JIRA Research Report

Feasibility Study and Implementation Guide
Project G22-2025FP (TZQI)

Jira Research Team

January 5, 2026

Abstract

This report is a feasibility study on using Jira Software for project management. Our team (G22-2025FP) has been using Redmine, but to test Jira we have two members assigned in a Jira research group to test Jira and see if it is better than Redmine.

We migrated some of our project data from Redmine into Jira and tried out the main features like Timeline views, Backlog management, Sprint planning, the board, and the built-in Pages for documentation. The Jira Research Team (Ghosh Deb Kumar and Shrestha Anjal) did this study separately from the main project work.

This guide walks through how we set everything up, with screenshots from our actual TZQI(code of our project which randomly give by Jira) project board. By the end, we found that Jira does have some real advantages over Redmine, especially when it comes to visualizing timelines, keeping documentation in one place, and working in sprints.

Project Team Members

Name	Role(s)
FUJITA Ryusei	Project Manager (PM)
DAKE Parth	Project Leader (PL), UML
JONATHAN Setiawan	Technical Developer
NICHOLAS Gallen Efendi	Technical Developer
YAMAMOTO Yuto	Technical Developer
STOLBOVOY Dennis Victor	Wiki Lead, Technical Developer
NAKAMURA Yukiya	Wiki Editor
YAMANE Leon	Wiki Editor
KIM Byeolha	Documentation Lead (Doxygen)
ANANNA Taslima Hossain	Documentation Contributor (Doxygen)
GHOSH Deb Kumar	Jira Research Team
SHRESTHA Anjal	Jira Research Team

Contents

List of Figures

1 Introduction

1.1 Project Overview

This report looks at whether Jira Software could be a good fit for managing our G22-2025FP project. Right now we use Redmine, which works fine for basic issue tracking, but honestly it gets messy when things start piling up. Everything shows up as one big list, and it is hard to tell what is urgent versus what can wait. We wanted to see if Jira's features would help us stay more organized, especially since our project covers a lot of different areas like UML diagrams, developer setup, wiki pages, and code documentation.

Our project runs from December 11–25, 2025 (basically a 2-week sprint). During this time we need to get through creating UML diagrams, setting up the developer environment, configuring workflows, building out wiki pages, setting up Doxygen for code docs, and then going back to revise the UML stuff.

1.2 Current System Limitations (Redmine)

The main issue with Redmine is that everything sits in one long list. Urgent bugs, future ideas, tasks in progress they all look the same. You cannot quickly see what someone is working on or if a task is stuck waiting for something else. The Gantt chart exists but you have to manually type in all the dates, and it does not really update on its own.

1.3 Jira Advantages

Jira, by contrast, offers specialized views:

- **Backlog** for planning future work
- **Board** for daily execution with drag-and-drop
- **Timeline** for visual roadmap with automatic bars
- **Pages** for integrated documentation
- **Reports** for burndown charts and velocity tracking

1.4 Research Scope and Team

The two of us (Ghosh Deb Kumar and Shrestha Anjal) took on this research task. We basically moved everything from our Redmine database into Jira to see how it handles real project data. Here is what we migrated:

- **One Epic**: G22-2025-Final Project (TZQI-13)
- **Six Main Tasks**:
  - TZQI-2: Feature #1489: UML
  - TZQI-3: Feature #1491: Developer
  - TZQI-4: Feature #1494: Jira
  - TZQI-5: Feature #1496: Wiki
  - TZQI-6: Feature #1495: Doxygen
  - TZQI-7: Support #1515: UML revise
- **Five Sub-tasks** under Wiki (TZQI-8 through TZQI-12):

  - Project Details
  - Group Members
  - UML Diagrams
  - Code
  - Results

We tested out all the main screens and features to see how Jira handles this kind of workload. We assigned all 13 items to ourselves, set dates between December 11, 2025 and January 5, 2026, and linked everything to the main Epic so we could track it all together.

The report starts with a section explaining what each Jira screen does and why it matters. After that, we go through the actual steps we took to set everything up, with screenshots from our real project board.

2 Jira Screens and Concepts Explained

Before jumping into the practical steps, let us go over what each Jira screen actually does. This section breaks down the main views you will use and explains them in plain terms.

2.1 The Timeline Screen (Roadmap)

2.1.1 What is it?

The Timeline (sometimes called Roadmap) is basically a Gantt chart. Instead of reading dates as plain text, you actually see bars stretching across a calendar. It makes it way easier to understand when things are happening.

2.1.2 What appears on this screen?

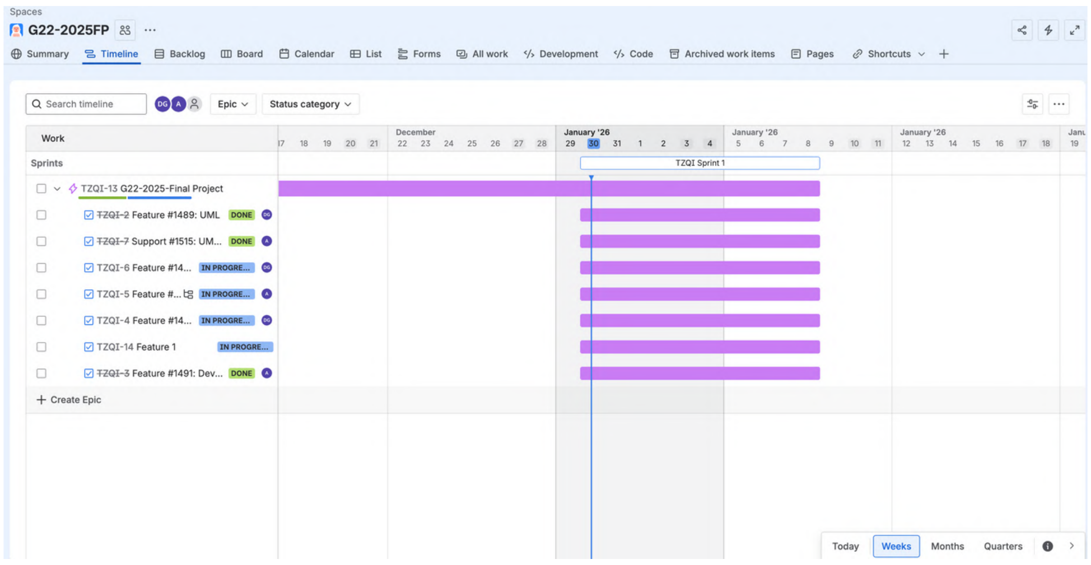
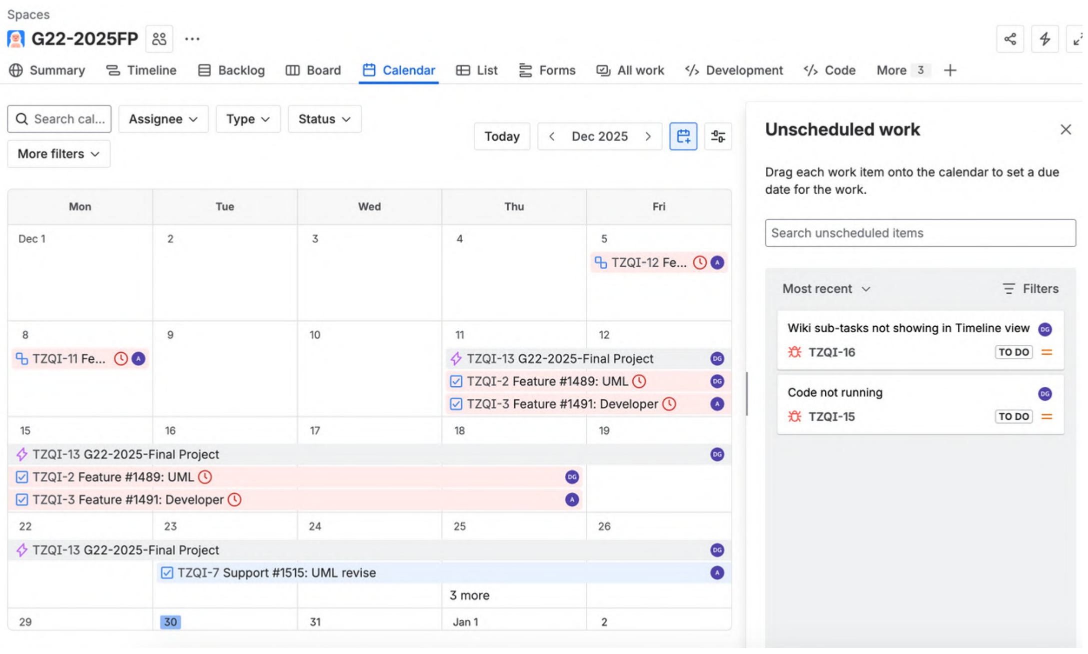

Spaces
G22-2025FP
Summary
Timeline
Backlog
Board
Calendar
List
Forms
All work
Development
Code
Archived work items
Pages
Shortcuts
Search timeline
Epic
Status category
Work
December
January '26
Sprints
TZQI Sprint 1
TZQI-13 G22-2025-Final Project
~~TZQI-2~~ Feature #1489: UML DONE
~~TZQI-7~~ Support #1515: UM... DONE
TZQI-6 Feature #14... IN PROGRE...
TZQI-5 Feature #... IN PROGRE...
TZQI-4 Feature #14... IN PROGRE...
TZQI-14 Feature 1 IN PROGRE...
~~TZQI-3~~ Feature #1491: Dev... DONE
Create Epic
Today
Weeks
Months
Quarters

Figure 1: Timeline view showing Epic and task bars across the project schedule

EPIC BAR (long one): Our Epic TZQI-13 "G22-2025-Final Project" shows up as a long purple bar going from December 11–25, 2025. You can see the whole project duration right away.

TASK BARS (short ones under the long epic): Each task (TZQI-2 through TZQI-7) gets its own bar. Where the bar sits on the calendar shows the start and due dates, and how long the bar is shows the duration. For example, TZQI-2 (UML) appears as a bar from Dec 11–15, so you know it is a 5-day task.



Mon	Tue	Wed	Thu	Fri
Dec 1	2	3	4	5 TZQI-12 Fe...
8 TZQI-11 Fe...	9	10	11 TZQI-13 G22-2025-Final Project; TZQI-2 Feature #1489: UML; TZQI-3 Feature #1491: Developer	12
15 TZQI-13 G22-2025-Final Project; TZQI-2 Feature #1489: UML; TZQI-3 Feature #1491: Developer	16	17	18	19
22 TZQI-13 G22-2025-Final Project	23 TZQI-7 Support #1515: UML revise	24	25 3 more	26
29	30	31	Jan 1	2

Unscheduled work

Drag each work item onto the calendar to set a due date for the work.

Search unscheduled items

Most recent Filters

Wiki sub-tasks not showing in Timeline view
TZQI-16 TO DO

Code not running
TZQI-15 TO DO

Figure 2: Calendar axis with zoom controls for different time-scale views

CALENDAR AXIS: Horizontal timeline showing dates, with zoom capability: day view, week view, month view, and today marker showing current date.

2.1.3 Why is this better than Redmine?

In Redmine, you just see text like "Due Date: 2025-12-15". In Jira, you actually see the bar ending on Dec 15. At a glance you can tell:

- Which tasks are running at the same time
- Which tasks come one after another
- Whether we will finish on time
- If there are any gaps where nothing is scheduled

2.1.4 Critical Feature for G22-2025FP

All 6 main tasks show up as bars on the timeline. The 5 sub-tasks under Wiki (TZQI-8 to TZQI-12) can be expanded to show their own bars within the Wiki task timeframe.

IMPORTANT: Timeline bars only show up if you set BOTH the start date AND the due date. When we first migrated our data, we saw tasks listed on the left side but no bars on the chart. Took us a while to figure out we had to go back and add dates to every task. That was a lesson learned.

2.2 The Planning Engine: Backlog and Sprint Management

This is basically the control center of your project. Here is where you organize all your work and decide what to do now versus what to save for later.

2.2.1 Concept: Backlog vs. Sprint (Storage vs. Focus)

One of the biggest differences between Redmine and Jira is how they handle task lists. In Redmine, everything urgent bugs, future ideas, current work all sits in one giant list. Jira splits things into two buckets:

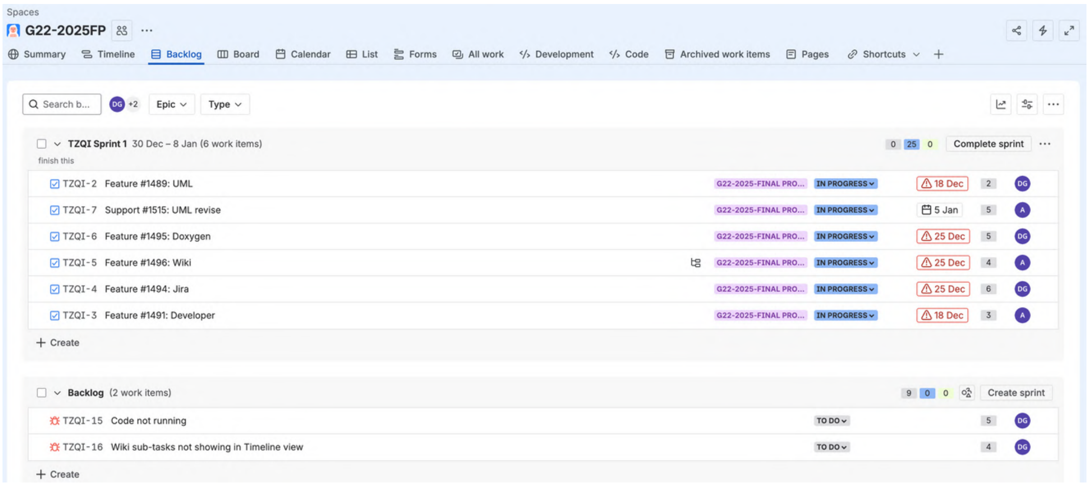
The Backlog (Storage): This is your master list with everything in the project. Tasks here are not active yet. It is just a place to keep track of ideas so you do not forget them.

The Sprint (Focus): This is your action list for a fixed period (usually 2 weeks for us). Tasks here are what you are actually working on right now. The idea is to keep the team focused on a small set of tasks instead of being overwhelmed by 50 things at once.

Why split things up? If you only had a Backlog (like Redmine), looking at a huge list gets overwhelming. If you only had a Sprint, you would lose track of future ideas. You need both: store all your ideas in the Backlog, pick the top priorities, and move them into the Sprint.

2.2.2 The Interface: How the Screen Looks

The Backlog screen is split into two parts.



TZQI Sprint 1 30 Dec – 8 Jan (6 work items) 0 25 0

finish this

TZQI-2	Feature #1489: UML	G22-2025-FINAL PRO...	IN PROGRESS	18 Dec	2	DG
TZQI-7	Support #1515: UML revise	G22-2025-FINAL PRO...	IN PROGRESS	5 Jan	5	A
TZQI-6	Feature #1495: Doxygen	G22-2025-FINAL PRO...	IN PROGRESS	25 Dec	5	DG
TZQI-5	Feature #1496: Wiki	G22-2025-FINAL PRO...	IN PROGRESS	25 Dec	4	A
TZQI-4	Feature #1494: Jira	G22-2025-FINAL PRO...	IN PROGRESS	25 Dec	6	DG
TZQI-3	Feature #1491: Developer	G22-2025-FINAL PRO...	IN PROGRESS	18 Dec	3	A

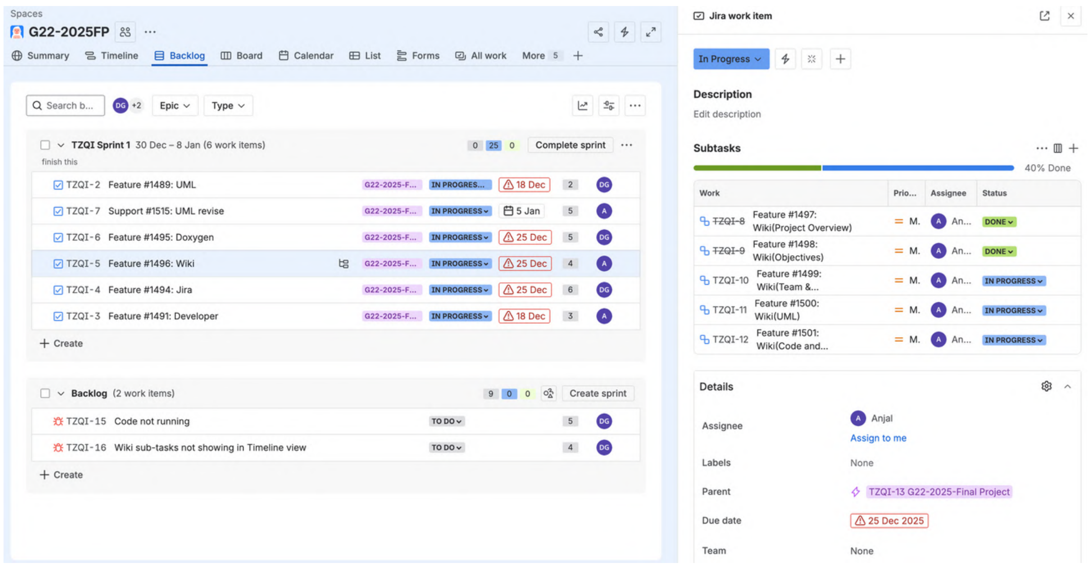
Backlog (2 work items) 9 0 0

TZQI-15	Code not running	TO DO	5	DG
TZQI-16	Wiki sub-tasks not showing in Timeline view	TO DO	4	DG

Figure 3: Backlog screen showing the Sprint section (top, active) and Backlog section (bottom, future work)

SPRINT SECTION (Top): This is your "focus" zone. You will see a box labeled "TZQI Sprint 1" at the top containing whatever you have committed to for the current cycle. For our project, this shows the 6 main tasks plus 5 sub-tasks. It also displays the sprint dates (December 11–25, 2025) and total story points.

BACKLOG SECTION (Bottom): This is your "storage" zone. Shows a list of future work that is not part of the current sprint. For us, this had 3 items including a couple of bugs (TZQI-15 and TZQI-16) that we are saving for later.



Spaces
G22-2025FP

Summary | Timeline | Backlog | Board | Calendar | List | Forms | All work | More 5

Search b... | DG +2 | Epic | Type

TZQI Sprint 1 30 Dec – 8 Jan (6 work items) 0 25 0 Complete sprint
finish this

TZQI-2	Feature #1489: UML	G22-2025-F...	IN PROGRES...	18 Dec	2	DG
TZQI-7	Support #1515: UML revise	G22-2025-F...	IN PROGRESS	5 Jan	5	A
TZQI-6	Feature #1495: Doxygen	G22-2025-F...	IN PROGRESS	25 Dec	5	DG
TZQI-5	Feature #1496: Wiki	G22-2025-F...	IN PROGRESS	25 Dec	4	A
TZQI-4	Feature #1494: Jira	G22-2025-F...	IN PROGRESS	25 Dec	6	DG
TZQI-3	Feature #1491: Developer	G22-2025-F...	IN PROGRESS	18 Dec	3	A

+ Create

Backlog (2 work items) 9 0 0 Create sprint

TZQI-15	Code not running	TO DO	5	DG
TZQI-16	Wiki sub-tasks not showing in Timeline view	TO DO	4	DG

+ Create

Jira work item
In Progress

Description
Edit description

Subtasks
40% Done

Work		Prio...	Assignee	Status
~~TZQI-8~~	Feature #1497: Wiki(Project Overview)	M.	An...	DONE
~~TZQI-9~~	Feature #1498: Wiki(Objectives)	M.	An...	DONE
TZQI-10	Feature #1499: Wiki(Team &...	M.	An...	IN PROGRESS
TZQI-11	Feature #1500: Wiki(UML)	M.	An...	IN PROGRESS
TZQI-12	Feature #1501: Wiki(Code and...	M.	An...	IN PROGRESS

Details

Assignee	Anjal Assign to me
Labels	None
Parent	TZQI-13 G22-2025-Final Project
Due date	25 Dec 2025
Team	None

Figure 4: Issue rows in the Backlog showing keys, types, assignees, status, and story points

ISSUE DETAILS (Row View): Each row displays:

- Issue Key: Unique IDs like TZQI-2, TZQI-3
- Icon: Visual indicators for task types (Story, Bug, Sub-task)
- Summary: The task title (e.g., “Feature #1489: UML”)
- Assignee: An avatar icon showing who is responsible
- Status: Current state of the issue
- Story Points: A number estimating complexity
- Epic Link: A label showing which module this belongs to

2.2.3 Comparison: Why is this better than Redmine?

Redmine just shows one big list with everything mixed together. Hard to tell what is urgent versus what is planned for next month.

In Jira:

- **Separation**: Sprint items are at the TOP (stuff you are doing now), Backlog items are at the BOTTOM (stuff for later)
- **Capacity Planning**: Jira automatically adds up all the Story Points in your sprint (ours showed 30 points). If you are taking on too much, you will see it right away. Redmine does not do this at all
- **Flexibility**: Just drag and drop to move items between sections

2.2.4 Critical Feature for G22-2025FP (Sub-tasks)

A specific advantage found for our project is the visual hierarchy. The sub-tasks (TZQI-8 to TZQI-12) appear indented directly under their parent task TZQI-5 (Wiki). Jira allows you to collapse or expand this parent task. This keeps the view clean while still maintaining the full breakdown structure, solving the clutter issue we faced in Redmine.

2.2.5 Workflow Summary

- **Creation**: New tasks are created and land in the Backlog Section (Bottom)
- **Planning**: We drag tasks UP into the Sprint Section (Top)
- **Activation**: We click "Start Sprint" to lock the scope
- **Execution**: The Sprint items move to the Active Board
- **Collection**: During the sprint, new ideas land in the Backlog section, waiting for the next planning cycle

2.3 The Active Board Screen (Execution Area)

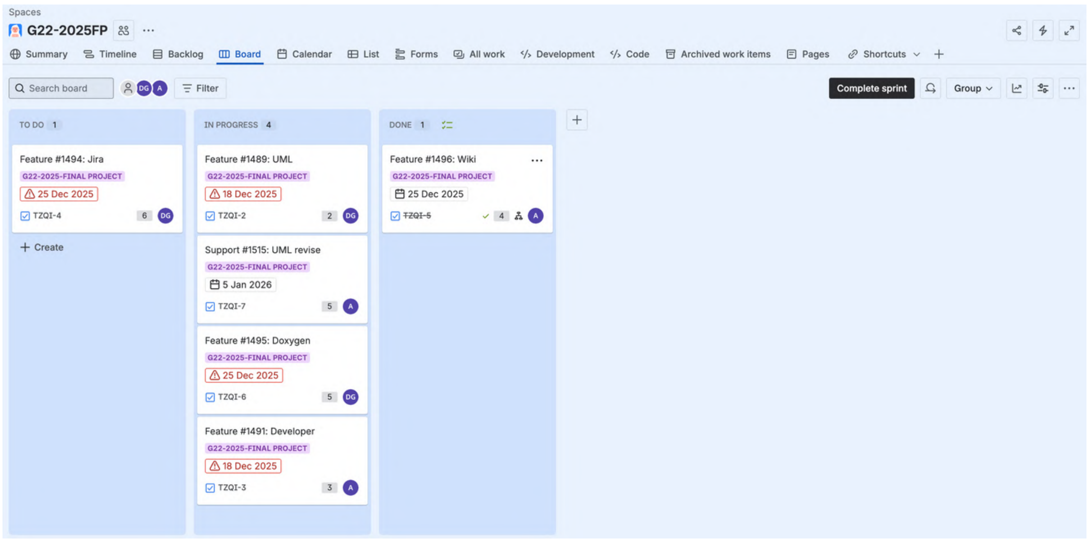
2.3.1 What is it?

The Board is where the actual day-to-day work happens. Once you start a sprint, all your committed tasks show up here as cards in columns. You will probably check this screen several times a day to see how things are going and move tasks along.

2.3.2 What appears on this screen?

The Board has THREE COLUMNS:

- **TO DO (Left)**: Tasks that have not been started yet. Everything lands here when the sprint begins
- **IN PROGRESS (Middle)**: Tasks someone is actively working on right now
- **DONE (Right)**: Finished tasks



TO DO 1

Feature #1494: Jira
G22-2025-FINAL PROJECT
25 Dec 2025
TZQI-4 6 DG

IN PROGRESS 4

Feature #1489: UML
G22-2025-FINAL PROJECT
18 Dec 2025
TZQI-2 2 DG

Support #1515: UML revise
G22-2025-FINAL PROJECT
5 Jan 2026
TZQI-7 5 A

Feature #1495: Doxygen
G22-2025-FINAL PROJECT
25 Dec 2025
TZQI-6 5 DG

Feature #1491: Developer
G22-2025-FINAL PROJECT
18 Dec 2025
TZQI-3 3 A

DONE 1

Feature #1496: Wiki
G22-2025-FINAL PROJECT
25 Dec 2025
~~TZQI-5~~ 4 A

Figure 5: Board view showing TO DO, IN PROGRESS, and DONE columns for visual task management

2.3.3 Why is this better than Redmine?

- **Redmine**: Click on task, find the status dropdown, select new status, save, wait for page to reload. Takes forever for something so simple
- **Jira**: Just grab the card and drag it to another column. Done in a second

2.4 Additional Views

Besides Timeline, Backlog, and Board, Jira has a few other views that come in handy.

2.4.1 The List View (Excel-Style)

List view shows all your issues in a table format, kind of like Excel or how Redmine looks by default. Good for when you need to see everything at once or make bulk changes.

COLUMNS (Customizable):

- Key: TZQI-2, TZQI-3, TZQI-4...
- Type: Icon showing Task/Bug/Sub-task
- Summary: Full task title
- Assignee: Name/avatar
- Status: To Do, In Progress, Done
- Priority: High, Medium, Low
- Due Date: Calendar date
- Story Points: Number
- Epic Link: G22-2025-Final Project

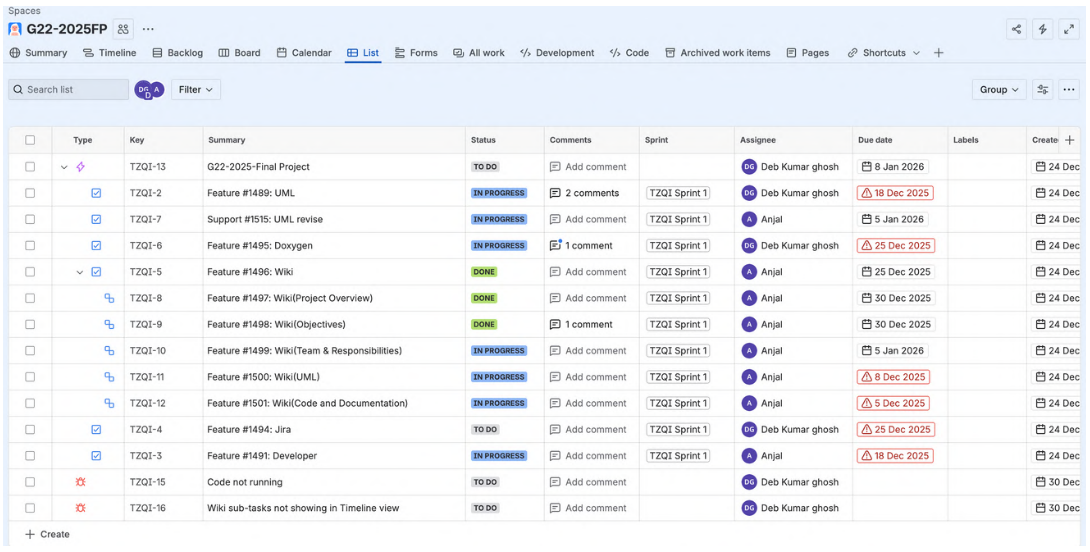
ROWS:

- Each row = one issue
- For G22-2025FP: Shows all 6 main tasks + 5 sub-tasks = 11 rows
- Can be filtered, sorted, grouped
- Inline editing available (click cell to edit)

When to use List view?

1. **Bulk Operations**: Select multiple issues, change assignee for 10 tasks at once, update due dates in bulk, add labels to multiple items
2. **Exporting Data**: Export to CSV/Excel for external reporting, university submission reports, management status updates
3. **Filtering and Search**: Filter by Assignee, Status, Priority, Epic, Sprint; quick search by text; save custom filters
4. **Comparison**: Compare due dates across all tasks, see which tasks are overdue (red dates), identify unassigned work

For G22-2025FP: List view shows all work items in sortable columns. We can quickly see that all items are assigned to members, all are linked to Epic TZQI-13, and dates span December 11–25.


Spaces
G22-2025FP

Type	Key	Summary	Status	Comments	Sprint	Assignee	Due date	Labels	Create
⚡	TZQI-13	G22-2025-Final Project	TO DO	Add comment		Deb Kumar ghosh	8 Jan 2026		24 Dec
☑	TZQI-2	Feature #1489: UML	IN PROGRESS	2 comments	TZQI Sprint 1	Deb Kumar ghosh	⚠ 18 Dec 2025		24 Dec
☑	TZQI-7	Support #1515: UML revise	IN PROGRESS	Add comment	TZQI Sprint 1	Anjal	5 Jan 2026		24 Dec
☑	TZQI-6	Feature #1495: Doxygen	IN PROGRESS	1 comment	TZQI Sprint 1	Deb Kumar ghosh	⚠ 25 Dec 2025		24 Dec
☑	TZQI-5	Feature #1496: Wiki	DONE	Add comment	TZQI Sprint 1	Anjal	25 Dec 2025		24 Dec
⧉	TZQI-8	Feature #1497: Wiki(Project Overview)	DONE	Add comment	TZQI Sprint 1	Anjal	30 Dec 2025		24 Dec
⧉	TZQI-9	Feature #1498: Wiki(Objectives)	DONE	1 comment	TZQI Sprint 1	Anjal	30 Dec 2025		24 Dec
⧉	TZQI-10	Feature #1499: Wiki(Team & Responsibilities)	IN PROGRESS	Add comment	TZQI Sprint 1	Anjal	5 Jan 2026		24 Dec
⧉	TZQI-11	Feature #1500: Wiki(UML)	IN PROGRESS	Add comment	TZQI Sprint 1	Anjal	⚠ 8 Dec 2025		24 Dec
⧉	TZQI-12	Feature #1501: Wiki(Code and Documentation)	IN PROGRESS	Add comment	TZQI Sprint 1	Anjal	⚠ 5 Dec 2025		24 Dec
☑	TZQI-4	Feature #1494: Jira	TO DO	Add comment	TZQI Sprint 1	Deb Kumar ghosh	⚠ 25 Dec 2025		24 Dec
☑	TZQI-3	Feature #1491: Developer	IN PROGRESS	Add comment	TZQI Sprint 1	Anjal	⚠ 18 Dec 2025		24 Dec
🐞	TZQI-15	Code not running	TO DO	Add comment		Deb Kumar ghosh			30 Dec
🐞	TZQI-16	Wiki sub-tasks not showing in Timeline view	TO DO	Add comment		Deb Kumar ghosh			30 Dec

+ Create

Figure 6: List view showing all issues in spreadsheet format with sortable columns and filtering options

2.4.2 The Pages View (Documentation/Wiki)

Pages (used to be called Confluence) is Jira's built-in documentation system. While issues track what needs to get done, Pages is where you explain the why and how. This is where you put project docs, meeting notes, technical write-ups, and so on.

PAGE TREE (Left Sidebar):

- Hierarchical structure showing parent/child pages
- For G22-2025FP: "G22-2025FP Home" is the parent
- 5 child pages nested underneath:
  - Project Details
  - Group Members
  - UML Diagrams
  - Code
  - Results

PAGE CONTENT (Main Area):

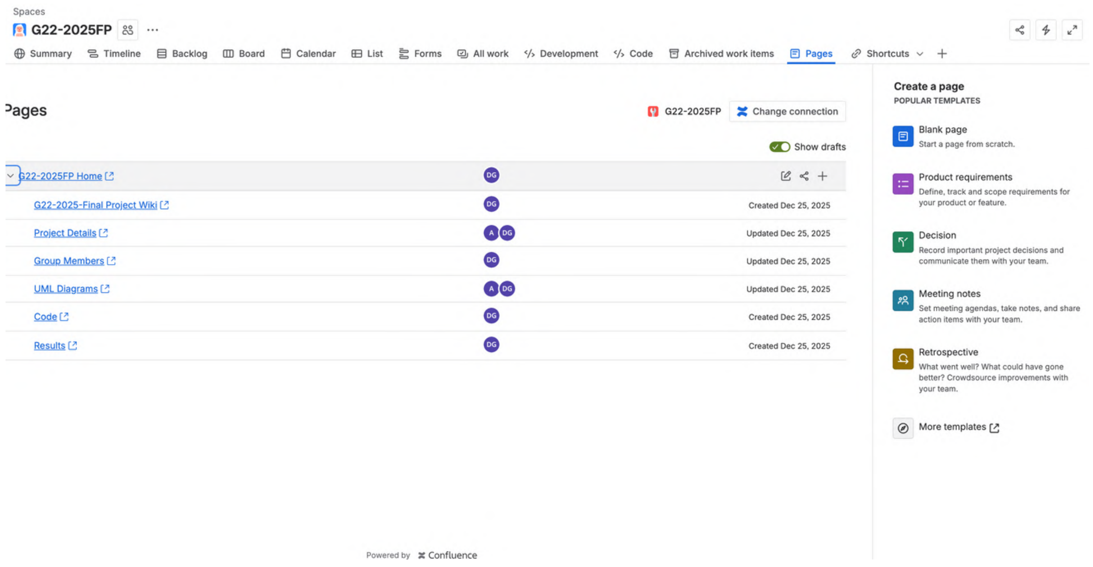
- Rich text editor (like Word)
- Formatting: Bold, italic, headings, lists, tables
- Code blocks with syntax highlighting
- Embedded images and diagrams
- Links to Jira issues (type "TZQI-" to link)
- Attachments (PDFs, images, files)
- Comments section for team discussion

2.4.3 Why is this better than Redmine Wiki?

INTEGRATION: In Redmine, the Wiki is completely separate and has no real connection to your issues. In Jira Pages, you can type "TZQI-5" and it automatically creates a clickable link. Hover over it to see issue details, click to open it.

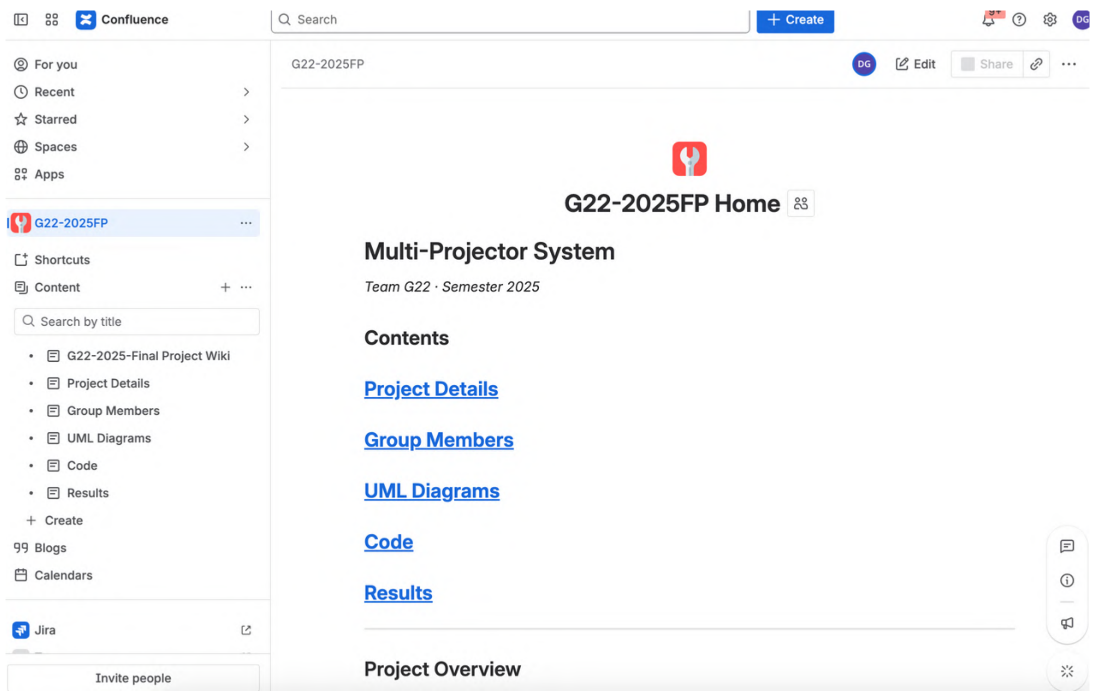
COLLABORATION: Redmine only shows who edited the page last. Jira Pages lets multiple people edit at the same time (like Google Docs), you can @mention teammates, and leave inline comments.

VERSION CONTROL: Both keep page history. But Jira shows you exactly who changed what with a visual comparison, and you can restore old versions with one click.


Spaces
G22-2025FP
Summary
Timeline
Backlog
Board
Calendar
List
Forms
All work
Development
Code
Archived work items
Pages
Shortcuts
Pages
G22-2025FP
Change connection
Show drafts
Create a page
POPULAR TEMPLATES
Blank page
Start a page from scratch.
Product requirements
Define, track and scope requirements for your product or feature.
Decision
Record important project decisions and communicate them with your team.
Meeting notes
Set meeting agendas, take notes, and share action items with your team.
Retrospective
What went well? What could have gone better? Crowdsource improvements with your team.
More templates
Powered by Confluence

G22-2025FP Home	DG	
G22-2025-Final Project Wiki	DG	Created Dec 25, 2025
Project Details	A DG	Updated Dec 25, 2025
Group Members	DG	Updated Dec 25, 2025
UML Diagrams	A DG	Updated Dec 25, 2025
Code	DG	Created Dec 25, 2025
Results	DG	Created Dec 25, 2025

Figure 7: Pages view showing hierarchical page tree in the left sidebar



G22-2025FP Home

Multi-Projector System

Team G22 · Semester 2025

Contents

Project Details

Group Members

UML Diagrams

Code

Results

Project Overview

Figure 8: Pages content area with rich text editor, formatting options, and collaboration features

3 Step-by-Step Implementation Guide

Now for the practical part. This section walks through exactly what we did to set up Jira and migrate our data from Redmine. Screenshots are from our actual TZQI project.

3.1 Phase 1: Environment Initialization

Before moving any tasks over, we had to set up the workspace first. This meant picking the right project template for our needs.

3.1.1 Step 1: Create the Project Space

We logged into Jira and found the "Create Project" button at the top.

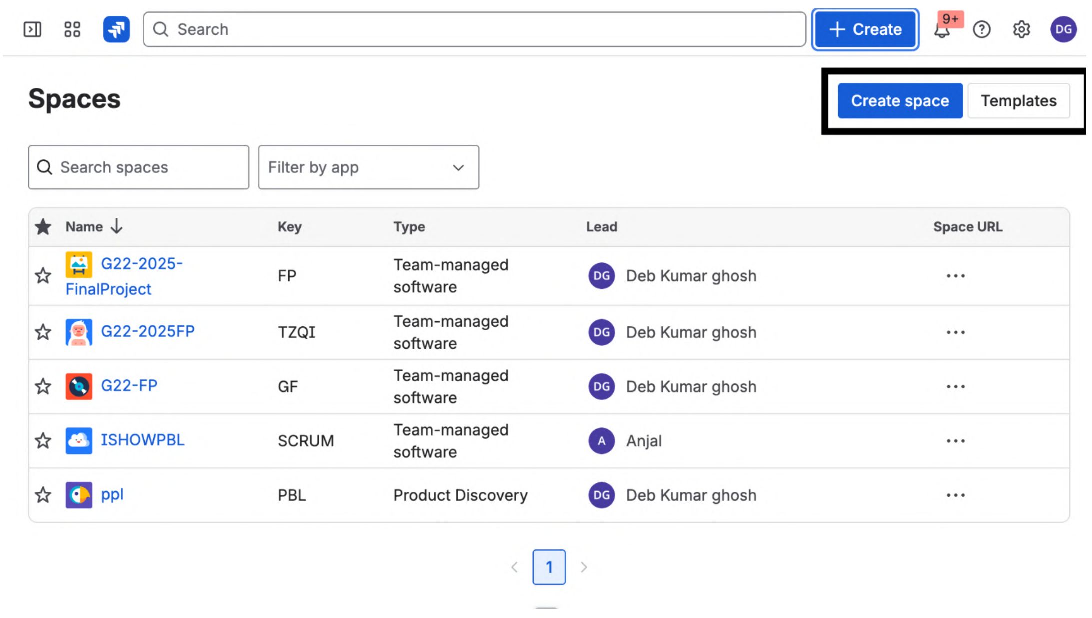
The screenshot shows the Jira Spaces dashboard. That blue "Create space" button in the top right is where you start. On the right side you can see the Templates panel we picked "Scrum" since we wanted to work in sprints.

Template Selection:

- Clicked "Create space"
- Got two options: Kanban or Scrum
- We went with Scrum
- Why? Our project has a hard deadline, so we needed the Sprint feature to break work into fixed 2-week chunks (December 11–25). Kanban is more of a continuous flow thing and does not have time-boxing built in

Project Details:

- Name: Entered our project name
- Key: Jira gave us "TZQI" automatically, so all our issues got numbered TZQI-1, TZQI-2, etc.
- Type: Team-managed software



Name	Key	Type	Lead	Space URL
G22-2025-FinalProject	FP	Team-managed software	Deb Kumar ghosh	···
G22-2025FP	TZQI	Team-managed software	Deb Kumar ghosh	···
G22-FP	GF	Team-managed software	Deb Kumar ghosh	···
ISHOWPBL	SCRUM	Team-managed software	Anjal	···
ppl	PBL	Product Discovery	Deb Kumar ghosh	···

Figure 9: Jira Spaces dashboard showing "Create space" button and Scrum template selection

3.1.2 Step 2: Team Onboarding (User Management)

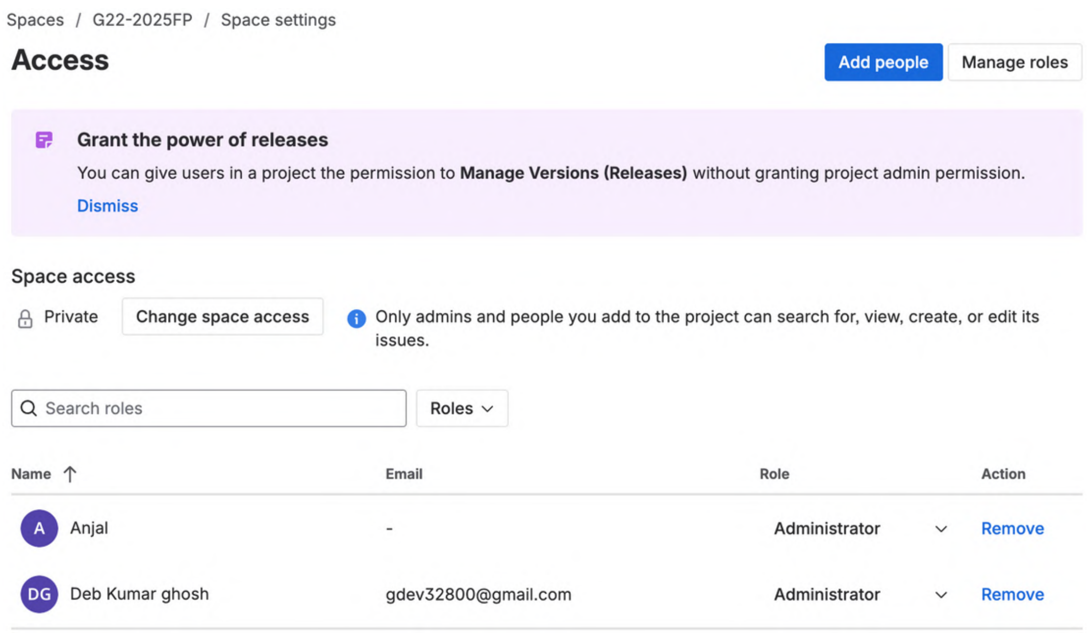
A project tool is not much use if you are the only one on it. We needed to add the team so we could assign tasks properly.

What we did:

1. Went to Project Settings (the gear icon) then People
2. Clicked "Add People"
3. Put in Shrestha Anjal's email
4. Set role to "Member" (so he can create and edit issues)
5. Hit "Add"

Result: Now both Ghosh Deb Kumar and Shrestha Anjal are in the system. This step matters because without it, you cannot assign tasks to anyone the Assignee dropdown would be empty.

Side note: In Redmine, tasks often got left as "Unassigned" and nobody knew who was supposed to do what. In Jira, you pretty much have to assign every task to someone, which keeps things clearer.


Spaces / G22-2025FP / Space settings

Access

Grant the power of releases

You can give users in a project the permission to **Manage Versions (Releases)** without granting project admin permission.

Space access

Private

Only admins and people you add to the project can search for, view, create, or edit its issues.

Name	Email	Role	Action
Anjal	-	Administrator	Remove
Deb Kumar ghosh	gdev32800@gmail.com	Administrator	Remove

Figure 10: Access settings page showing team members with "Add people" button and Administrator roles

3.2 Phase 2: Populating the Backlog (The Migration)

This is where we actually moved the data. We manually entered all the items from Redmine into the Jira Backlog to test out the workflow.

3.2.1 Step 1: Locate the Creation Tool

At the top of the screen there is a blue "Create" button. Click that and you get the "Create Issue" window, which is how you add anything to Jira.

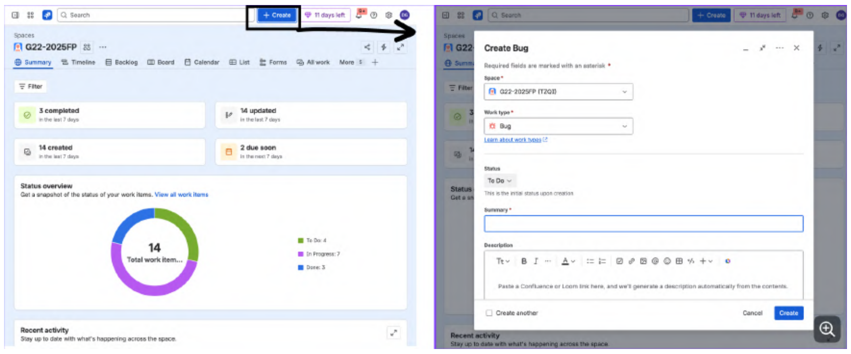


Figure 11: Create Issue modal dialog showing fields for Space, Work type, Status, Summary, and Description

The screenshot shows the Create Issue dialog. This is what we used to add everything. The main fields are:

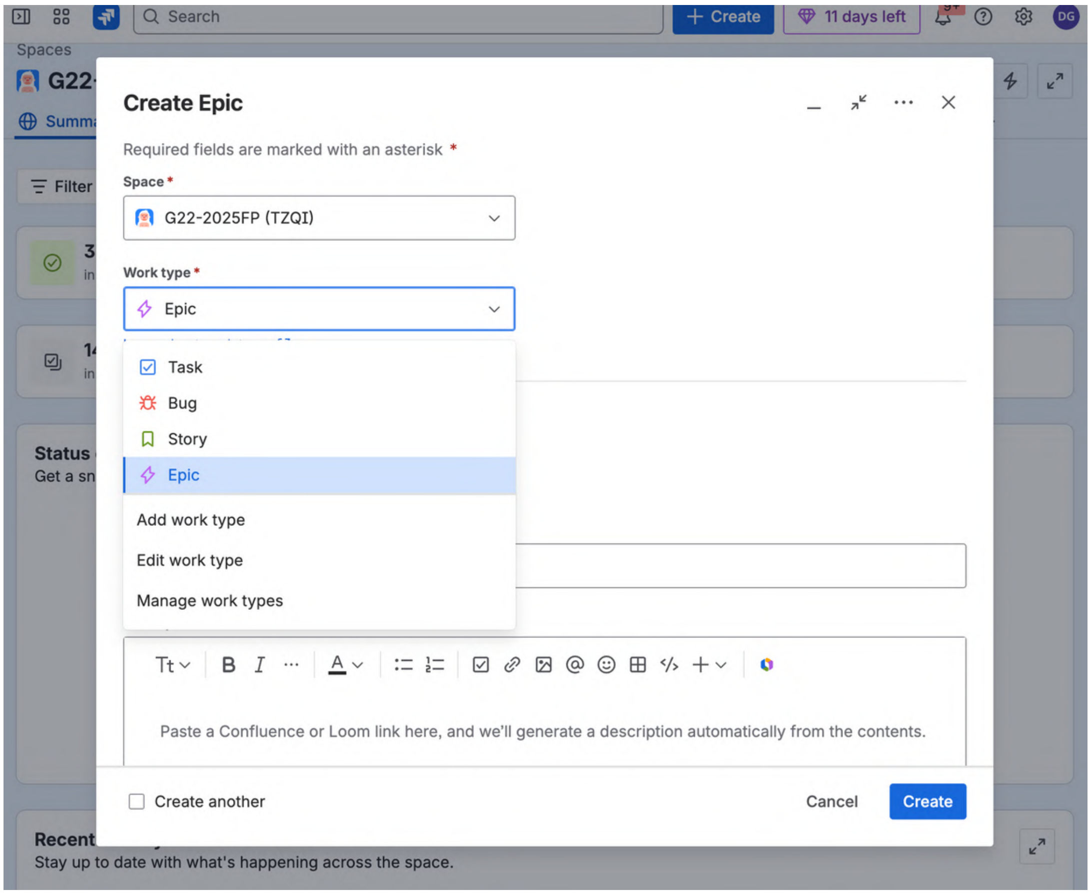
- Space: G22-2025FP (TZQI) our project
- Work type: dropdown where you pick Task, Story, Epic, Bug, Sub-task, etc.
- Status: starts as To Do
- Summary: title of the issue
- Description: text area with formatting options
- Blue "Create" button at the bottom to save it

3.2.2 Step 2: Creating the First Issue - The Epic

We started with the Epic since it is the top-level container that holds all the other tasks.

What we did:

1. In the Create dialog, clicked the "Work type" dropdown
2. Selected "Epic"
3. In Summary, typed "G22-2025-Final Project"
4. Scrolled down to add description, start date, and end date
5. Clicked "Create"


Create Epic

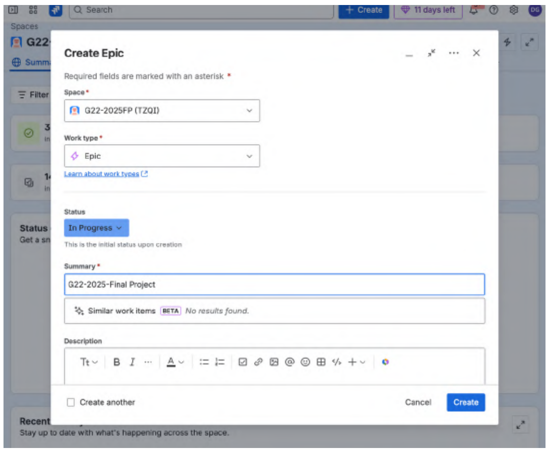
Required fields are marked with an asterisk *

Space *
G22-2025FP (TZQI)

Work type *
Epic

Paste a Confluence or Loom link here, and we'll generate a description automatically from the contents.

Figure 12: Create Issue dialog with Work type dropdown expanded showing Epic, Task, Story, Bug, Sub-task options


Create Epic

Required fields are marked with an asterisk *

Space *
G22-2025FP (TZQI)

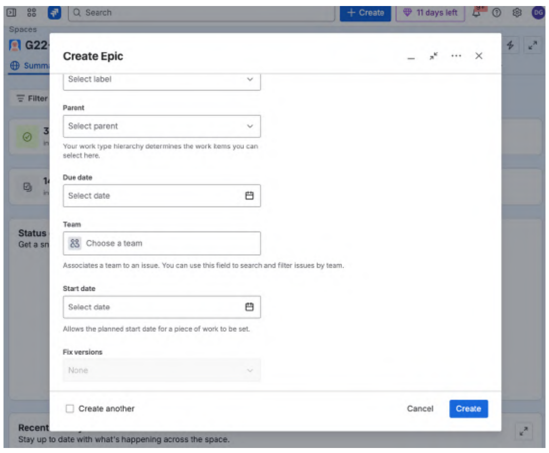
Work type *
Epic
Learn about work types

Status
In Progress
This is the initial status upon creation

Summary *
G22-2025-Final Project

Similar work items BETA No results found.

Description


Create Epic

Select label

Parent
Select parent
Your work type hierarchy determines the work items you can select here.

Due date
Select date

Team
Choose a team
Associates a team to an issue. You can use this field to search and filter issues by team.

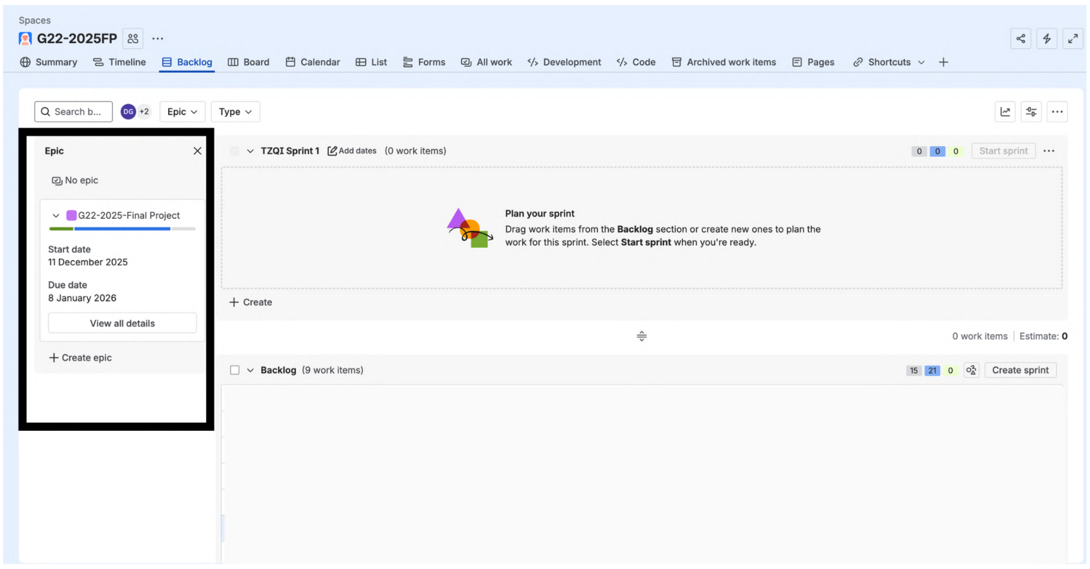
Start date
Select date
Allows the planned start date for a piece of work to be set.

Fix versions
None

Figure 13: Left: Epic form with Summary field filled. Right: Description and date fields after scrolling down

Result: Epic created. This becomes the container that links all our tasks together.

Why make an Epic first? Jira has a hierarchy: Epic > Task > Sub-task. The Epic is like the umbrella for the whole project. All the individual features (UML, Jira research, Wiki, etc.) get linked to this Epic, which makes them show up together on the Timeline.



Start date
11 December 2025

Due date
8 January 2026

Plan your sprint
Drag work items from the **Backlog** section or create new ones to plan the work for this sprint. Select **Start sprint** when you're ready.

0 work items | Estimate: 0

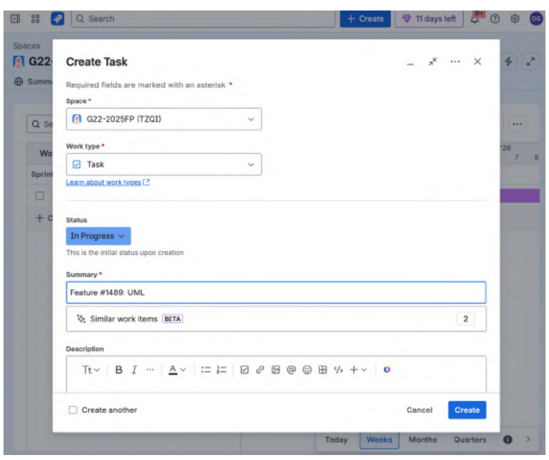
Figure 14: Backlog view showing Epic "G22-2025-Final Project" with purple Epic icon

3.2.3 Step 3: Creating Main Tasks

Next up, we created the 6 main tasks from our Redmine system.

For each task, we did this:

1. Click "Create"
2. Pick "Task" as the Work type
3. Enter the title from Redmine (like "Feature #1489: UML")
4. Add a description with details
5. Set Assignee to either Ghosh Deb Kumar or Shrestha Anjal
6. Set Story Points (we used 5 for most tasks)
7. Link to Epic by selecting "G22-2025-Final Project" from the Epic Link dropdown
8. Set Start Date and Due Date (you need both for the Timeline to work!)
9. Click "Create"


Create Task

Required fields are marked with an asterisk *

Space *: G22-2025FP (TZQI)

Work type *: Task

Learn about work types

Status: In Progress

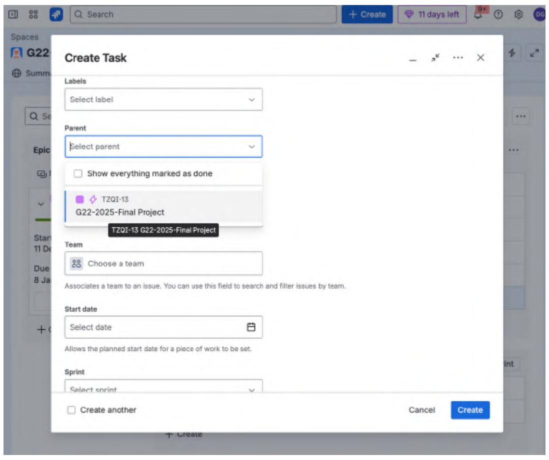
This is the initial status upon creation

Summary *: Feature #1489: UML

Similar work items BETA 2

Description

Create another Cancel Create


Create Task

Labels: Select label

Parent: Select parent

Show everything marked as done

TZQI-13 G22-2025-Final Project

Team: Choose a team

Associates a team to an issue. You can use this field to search and filter issues by team.

Start date: Select date

Allows the planned start date for a piece of work to be set.

Sprint: Select sprint

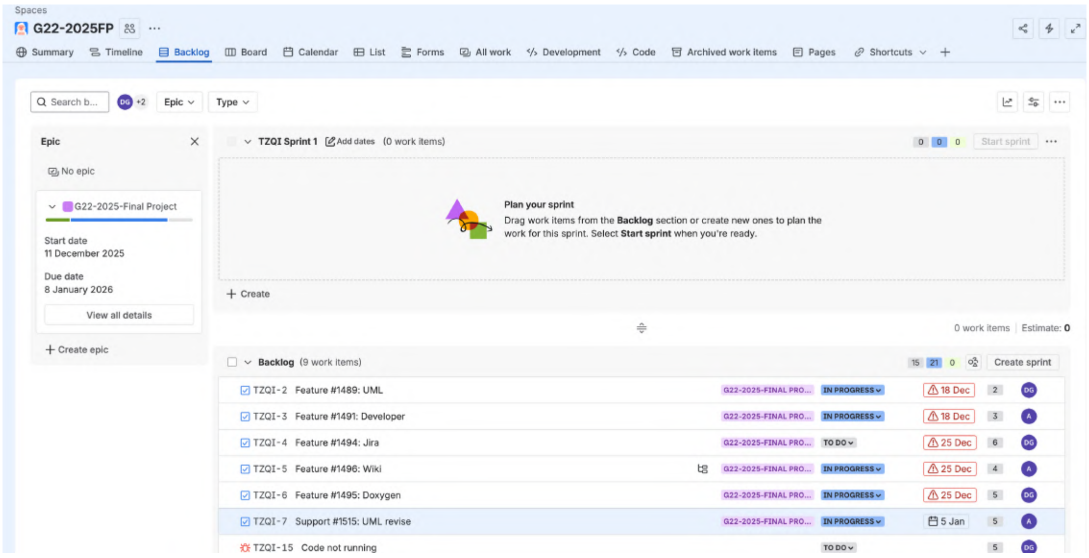
Create another Cancel Create

Figure 15: Task creation dialog showing fields for Summary, Assignee, Story Points, and Epic Link

Tasks Created:

- TZQI-2: Feature #1489: UML (5 points, Dec 11–15)
- TZQI-3: Feature #1491: Developer (3 points, Dec 11–18)
- TZQI-4: Feature #1494: Jira (6 points, Dec 16–25)
- TZQI-5: Feature #1496: Wiki (4 points, Dec 16–22)
- TZQI-6: Feature #1495: Doxygen (5 points, Dec 18–25)
- TZQI-7: Support #1515: UML revise (5 points, Dec 23–25)

After adding everything, you can see all the data sitting in the backlog all our tasks plus the Test1 item we created earlier.


Spaces

G22-2025FP

Summary · Timeline · Backlog · Board · Calendar · List · Forms · All work · Development · Code · Archived work items · Pages · Shortcuts

Epic

No epic

G22-2025-Final Project

Start date
11 December 2025

Due date
8 January 2026

View all details

Create epic

TZQI Sprint 1 Add dates (0 work items) 0 0 0 Start sprint

Plan your sprint

Drag work items from the **Backlog** section or create new ones to plan the work for this sprint. Select **Start sprint** when you're ready.

0 work items | Estimate: 0

Backlog (9 work items) 15 21 0 Create sprint

TZQI-2	Feature #1489: UML	G22-2025-FINAL PRO...	IN PROGRESS	18 Dec	2	DG
TZQI-3	Feature #1491: Developer	G22-2025-FINAL PRO...	IN PROGRESS	18 Dec	3	A
TZQI-4	Feature #1494: Jira	G22-2025-FINAL PRO...	TO DO	25 Dec	6	DG
TZQI-5	Feature #1496: Wiki	G22-2025-FINAL PRO...	IN PROGRESS	25 Dec	4	A
TZQI-6	Feature #1495: Doxygen	G22-2025-FINAL PRO...	IN PROGRESS	25 Dec	5	DG
TZQI-7	Support #1515: UML revise	G22-2025-FINAL PRO...	IN PROGRESS	5 Jan	5	A
TZQI-15	Code not running		TO DO		5	DG

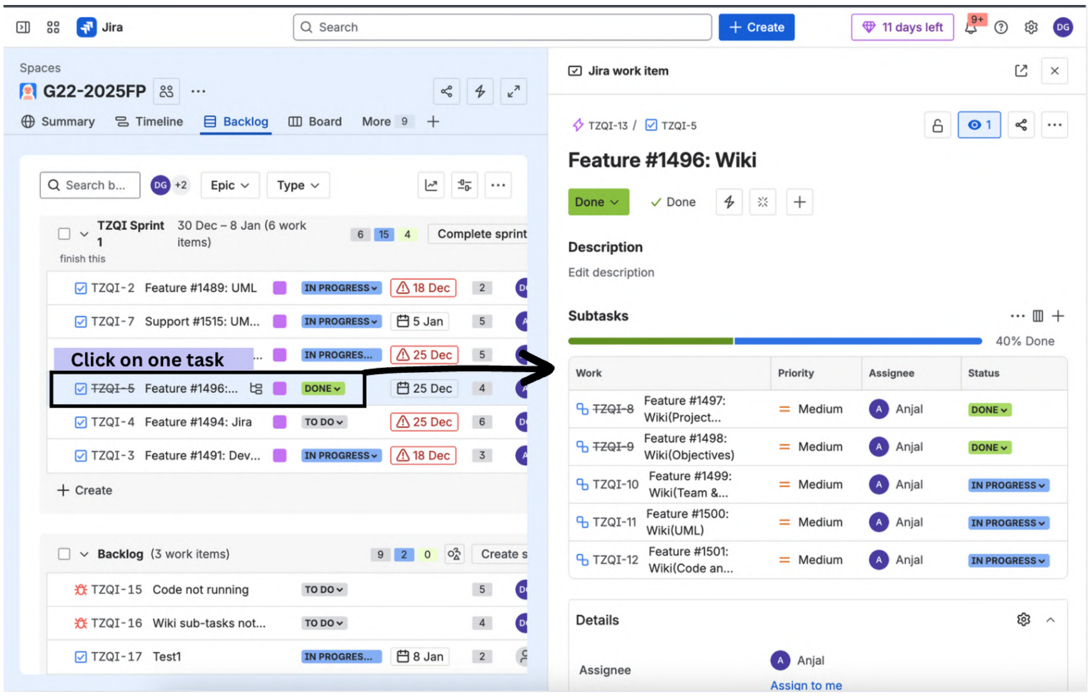
Figure 16: Backlog populated with all 6 main tasks and Epic linked together

3.2.4 Step 4: Creating Sub-tasks (Wiki Pages)

The Wiki task (TZQI-5) needed 5 child pages. In Jira, these are called "Sub-tasks" and they show up indented under their parent.

For each sub-task:

1. Clicked on TZQI-5 (Wiki) in the Backlog to open it
2. Inside the issue view, clicked "Create sub-task"
3. Work type was already set to Sub-task
4. Entered the title like "TZQI-8: Create Project Details page"
5. Parent was automatically linked to TZQI-5
6. Set Assignee, Story Points (1 point each), and dates
7. Clicked "Create"



TZQI Sprint 1 30 Dec – 8 Jan (6 work items) 6 15 4 Complete sprint
finish this

TZQI-2	Feature #1489: UML	IN PROGRESS	18 Dec	2	
TZQI-7	Support #1515: UM...	IN PROGRESS	5 Jan	5	
	...	IN PROGRES...	25 Dec	5	
~~TZQI-5~~	Feature #1496:...	DONE	25 Dec	4	
TZQI-4	Feature #1494: Jira	TO DO	25 Dec	6	
TZQI-3	Feature #1491: Dev...	IN PROGRESS	18 Dec	3	

+ Create

Backlog (3 work items) 9 2 0 Create s

TZQI-15	Code not running	TO DO		5
TZQI-16	Wiki sub-tasks not...	TO DO		4
TZQI-17	Test1	IN PROGRES...	8 Jan	2

TZQI-13 / TZQI-5

Feature #1496: Wiki

Done ✓ Done

Description

Edit description

Subtasks

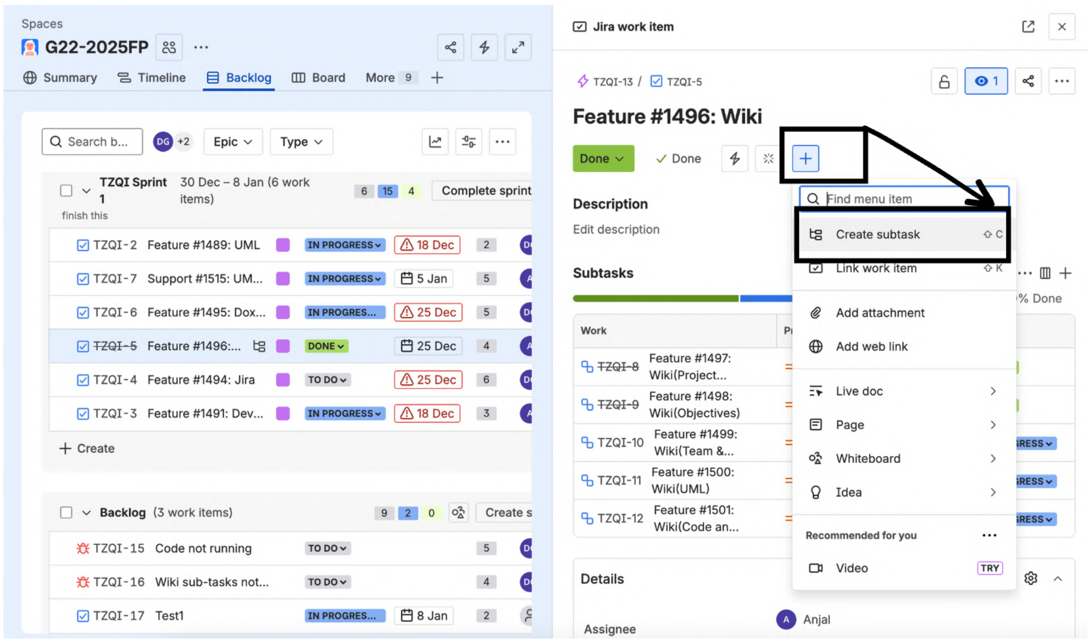
40% Done

Work		Priority	Assignee	Status
~~TZQI-8~~	Feature #1497: Wiki(Project...	Medium	Anjal	DONE
~~TZQI-9~~	Feature #1498: Wiki(Objectives)	Medium	Anjal	DONE
TZQI-10	Feature #1499: Wiki(Team &...	Medium	Anjal	IN PROGRESS
TZQI-11	Feature #1500: Wiki(UML)	Medium	Anjal	IN PROGRESS
TZQI-12	Feature #1501: Wiki(Code an...	Medium	Anjal	IN PROGRESS

Details

Assignee Anjal
Assign to me

Figure 17: Clicking on a task to open it and view "Create sub-task" option


Spaces

G22-2025FP

Summary | Timeline | Backlog | Board | More 9

Search b... | DG +2 | Epic | Type

TZQI Sprint 1 — 30 Dec – 8 Jan (6 work items) — 6 15 4 — Complete sprint

finish this

TZQI-2	Feature #1489: UML	IN PROGRESS	18 Dec	2	DG
TZQI-7	Support #1515: UM...	IN PROGRESS	5 Jan	5	A
TZQI-6	Feature #1495: Dox...	IN PROGRES...	25 Dec	5	DG
~~TZQI-5~~	Feature #1496:...	DONE	25 Dec	4	A
TZQI-4	Feature #1494: Jira	TO DO	25 Dec	6	DG
TZQI-3	Feature #1491: Dev...	IN PROGRESS	18 Dec	3	A

+ Create

Backlog (3 work items) — 9 2 0 — Create s

TZQI-15	Code not running	TO DO		5	DG
TZQI-16	Wiki sub-tasks not...	TO DO		4	DG
TZQI-17	Test1	IN PROGRES...	8 Jan	2	

Jira work item

TZQI-13 / TZQI-5

Feature #1496: Wiki

Done | ✓ Done

Description

Edit description

Subtasks

Work	
~~TZQI-8~~	Feature #1497: Wiki(Project...
~~TZQI-9~~	Feature #1498: Wiki(Objectives)
TZQI-10	Feature #1499: Wiki(Team &...
TZQI-11	Feature #1500: Wiki(UML)
TZQI-12	Feature #1501: Wiki(Code an...

Details

Assignee: A Anjal

Figure 18: Creating sub-task with Summary, Parent link, Assignee, and Story Points fields

Sub-tasks Created:

- TZQI-8: Create Project Details page (1 point)
- TZQI-9: Create Group Members page (1 point)
- TZQI-10: Create UML Diagrams page (1 point)
- TZQI-11: Create Code page (1 point)
- TZQI-12: Create Results page (1 point)

Result: The 5 sub-tasks now appear indented under TZQI-5 in the Backlog. You can clearly see the parent-child relationship. Total count: 1 Epic + 6 Tasks + 5 Sub-tasks = 12 issues migrated from Redmine.

3.3 Phase 3: The Planning Ceremony (Sprint Creation)

Okay so now all our issues are in the Backlog, but they are just sitting there not active yet. To actually start working on them, we need to put them into a Sprint. This is the sprint planning step.

3.3.1 What is a Sprint?

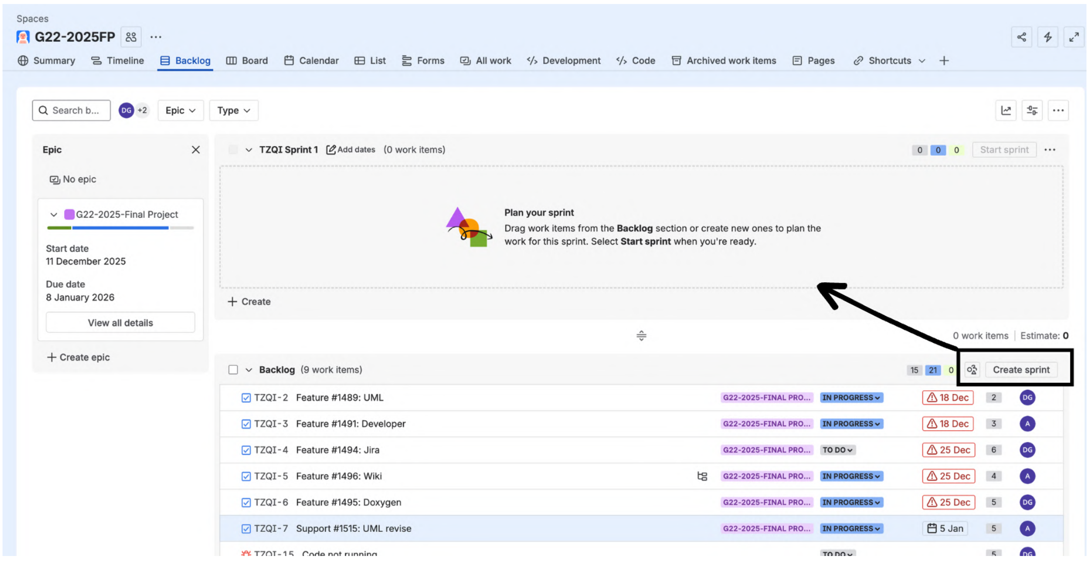
A Sprint is basically a time box (usually 1–4 weeks) where you commit to finishing certain tasks. For our project, we set up a 2-week sprint running December 11–25, 2025.

3.3.2 Step 1: Create the Sprint Container

On the Backlog screen, there is a "Create Sprint" button in the top-right corner.

What we did:

1. Clicked "Create Sprint"
2. A new grey box appeared at the top of the Backlog labeled "TZQI Sprint 1"
3. This box was empty, waiting for us to drag tasks into it



Start date
11 December 2025

Due date
8 January 2026

TZQI Sprint 1 (0 work items) 0 0 0

Plan your sprint
Drag work items from the **Backlog** section or create new ones to plan the work for this sprint. Select **Start sprint** when you're ready.

0 work items | Estimate: **0**

Backlog (9 work items) 15 21 0

TZQI-2	Feature #1489: UML	G22-2025-FINAL PRO...	IN PROGRESS	18 Dec	2	DG
TZQI-3	Feature #1491: Developer	G22-2025-FINAL PRO...	IN PROGRESS	18 Dec	3	A
TZQI-4	Feature #1494: Jira	G22-2025-FINAL PRO...	TO DO	25 Dec	6	DG
TZQI-5	Feature #1496: Wiki	G22-2025-FINAL PRO...	IN PROGRESS	25 Dec	4	A
TZQI-6	Feature #1495: Doxygen	G22-2025-FINAL PRO...	IN PROGRESS	25 Dec	5	DG
TZQI-7	Support #1515: UML revise	G22-2025-FINAL PRO...	IN PROGRESS	5 Jan	5	A
TZQI-15	Code not running		TO DO		5	DG

Figure 19: Backlog screen with "Create Sprint" button visible in top-right corner

3.3.3 Step 2: Drag and Drop Planning

This is the key part dragging tasks from the Backlog into the Sprint means you are committing to finish them.

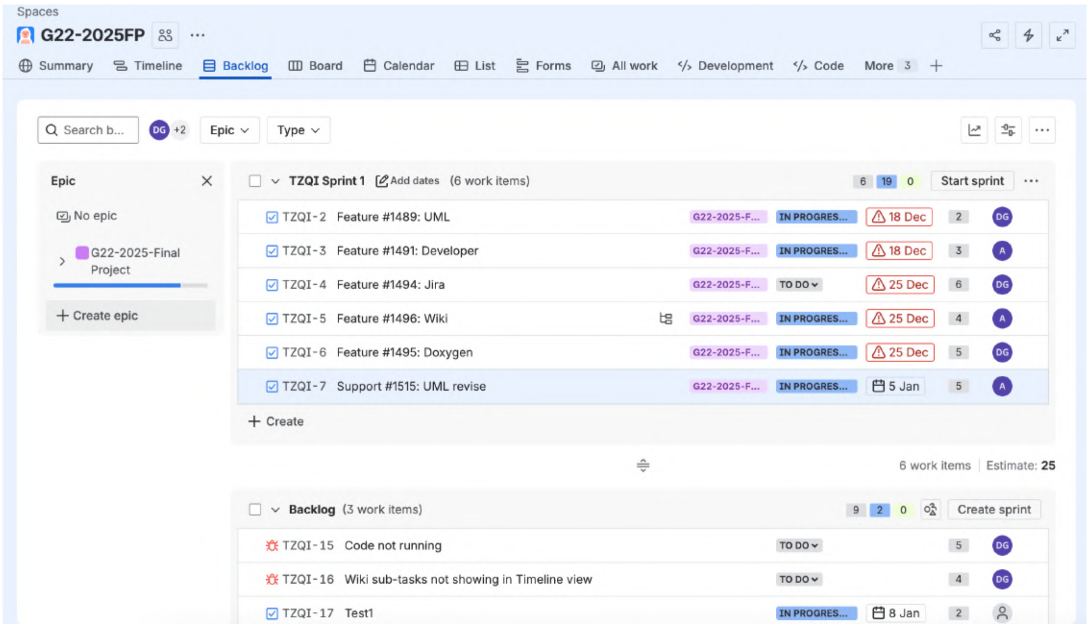
What we did:

1. Found our Epic TZQI-13 in the Backlog section (at the bottom)
2. Clicked and held the card
3. Dragged it up into the "TZQI Sprint 1" box
4. Let go and it moved into the Sprint
5. Did the same for all 6 main tasks (TZQI-2 through TZQI-7)

6. And again for all 5 sub-tasks (TZQI-8 through TZQI-12)

Something cool: As we dragged each item in, Jira automatically added up the Story Points at the bottom of the Sprint box. Our total came to 30 points (5+3+6+4+5+5 for tasks plus 1+1+1+1+1 for sub-tasks).

This total helps you see if you are biting off more than you can chew.



TZQI Sprint 1 Add dates (6 work items) 6 19 0 Start sprint

TZQI-2	Feature #1489: UML	G22-2025-F...	IN PROGRES...	18 Dec	2	DG
TZQI-3	Feature #1491: Developer	G22-2025-F...	IN PROGRES...	18 Dec	3	A
TZQI-4	Feature #1494: Jira	G22-2025-F...	TO DO	25 Dec	6	DG
TZQI-5	Feature #1496: Wiki	G22-2025-F...	IN PROGRES...	25 Dec	4	A
TZQI-6	Feature #1495: Doxygen	G22-2025-F...	IN PROGRES...	25 Dec	5	DG
TZQI-7	Support #1515: UML revise	G22-2025-F...	IN PROGRES...	5 Jan	5	A

+ Create

6 work items | Estimate: **25**

Backlog (3 work items) 9 2 0 Create sprint

TZQI-15	Code not running	TO DO		5	DG
TZQI-16	Wiki sub-tasks not showing in Timeline view	TO DO		4	DG
TZQI-17	Test1	IN PROGRES...	8 Jan	2	

Figure 20: Sprint 1 filled with all 12 items (Epic + Tasks + Sub-tasks) showing 30 Story Points total. Backlog section below shows 2 bugs and one task for later

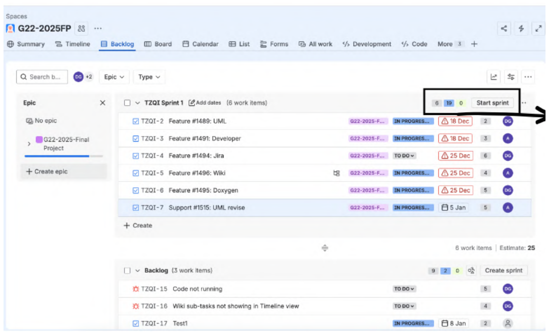
3.3.4 Step 3: Start the Sprint (Setting Dates)

With everything in the Sprint container, time to actually kick it off. This switches the tasks from "planned" to "active".

What we did:

1. Clicked the "Start Sprint" button at the top-right of the Sprint 1 box
2. A dialog popped up asking for:
   - Sprint Name: kept it as "TZQI Sprint 1"
   - Duration: picked "2 weeks" from the dropdown
   - Start Date: December 11, 2025 (set automatically)
   - End Date: December 25, 2025 (calculated automatically)
3. Clicked "Start"

Result: The screen jumped from Backlog view to Board view. All 12 items moved from planning mode into execution mode.



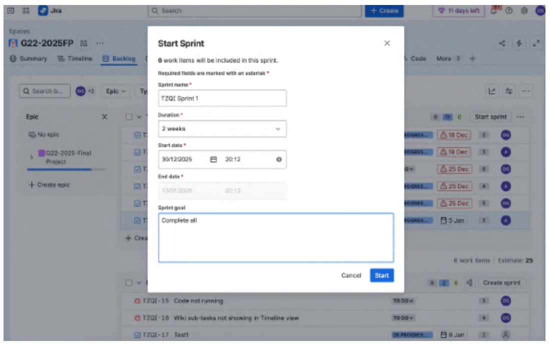
TZQI Sprint 1 Add dates (6 work items) 6 19 0

TZQI-2	Feature #1489: UML	G22-2025-F...	IN PROGRES...	18 Dec	2	DG
TZQI-3	Feature #1491: Developer	G22-2025-F...	IN PROGRES...	18 Dec	3	A
TZQI-4	Feature #1494: Jira	G22-2025-F...	TO DO	25 Dec	6	DG
TZQI-5	Feature #1496: Wiki	G22-2025-F...	IN PROGRES...	25 Dec	4	A
TZQI-6	Feature #1495: Doxygen	G22-2025-F...	IN PROGRES...	25 Dec	5	DG
TZQI-7	Support #1515: UML revise	G22-2025-F...	IN PROGRES...	5 Jan	5	A

6 work items | Estimate: 25

Backlog (3 work items) 9 2 0

TZQI-15	Code not running	TO DO		5	DG
TZQI-16	Wiki sub-tasks not showing in Timeline view	TO DO		4	DG
TZQI-17	Test1	IN PROGRES...	8 Jan	2	



Start Sprint

6 work items will be included in this sprint.

Required fields are marked with an asterisk *

Sprint name *: TZQI Sprint 1

Duration *: 2 weeks

Start date *: 30/12/2025 20:12

End date *: 13/01/2026 20:12

Sprint goal: Complete all

Figure 21: Left: Sprint filling with items being dragged. Right: "Start Sprint" dialog with duration set to "2 weeks" and dates visible

Heads up: Once you start a sprint, you cannot easily undo it without completing or canceling the whole thing. So make sure you have everything in there before clicking Start.

3.4 Phase 4: Daily Execution (Using the Board)

After starting the sprint, Jira took us straight to the Board view. This is where day-to-day work gets done.

What we saw:

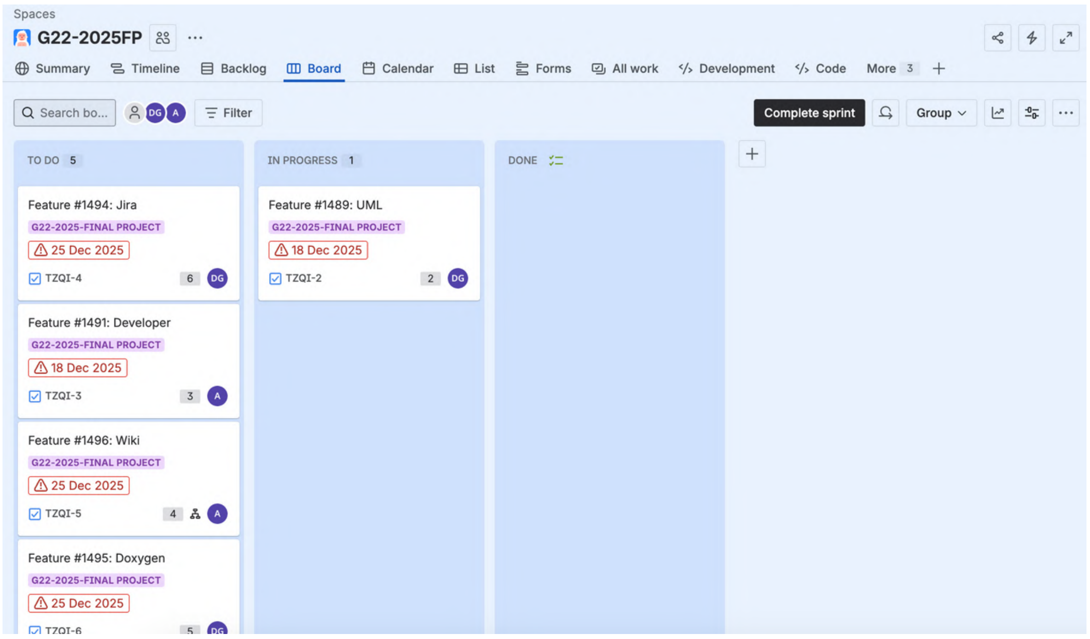
- TO DO (Left): All 12 items started here
- IN PROGRESS (Middle): Empty, waiting for someone to start working
- DONE (Right): Empty, will fill up as tasks get finished

3.4.1 Step 1: Moving a Task to "In Progress"

When one of us started working on TZQI-2 (UML), here is what happened:

1. Found the TZQI-2 card in the TO DO column
2. Clicked and held it
3. Dragged it over to the IN PROGRESS column
4. Let go

Result: Card moved right away. Now anyone who opens Jira can see that someone is working on UML.



TO DO 5

Feature #1494: Jira
G22-2025-FINAL PROJECT
25 Dec 2025
TZQI-4 6 DG

Feature #1491: Developer
G22-2025-FINAL PROJECT
18 Dec 2025
TZQI-3 3 A

Feature #1496: Wiki
G22-2025-FINAL PROJECT
25 Dec 2025
TZQI-5 4 A

Feature #1495: Doxygen
G22-2025-FINAL PROJECT
25 Dec 2025
TZQI-6 5 DG

IN PROGRESS 1

Feature #1489: UML
G22-2025-FINAL PROJECT
18 Dec 2025
TZQI-2 2 DG

DONE

Figure 22: Board showing TZQI-2 in the IN PROGRESS column, other items in TO DO, and DONE empty

Why this is nice: In Redmine, updating status meant clicking into the issue, finding the dropdown, picking the new status, saving, then waiting for the page to reload. In Jira you just drag and drop.

3.4.2 Step 2: Team Collaboration (Comments)

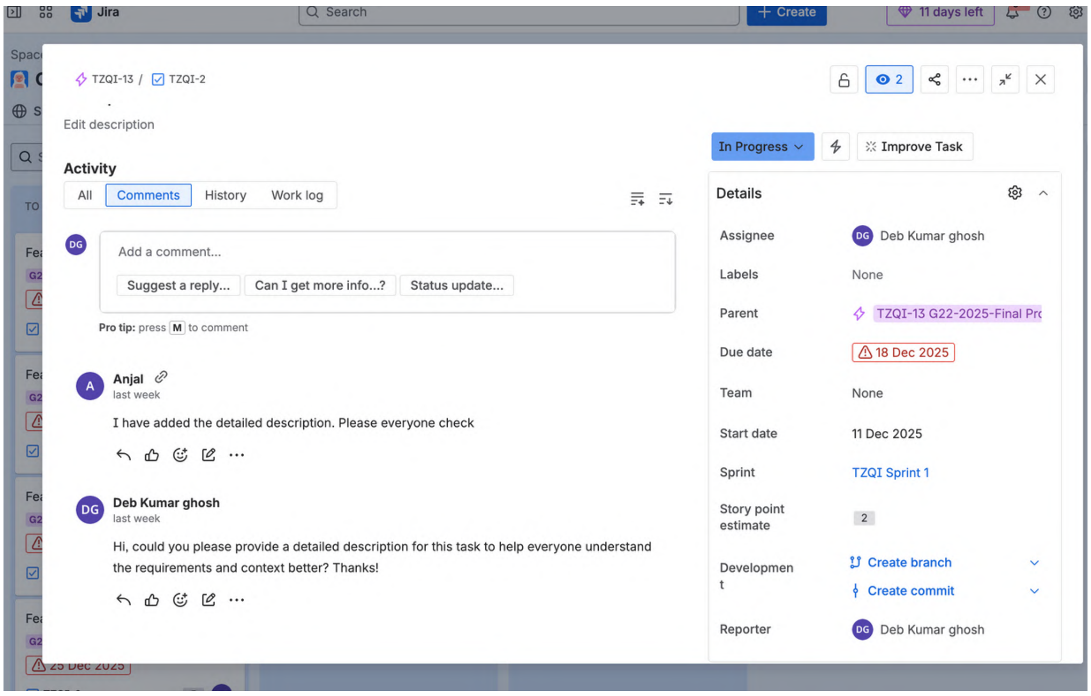
Sometimes you need to talk to teammates about a task. Jira has built-in commenting for this.

Example: One of us realized we needed Lucidchart for the UML diagrams and the other person needed to set up an account.

What we did:

1. Clicked on the TZQI-2 card to open the detail panel
2. Scrolled to the "Comments" section
3. Typed: "@Anjal Please create Lucidchart account and share access for UML diagrams"
4. Clicked "Save"

Result: The other person got an email notification plus an alert in Jira. He replied: "Done. Account shared." The whole conversation stays saved in the task history.



Anjal
last week
I have added the detailed description. Please everyone check

Deb Kumar ghosh
last week
Hi, could you please provide a detailed description for this task to help everyone understand the requirements and context better? Thanks!

Details

Assignee	Deb Kumar ghosh
Labels	None
Parent	TZQI-13 G22-2025-Final Pro
Due date	18 Dec 2025
Team	None
Start date	11 Dec 2025
Sprint	TZQI Sprint 1
Story point estimate	2
Development	Create branch Create commit
Reporter	Deb Kumar ghosh

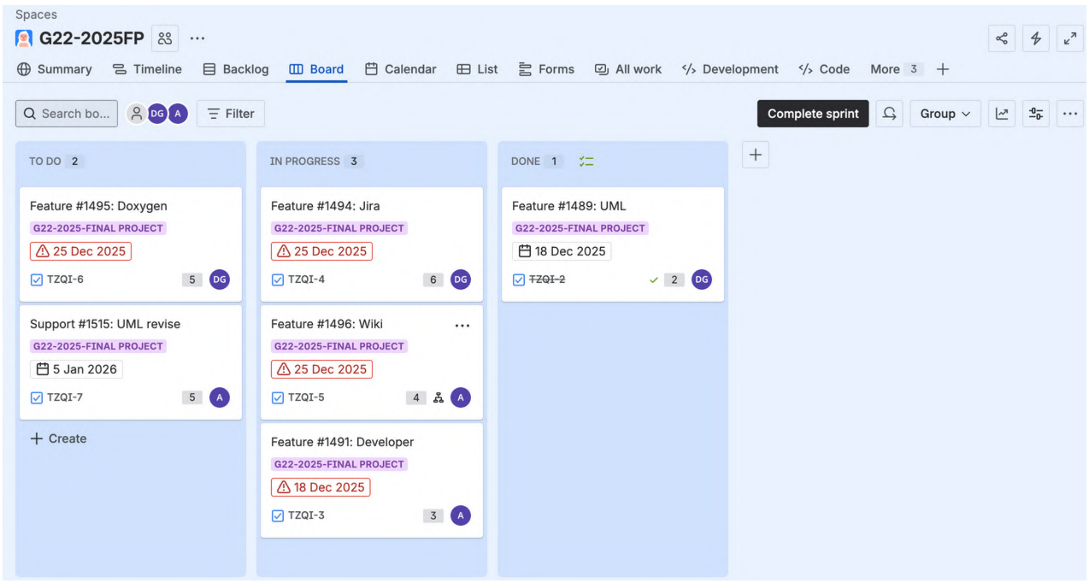
Figure 23: Task detail panel showing Comments section with conversation between team members using @mentions

3.4.3 Step 3: Completing Work (Moving to Done)

When a task is finished, you update the status:

1. Finished the UML diagrams
2. Dragged TZQI-2 from IN PROGRESS to DONE
3. Card turned green
4. Progress bar at the top updated to "1 of 12 items completed"

Result: Visual progress tracking. You can see at a glance how much is done without asking anyone for a status update.


Spaces
G22-2025FP
Summary Timeline Backlog Board Calendar List Forms All work Development Code More 3
Search bo... DG A Filter Complete sprint Group

TO DO 2

Feature #1495: Doxygen
G22-2025-FINAL PROJECT
25 Dec 2025
TZQI-6 5 DG

Support #1515: UML revise
G22-2025-FINAL PROJECT
5 Jan 2026
TZQI-7 5 A

Create

IN PROGRESS 3

Feature #1494: Jira
G22-2025-FINAL PROJECT
25 Dec 2025
TZQI-4 6 DG

Feature #1496: Wiki
G22-2025-FINAL PROJECT
25 Dec 2025
TZQI-5 4 A

Feature #1491: Developer
G22-2025-FINAL PROJECT
18 Dec 2025
TZQI-3 3 A

DONE 1

Feature #1489: UML
G22-2025-FINAL PROJECT
18 Dec 2025
~~TZQI-2~~ ✓ 2 DG

Figure 24: Board showing TZQI-2 moved to DONE column (green), progress bar updated to "1 of 12 items completed"

3.5 Phase 5: Creating Pages (The Wiki)

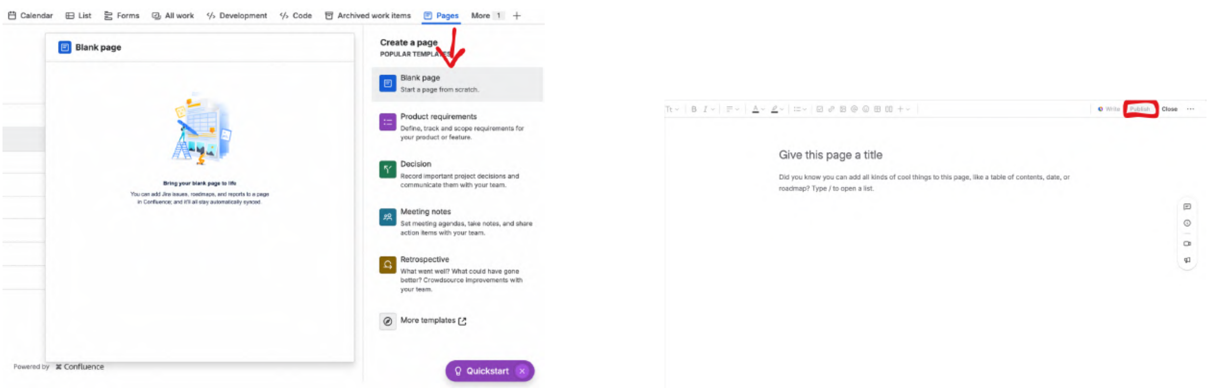
With tasks and sprint set up, the last part was setting up our documentation. In Redmine, we had a separate Wiki tab. In Jira, you use "Pages" which ties directly into your project.

3.5.1 Step 1: Creating Main Pages (Wiki Pages)

We created 1 main page plus 5 sub-pages to match what we had in Redmine.

For each page:

1. Clicked the "Pages" button
2. Picked "Blank Page" (or whatever template you want) from the "Create a Page" panel on the right
3. Entered the title
4. Added content, links to sub-pages, project overview, etc.
5. Clicked "Publish"



Bring your blank page to life

You can add Jira issues, roadmaps, and reports to a page in Confluence; and it'll all stay automatically synced.

Blank page — Start a page from scratch.

Product requirements — Define, track and scope requirements for your product or feature.

Decision — Record important project decisions and communicate them with your team.

Meeting notes — Set meeting agendas, take notes, and share action items with your team.

Retrospective — What went well? What could have gone better? Crowdsource improvements with your team.

Give this page a title

Did you know you can add all kinds of cool things to this page, like a table of contents, date, or roadmap? Type / to open a list.

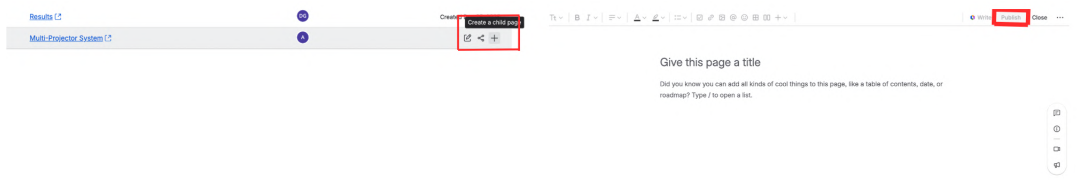
Figure 25: Left: Pages button and "Create a Page" interface. Right: Selecting Blank Page template

3.5.2 Step 2: Creating Sub-Pages (Wiki Pages)

Next we made the 5 sub-pages.

For each sub-page:

1. Clicked on the title of the parent page
2. Found the "+" mark on the right side and clicked "Create a child page"
3. Entered the title ("Project Details", "Group Members", etc.)
4. Added content
5. Clicked "Publish"



Give this page a title

Did you know you can add all kinds of cool things to this page, like a table of contents, date, or roadmap? Type / to open a list.

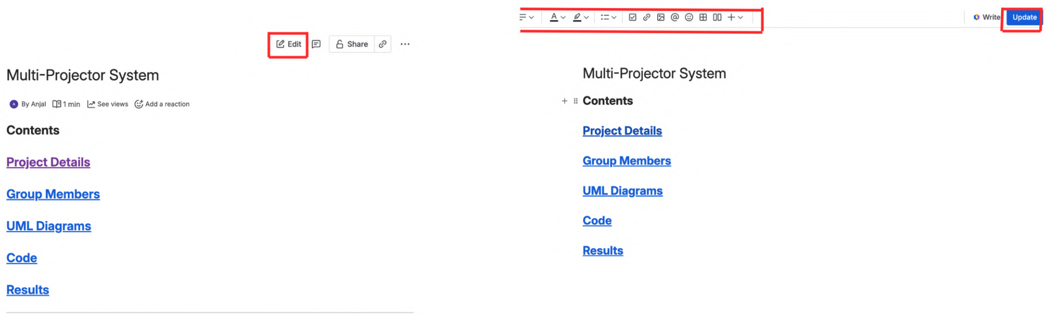
Figure 26: Page tree with "+" mark showing "Create a child page" option visible

3.5.3 Step 3: Editing Pages (Wiki Pages)

To edit existing pages:

For each page:

1. Click on the page title to open it
2. Click "Edit" on the right side
3. Use the toolbar at the top to change font size, add links, emojis, pictures, etc.
4. Click "Update"



Multi-Projector System

Contents

Project Details

Group Members

UML Diagrams

Code

Results

Figure 27: Pages editor with Edit button and formatting toolbar with font size, links, emoji, and picture options

Result: We got the full documentation structure moved over: 1 parent page with 5 child pages underneath. The nice thing compared to Redmine is the Page Tree in the sidebar you can see the whole document structure without clicking around. Plus you can link directly to tasks (like TZQI-13) inside the page text, so docs and tasks stay connected.

The other big wins: multiple people can edit at the same time (like Google Docs), you can @mention teammates, and you can restore old versions with one click. Redmine's wiki cannot do any of that.

4 Summary of Migration Phases

4.1 Phase 1: Environment Initialization

- Created project with Scrum template
- Added team members (Ghosh Deb Kumar, Shrestha Anjal) with Administrator roles

4.2 Phase 2: Populating the Backlog

- Created 1 Epic (TZQI-13): "G22-2025-Final Project"
- Created 6 Main Tasks (TZQI-2 to TZQI-7): UML, Developer, Jira, Wiki, Doxygen, UML Revision
- Created 5 Sub-tasks (TZQI-8 to TZQI-12): Wiki page creation tasks
- Linked all tasks to the Epic
- Set assignees, story points (total 30 points), and dates (Dec 11–25, 2025)

4.3 Phase 3: Sprint Planning

- Created Sprint 1 container
- Dragged all 12 items into the sprint
- Started sprint with 2-week duration (December 11–25, 2025)
- System automatically switched to Board view

4.4 Phase 4: Daily Execution

- Moved tasks through workflow: To Do → In Progress → Done
- Used @mentions for team communication
- Tracked progress visually on the Board

4.5 Phase 5: Creating Documentation

- Created main project page (G22-2025FP Home)
- Created 5 child pages (Project Details, Group Members, UML Diagrams, Code, Results)
- Edited pages with rich-text content and embedded links
- Published documentation accessible to all team members

5 Conclusion

After going through this whole process, we can say that Jira does have some real advantages over Redmine, especially if your team works in sprints:

- **Visual Management**: The Timeline and Board views let you see status at a glance instead of clicking through lists
- **Integrated Documentation**: Pages is built right in and links to your issues, so you do not need a separate wiki
- **Agile Support**: The Sprint, Backlog, and story points features actually support how agile teams work
- **Team Collaboration**: Comments, @mentions, and version history make it easier to communicate and track changes
- **Scalability**: The Epic > Task > Sub-task hierarchy handles complexity better than flat lists

Based on our testing, switching from Redmine to Jira seems doable. The G22-2025FP project could benefit from better visualization, tighter documentation integration, and proper sprint management.