Experiment : Create a Kanban Board for Hospital Management System using Jira

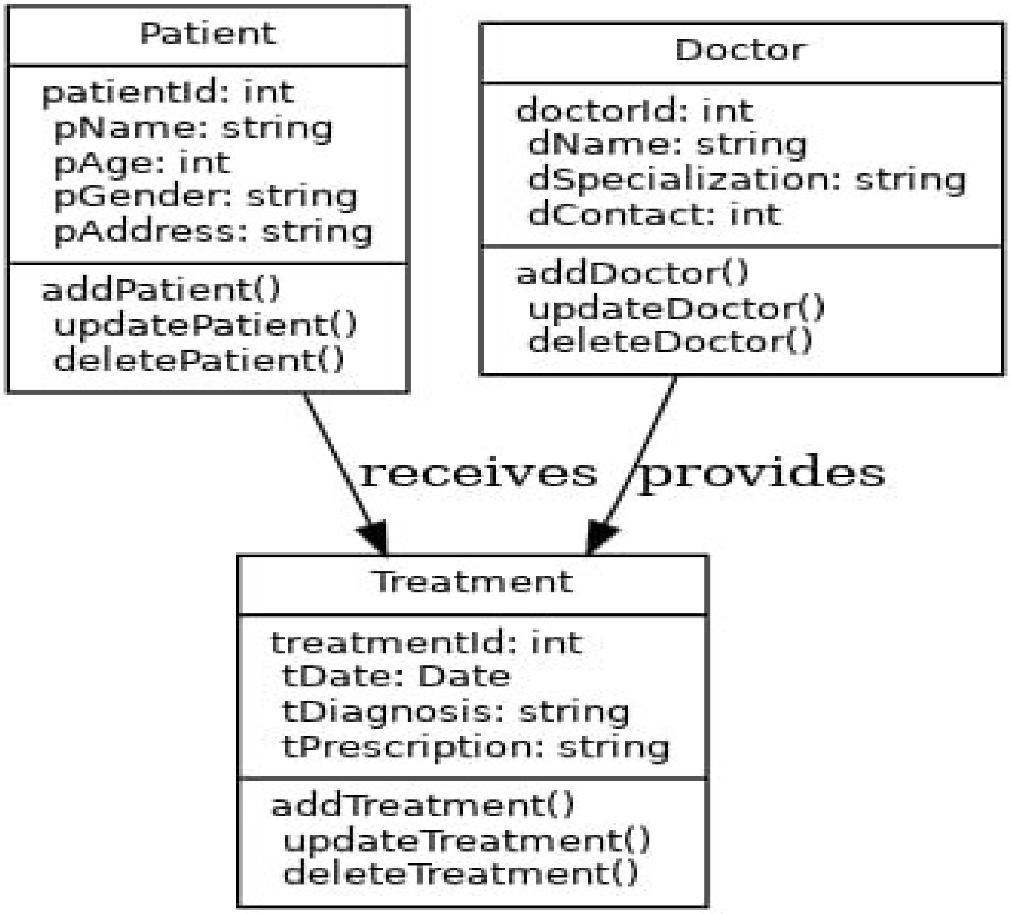
Aim:

To create a Kanban board for a Hospital Management System using Jira, by defining projects, teams, epics, stories, tasks, and releases.

Description:

Agile methodology emphasizes iterative and incremental development. **Kanban**, a popular Agile framework, helps visualize workflow, limit work-in-progress, and improve efficiency. Using **Jira**, hospital operations such as patient registration, doctor assignment, treatment, and billing can be managed effectively by organizing them into epics, stories, and tasks on a Kanban board.

Class Diagram: Hospital Management System



Procedure:

Step 1: Create a Project

- Start Jira → Click **Projects** → **Create Project**.
- Select **Software Development** → **Kanban** → **Use Template**.
- Choose **Company Managed Project**.
- Enter project name **"Hospital Management System"** → **Click Create Project**.
- Skip recommended options. → Project created successfully.

Step 2: Create a Team

a) **Invite Members**

- Enter email → Select name → Invite.

b) **Create a Team**

- Click **Teams** → **Create Team** → **Add Members** → **Click Create**.
- Team created successfully

Step 3: Create Epics

- Click **Create** → **Work Type: Epic**.
- Enter Epic name (e.g., *Patient*), description, select team, start & due dates.
- Click **Create**.
- Similarly create other Epics: *Doctor*, *Treatment*.

Step 4: Create Stories

- Click **Create** → **Work Type: Story**.
- Enter story name (e.g., *Add Patient*) under parent Epic (*Patient*)
- Add description, assign team, set dates → Click **Create**.
- Similarly create stories like *Update Patient*, *Delete Patient*, *Doctor Consultation*, *Treatment Details*.

Step 5: Create Tasks

- Click **Create** → **Work Type: Task**.
- Enter task name (e.g., *Patient ID*) under parent Epic.
- Add description, assign team, set dates → Click **Create**.
- Similarly create tasks under *Doctor* and *Treatment*.

Step 6: Create Version and Release

Create Version:

- In the sidebar, select **More actions (•••)** and select **Releases**
- Click on the **"Create version"** button, typically found at the top right of the Releases page.
- **Enter Version Details:**

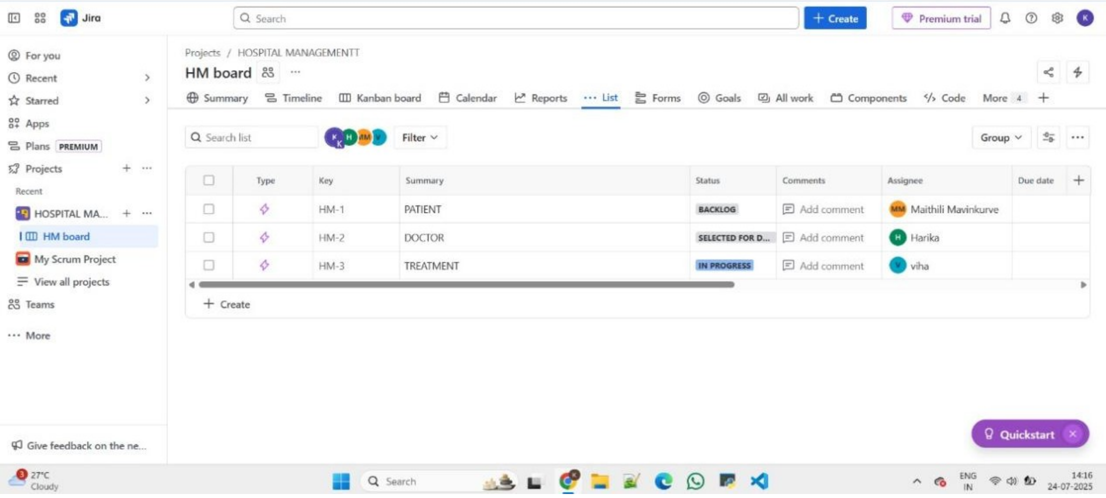
  1) **Name:** Provide an identifiable name for the version (e.g., *"Patient Module v1.0"*, *"Doctor Module v1.0"*, *"Treatment Module v1.0"*).
  2) **Start date (Optional):** Specify the planned start date for work related to this version.
  3) **Release date (Optional):** Define the target release date for the version.
  4) **Description (Optional):** Add a brief description outlining the scope or purpose of the version (e.g., *"Includes epics: Patient Registration, Update Patient, and Delete Patient"*).

- **Save:** Click **"Save"** to create the new version

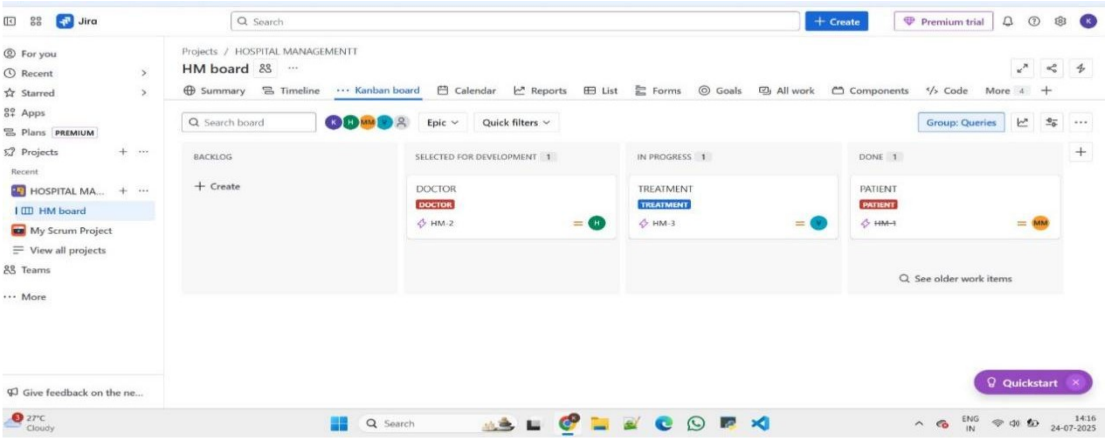
Releasing a Version:

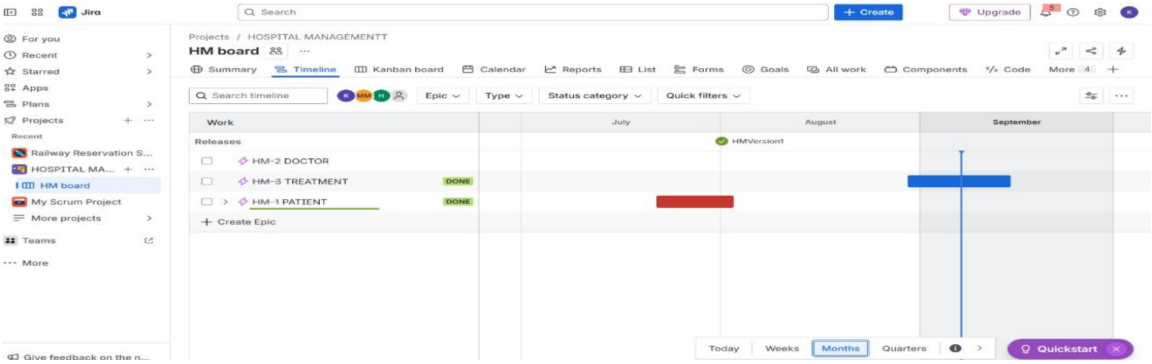
- **Initiate Release:** On the individual version's page, click the **"Release"** button. This button is usually prominent.
- **Confirm Release Details:** A dialog box will appear, allowing you to confirm or add details related to the release, such as the actual release date (e.g., *Releasing Doctor Consultation and Treatment Details*).
- **Execute Release:** Click the **"Release"** button within the dialog to finalize the release of the version.

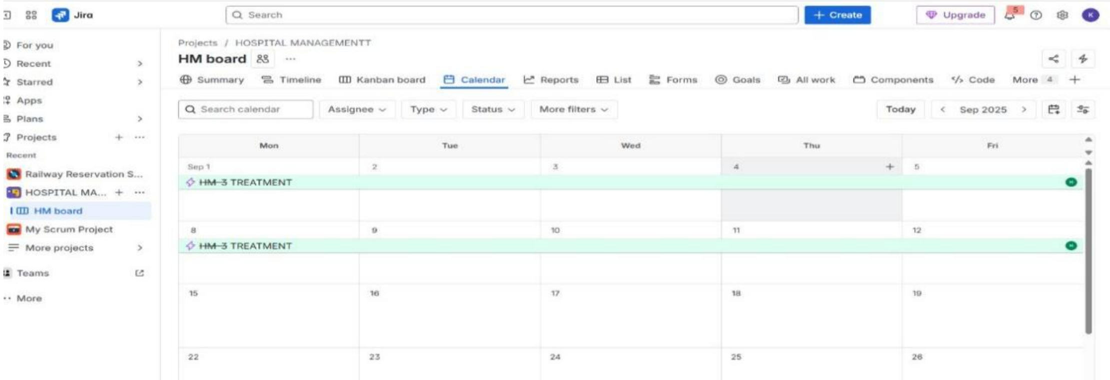
Work Flow Images:



☐	Type	Key	Summary	Status	Comments	Assignee	Due date
☐		HM-1	PATIENT	BACKLOG	Add comment	Maithili Mavinkurve	
☐		HM-2	DOCTOR	SELECTED FOR D...	Add comment	Harika	
☐		HM-3	TREATMENT	IN PROGRESS	Add comment	viha	







Result:

By using Jira and Kanban methodology, the hospital workflow can be efficiently managed. Tasks are visualized, responsibilities are assigned, and progress is easily tracked, ensuring better coordination and timely delivery of healthcare services.

Experiment : Create a Scrum Board for Railway Reservation System using Jira

Aim:

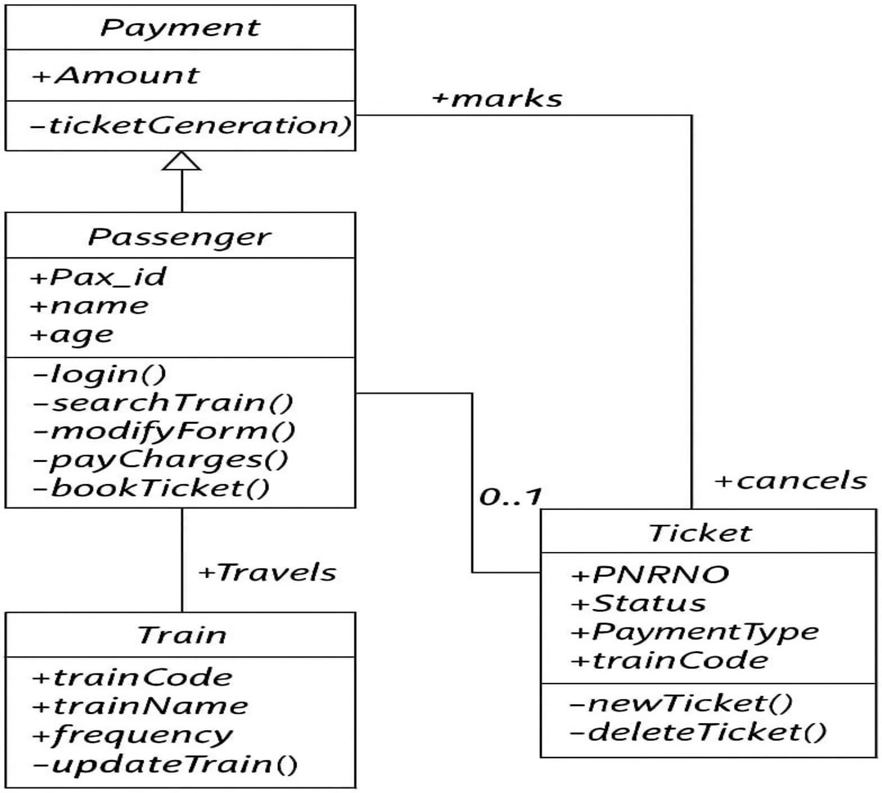
To create a **Scrum Board** for a Railway Reservation System using **Jira**, by organizing workflow into sprints and managing product backlog items such as Passenger, Train, Ticket, and Payment modules.

Description:

Scrum is an Agile framework that organizes work into **time-boxed iterations (sprints)**. A **Scrum Board** helps visualize backlog items, sprint progress, and completed tasks.

For a **Railway Reservation System**, Scrum methodology is applied to handle operations like passenger management, train scheduling, ticket booking, and payment processing. Jira Scrum Board allows tracking of **Epics, Stories, and Tasks** for each sprint, ensuring systematic and timely development of the system.

Class Diagram: Railway Reservation System



Procedure:

Step 1: Create a Project

- Start Jira → Click **Projects** → **Create Project**.
- Select **Software Development** → **Scrum** → **Use Template**.
- Choose **Company Managed Project**.
- Enter project name **“Railway Reservation System”** → **Create Project**.

Step 2: Create a Team

- Invite members by entering email IDs.
- Create a team and assign members.
- Team created successfully.

Step 3: Create Epics (Product Backlog)

- Create Epics for the main modules from the class diagram:

1) **Passenger** (login, search train, modify form, book ticket)
2) **Train** (update train, cancel train)
3) **Ticket** (new ticket, delete ticket, status updates)
4) **Payment** (pay charges, generate ticket, amount handling).

Step 4: Create Stories

- Under each Epic, create User Stories:

1) *Passenger Epic*: Add Passenger, Modify Passenger Form, Book Ticket.
2) *Train Epic*: Update Train Info, Cancel Train.
3) *Ticket Epic*: Generate New Ticket, Delete Ticket, Update Status.
4) *Payment Epic*: Make Payment, Confirm Ticket Generation.

Step 5: Create Tasks

- Break stories into smaller tasks, e.g.:

1) For *Passenger*: Create Passenger ID, Validate Login, Implement Search
2) For *Train*: Create Train Code, Update Frequency.
3) For *Ticket*: Assign PNR Number, Track Status.
4) For *Payment*: Calculate Amount, Validate Payment Type.

Step 6: Manage Scrum Board

- Move backlog items into the sprint backlog.

- ➢ Start a new sprint (set sprint goal and duration).
- ➢ Move tasks across columns: *To Do → In Progress → Done*.
- ➢ Monitor burndown chart to track sprint progress.

Step 7: Create Version and Release

Create Version:

- ➢ In the sidebar, select **More actions (•••)** and select **Releases**
- ➢ Click on the **"Create version"** button, typically found at the top right of the Releases page.
- ➢ **Enter Version Details:**

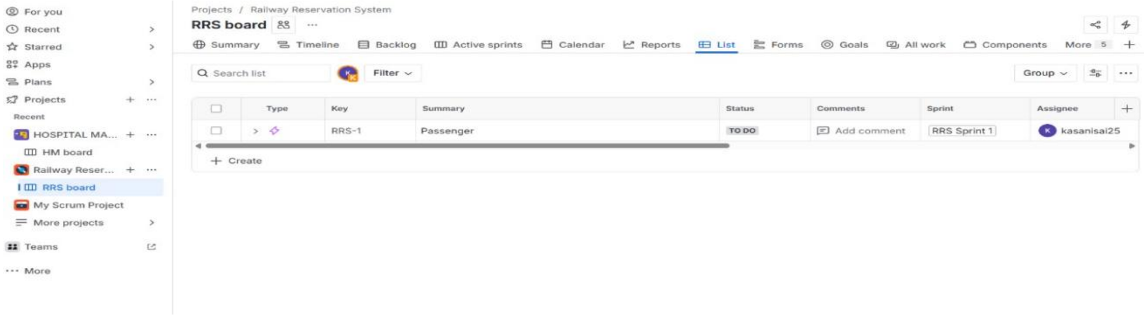
    1. **Name:** Provide an identifiable name for the version (e.g., *"Passenger Module v1.0"*, *"Train Module v1.0"*, *"Ticket Module v1.0"*, *"Payment Module v1.0"*).
    2. **Start date (Optional):** Specify the planned start date for work related to this version.
    3. **Release date (Optional):** Define the target release date for the version.
    4. **Description (Optional):** Add a brief description outlining the scope or purpose of the version (e.g., *"Includes epics: Add Passenger, Modify Passenger Form, and Book Ticket"* or *"Includes epics: Make Payment and Confirm Ticket Generation"*).

- ➢ **Save:** Click **"Save"** to create the new version.

Releasing a Version:

- ➢ **Initiate Release:** On the individual version's page, click the **"Release"** button. This button is usually prominent.
- ➢ **Confirm Release Details:** A dialog box will appear, allowing you to confirm or add details related to the release, such as the actual release date (e.g., *Releasing Train Module with Update Train Info and Cancel Train stories*).
- ➢ **Execute Release:** Click the **"Release"** button within the dialog to finalize the release of the version.

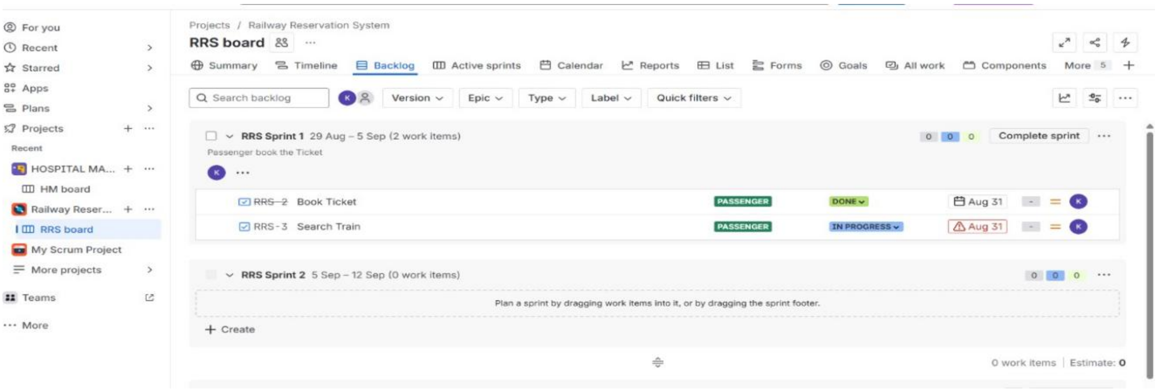
Work Flow Images:



Projects / Railway Reservation System

RRS board

□	Type	Key	Summary	Status	Comments	Sprint	Assignee
□	>	RRS-1	Passenger	TO DO	Add comment	RRS Sprint 1	kasanisai25

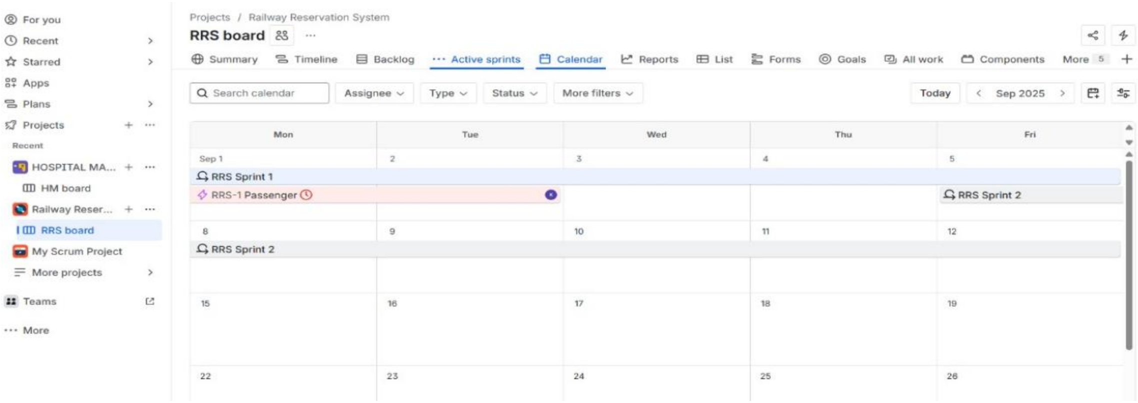


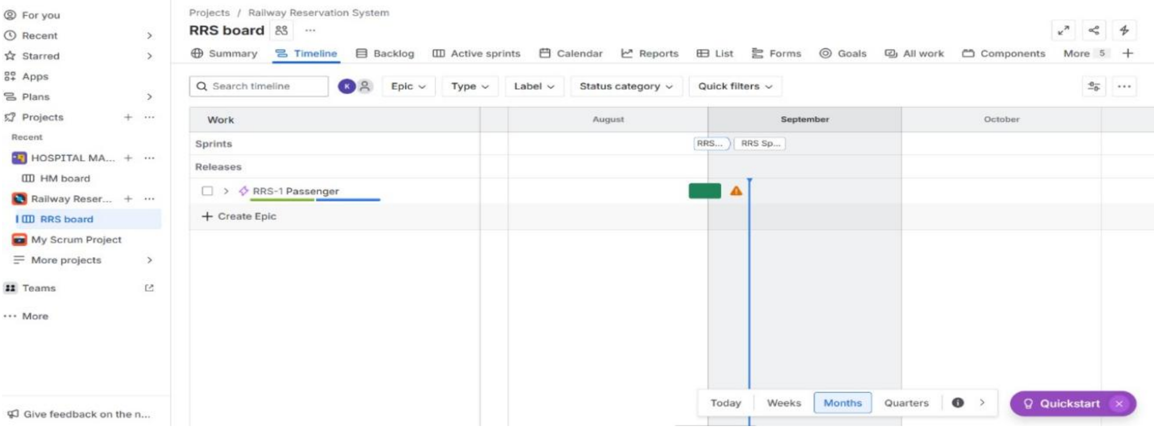
RRS Sprint 1 29 Aug – 5 Sep (2 work items)

Passenger book the Ticket

~~RRS-2~~	Book Ticket	PASSENGER	DONE	Aug 31
RRS-3	Search Train	PASSENGER	IN PROGRESS	Aug 31

RRS Sprint 2 5 Sep – 12 Sep (0 work items)





Result:

The Scrum Board for the Railway Reservation System was successfully created using Jira. By dividing work into Epics, Stories, and Tasks aligned with the modules *Passenger, Train, Ticket, and Payment*, the project can be developed efficiently with iterative sprint planning and tracking.

Experiment : Create a Kanban Board for Library Management System using Jira

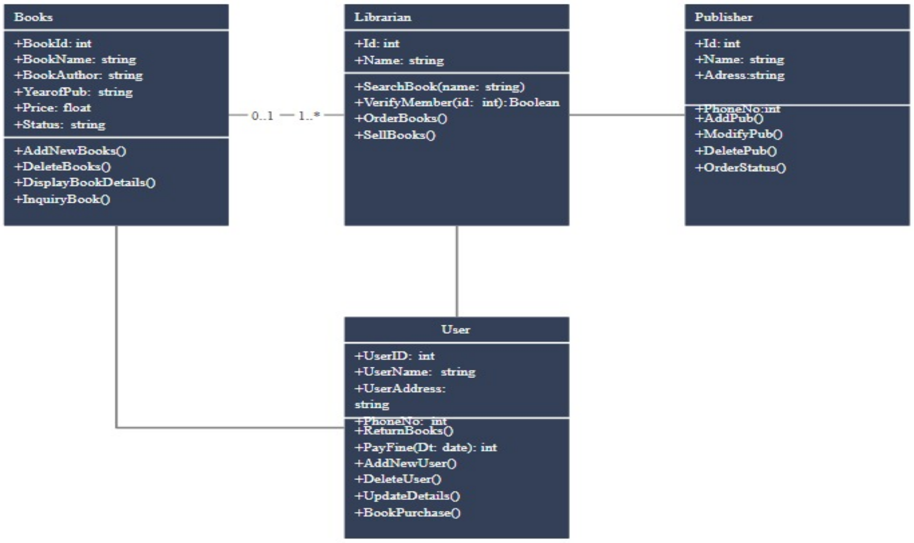
Aim:

To create a **Kanban Board** for a Library Management System using **Jira**, by defining epics, stories, tasks, and releases to manage the library workflow effectively.

Description:

The Library Management System is designed to manage books, librarians, publishers, and users efficiently. The system keeps track of available books, manages user memberships, handles fines, and facilitates book purchase and sales. By implementing the Kanban methodology in Jira, we can visualize the workflow, track tasks, and ensure smooth project delivery.

Class Diagram: Library Management System



Procedure:

Step 1: Create a Project

- Start Jira → Click **Projects** → **Create Project**.
- Select **Software Development** → **Kanban** → **Use Template**.
- Choose **Company Managed Project**.
- Enter project name **"Library Management System"** → **Create Project**.

Step 2: Create a Team

- Invite members by entering their email IDs.
- Create a project team and assign roles (e.g., Developer, Tester, Admin).
- Team created successfully.

Step 3: Create Epics

- **Books** (Add Books, Delete Books, Display Book Details, Inquiry Book).
- **Librarian** (Search Book, Verify Member, Order Books, Sell Books).
- **Publisher** (Add Publisher, Modify Publisher, Delete Publisher, Order Status).
- **User** (Add User, Delete User, Update Details, Book Purchase, Return Books, Pay Fine).

Step 4: Create Stories

- Under each Epic, create user stories:

1) **Books Epic**: Add new book, Delete book record, Display book details, Search/Inquiry book.
2) **Librarian Epic**: Search for a book, Verify member ID, Place book order, Sell books.
3) **Publisher Epic**: Add publisher details, Modify publisher record, Delete publisher, Track order status.
4) **User Epic**: Add new user, Delete user, Update user details, Purchase book, Return book, Pay fine

Step 5: Create Tasks

- Break stories into smaller technical tasks:

1) **Books**: Implement BookID, Validate Book Status, Update Price field.
2) **Librarian**: Create Search Function, Implement Verify Member Logic, Automate Order Placement.
3) **Publisher**: Build Publisher ID system, Enable Modify/Delete Operations, Integrate Order Tracking.
4) **User**: Generate UserID, Validate Phone Number, Build Fine Calculation, Implement Book Purchase Flow.

Step 6: Manage Kanban Board

- Use Kanban board columns: **To Do → In Progress → Done**.
- Add backlog items and assign them to members.
- Move tasks across the board as progress is made.
- Track workflow efficiency to ensure smooth delivery.

Step 7: Create Version and Release

Create Version:

- In the sidebar, select **More actions (•••)** and select **Releases**
- Click on the **"Create version"** button, typically found at the top right of the Releases page.
- **Enter Version Details:**

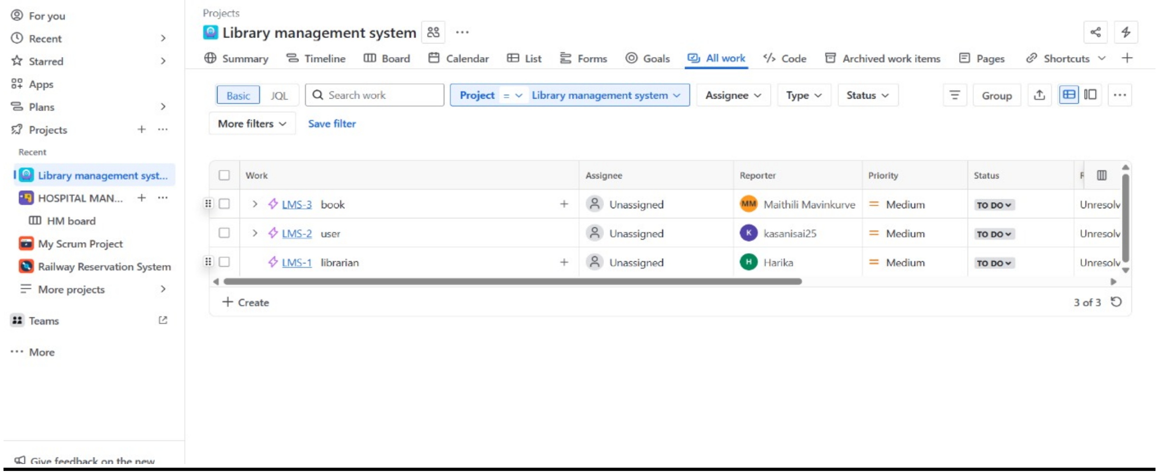
    1. **Name:** Provide an identifiable name for the version (e.g., *"Books Module v1.0"*, *"Librarian Module v1.0"*, *"Publisher Module v1.0"*, *"User Module v1.0"*).
    2. **Start date (Optional):** Specify the planned start date for work related to this version.
    3. **Release date (Optional):** Define the target release date for the version.
    4. **Description (Optional):** Add a brief description outlining the scope or purpose of the version (e.g., *"Includes epics: Add Books, Delete Books, Display Book Details, and Inquiry Book"* or *"Includes epics: Add User, Delete User, Purchase Book, and Pay Fine"*)

- **Save:** Click **"Save"** to create the new version.

Releasing a Version:

- **Initiate Release:** On the individual version's page, click the **"Release"** button. This button is usually prominent.
- **Confirm Release Details:** A dialog box will appear, allowing you to confirm or add details related to the release, such as the actual release date (e.g., *Releasing Publisher Module with Add/Modify/Delete Publisher and Track Order Status stories*).
- **Execute Release:** Click the **"Release"** button within the dialog to finalize the release of the version

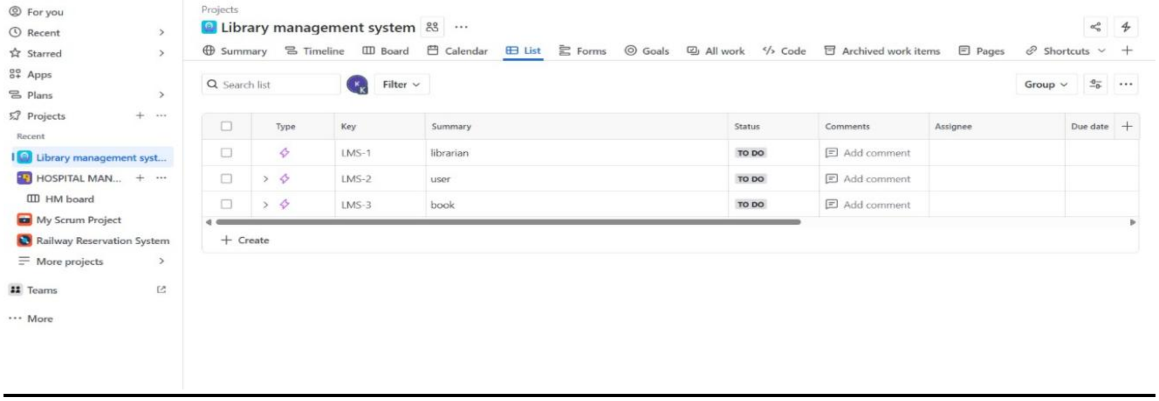
Work Flow Images:



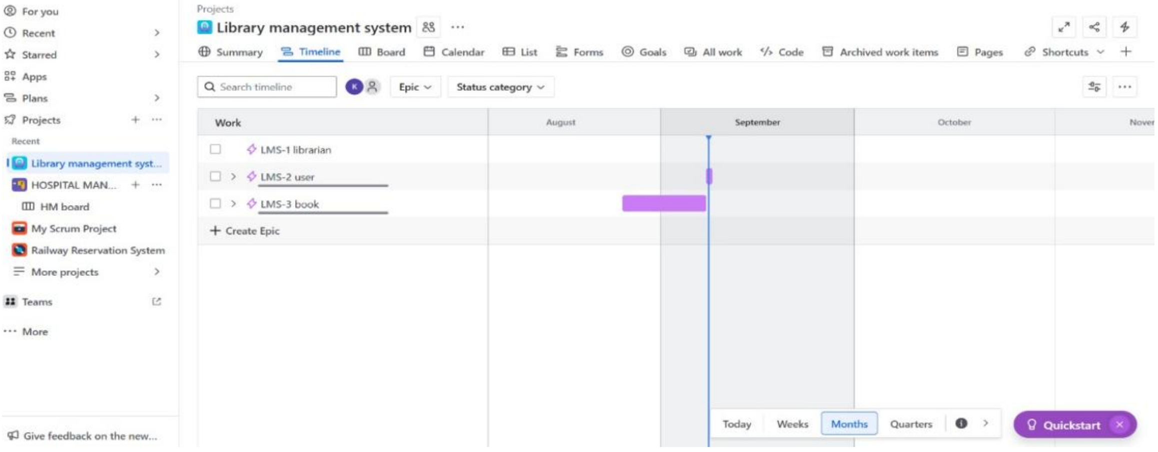
Project = Library management system

	Work		Assignee	Reporter	Priority	Status	
□	LMS-3	book	Unassigned	Maithili Mavinkurve	Medium	TO DO	Unresolv
□	LMS-2	user	Unassigned	kasanisai25	Medium	TO DO	Unresolv
□	LMS-1	librarian	Unassigned	Harika	Medium	TO DO	Unresolv

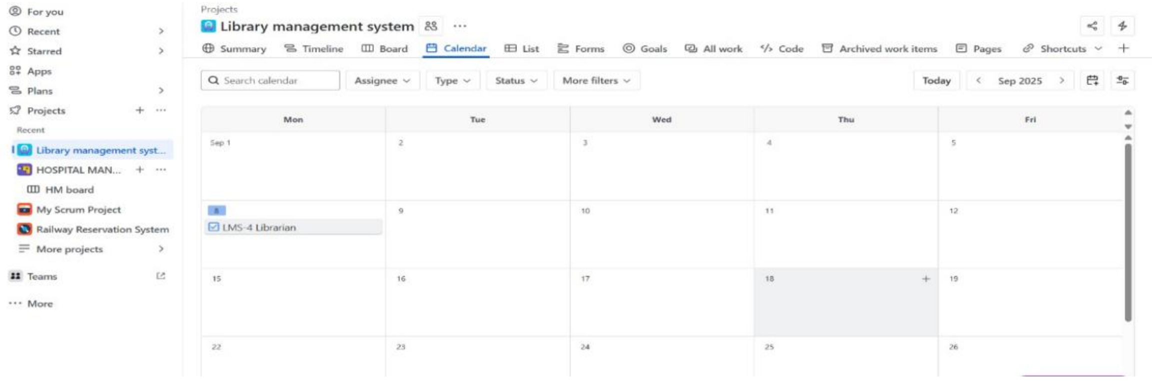
3 of 3



	Type	Key	Summary	Status	Comments	Assignee	Due date
□		LMS-1	librarian	TO DO	Add comment		
□	>	LMS-2	user	TO DO	Add comment		
□	>	LMS-3	book	TO DO	Add comment		



Work
- LMS-1 librarian
- LMS-2 user
- LMS-3 book
- Create Epic



Mon	Tue	Wed	Thu	Fri
Sep 1	2	3	4	5
8 ☑ LMS-4 Librarian	9	10	11	12
15	16	17	18	19
22	23	24	25	26

Result:

The Kanban Board for the Library Management System was successfully created using Jira, including projects, teams, epics, stories, tasks, and releases, allowing efficient management of library operations.

Experiment : Create a Scrum Board for Online Job Application System using Jira

Aim:

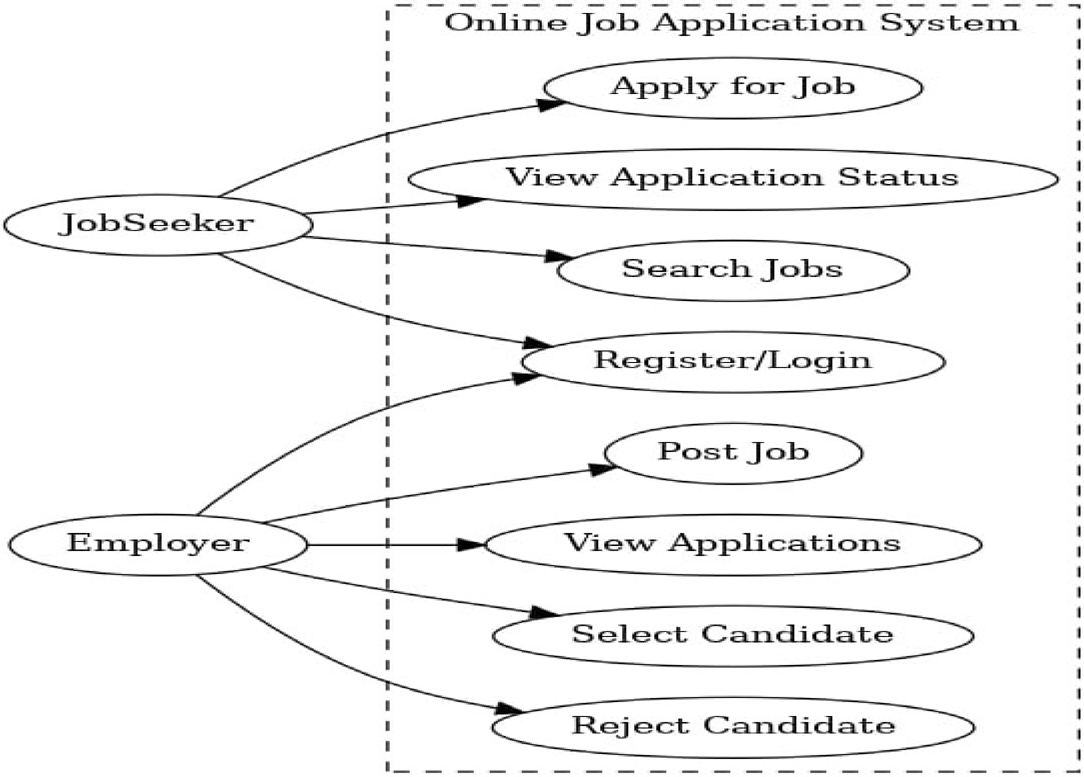
To create a Scrum Board for an Online Job Application System using Jira, by organizing workflow into sprints and managing product backlog items such as JobSeeker and Employer modules.

Description:

Scrum is an Agile framework that organizes work into time-boxed iterations (sprints). A Scrum Board helps visualize backlog items, sprint progress, and completed tasks.

For the Online Job Application System, Scrum methodology is applied to handle operations like job posting, job applications, candidate selection, and tracking application status. Jira Scrum Board allows tracking of **Epics, Stories, and Tasks** for each sprint, ensuring systematic and timely development of the system.

Use Case Diagram: Online Job Application System



Procedure:

Step 1: Create a Project

- Start Jira → Click **Projects → Create Project**.
- Select **Software Development → Scrum → Use Template**.
- Choose **Company Managed Project**.
- Enter project name **"Online Job Application System"** → Create Project.

Step 2: Create a Team

- Invite members by entering email IDs.
- Create a team and assign members (Developer, Tester, Admin).
- Team created successfully.

Step 3: Create Epics (Product Backlog)

- Create Epics for the main modules from the system:

  1. **JobSeeker** (Register/Login, Search Jobs, Apply for Job, View Application Status).
  2. **Employer** (Post Job, View Applications, Select Candidate, Reject Candidate).

Step 4: Create Stories

- Under each Epic, create User Stories:

JobSeeker Epic

1. Register/Login
2. Search Jobs
3. Apply for Job
4. View Application Status

- **Employer Epic**

1. Post Job
2. View Applications
3. Select Candidate
4. Reject Candidate

Step 5: Create Tasks

- Break stories into smaller technical tasks:

JobSeeker Epic

1. Create JobSeeker ID
2. Validate Login
3. Implement Job Search Filter
4. Develop Apply for Job Form
5. Track and Display Application Status

Employer Epic

6. Generate Employer ID
7. Build Post Job Form
8. Integrate Applications List View
9. Implement Candidate Selection Logic
10. Implement Candidate Rejection Function

Step 6: Manage Scrum Board (Sprint Backlog)

- Move backlog items into the sprint backlog.
- Start a new sprint (set sprint goal and duration).
- Move tasks across columns: **To Do → In Progress → Done**.
- Monitor **burndown chart** to track sprint progress.

Step 7: Create Version and Release

Create Version:

- In the sidebar, select **More actions (•••) → Releases**.
- Click on the **"Create version"** button.
- **Enter Version Details:**

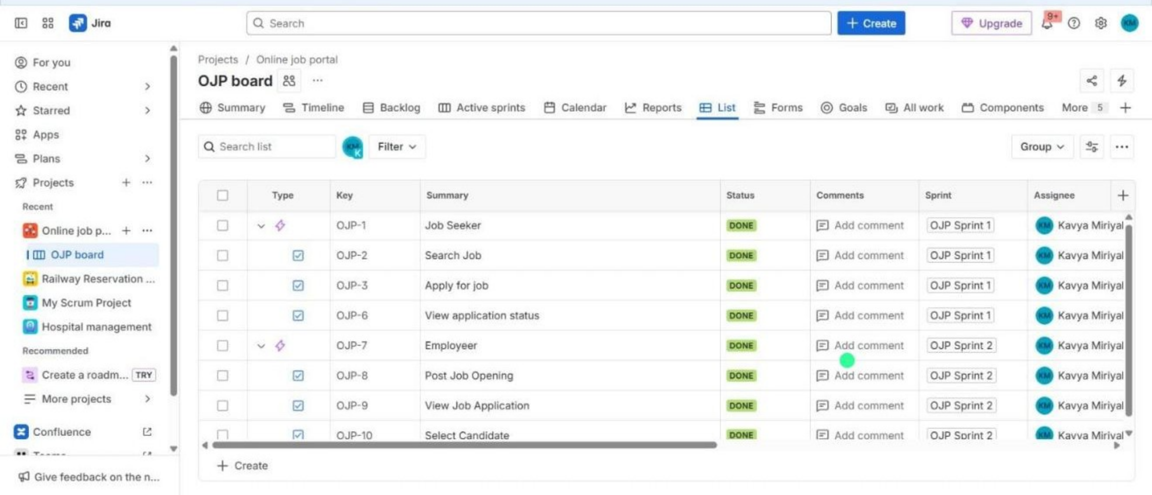
  1. **Name:** Provide an identifiable name (e.g., *"JobSeeker Module v1.0"*, *"Employer Module v1.0"*).
  2. **Start date (Optional):** Planned start date for the version.
  3. **Release date (Optional):** Define the target release date.
  4. **Description (Optional):** Briefly outline the scope (e.g., *"Includes epics: Register/Login, Search Jobs, Apply for Job, and View Application Status"*).

- **Save:** Click **"Save"** to create the version.

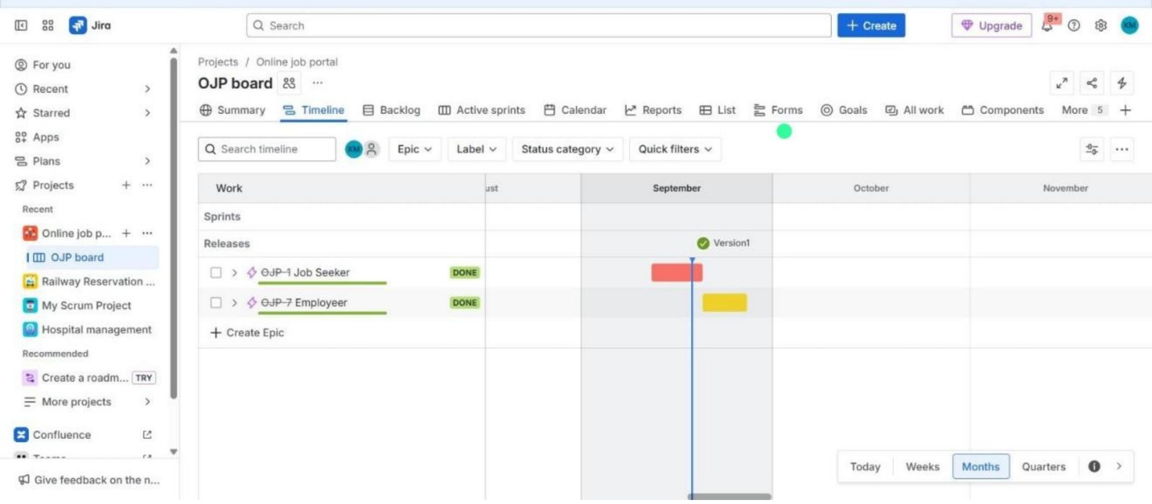
Releasing a Version:

- **Initiate Release:** On the version page, click **"Release"**.
- **Confirm Release Details:** Add actual release date (e.g., *Releasing Employer Module with Post Job, View Applications, Select and Reject Candidate stories*).
- **Execute Release:** Click **"Release"** to finalize.

<u>Work Flow Images:</u>

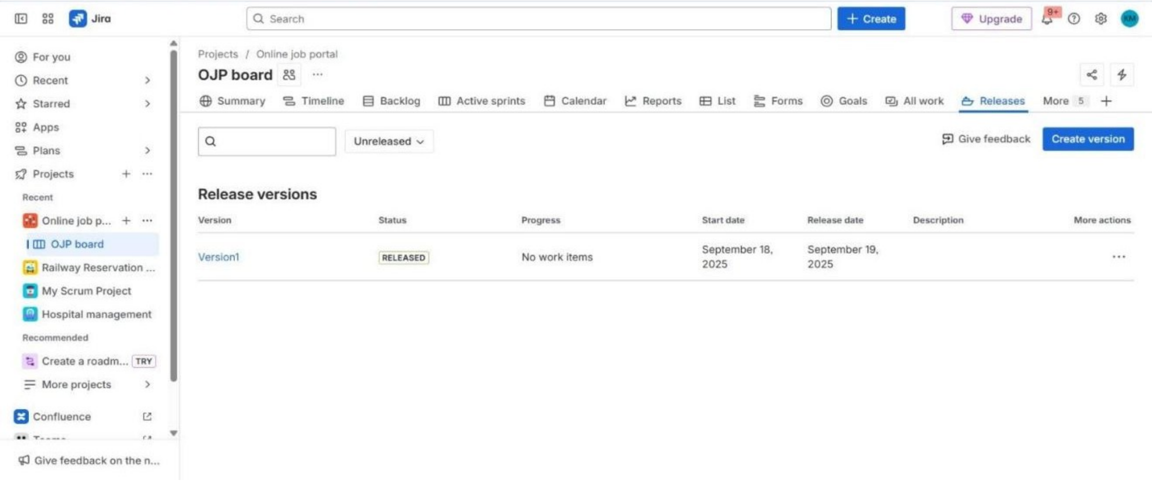


☐	Type	Key	Summary	Status	Comments	Sprint	Assignee
☐	⌄ ⚡	OJP-1	Job Seeker	DONE	Add comment	OJP Sprint 1	Kavya Miriyal
☐	☑	OJP-2	Search Job	DONE	Add comment	OJP Sprint 1	Kavya Miriyal
☐	☑	OJP-3	Apply for job	DONE	Add comment	OJP Sprint 1	Kavya Miriyal
☐	☑	OJP-6	View application status	DONE	Add comment	OJP Sprint 1	Kavya Miriyal
☐	⌄ ⚡	OJP-7	Employeer	DONE	Add comment	OJP Sprint 2	Kavya Miriyal
☐	☑	OJP-8	Post Job Opening	DONE	Add comment	OJP Sprint 2	Kavya Miriyal
☐	☑	OJP-9	View Job Application	DONE	Add comment	OJP Sprint 2	Kavya Miriyal
☐	☑	OJP-10	Select Candidate	DONE	Add comment	OJP Sprint 2	Kavya Miriyal



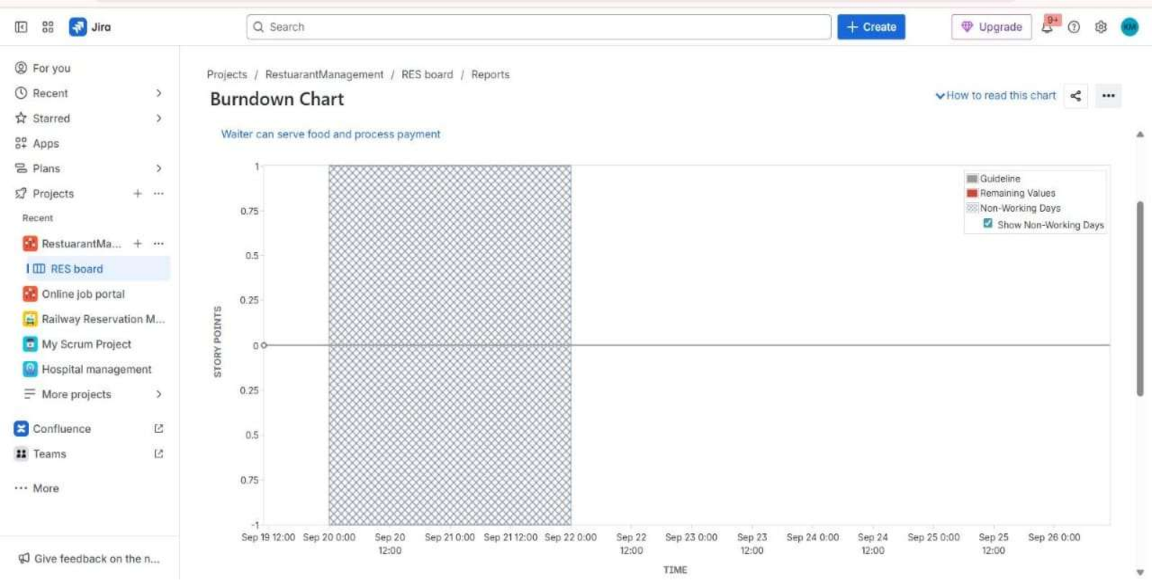
☐ > ⚡ ~~OJP-1~~ Job Seeker DONE

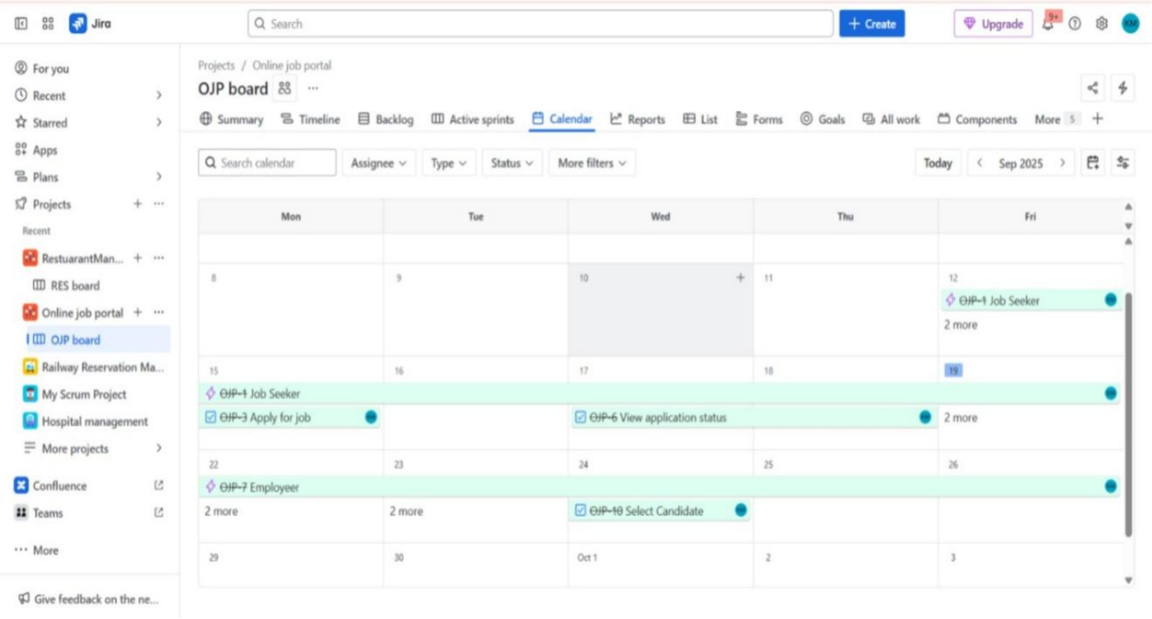
☐ > ⚡ ~~OJP-7~~ Employeer DONE



Release versions

Version	Status	Progress	Start date	Release date	Description	More actions
Version1	RELEASED	No work items	September 18, 2025	September 19, 2025		···





Result:

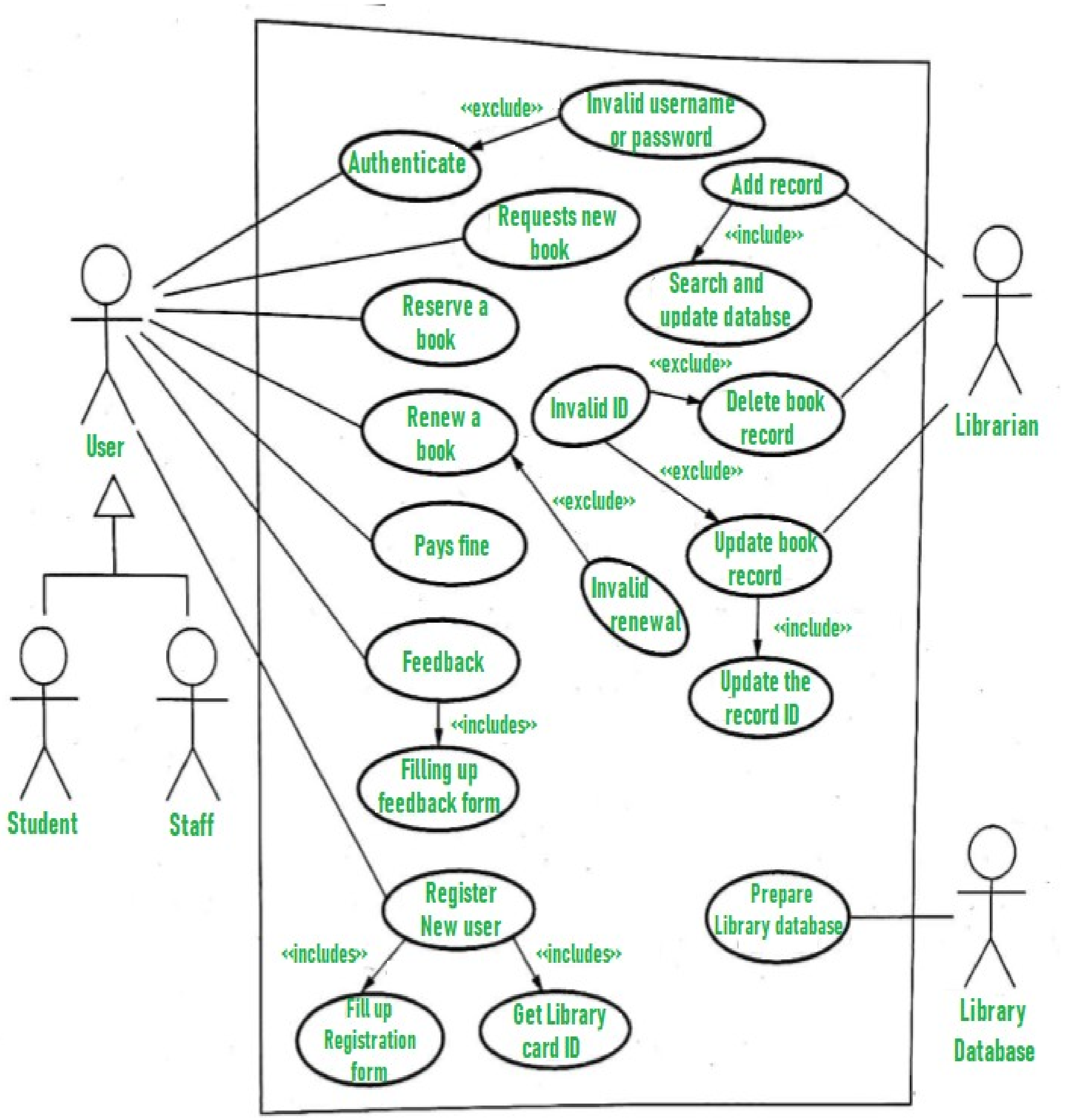
The Scrum Board for the Online Job Application System was successfully created using Jira. By dividing work into Epics, Stories, and Tasks aligned with the modules JobSeeker and Employer, the project can be developed efficiently with iterative sprint planning and tracking.

Experiment : **Create an Agile Development Plan for the Library Management System.**

Aim: To Create an Agile Development Plan for the Library Management System.

Procedure:

Use case diagram



Assuming 10 sprints with each sprint lasting 10 working days:

Sprint 1 (Days 1-10):

- Conduct project kickoff meeting
- Develop user stories and prioritize backlog
- Create wireframes for the main screens
- Set up development environment
- Begin development of user authentication and authorization system

Sprint 2 (Days 11-20):

- Complete development of user authentication and authorization system
- Begin development of book search functionality
- Begin development of book borrowing functionality
- Review wireframes with stakeholders and make necessary changes

Sprint 3 (Days 21-30):

- Complete development of book search functionality
- Complete development of book borrowing functionality
- Begin development of book return functionality
- Begin development of book reservation functionality

Sprint 4 (Days 31-40):

- Complete development of book return functionality
- Complete development of book reservation functionality
- Begin development of user profile functionality
- Begin development of book recommendation functionality

Sprint 5 (Days 41-50):

- Complete development of user profile functionality
- Complete development of book recommendation functionality
- Begin development of book review and rating functionality
- Begin development of administrative dashboard for librarians

Sprint 6 (Days 51-60):

- Complete development of book review and rating functionality
- Complete development of administrative dashboard forlibrarians
- Begin development of book purchase and inventory management functionality
- Begin development of fine management functionality

Sprint 7 (Days 61-70):

- Complete development of book purchase and inventory management functionality
- Complete development of fine management functionality
- Begin development of reporting and analytics functionality
- Begin development of mobile application

Sprint 8 (Days 71-80):

- Complete development of reporting and analytics functionality
- Complete development of mobile application
- Begin development of integration with external systems (e.g. payment gateway)

Sprint 9 (Days 81-90):

- Complete development of integration with external systems
- Begin testing and bug fixing
- Begin user acceptance testing
- Begin documentation and training material development

Sprint 10 (Days 91-100):

- Complete testing and bug fixing
- Complete user acceptance testing
- Complete documentation and training material development
- Conduct system deployment
- Conduct final review and retrospective

Result:

This is just an example of an Agile development plan for the Library system, and the actual plan may vary depending on the specific needs of the project and the team’s progress during each sprint

Experiment : Create a Kanban Board for Hospital Management System using Jira

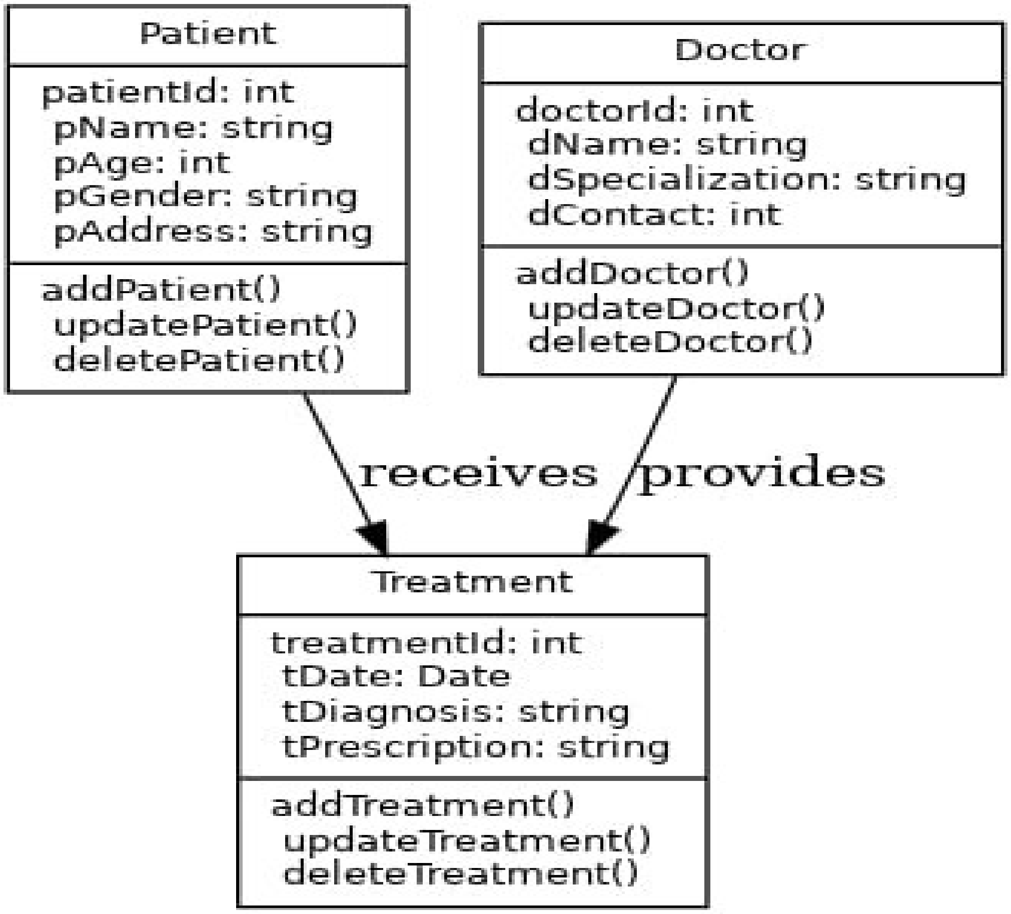
Aim:

To create a Kanban board for a Hospital Management System using Jira, by defining projects, teams, epics, stories, tasks, and releases.

Description:

Agile methodology emphasizes iterative and incremental development. **Kanban**, a popular Agile framework, helps visualize workflow, limit work-in-progress, and improve efficiency. Using **Jira**, hospital operations such as patient registration, doctor assignment, treatment, and billing can be managed effectively by organizing them into epics, stories, and tasks on a Kanban board.

Class Diagram: Hospital Management System



Procedure:

Step 1: Create a Project

- Start Jira → Click **Projects** → **Create Project**.
- Select **Software Development** → **Kanban** → **Use Template**.
- Choose **Company Managed Project**.
- Enter project name **“Hospital Management System”** → **Click Create Project**.
- Skip recommended options. → Project created successfully.

Step 2: Create a Team

a) **Invite Members**

- Enter email → Select name → Invite.

b) **Create a Team**

- Click **Teams** → **Create Team** → **Add Members** → **Click Create**.
- Team created successfully

Step 3: Create Epics

- Click **Create** → **Work Type: Epic**.
- Enter Epic name (e.g., *Patient*), description, select team, start & due dates.
- Click **Create**.
- Similarly create other Epics: *Doctor*, *Treatment*.

Step 4: Create Stories

- Click **Create** → **Work Type: Story**.
- Enter story name (e.g., *Add Patient*) under parent Epic (*Patient*)
- Add description, assign team, set dates → Click **Create**.
- Similarly create stories like *Update Patient*, *Delete Patient*, *Doctor Consultation*, *Treatment Details*.

Step 5: Create Tasks

- Click **Create** → **Work Type: Task**.
- Enter task name (e.g., *Patient ID*) under parent Epic.
- Add description, assign team, set dates → Click **Create**.
- Similarly create tasks under *Doctor* and *Treatment*.

Step 6: Create Version and Release

Create Version:

- In the sidebar, select **More actions (•••)** and select **Releases**
- Click on the **"Create version"** button, typically found at the top right of the Releases page.
- **Enter Version Details:**

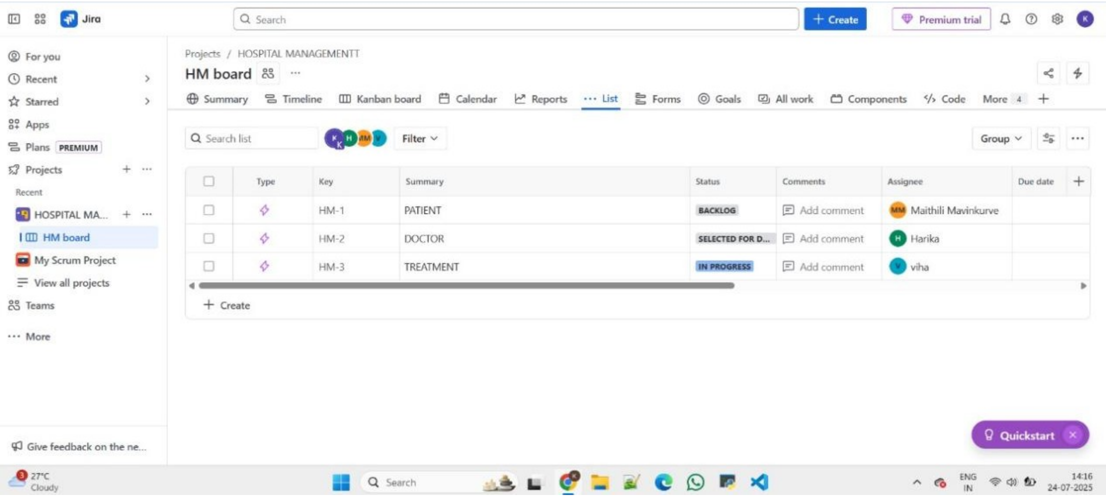
  1) **Name:** Provide an identifiable name for the version (e.g., *"Patient Module v1.0"*, *"Doctor Module v1.0"*, *"Treatment Module v1.0"*).
  2) **Start date (Optional):** Specify the planned start date for work related to this version.
  3) **Release date (Optional):** Define the target release date for the version.
  4) **Description (Optional):** Add a brief description outlining the scope or purpose of the version (e.g., *"Includes epics: Patient Registration, Update Patient, and Delete Patient"*).

- **Save:** Click **"Save"** to create the new version

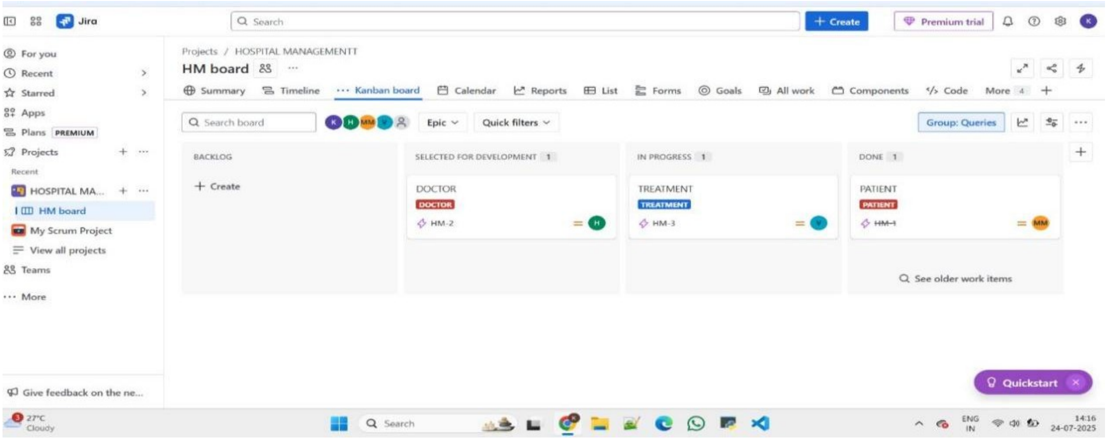
Releasing a Version:

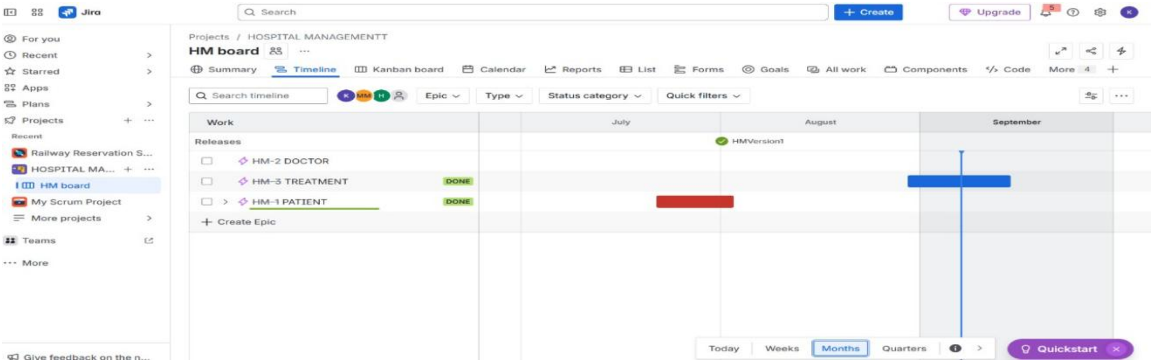
- **Initiate Release:** On the individual version's page, click the **"Release"** button. This button is usually prominent.
- **Confirm Release Details:** A dialog box will appear, allowing you to confirm or add details related to the release, such as the actual release date (e.g., *Releasing Doctor Consultation and Treatment Details*).
- **Execute Release:** Click the **"Release"** button within the dialog to finalize the release of the version.

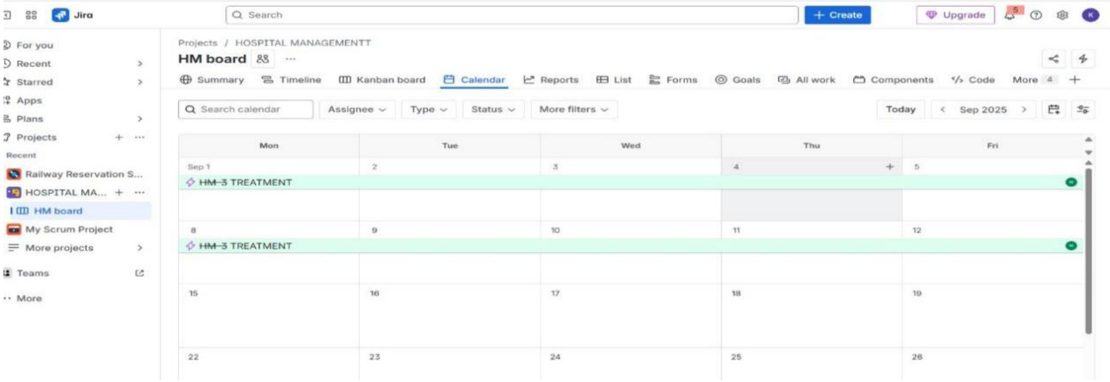
Work Flow Images:



☐	Type	Key	Summary	Status	Comments	Assignee	Due date
☐		HM-1	PATIENT	BACKLOG	Add comment	Maithili Mavinkurve	
☐		HM-2	DOCTOR	SELECTED FOR D...	Add comment	Harika	
☐		HM-3	TREATMENT	IN PROGRESS	Add comment	viha	







Result:

By using Jira and Kanban methodology, the hospital workflow can be efficiently managed. Tasks are visualized, responsibilities are assigned, and progress is easily tracked, ensuring better coordination and timely delivery of healthcare services.

Experiment : Create a Scrum Board for Railway Reservation System using Jira

Aim:

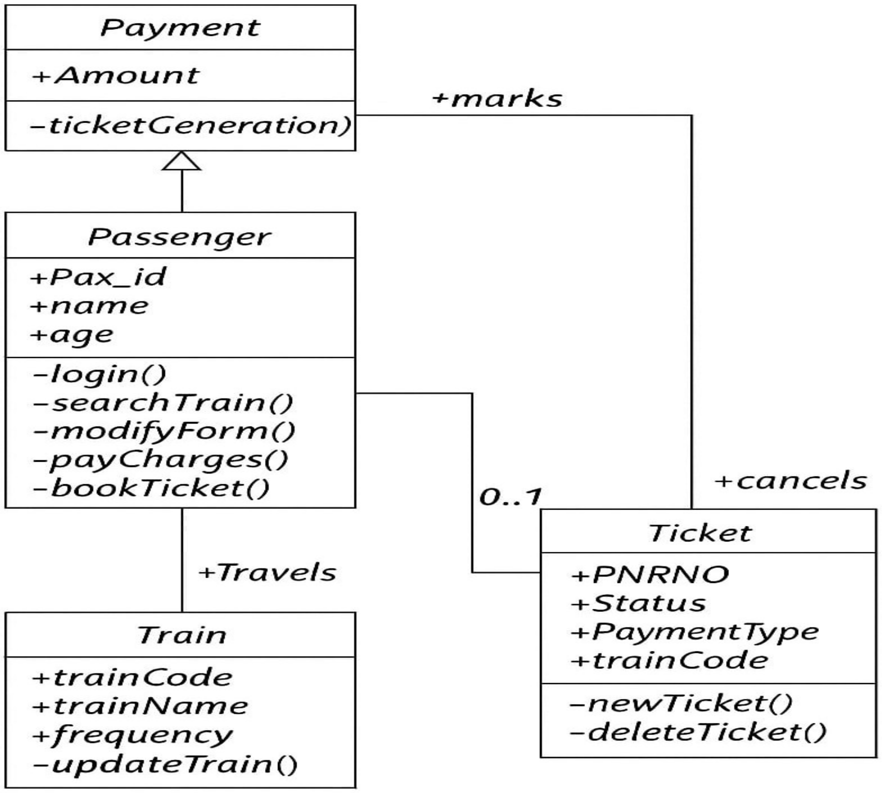
To create a **Scrum Board** for a Railway Reservation System using **Jira**, by organizing workflow into sprints and managing product backlog items such as Passenger, Train, Ticket, and Payment modules.

Description:

Scrum is an Agile framework that organizes work into **time-boxed iterations (sprints)**. A **Scrum Board** helps visualize backlog items, sprint progress, and completed tasks.

For a **Railway Reservation System**, Scrum methodology is applied to handle operations like passenger management, train scheduling, ticket booking, and payment processing. Jira Scrum Board allows tracking of **Epics, Stories, and Tasks** for each sprint, ensuring systematic and timely development of the system.

Class Diagram: Railway Reservation System



Procedure:

Step 1: Create a Project

- Start Jira → Click **Projects → Create Project**.
- Select **Software Development → Scrum → Use Template**.
- Choose **Company Managed Project**.
- Enter project name **“Railway Reservation System” → Create Project**.

Step 2: Create a Team

- Invite members by entering email IDs.
- Create a team and assign members.
- Team created successfully.

Step 3: Create Epics (Product Backlog)

- Create Epics for the main modules from the class diagram:

1) **Passenger** (login, search train, modify form, book ticket)
2) **Train** (update train, cancel train)
3) **Ticket** (new ticket, delete ticket, status updates)
4) **Payment** (pay charges, generate ticket, amount handling).

Step 4: Create Stories

- Under each Epic, create User Stories:

1) *Passenger Epic*: Add Passenger, Modify Passenger Form, Book Ticket.
2) *Train Epic*: Update Train Info, Cancel Train.
3) *Ticket Epic*: Generate New Ticket, Delete Ticket, Update Status.
4) *Payment Epic*: Make Payment, Confirm Ticket Generation.

Step 5: Create Tasks

- Break stories into smaller tasks, e.g.:

1) For *Passenger*: Create Passenger ID, Validate Login, Implement Search
2) For *Train*: Create Train Code, Update Frequency.
3) For *Ticket*: Assign PNR Number, Track Status.
4) For *Payment*: Calculate Amount, Validate Payment Type.

Step 6: Manage Scrum Board

- Move backlog items into the sprint backlog.

- Start a new sprint (set sprint goal and duration).
- Move tasks across columns: *To Do → In Progress → Done*.
- Monitor burndown chart to track sprint progress.

Step 7: Create Version and Release

Create Version:

- In the sidebar, select **More actions (•••)** and select **Releases**
- Click on the **"Create version"** button, typically found at the top right of the Releases page.
- **Enter Version Details:**

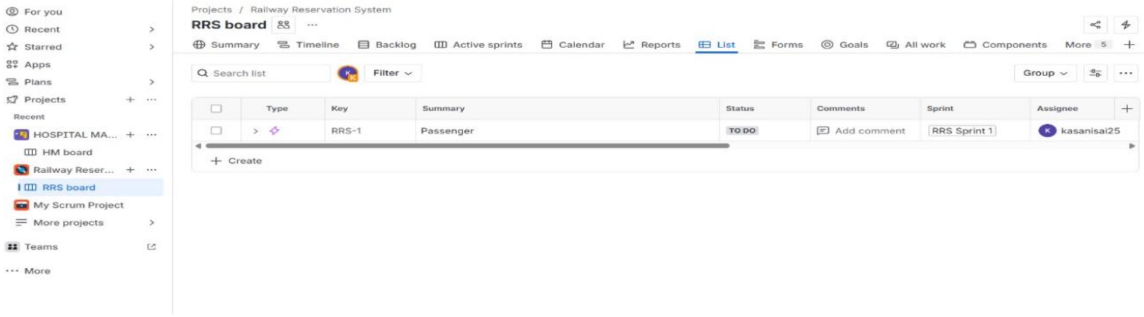
    1. **Name:** Provide an identifiable name for the version (e.g., *"Passenger Module v1.0"*, *"Train Module v1.0"*, *"Ticket Module v1.0"*, *"Payment Module v1.0"*).
    2. **Start date (Optional):** Specify the planned start date for work related to this version.
    3. **Release date (Optional):** Define the target release date for the version.
    4. **Description (Optional):** Add a brief description outlining the scope or purpose of the version (e.g., *"Includes epics: Add Passenger, Modify Passenger Form, and Book Ticket"* or *"Includes epics: Make Payment and Confirm Ticket Generation"*).

- **Save:** Click **"Save"** to create the new version.

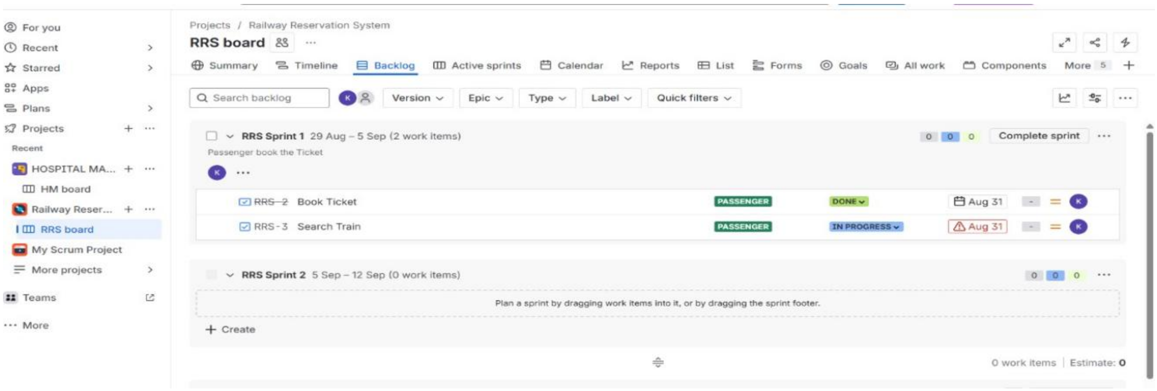
Releasing a Version:

- **Initiate Release:** On the individual version's page, click the **"Release"** button. This button is usually prominent.
- **Confirm Release Details:** A dialog box will appear, allowing you to confirm or add details related to the release, such as the actual release date (e.g., *Releasing Train Module with Update Train Info and Cancel Train stories*).
- **Execute Release:** Click the **"Release"** button within the dialog to finalize the release of the version.

Work Flow Images:



□	Type	Key	Summary	Status	Comments	Sprint	Assignee
□	>	RRS-1	Passenger	TO DO	Add comment	RRS Sprint 1	kasanisai25



Projects / Railway Reservation System

RRS board

RRS Sprint 1 29 Aug – 5 Sep (2 work items) 0 0 0

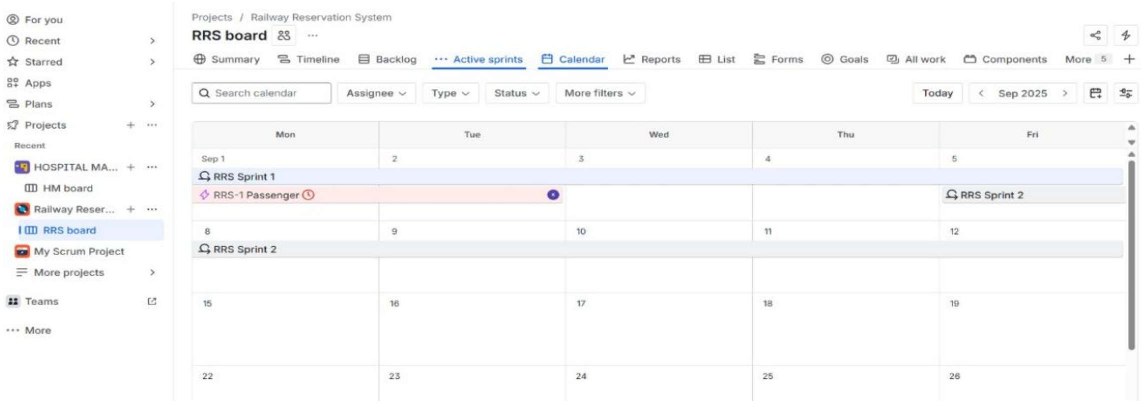
Passenger book the Ticket

~~RRS-2~~ Book Ticket	PASSENGER	DONE	Aug 31	K
RRS-3 Search Train	PASSENGER	IN PROGRESS	Aug 31	K

RRS Sprint 2 5 Sep – 12 Sep (0 work items) 0 0 0

Plan a sprint by dragging work items into it, or by dragging the sprint footer.

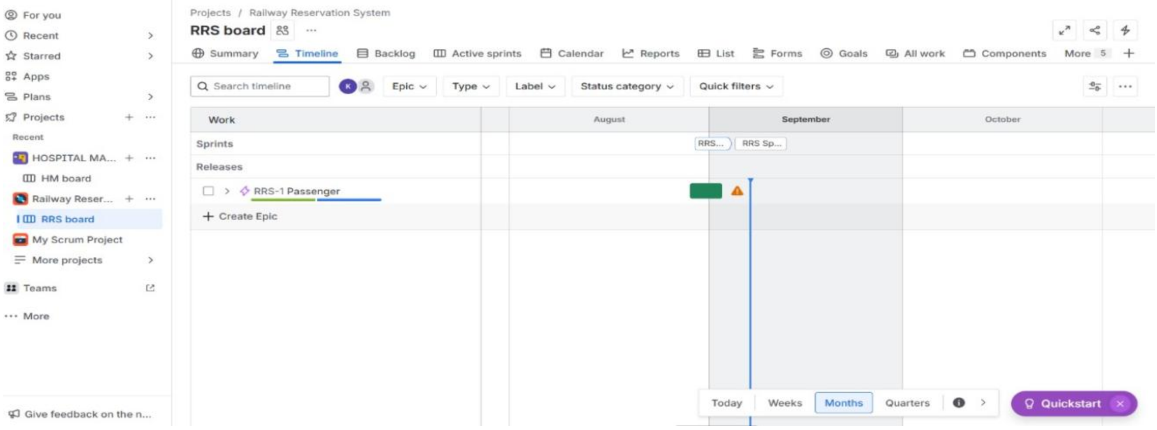
0 work items | Estimate: 0



Projects / Railway Reservation System

RRS board

Mon	Tue	Wed	Thu	Fri
Sep 1 RRS Sprint 1 RRS-1 Passenger	2	3	4	5 RRS Sprint 2
8 RRS Sprint 2	9	10	11	12
15	16	17	18	19
22	23	24	25	26



Projects / Railway Reservation System

RRS board

Work	August	September	October
Sprints		RRS... RRS Sp...	
Releases			
RRS-1 Passenger			
+ Create Epic			

Result:

The Scrum Board for the Railway Reservation System was successfully created using Jira. By dividing work into Epics, Stories, and Tasks aligned with the modules *Passenger, Train, Ticket, and Payment*, the project can be developed efficiently with iterative sprint planning and tracking.

Experiment : Create a Kanban Board for Library Management System using Jira

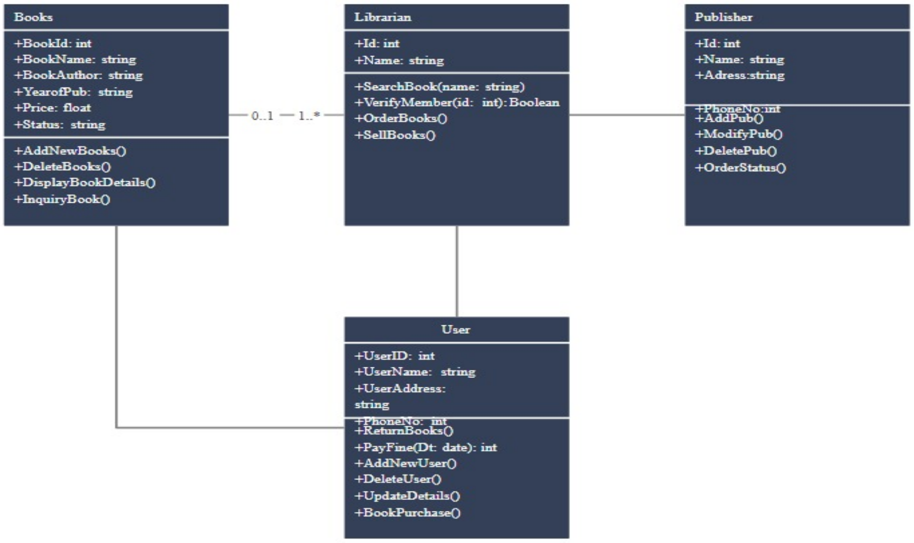
Aim:

To create a **Kanban Board** for a Library Management System using **Jira**, by defining epics, stories, tasks, and releases to manage the library workflow effectively.

Description:

The Library Management System is designed to manage books, librarians, publishers, and users efficiently. The system keeps track of available books, manages user memberships, handles fines, and facilitates book purchase and sales. By implementing the Kanban methodology in Jira, we can visualize the workflow, track tasks, and ensure smooth project delivery.

Class Diagram: Library Management System



Procedure:

Step 1: Create a Project

- Start Jira → Click **Projects** → **Create Project**.
- Select **Software Development** → **Kanban** → **Use Template**.
- Choose **Company Managed Project**.
- Enter project name **“Library Management System”** → **Create Project**.

Step 2: Create a Team

- Invite members by entering their email IDs.
- Create a project team and assign roles (e.g., Developer, Tester, Admin).
- Team created successfully.

Step 3: Create Epics

- **Books** (Add Books, Delete Books, Display Book Details, Inquiry Book).
- **Librarian** (Search Book, Verify Member, Order Books, Sell Books).
- **Publisher** (Add Publisher, Modify Publisher, Delete Publisher, Order Status).
- **User** (Add User, Delete User, Update Details, Book Purchase, Return Books, Pay Fine).

Step 4: Create Stories

- Under each Epic, create user stories:

1) **Books Epic**: Add new book, Delete book record, Display book details, Search/Inquiry book.
2) **Librarian Epic**: Search for a book, Verify member ID, Place book order, Sell books.
3) **Publisher Epic**: Add publisher details, Modify publisher record, Delete publisher, Track order status.
4) **User Epic**: Add new user, Delete user, Update user details, Purchase book, Return book, Pay fine

Step 5: Create Tasks

- Break stories into smaller technical tasks:

1) **Books**: Implement BookID, Validate Book Status, Update Price field.
2) **Librarian**: Create Search Function, Implement Verify Member Logic, Automate Order Placement.
3) **Publisher**: Build Publisher ID system, Enable Modify/Delete Operations, Integrate Order Tracking.
4) **User**: Generate UserID, Validate Phone Number, Build Fine Calculation, Implement Book Purchase Flow.

Step 6: Manage Kanban Board

- Use Kanban board columns: **To Do → In Progress → Done**.
- Add backlog items and assign them to members.
- Move tasks across the board as progress is made.
- Track workflow efficiency to ensure smooth delivery.

Step 7: Create Version and Release

Create Version:

- In the sidebar, select **More actions (•••)** and select **Releases**
- Click on the **"Create version"** button, typically found at the top right of the Releases page.
- **Enter Version Details:**

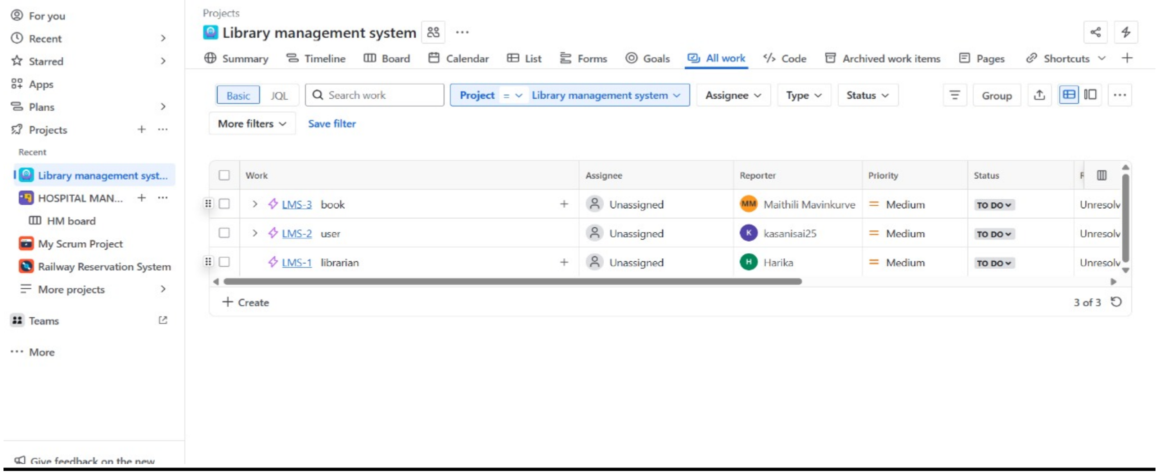
  1. **Name:** Provide an identifiable name for the version (e.g., *"Books Module v1.0"*, *"Librarian Module v1.0"*, *"Publisher Module v1.0"*, *"User Module v1.0"*).
  2. **Start date (Optional):** Specify the planned start date for work related to this version.
  3. **Release date (Optional):** Define the target release date for the version.
  4. **Description (Optional):** Add a brief description outlining the scope or purpose of the version (e.g., *"Includes epics: Add Books, Delete Books, Display Book Details, and Inquiry Book"* or *"Includes epics: Add User, Delete User, Purchase Book, and Pay Fine"*)

- **Save:** Click **"Save"** to create the new version.

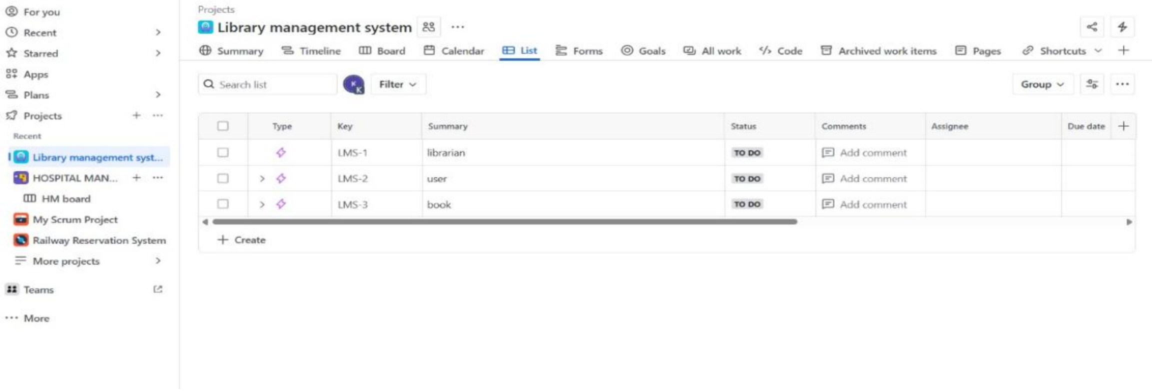
Releasing a Version:

- **Initiate Release:** On the individual version's page, click the **"Release"** button. This button is usually prominent.
- **Confirm Release Details:** A dialog box will appear, allowing you to confirm or add details related to the release, such as the actual release date (e.g., *Releasing Publisher Module with Add/Modify/Delete Publisher and Track Order Status stories*).
- **Execute Release:** Click the **"Release"** button within the dialog to finalize the release of the version

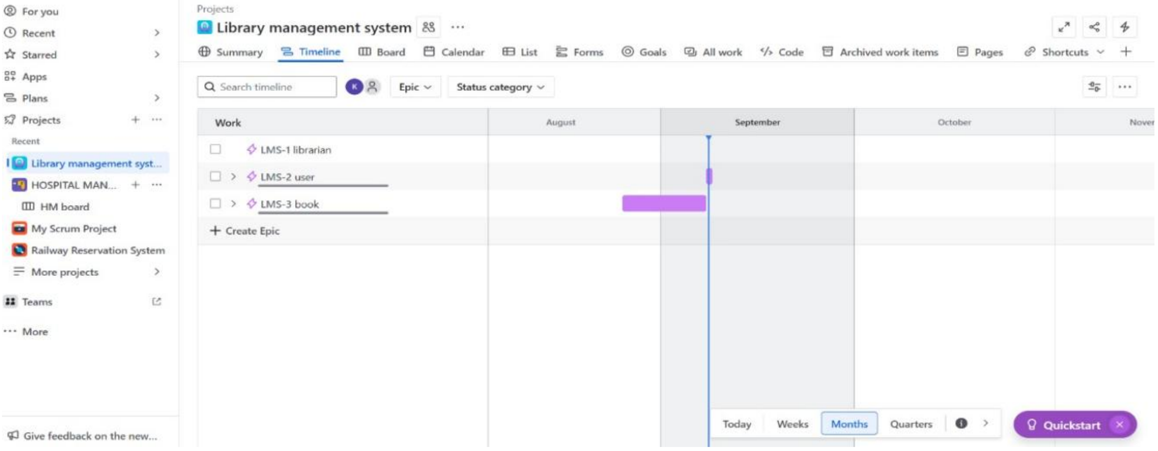
Work Flow Images:



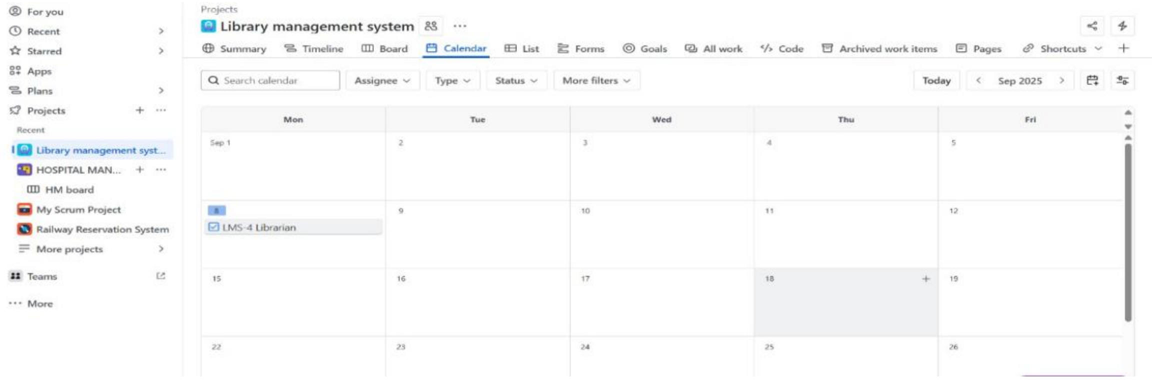
	Work	Assignee	Reporter	Priority	Status	R
□	LMS-3 book	Unassigned	Maithili Mavinkurve	Medium	TO DO	Unresolv
□	LMS-2 user	Unassigned	kasanisai25	Medium	TO DO	Unresolv
□	LMS-1 librarian	Unassigned	Harika	Medium	TO DO	Unresolv



Type	Key	Summary	Status	Comments	Assignee	Due date
	LMS-1	librarian	TO DO	Add comment		
	LMS-2	user	TO DO	Add comment		
	LMS-3	book	TO DO	Add comment		



LMS-1 librarian
LMS-2 user
LMS-3 book



LMS-4 Librarian

Result:

The Kanban Board for the Library Management System was successfully created using Jira, including projects, teams, epics, stories, tasks, and releases, allowing efficient management of library operations.

Experiment : Create a Scrum Board for Online Job Application System using Jira

Aim:

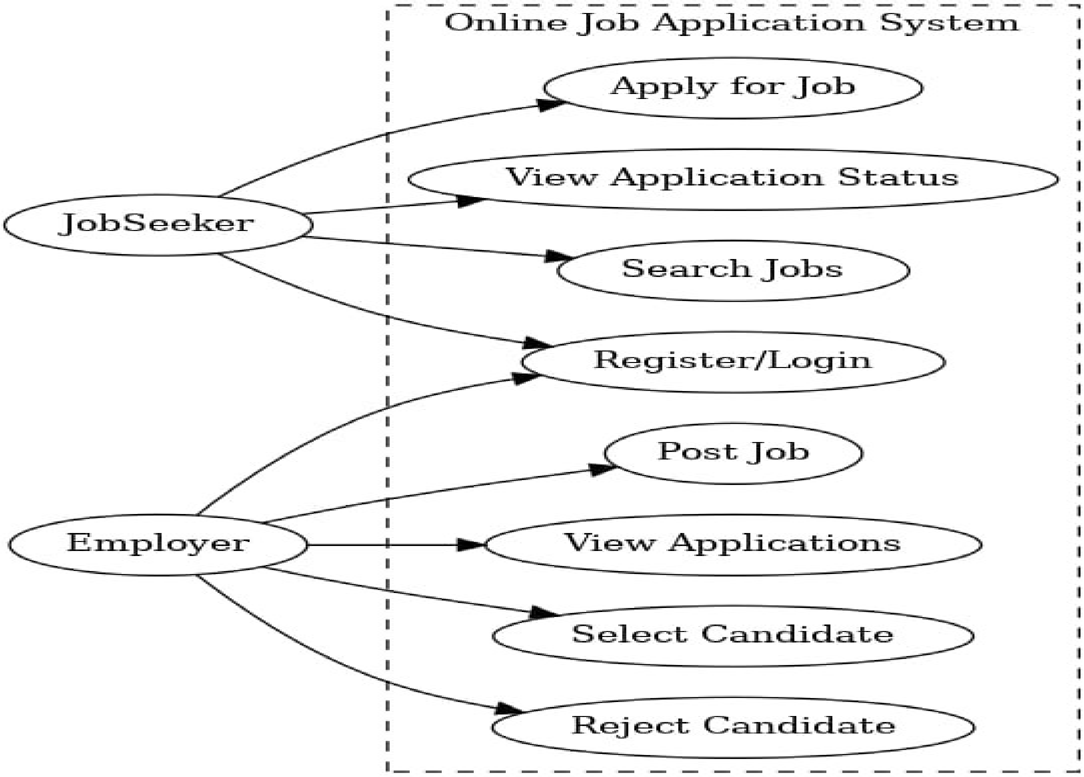
To create a Scrum Board for an Online Job Application System using Jira, by organizing workflow into sprints and managing product backlog items such as JobSeeker and Employer modules.

Description:

Scrum is an Agile framework that organizes work into time-boxed iterations (sprints). A Scrum Board helps visualize backlog items, sprint progress, and completed tasks.

For the Online Job Application System, Scrum methodology is applied to handle operations like job posting, job applications, candidate selection, and tracking application status. Jira Scrum Board allows tracking of **Epics, Stories, and Tasks** for each sprint, ensuring systematic and timely development of the system.

Use Case Diagram: Online Job Application System



Procedure:

Step 1: Create a Project

- Start Jira → Click **Projects** → **Create Project**.
- Select **Software Development** → **Scrum** → **Use Template**.
- Choose **Company Managed Project**.
- Enter project name **"Online Job Application System"** → Create Project.

Step 2: Create a Team

- Invite members by entering email IDs.
- Create a team and assign members (Developer, Tester, Admin).
- Team created successfully.

Step 3: Create Epics (Product Backlog)

- Create Epics for the main modules from the system:

  1. **JobSeeker** (Register/Login, Search Jobs, Apply for Job, View Application Status).
  2. **Employer** (Post Job, View Applications, Select Candidate, Reject Candidate).

Step 4: Create Stories

- Under each Epic, create User Stories:

JobSeeker Epic

1. Register/Login
2. Search Jobs
3. Apply for Job
4. View Application Status

- **Employer Epic**

1. Post Job
2. View Applications
3. Select Candidate
4. Reject Candidate

Step 5: Create Tasks

- Break stories into smaller technical tasks:

JobSeeker Epic

1. Create JobSeeker ID
2. Validate Login
3. Implement Job Search Filter
4. Develop Apply for Job Form
5. Track and Display Application Status

Employer Epic

6. Generate Employer ID
7. Build Post Job Form
8. Integrate Applications List View
9. Implement Candidate Selection Logic
10. Implement Candidate Rejection Function

Step 6: Manage Scrum Board (Sprint Backlog)

- Move backlog items into the sprint backlog.
- Start a new sprint (set sprint goal and duration).
- Move tasks across columns: **To Do → In Progress → Done**.
- Monitor **burndown chart** to track sprint progress.

Step 7: Create Version and Release

Create Version:

- In the sidebar, select **More actions (•••) → Releases**.
- Click on the **"Create version"** button.
- **Enter Version Details:**

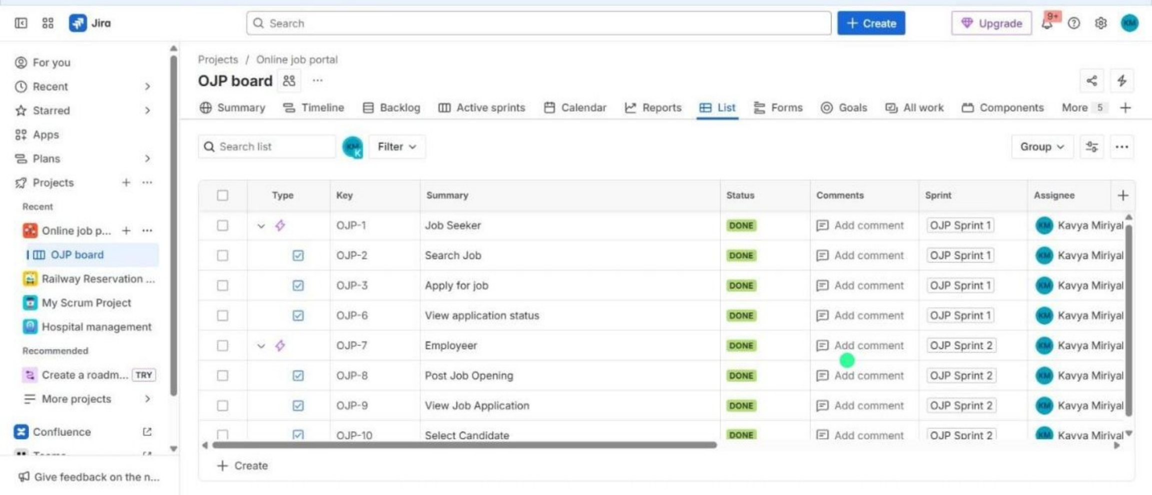
  1. **Name:** Provide an identifiable name (e.g., *"JobSeeker Module v1.0"*, *"Employer Module v1.0"*).
  2. **Start date (Optional):** Planned start date for the version.
  3. **Release date (Optional):** Define the target release date.
  4. **Description (Optional):** Briefly outline the scope (e.g., *"Includes epics: Register/Login, Search Jobs, Apply for Job, and View Application Status"*).

- **Save:** Click **"Save"** to create the version.

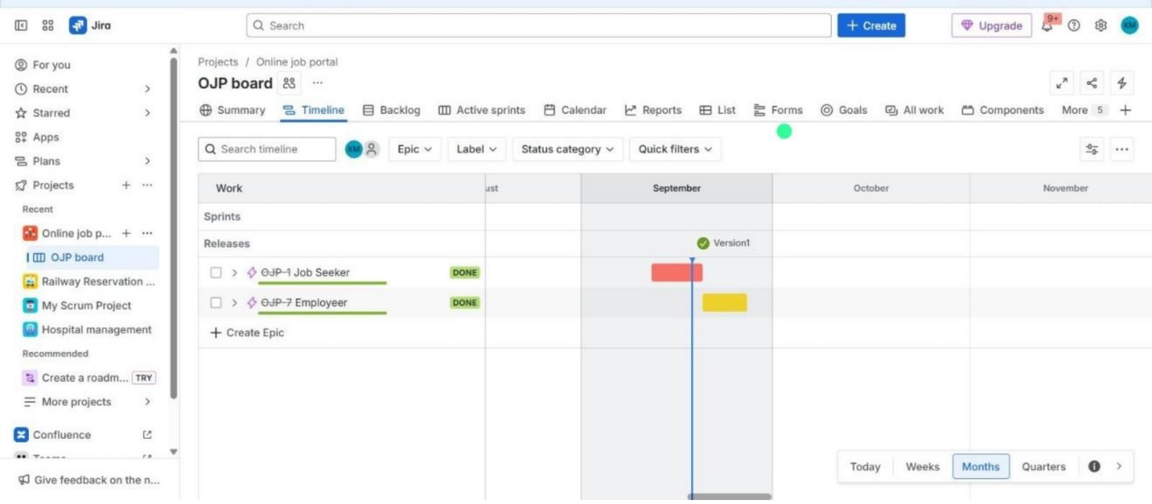
Releasing a Version:

- **Initiate Release:** On the version page, click **"Release"**.
- **Confirm Release Details:** Add actual release date (e.g., *Releasing Employer Module with Post Job, View Applications, Select and Reject Candidate stories*).
- **Execute Release:** Click **"Release"** to finalize.

Work Flow Images:

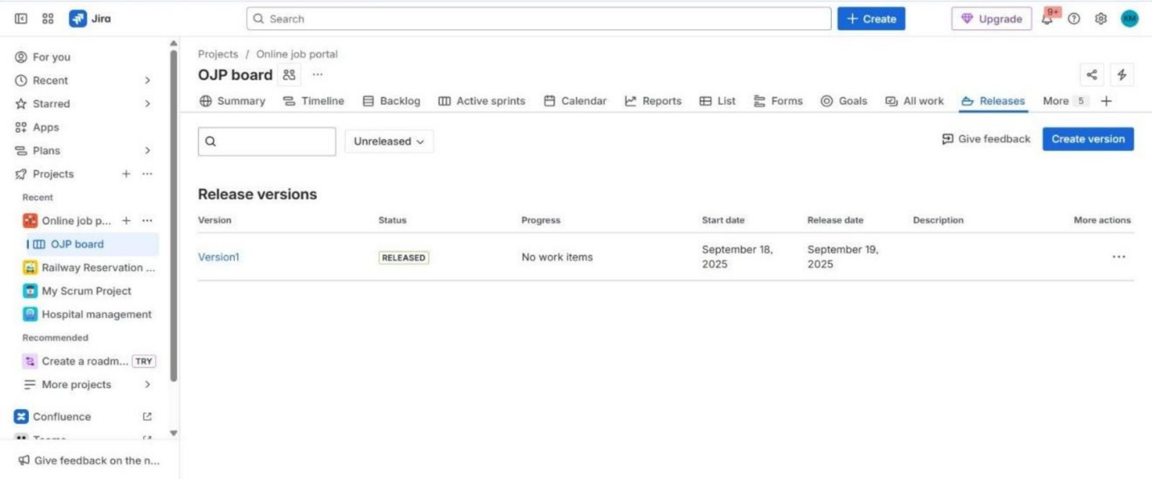


☐	Type	Key	Summary	Status	Comments	Sprint	Assignee
☐	⌄ ⚡	OJP-1	Job Seeker	DONE	Add comment	OJP Sprint 1	Kavya Miriyal
☐	☑	OJP-2	Search Job	DONE	Add comment	OJP Sprint 1	Kavya Miriyal
☐	☑	OJP-3	Apply for job	DONE	Add comment	OJP Sprint 1	Kavya Miriyal
☐	☑	OJP-6	View application status	DONE	Add comment	OJP Sprint 1	Kavya Miriyal
☐	⌄ ⚡	OJP-7	Employeer	DONE	Add comment	OJP Sprint 2	Kavya Miriyal
☐	☑	OJP-8	Post Job Opening	DONE	Add comment	OJP Sprint 2	Kavya Miriyal
☐	☑	OJP-9	View Job Application	DONE	Add comment	OJP Sprint 2	Kavya Miriyal
☐	☑	OJP-10	Select Candidate	DONE	Add comment	OJP Sprint 2	Kavya Miriyal



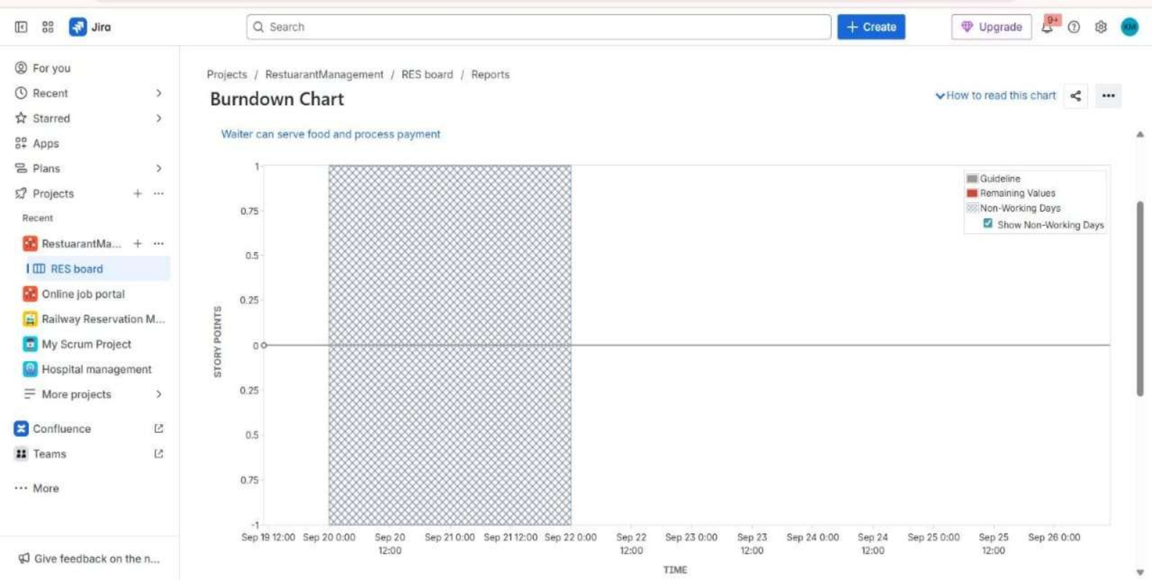
☐ > ~~OJP-1~~ Job Seeker DONE

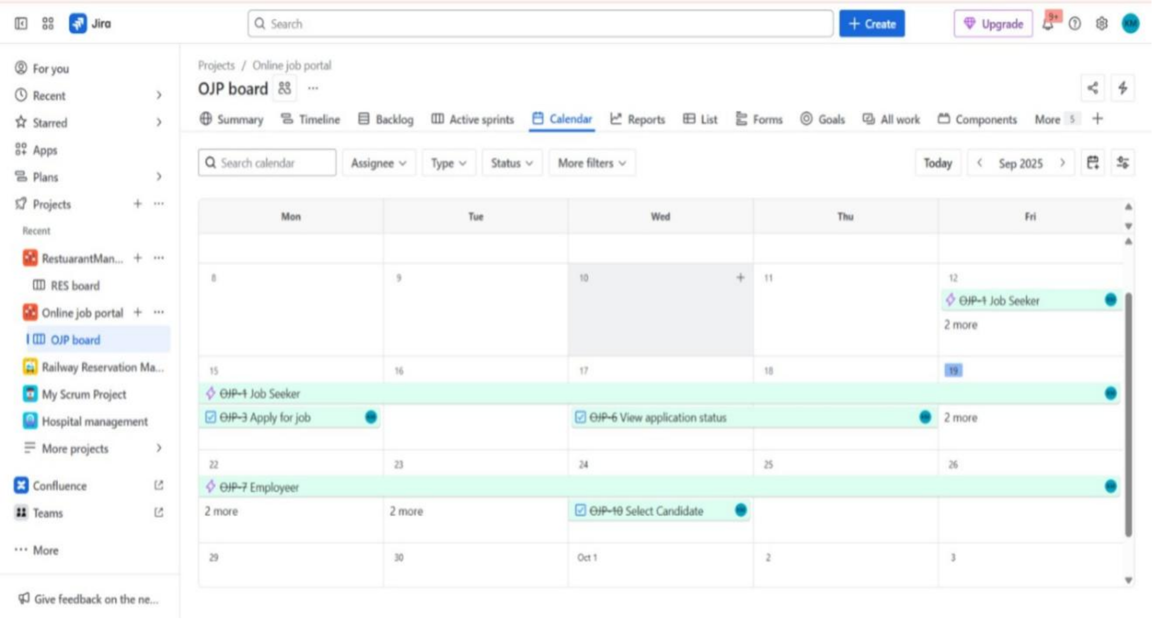
☐ > ~~OJP-7~~ Employeer DONE



Release versions

Version	Status	Progress	Start date	Release date	Description	More actions
Version1	RELEASED	No work items	September 18, 2025	September 19, 2025		···





Mon	Tue	Wed	Thu	Fri
8	9	10	11	12 ~~OJP-1~~ Job Seeker 2 more
15 ~~OJP-1~~ Job Seeker ~~OJP-3~~ Apply for job	16	17 ~~OJP-6~~ View application status	18	19 2 more
22 ~~OJP-7~~ Employeer 2 more	23 2 more	24 ~~OJP-10~~ Select Candidate	25	26
29	30	Oct 1	2	3

Result:

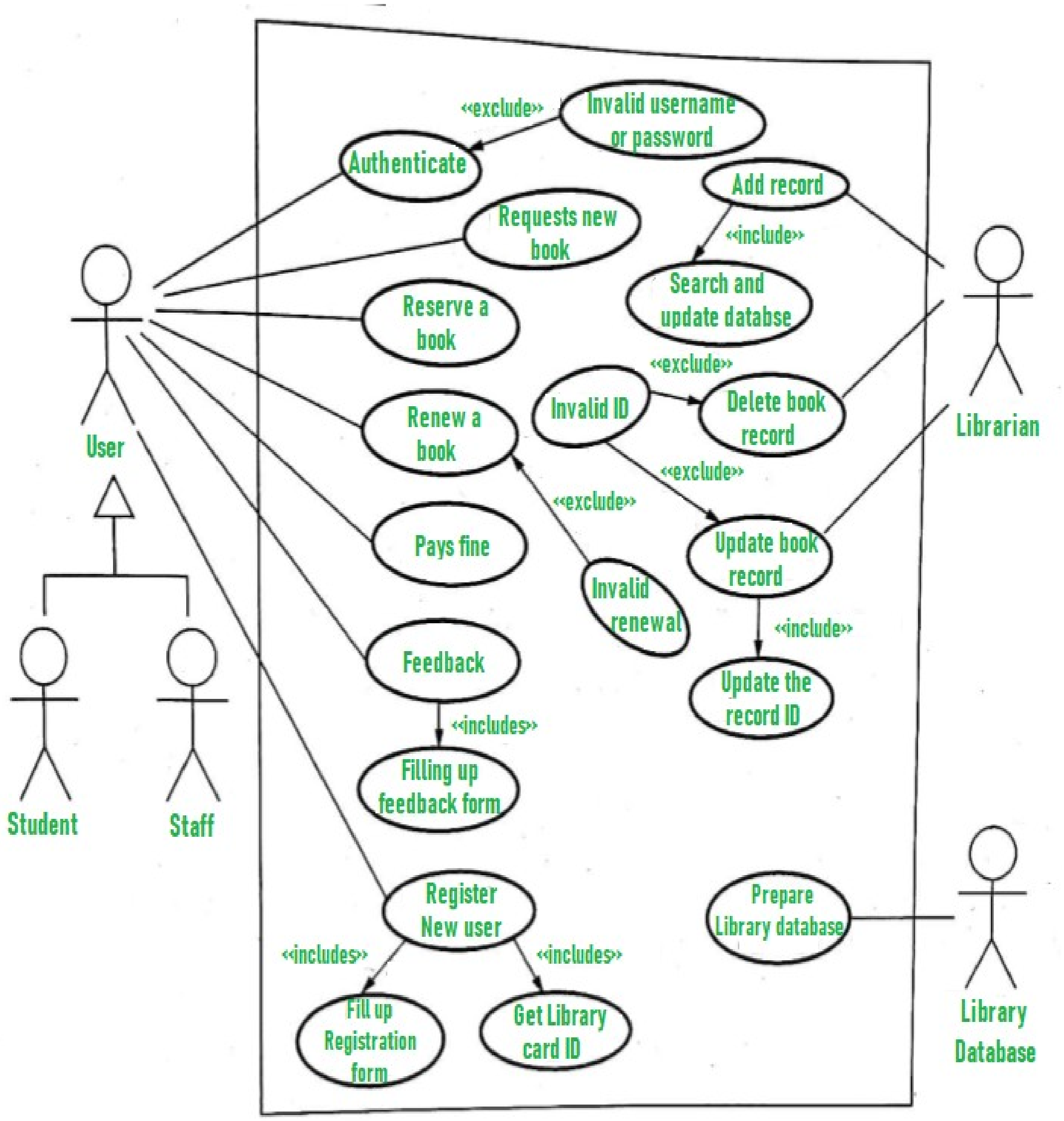
The Scrum Board for the Online Job Application System was successfully created using Jira. By dividing work into Epics, Stories, and Tasks aligned with the modules JobSeeker and Employer, the project can be developed efficiently with iterative sprint planning and tracking.

Experiment : **Create an Agile Development Plan for the Library Management System.**

Aim: **To Create an Agile Development Plan for the Library Management System.**

Procedure:

Use case diagram



Assuming 10 sprints with each sprint lasting 10 working days:

Sprint 1 (Days 1-10):

- Conduct project kickoff meeting
- Develop user stories and prioritize backlog
- Create wireframes for the main screens
- Set up development environment
- Begin development of user authentication and authorization system

Sprint 2 (Days 11-20):

- Complete development of user authentication and authorization system
- Begin development of book search functionality
- Begin development of book borrowing functionality
- Review wireframes with stakeholders and make necessary changes

Sprint 3 (Days 21-30):

- Complete development of book search functionality
- Complete development of book borrowing functionality
- Begin development of book return functionality
- Begin development of book reservation functionality

Sprint 4 (Days 31-40):

- Complete development of book return functionality
- Complete development of book reservation functionality
- Begin development of user profile functionality
- Begin development of book recommendation functionality

Sprint 5 (Days 41-50):

- Complete development of user profile functionality
- Complete development of book recommendation functionality
- Begin development of book review and rating functionality
- Begin development of administrative dashboard for librarians

Sprint 6 (Days 51-60):

- Complete development of book review and rating functionality
- Complete development of administrative dashboard forlibrarians
- Begin development of book purchase and inventory management functionality
- Begin development of fine management functionality

Sprint 7 (Days 61-70):

- Complete development of book purchase and inventory management functionality
- Complete development of fine management functionality
- Begin development of reporting and analytics functionality
- Begin development of mobile application

Sprint 8 (Days 71-80):

- Complete development of reporting and analytics functionality
- Complete development of mobile application
- Begin development of integration with external systems (e.g. payment gateway)

Sprint 9 (Days 81-90):

- Complete development of integration with external systems
- Begin testing and bug fixing
- Begin user acceptance testing
- Begin documentation and training material development

Sprint 10 (Days 91-100):

- Complete testing and bug fixing
- Complete user acceptance testing
- Complete documentation and training material development
- Conduct system deployment
- Conduct final review and retrospective

Result:

This is just an example of an Agile development plan for the Library system, and the actual plan may vary depending on the specific needs of the project and the team's progress during each sprint