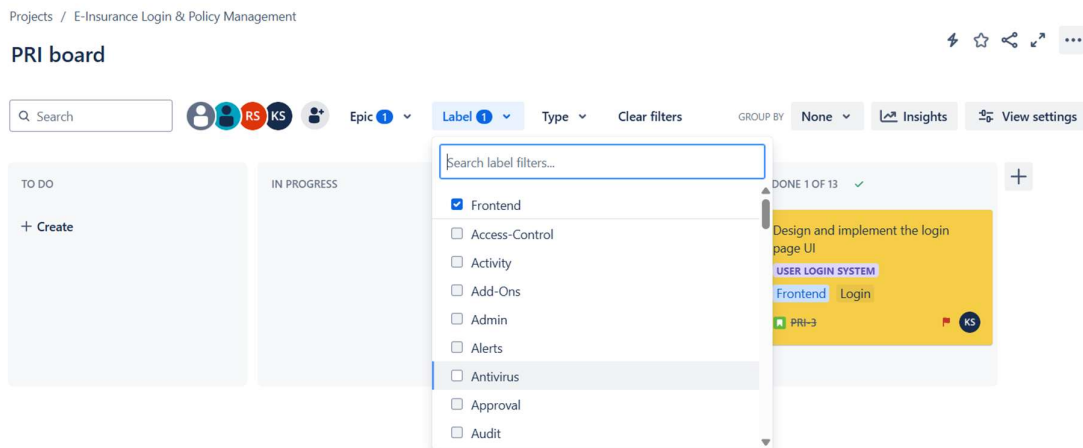
Example: For “User Login”

- **Task:** Backend API Integration
  - **Sub-tasks:**
    - Input validation logic
    - Generate and return JWT tokens
    - Connect with database for user credentials
    - API documentation for frontend consumption
- **Task:** Frontend UI Design
  - **Sub-tasks:**
    - Create login form (Figma)
    - Integrate with Auth API
    - Show error messages and validation



## 8. Labels Used

- login, security, backend, authentication, UI, testing



## 9. Issue Assignments

- Issues were assigned to relevant team members based on skill sets (e.g., UI → frontend developer, JWT handling → backend).

## 10. Workflow Transitions Practiced

Tracked progress through these stages:

- To Do → In Progress → In Review → Done

### *View of the Jira Board*

Jira's board view facilitates visual task management and stage-by-stage progress tracking. Usually, it has columns such as:

**To Do:** I haven't started any newly created tasks yet.

**In Progress:** The things that are being worked on right now.

**Code Review/QA (optional):** Work that needs to be approved or tested.

**Done:** Tasks have been finished.

#### *Qualities:*

Tasks can be moved between columns using a drag-and-drop interface.

High-priority or flagged issues are indicated by color-coded indicators.

Instant access to task statuses, progress blockers, and who is doing what.

### *Handling Dependencies with Associated Problems*

I discovered how to associate problems with the following kinds of relationships:

**Blocks:** This problem stops another from happening.

**Is Blocked By:** Until another problem is fixed, this one cannot continue.

**Relates To:** A general link that conveys significance or a connection.

By using these connections, teams can manage dependencies and expedite resolution procedures.

### *Flagging Impediments*

If a task encounters an issue or needs clarification, I can Flag it.

Benefits of flagging:

- Managers and team members are visually alerted.
- On the board, flags are easily visible (usually with a red marker or symbol).
- promotes the prompt removal of obstacles.

### *What I Found Helpful*

#### **Feature**

#### **How It Helped**

Epics & Stories

Broke down complex features into manageable tasks.

| Feature          | How It Helped   |
|------------------|---|
| Boards           | Gave a clear visual of task progress and team workload. |
| Labels & Filters | Helped organize work and retrieve tasks quickly.        |
| Issue Linking    | Maintained context and tracked task dependencies.       |
| Status Updates   | Kept the team informed of real-time progress.           |
| Flagging         | Clearly marked impediments that needed attention.       |

### **Conclusion:**

My understanding of collaborative project tracking, task prioritization, and Agile workflows has greatly increased since learning Jira. I now have greater self-assurance in my ability to plan projects, recognize dependencies, and use Jira's features to facilitate team communication and delivery.