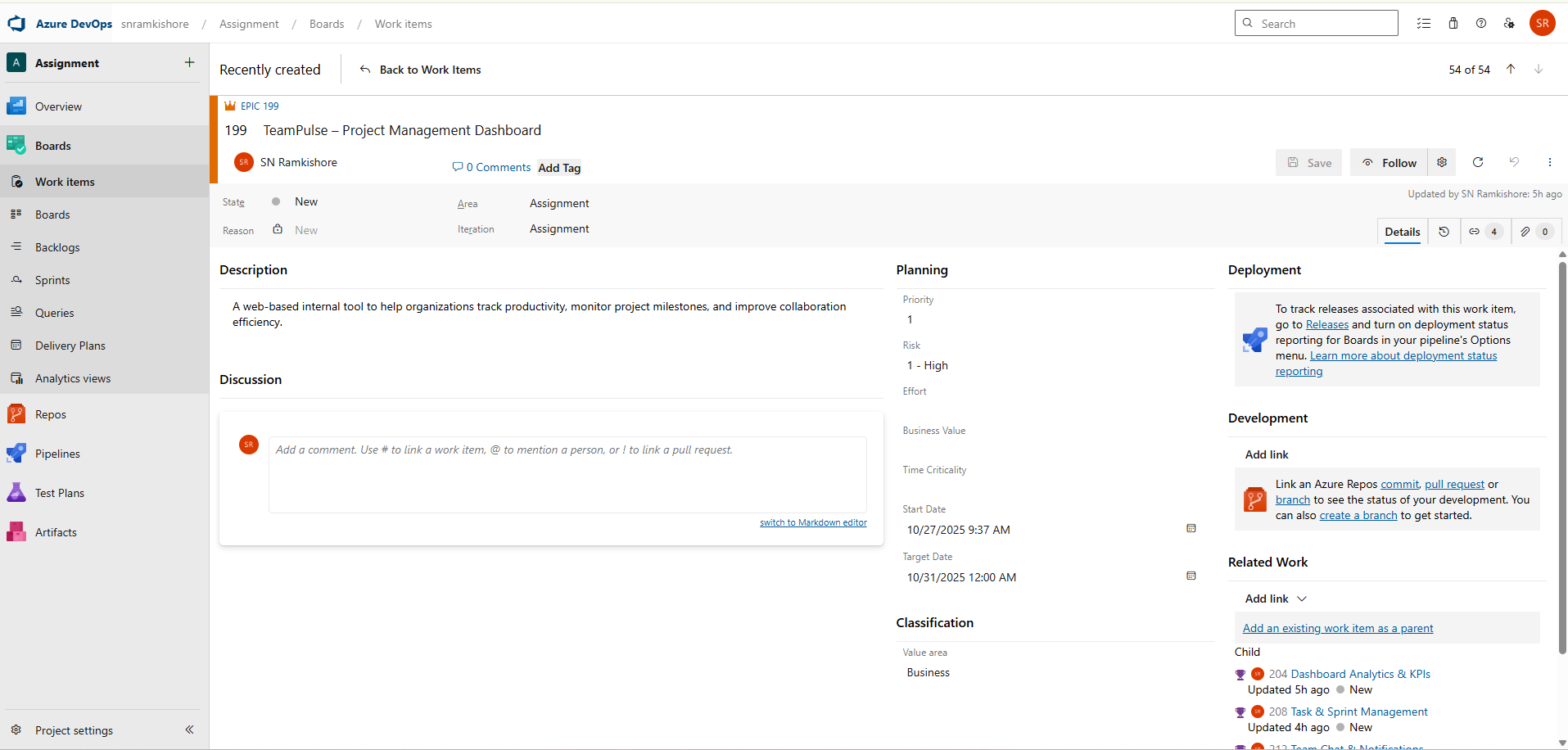
**Azure DevOps – Theoretical Assignment**

**Epic**

TeamPulse – Project Management Dashboard

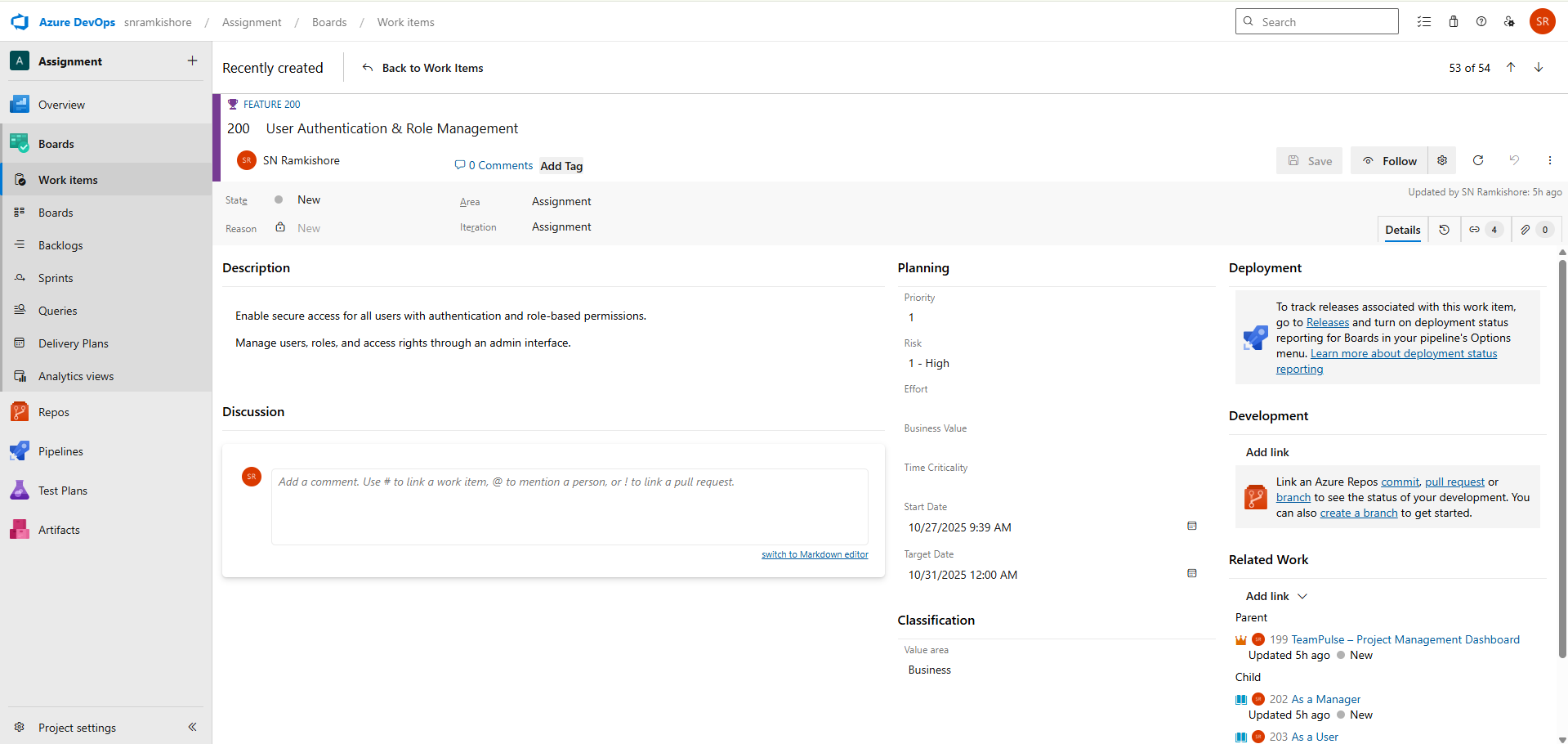
A web-based internal tool to help organizations track productivity, monitor project milestones, and improve collaboration efficiency.

****

**Features**

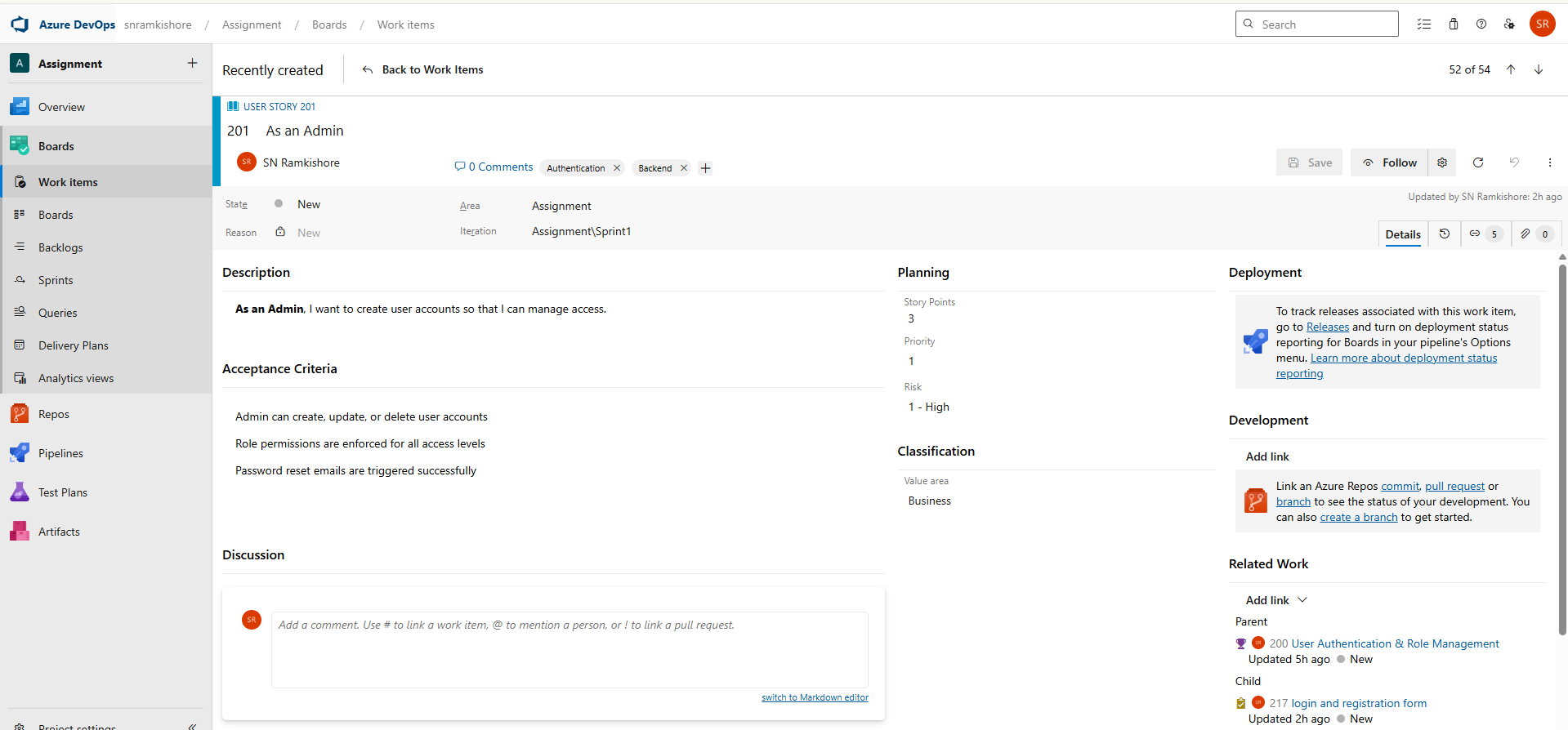
**Feature 1: User Authentication & Role Management**

* Enable secure access for all users with authentication and role-based permissions.
* Manage users, roles, and access rights through an admin interface.



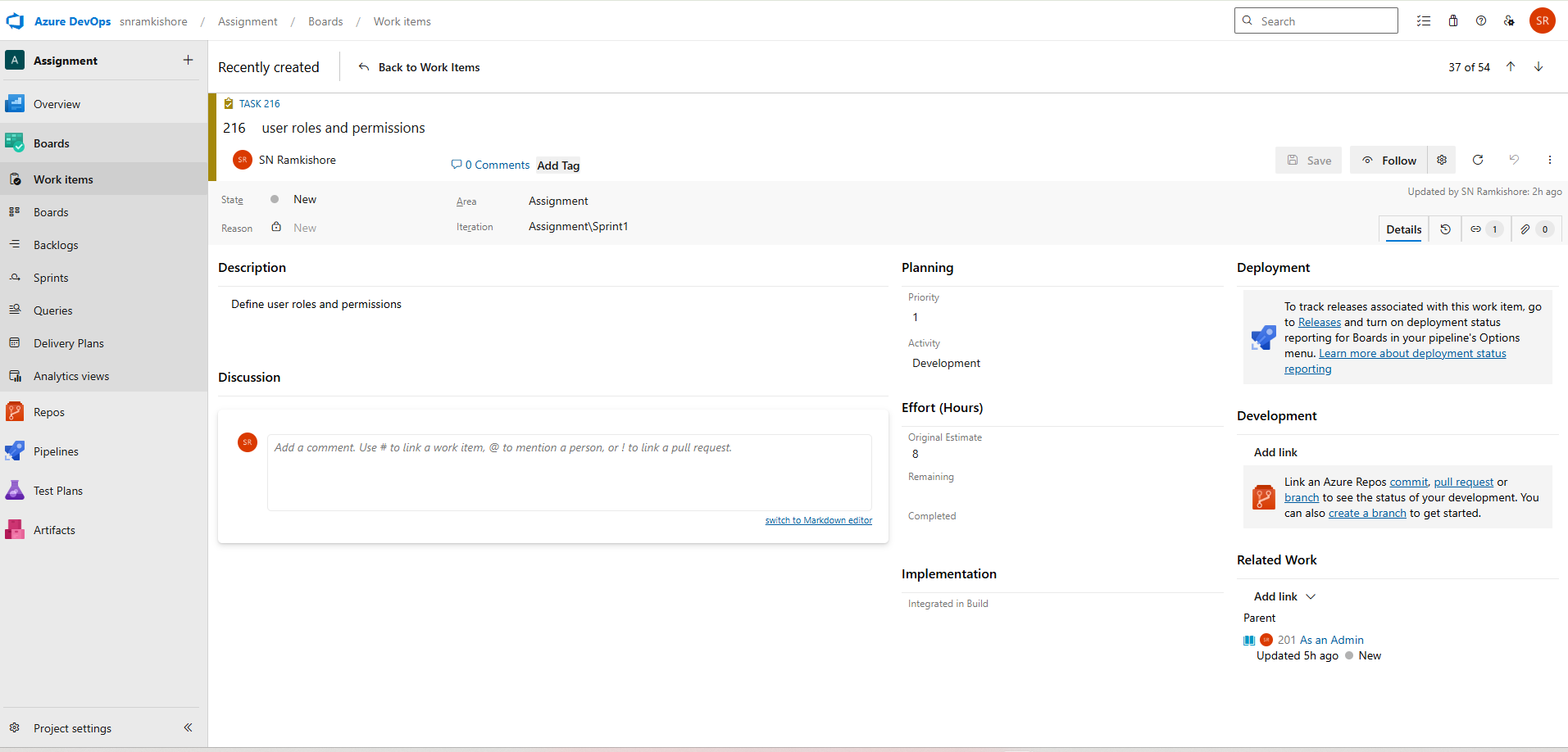
**User Stories**

1. **As an Admin**, I want to create user accounts so that I can manage access.

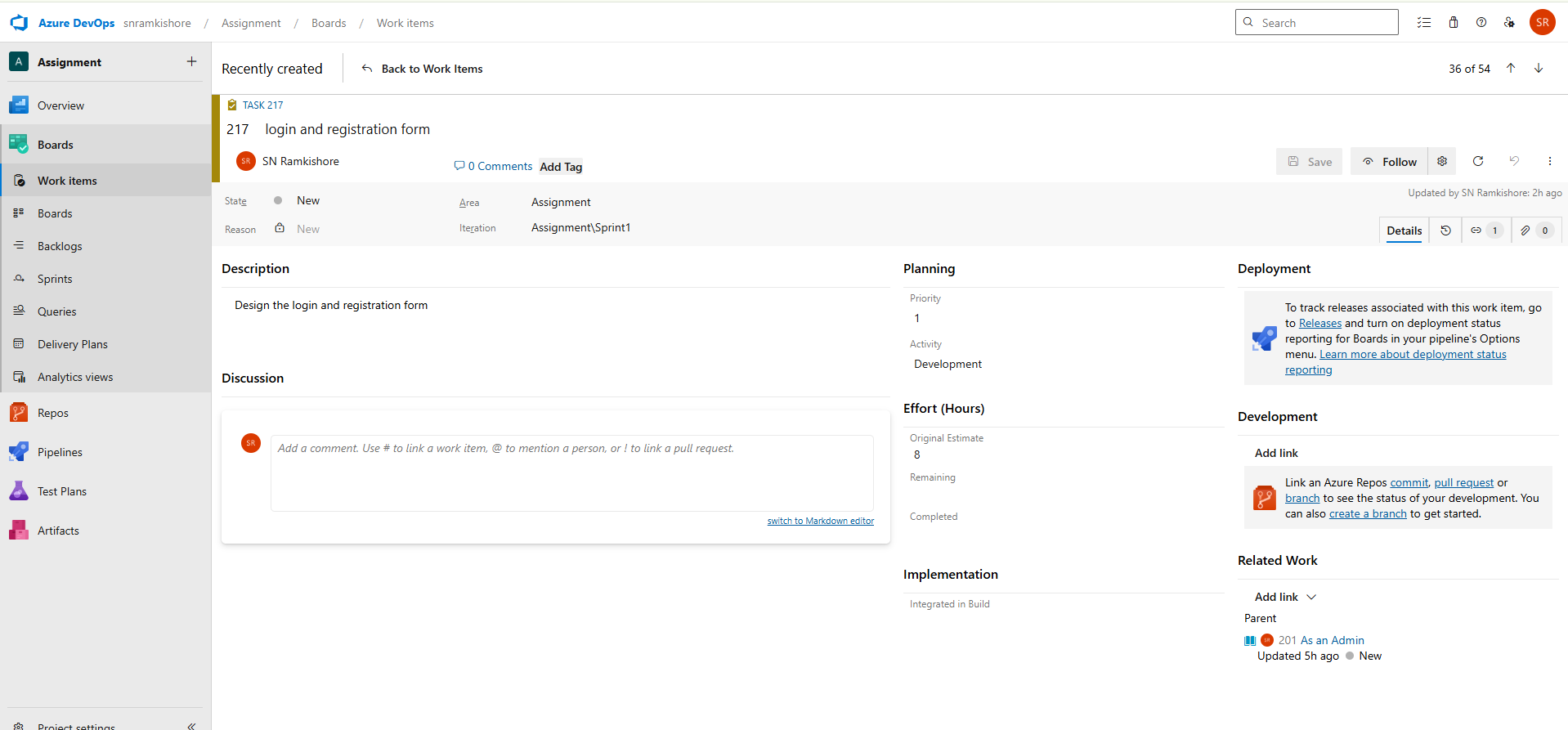


**Tasks:**

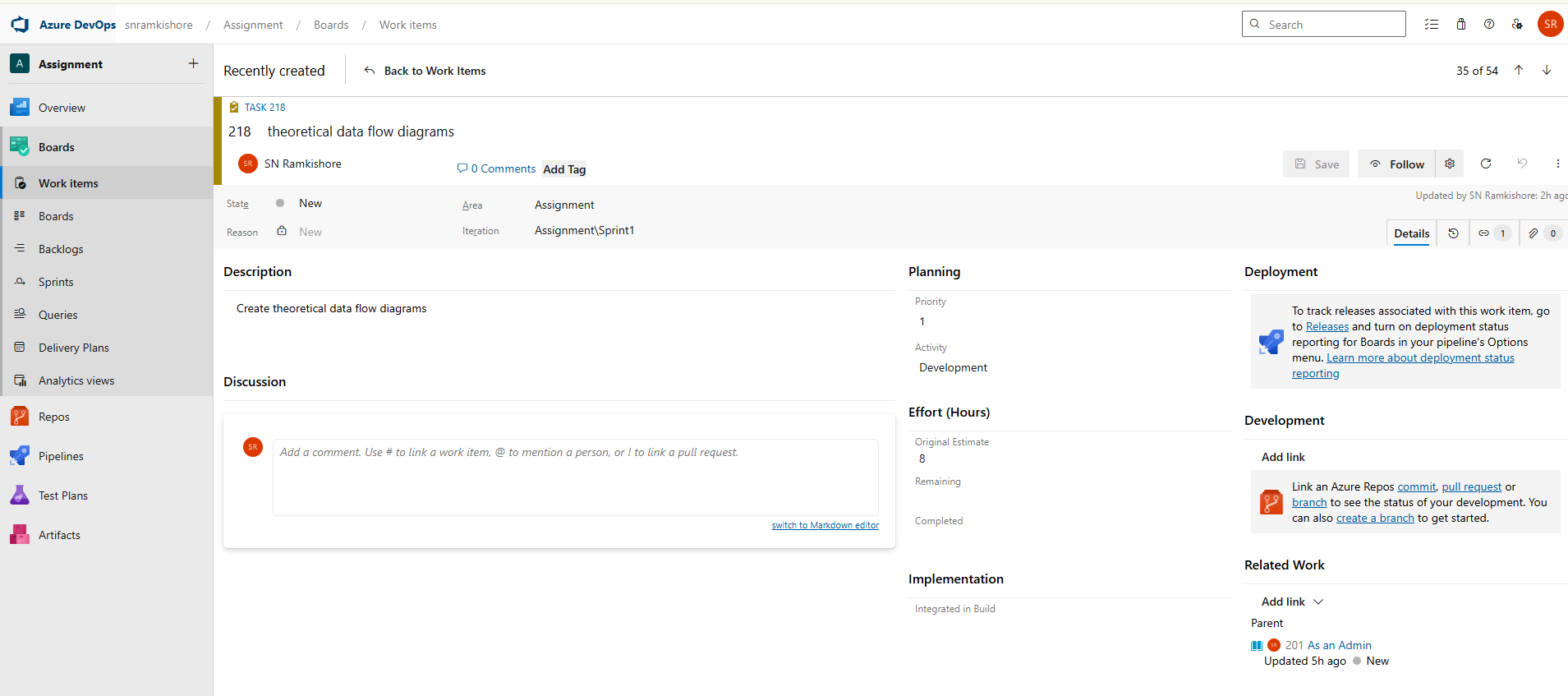
1. Define user roles and permissions



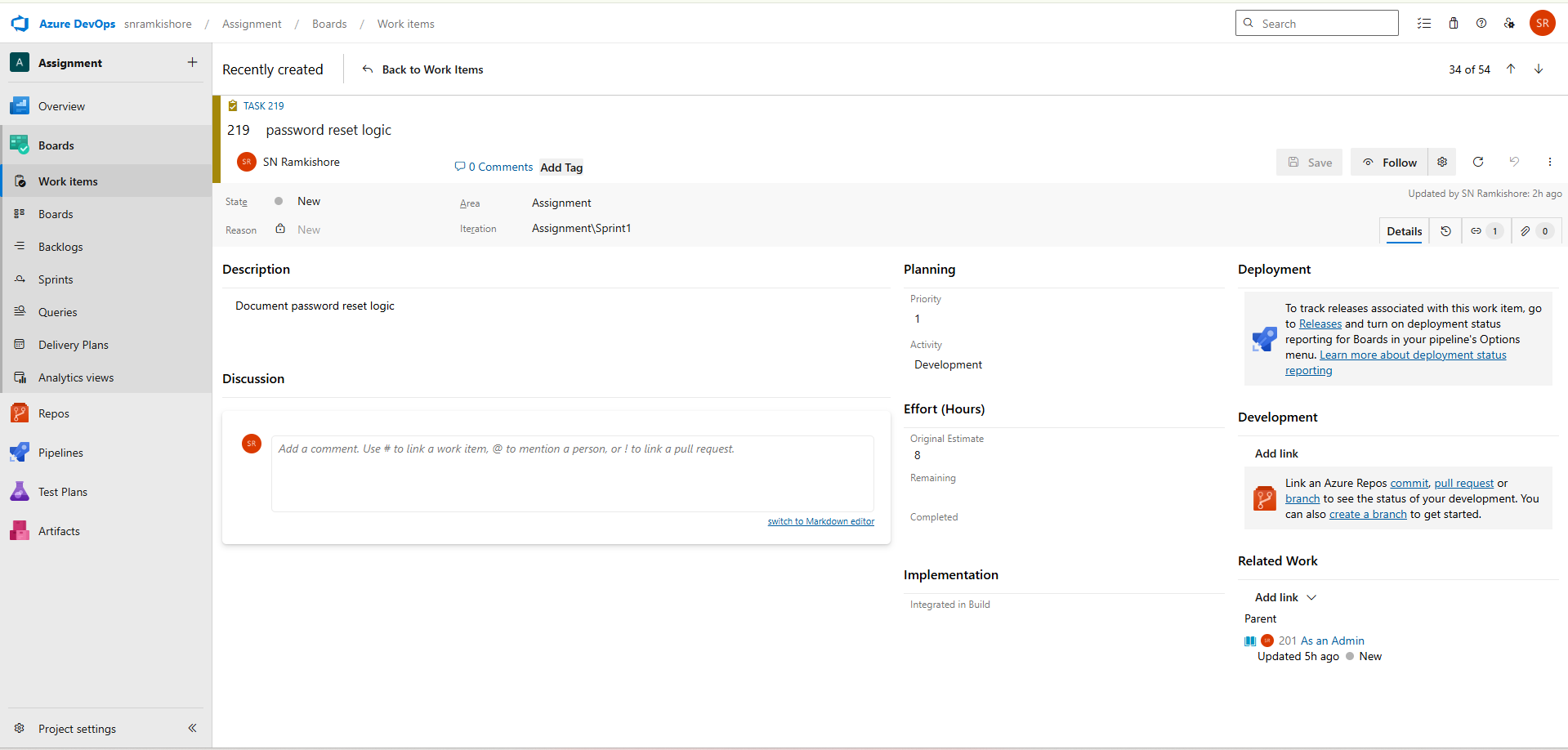
2.Design the login and registration form



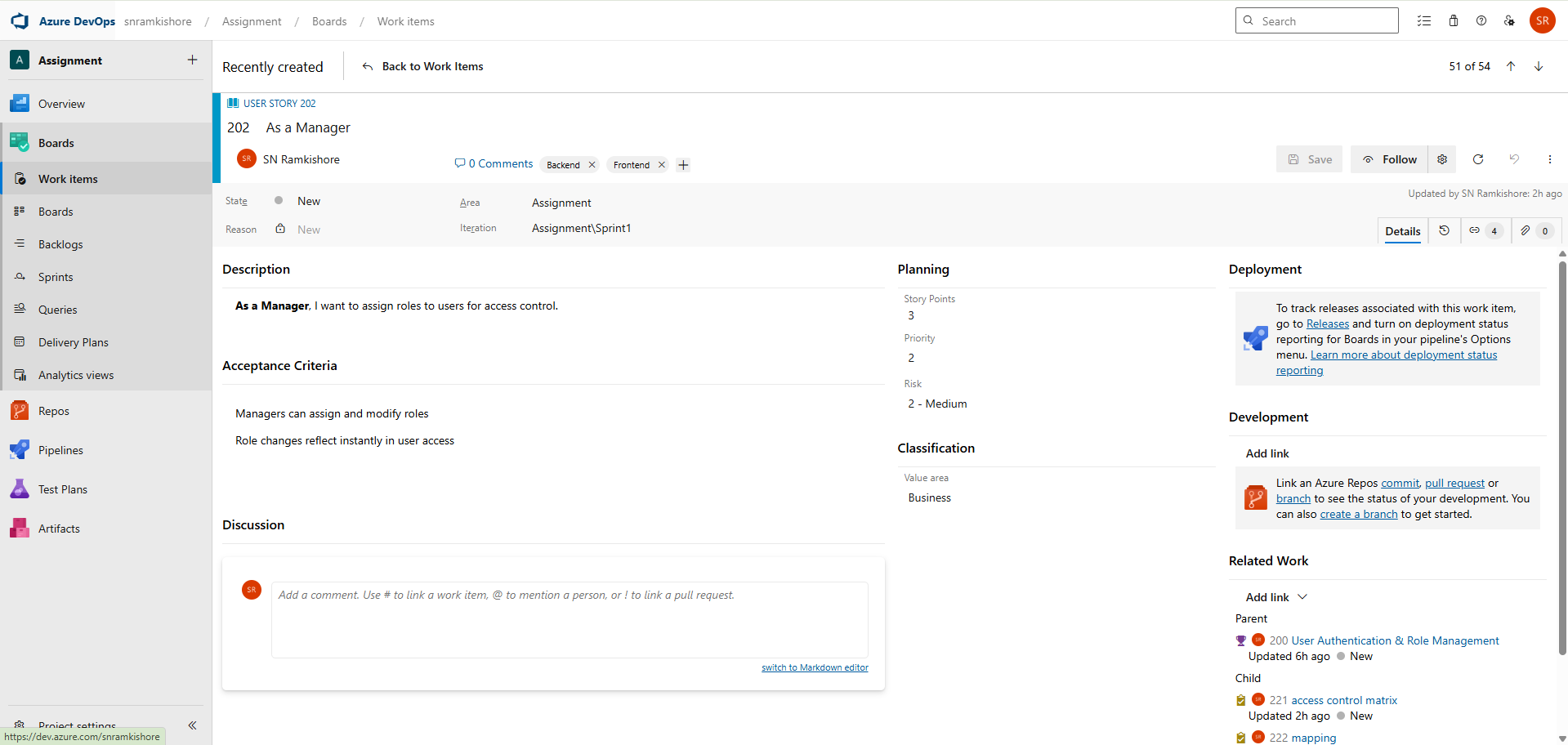
3.Create theoretical data flow diagrams



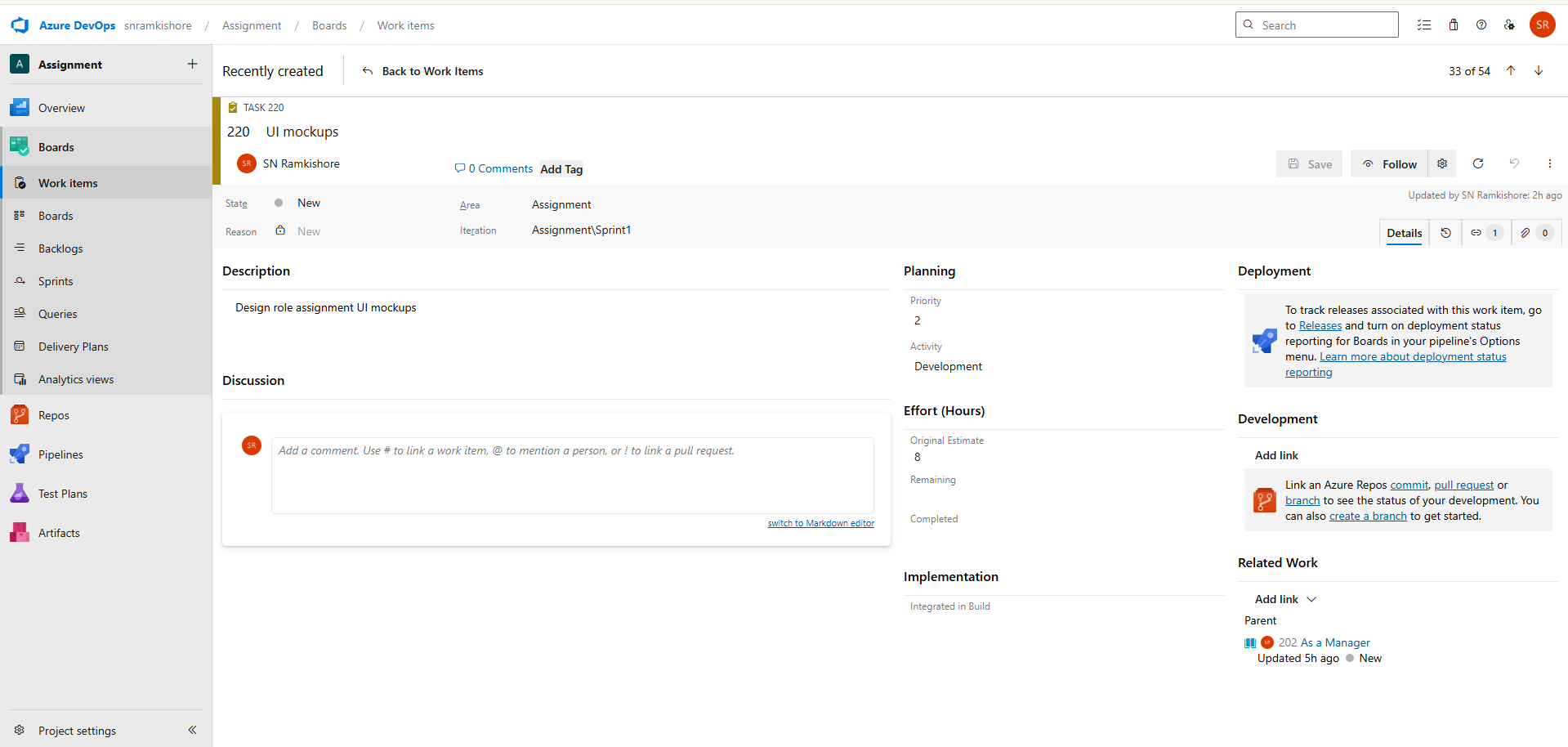
4.Document password reset logic



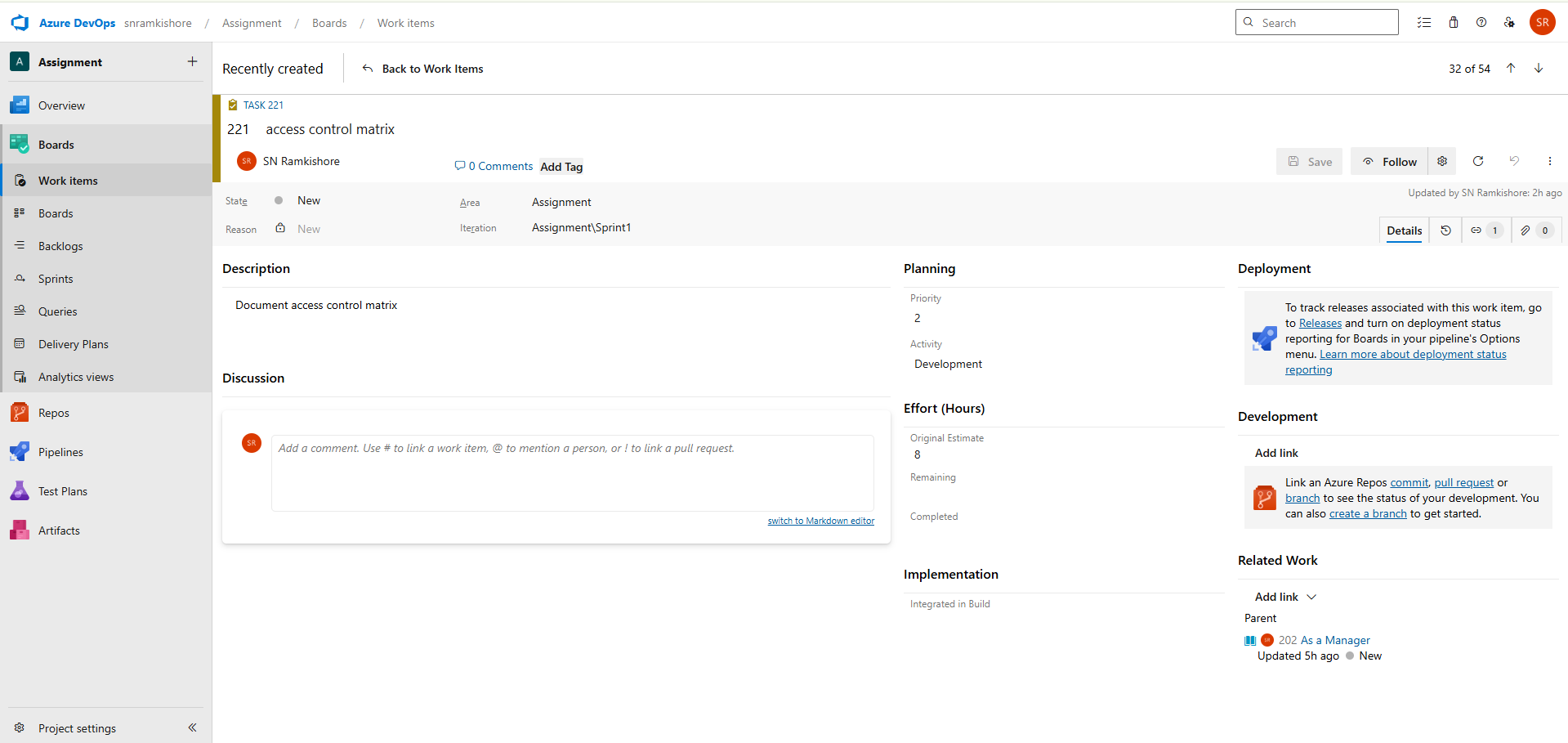
**2.As a Manager**, I want to assign roles to users for access control.



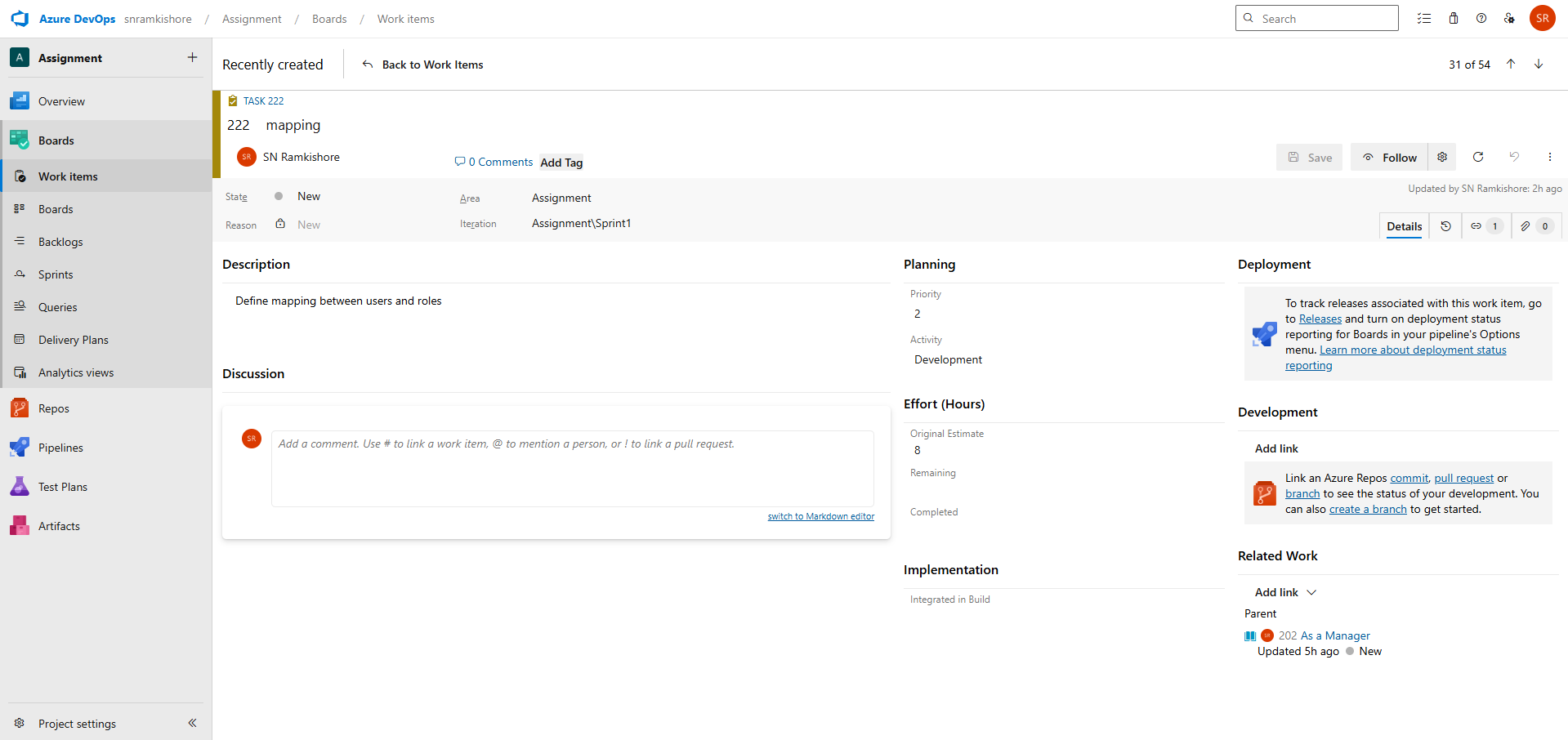
1.Design role assignment UI mockups



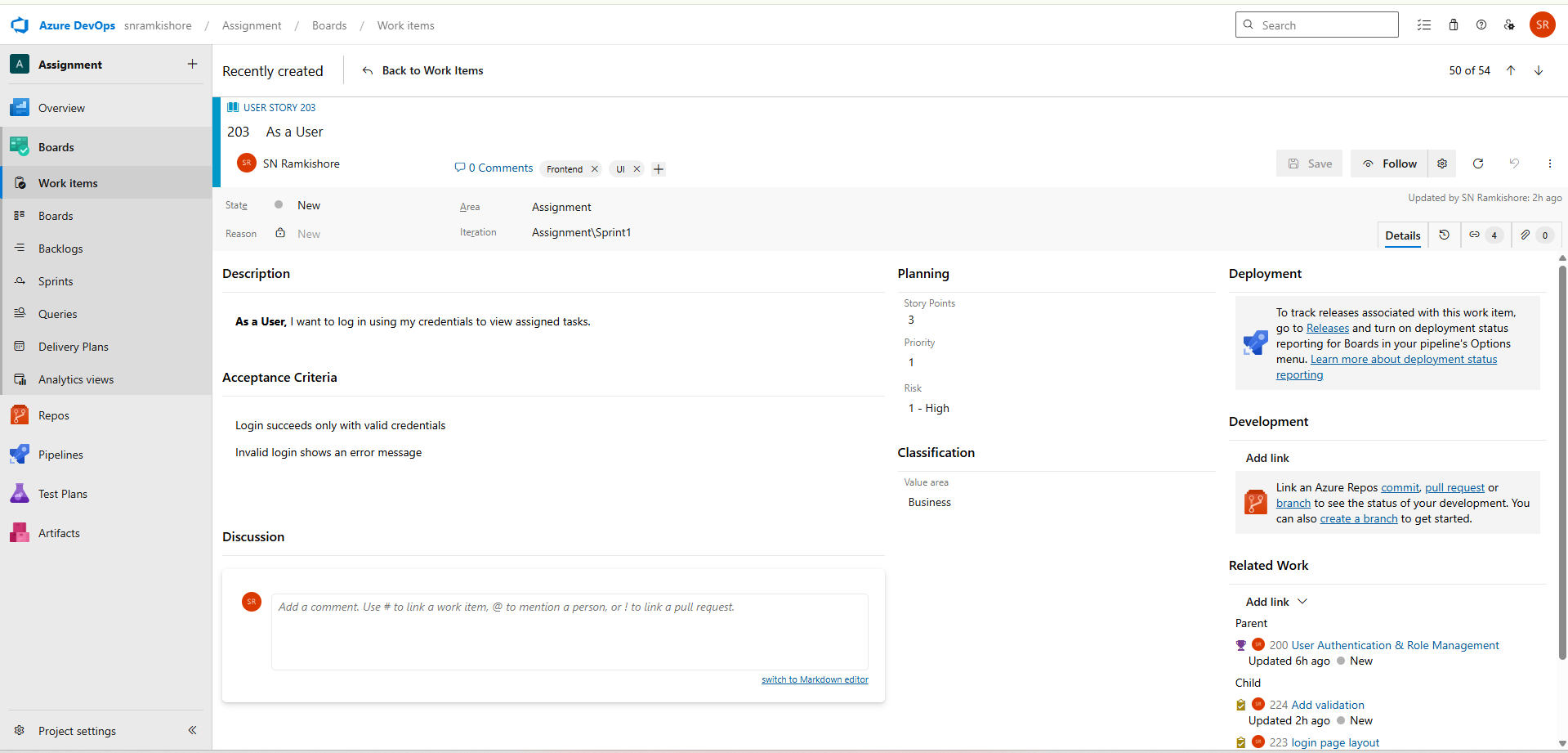
2. Document access control matrix



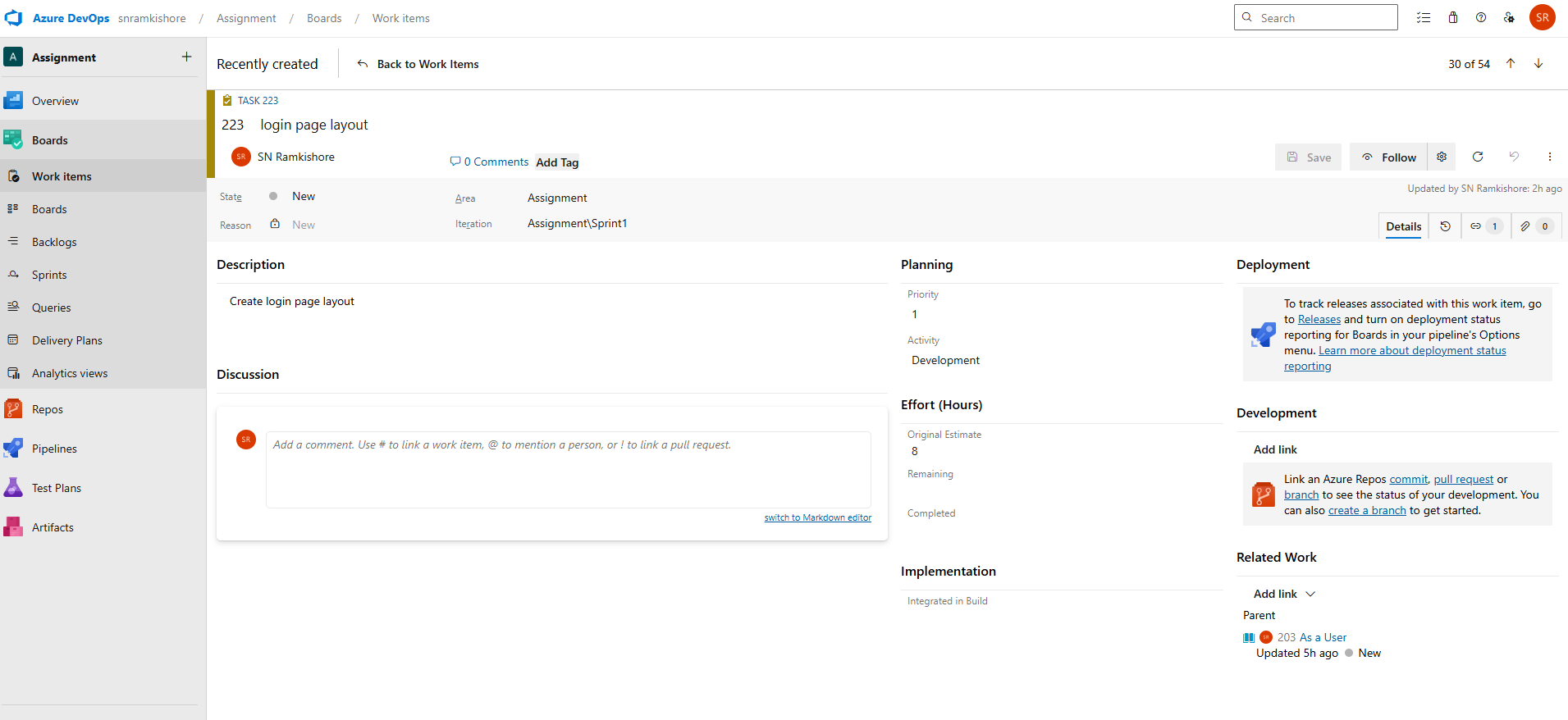
3. Define mapping between users and roles



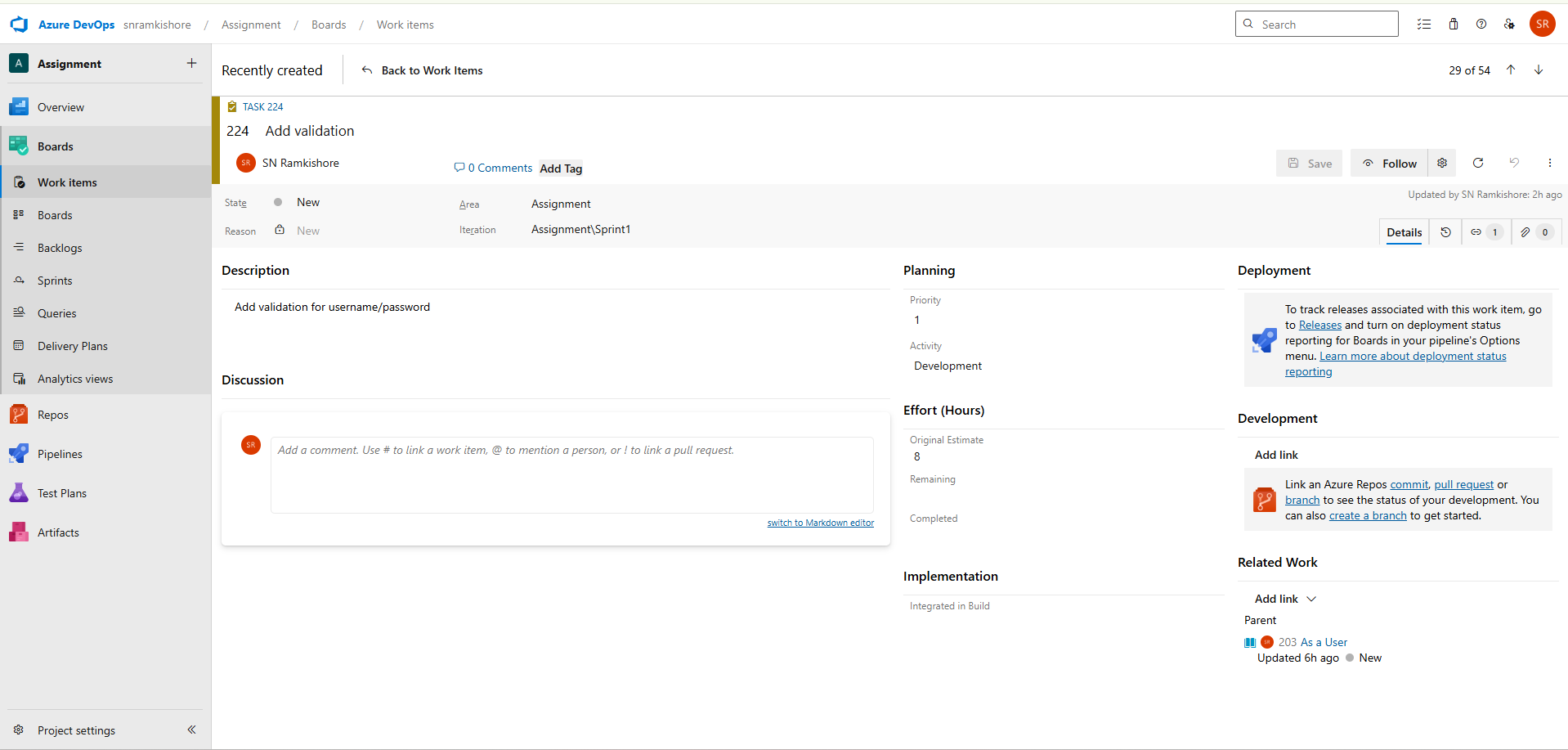
**3.As a User**, I want to log in using my credentials to view assigned tasks.



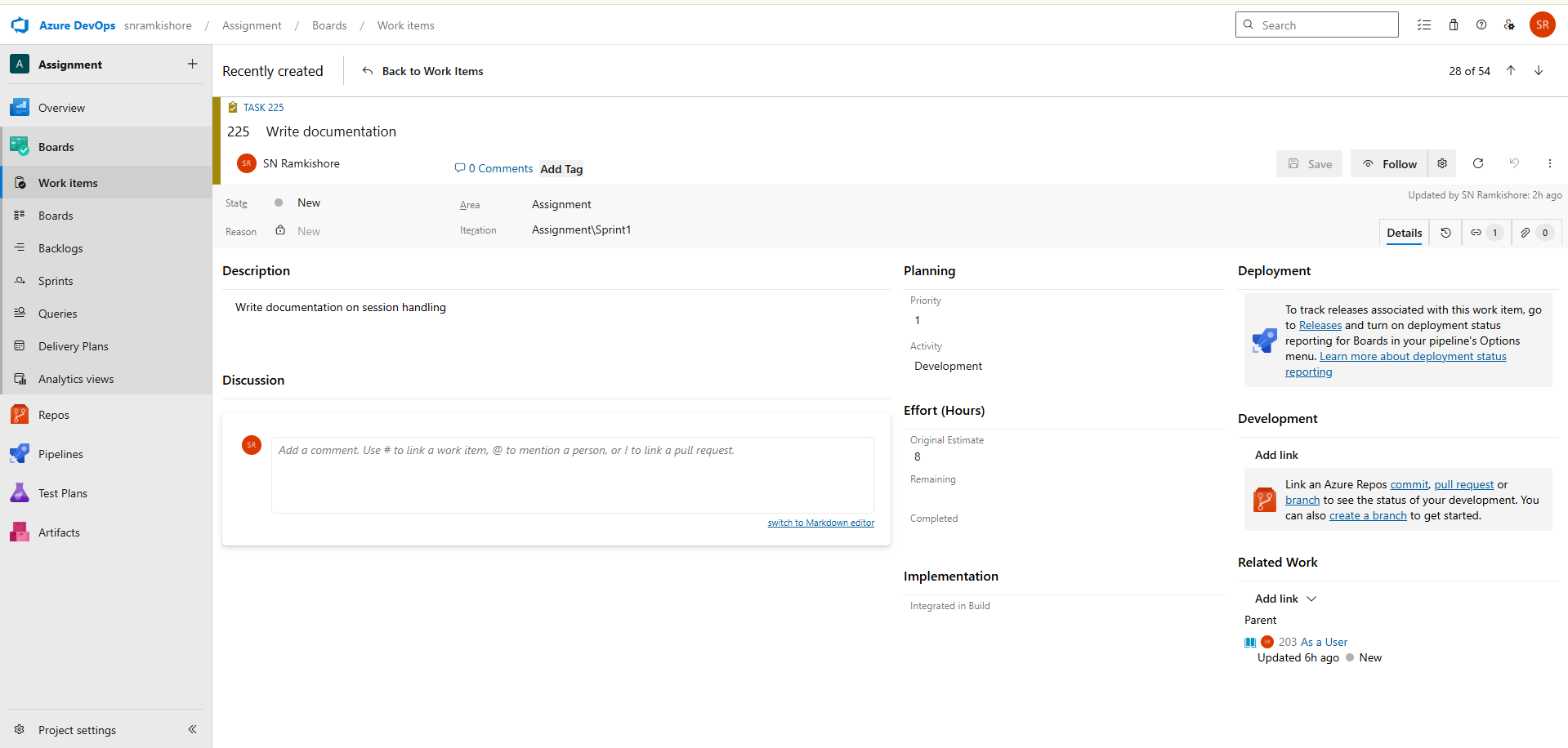
1.Create login page layout



2.Add validation for username/password

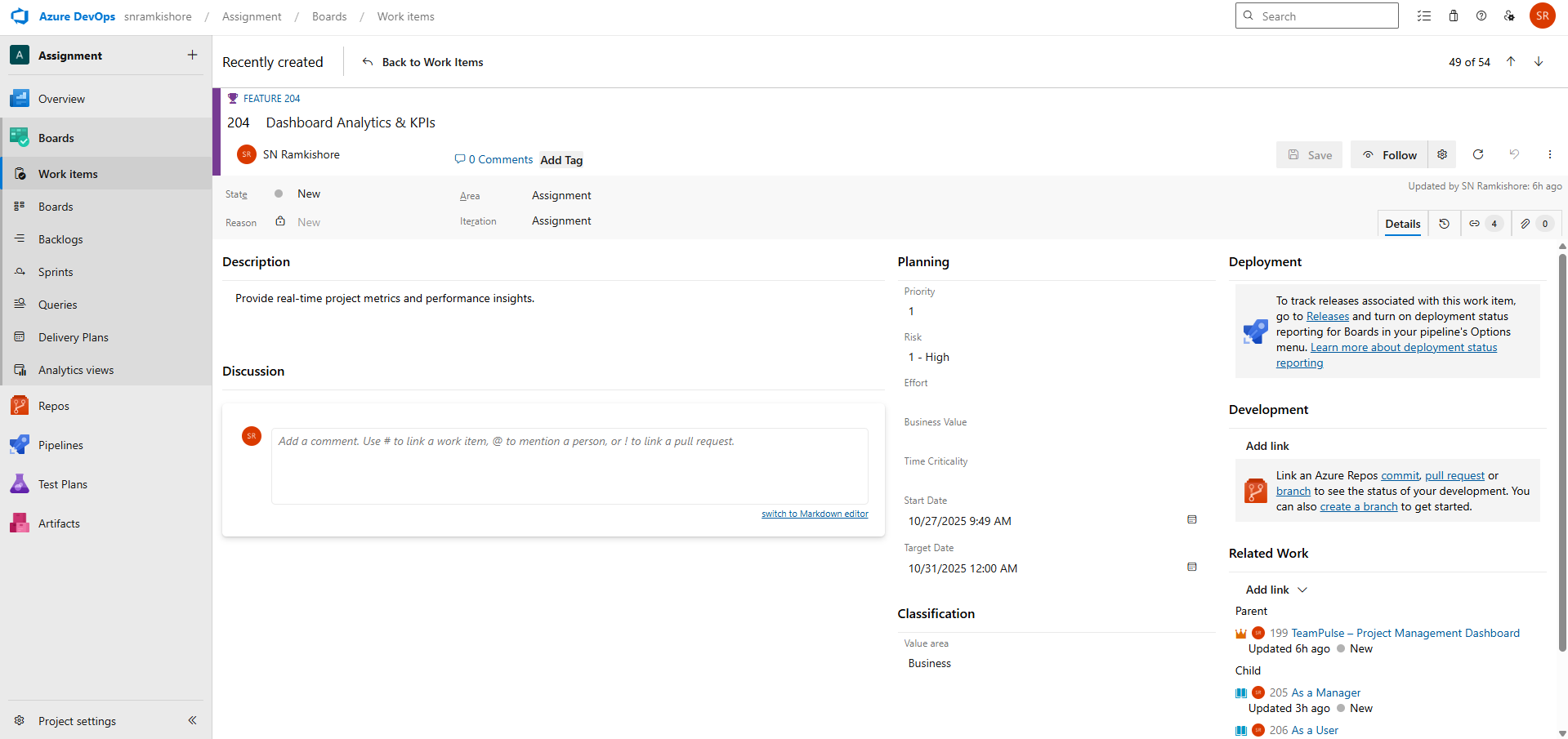


3.Write documentation on session handling



**Feature 2: Dashboard Analytics & KPIs**

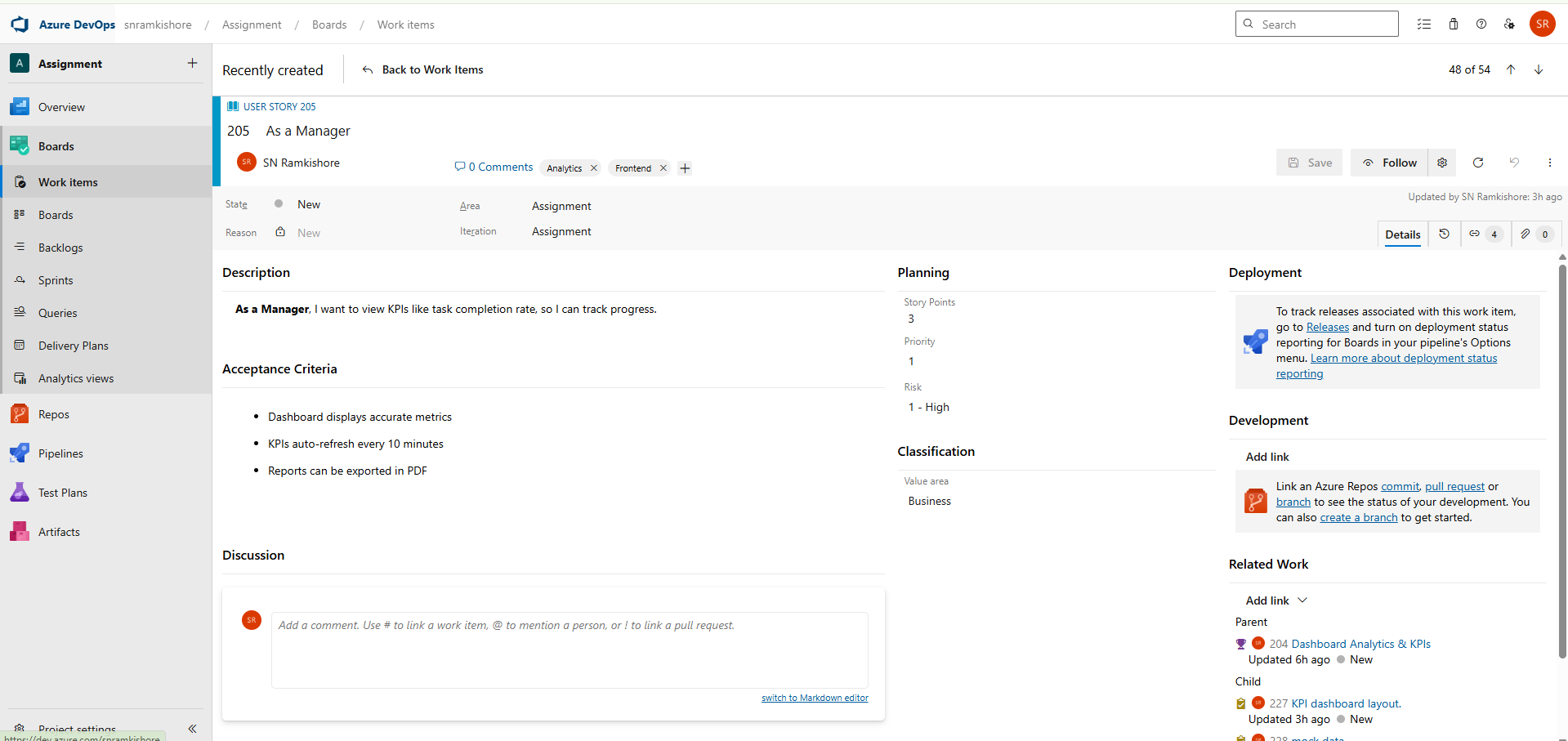
* Provide real-time project metrics and performance insights.



**User Stories**

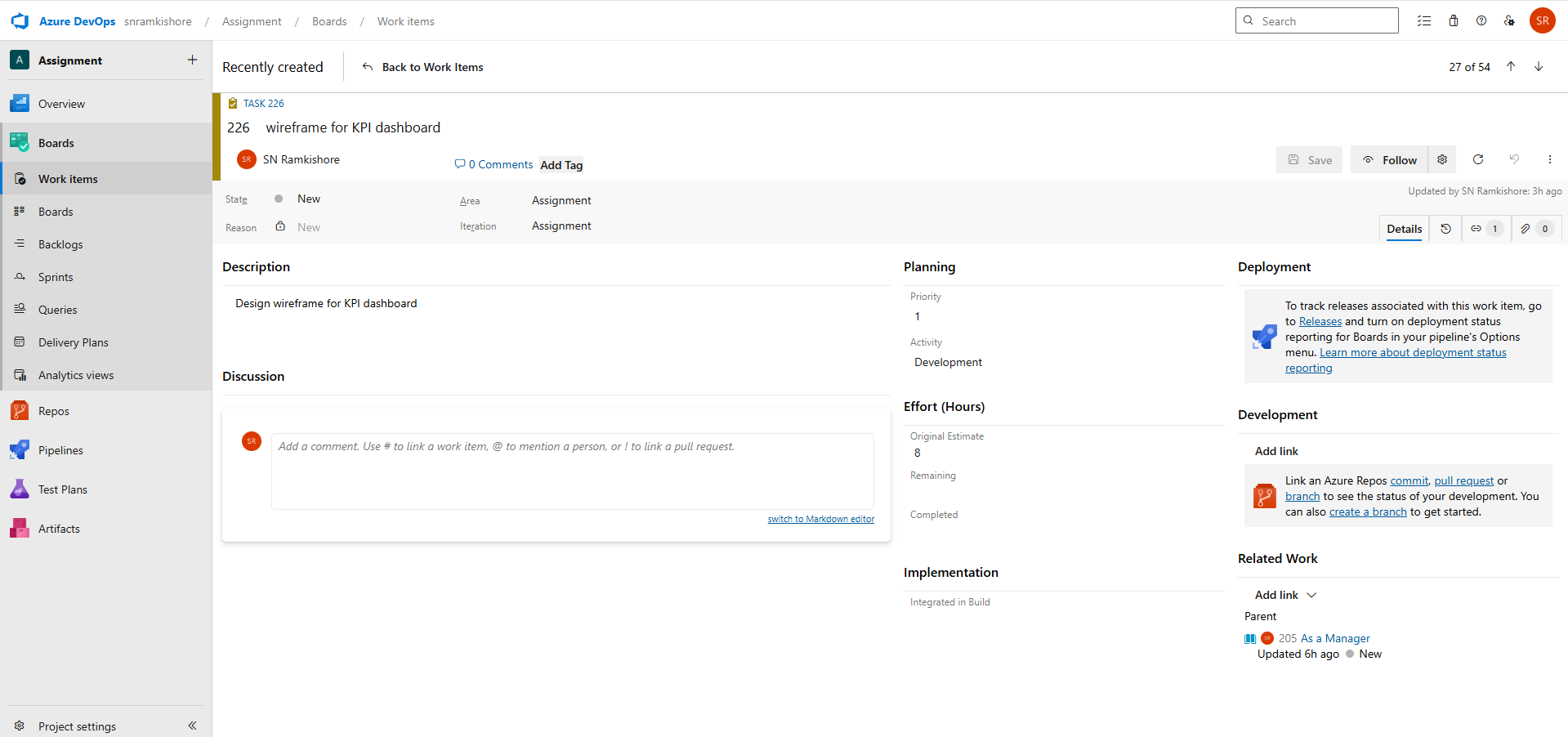
**User Story 1:**

**As a Manager,** I want to view KPIs like task completion rate, so I can track progress.



**Tasks:**

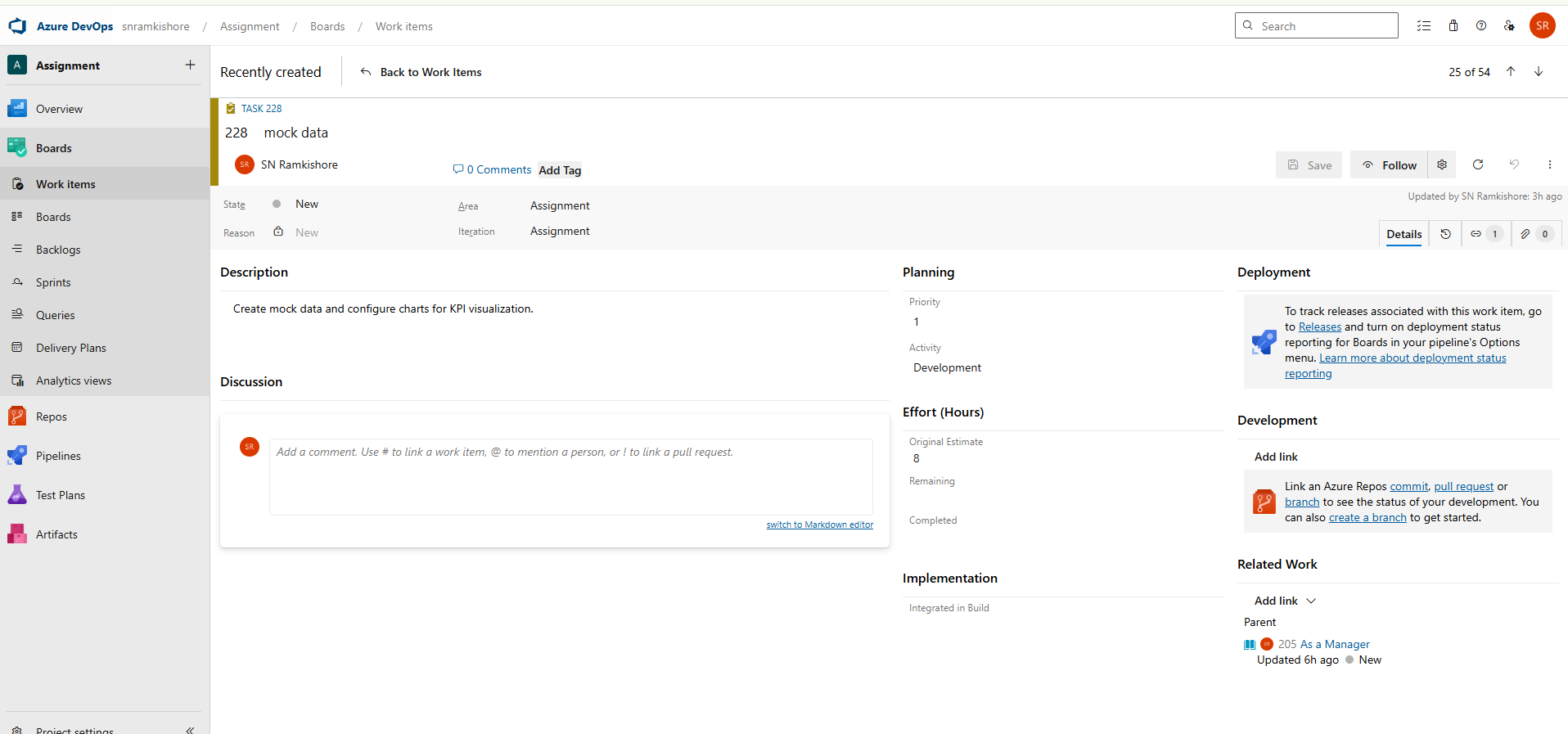
1. Identify and define key KPI metrics for the dashboard.



2. Design and document the KPI dashboard layout.

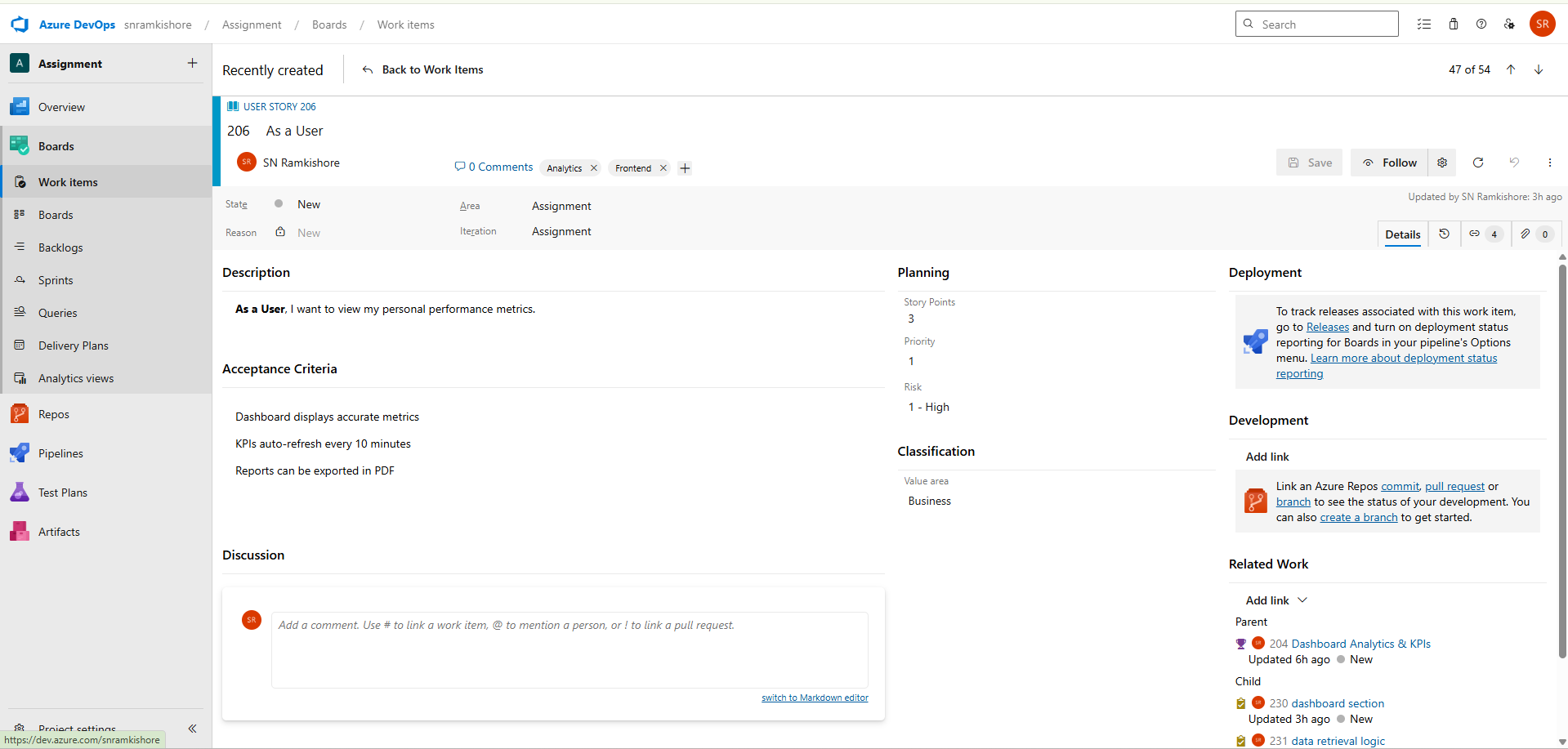


3.Create mock data and configure charts for KPI visualization.

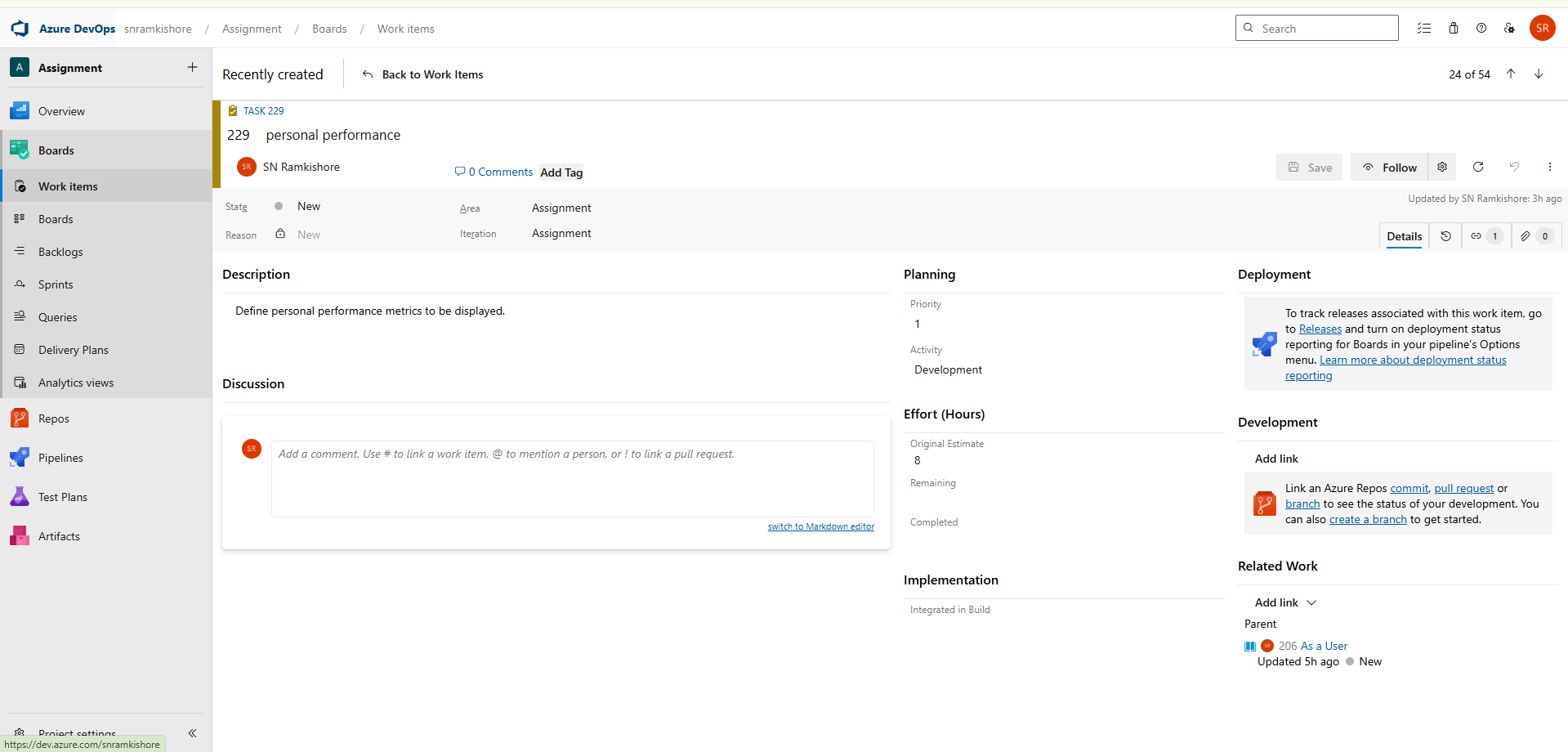


**User Story 2:**

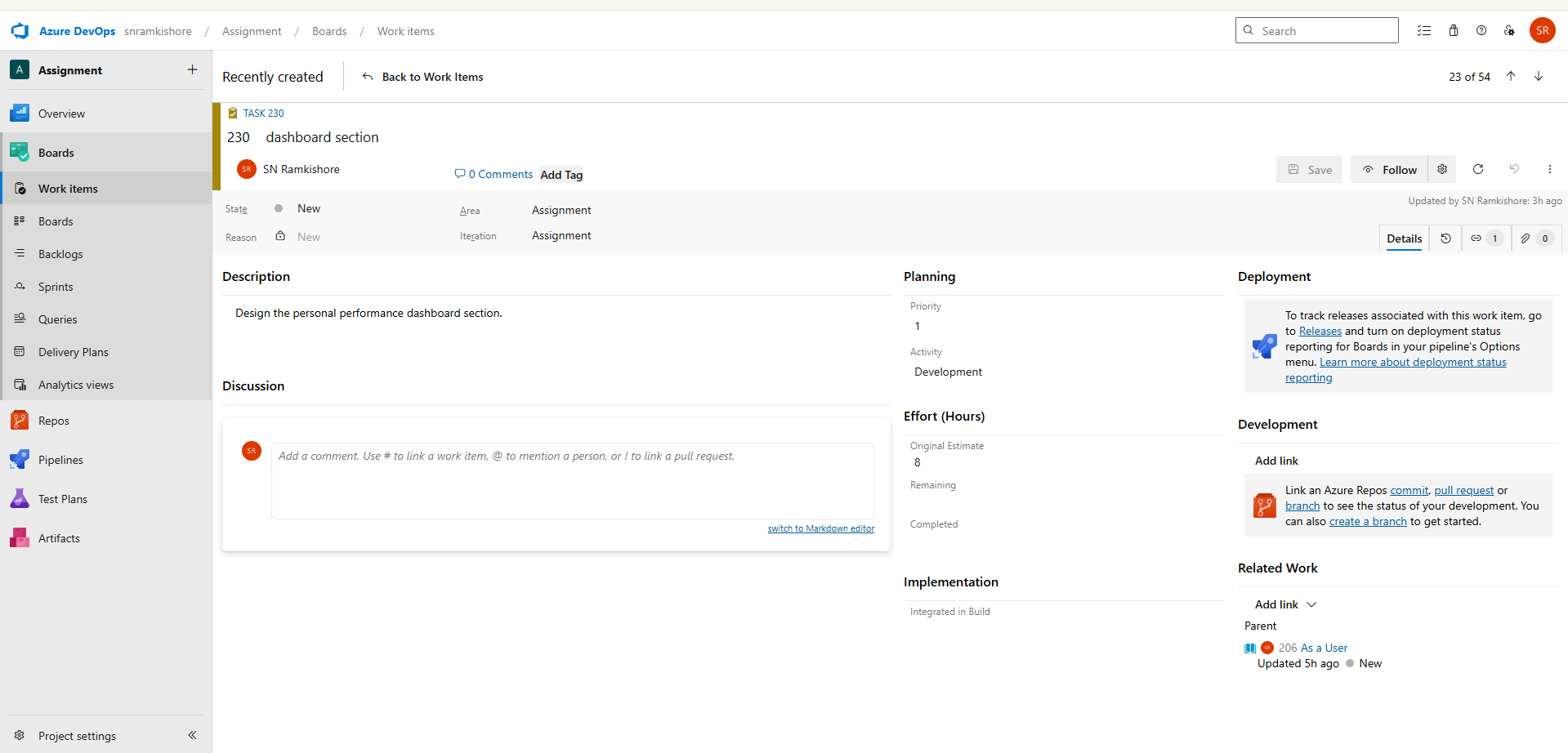
**As a User**, I want to view my personal performance metrics.

  
**Tasks:**

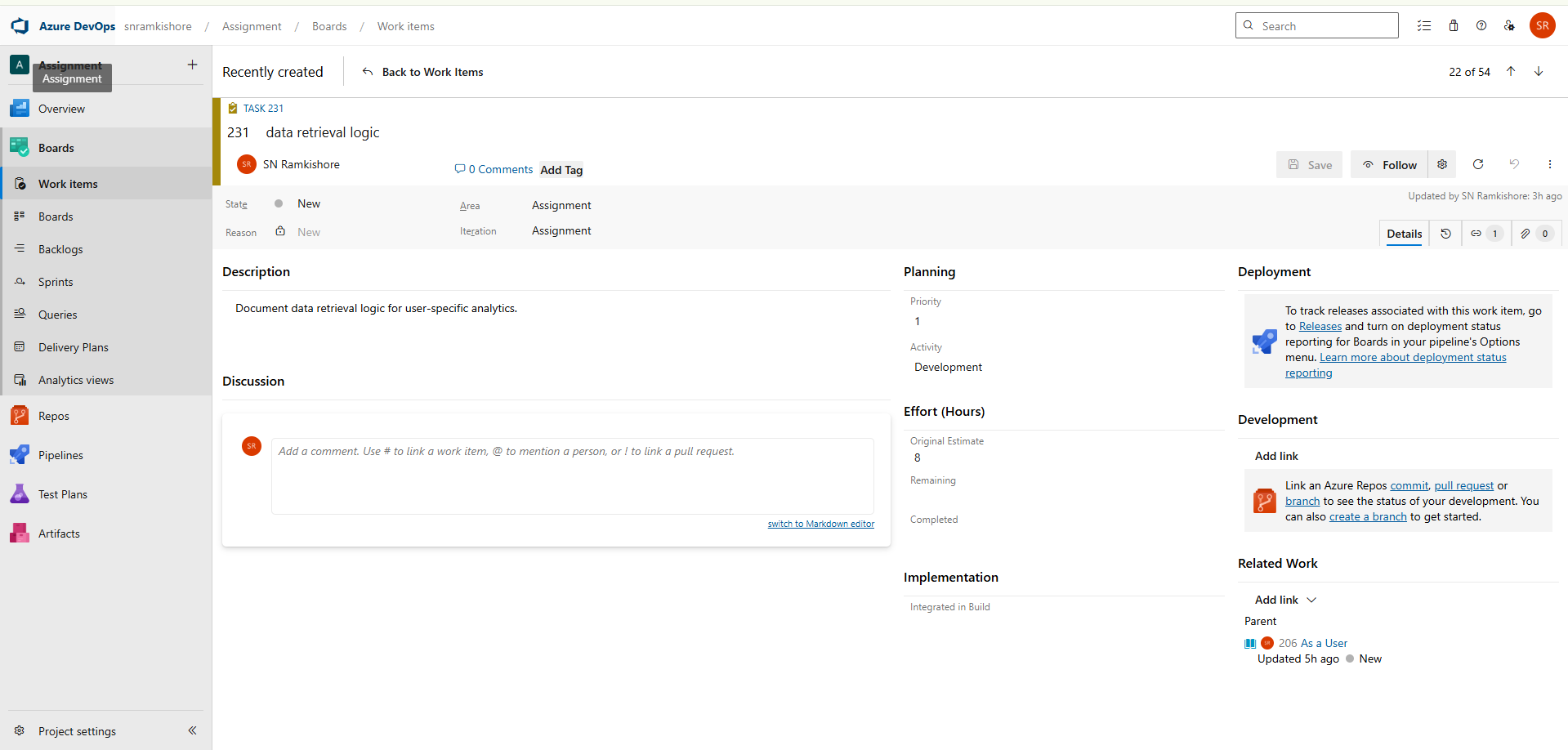
1. Define personal performance metrics to be displayed.



2.Design the personal performance dashboard section.

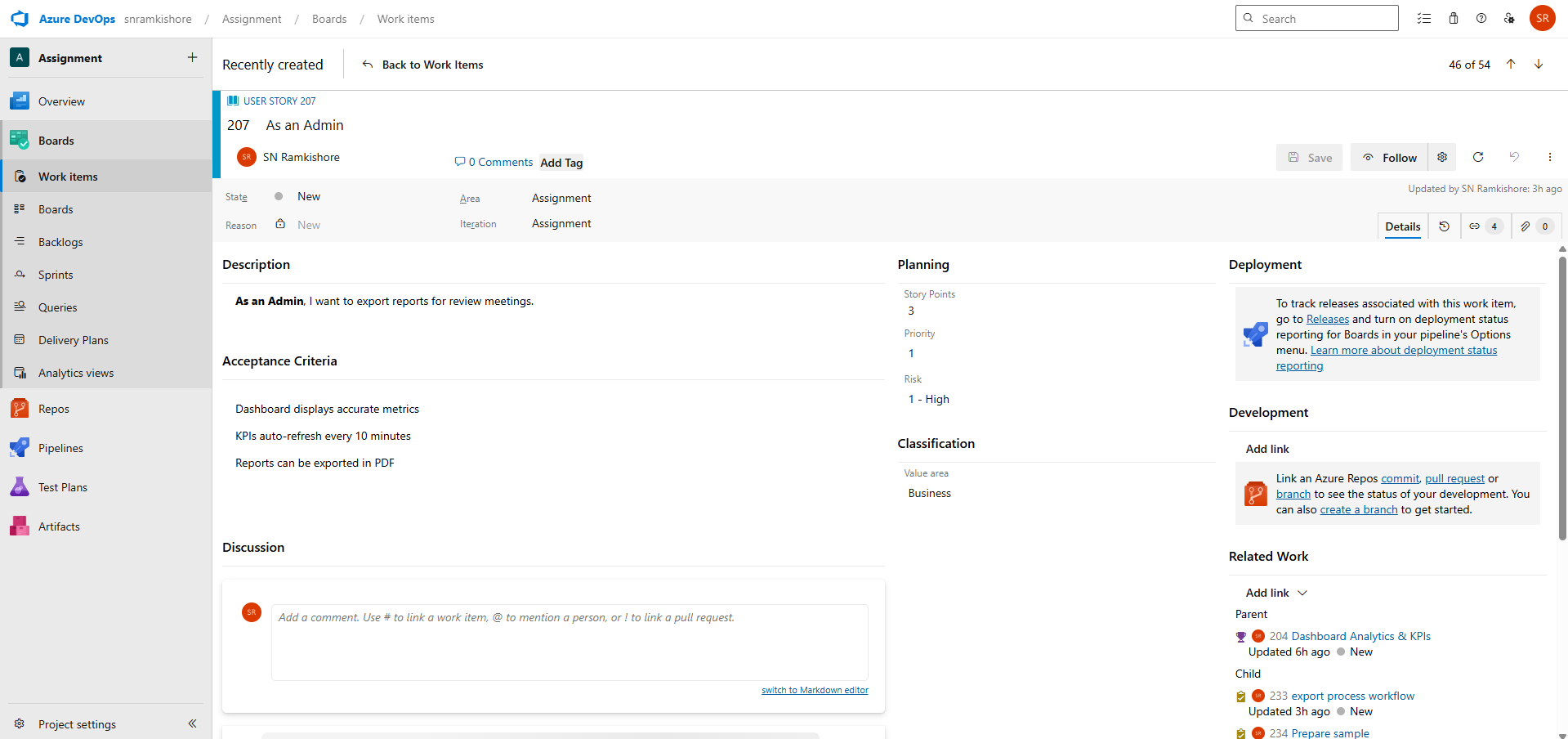


3.Document data retrieval logic for user-specific analytics.



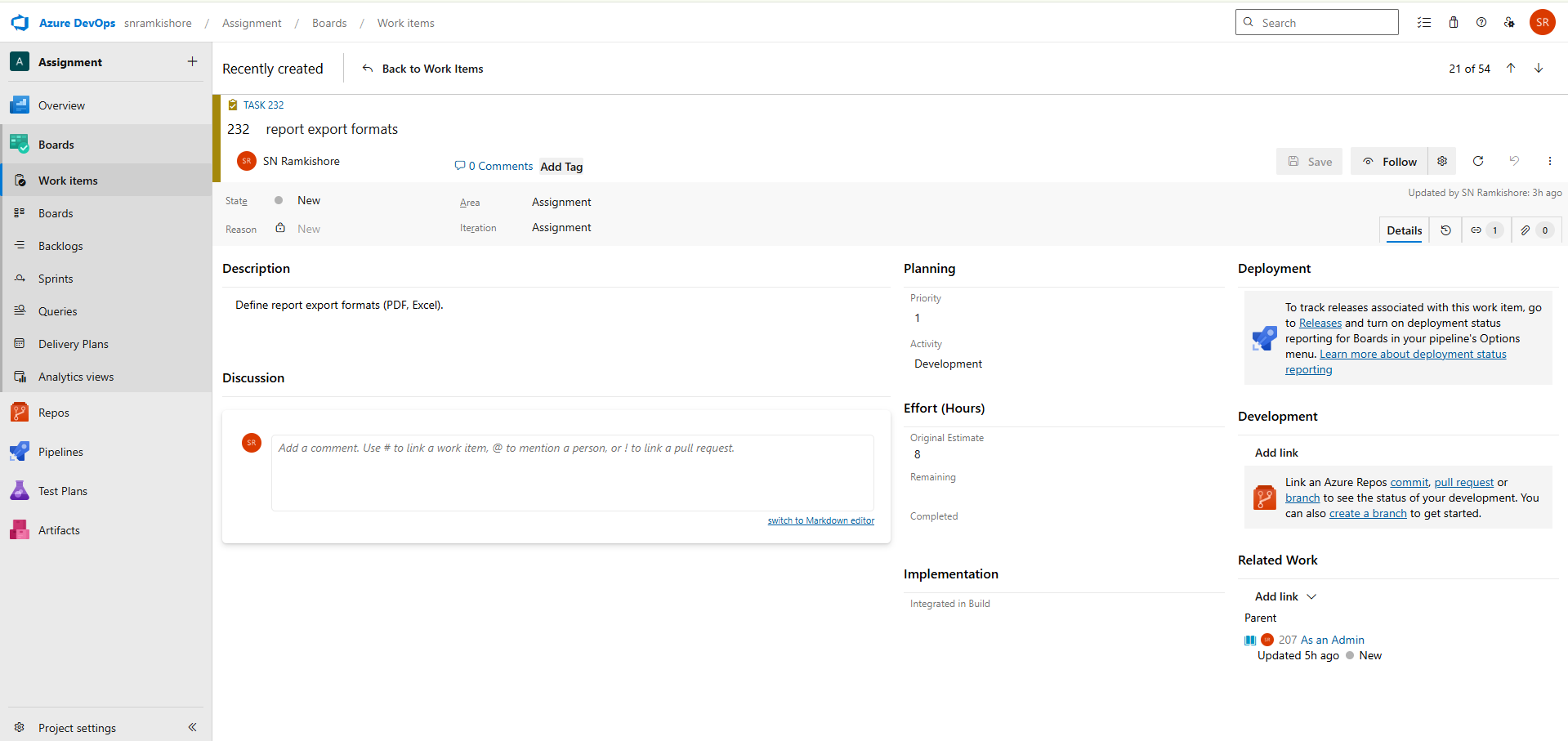
**User Story 3:**

**As an Admin**, I want to export reports for review meetings.

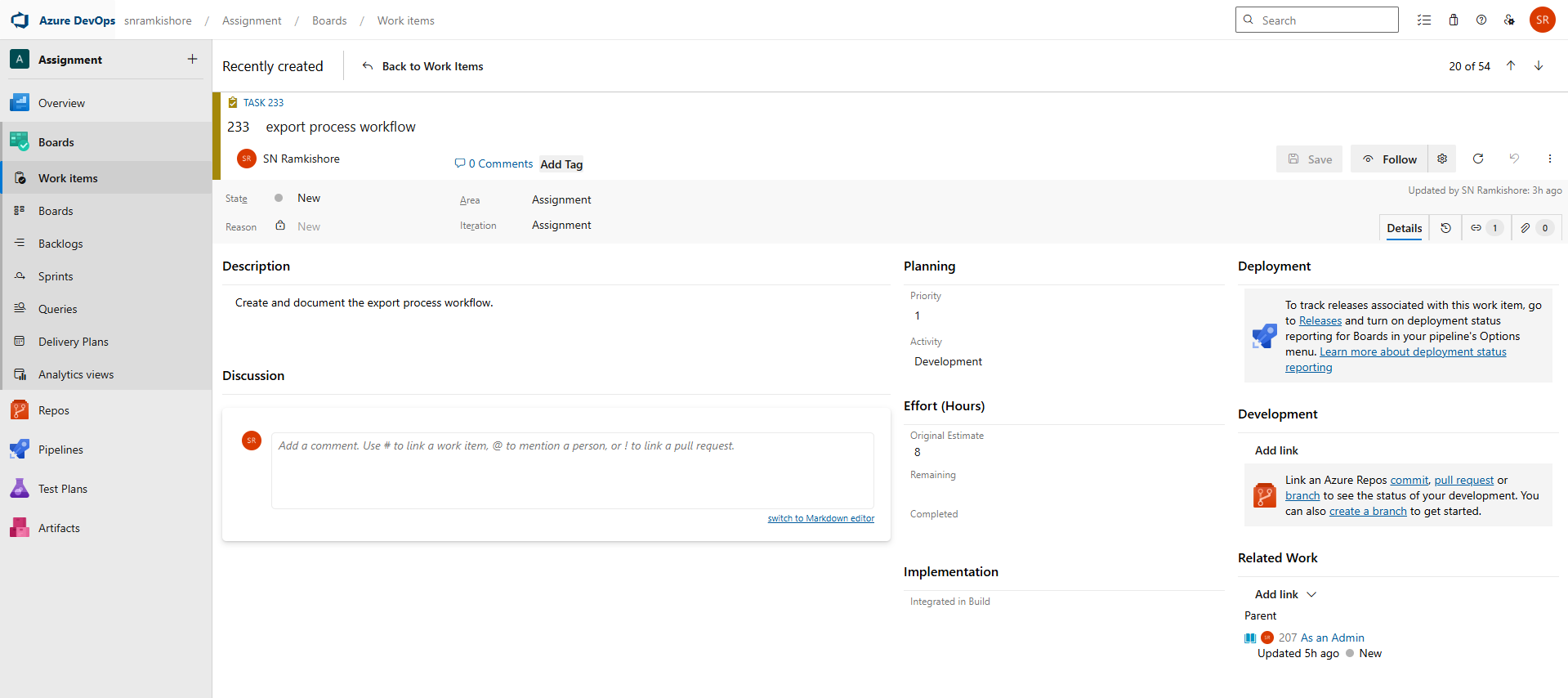


**Tasks:**

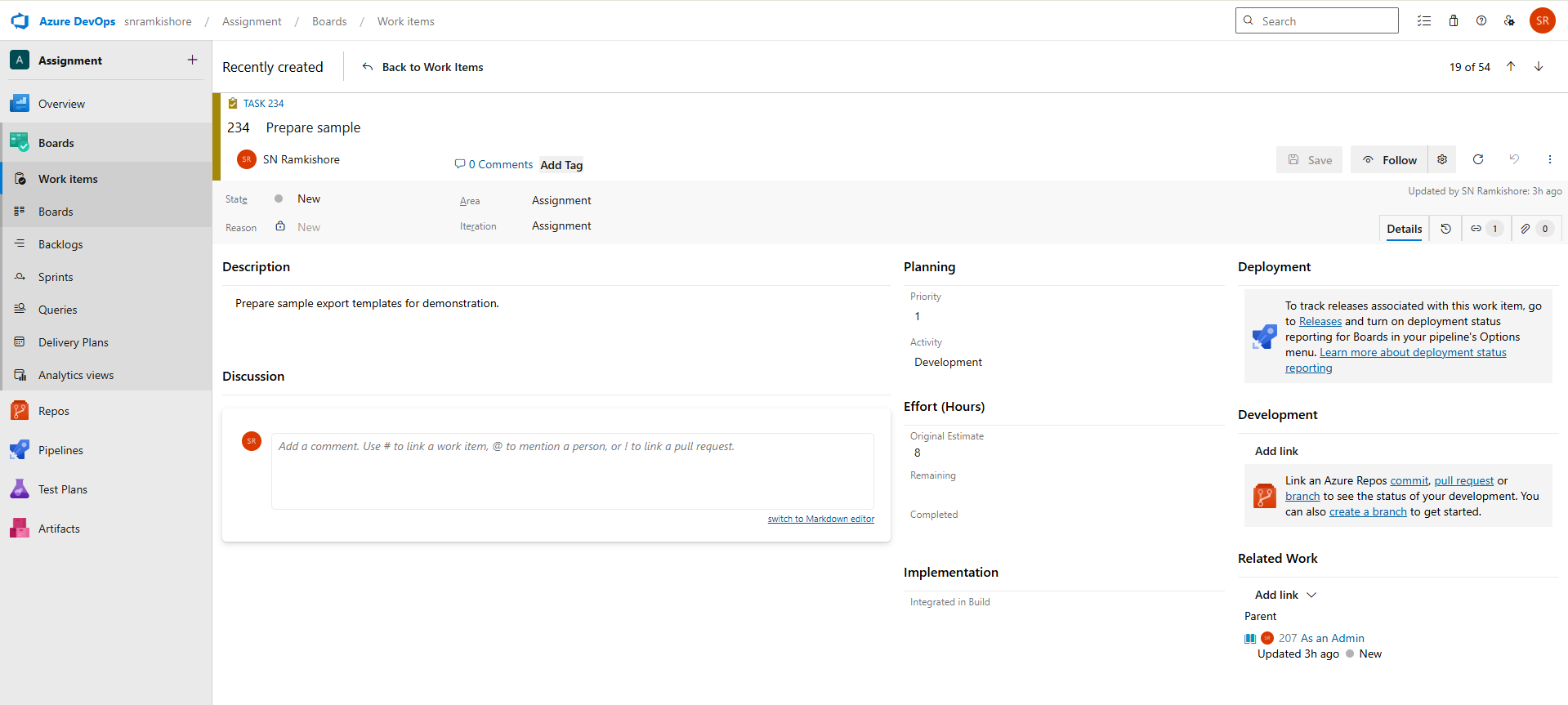
1. Define report export formats (PDF, Excel).



2.Create and document the export process workflow.

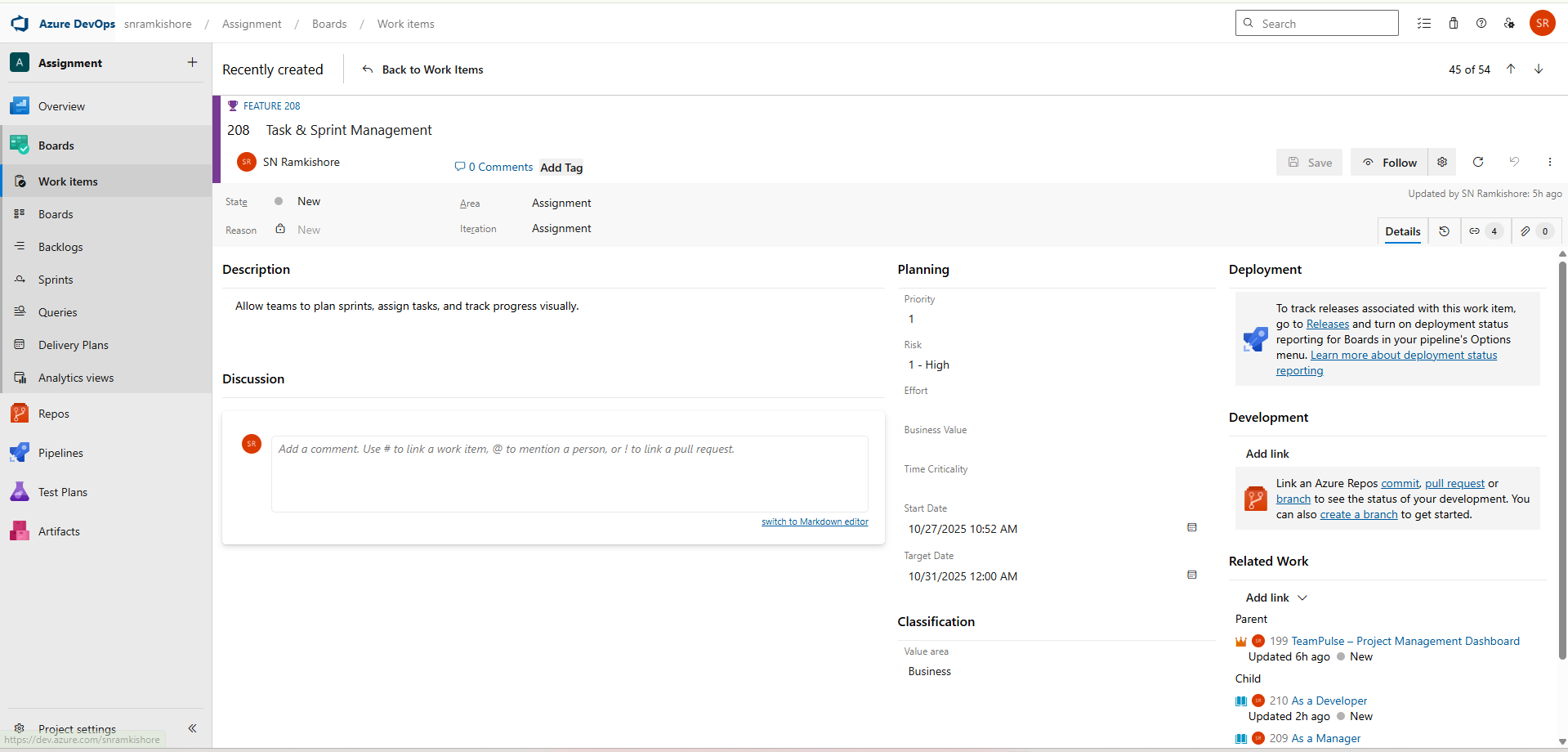


3.Prepare sample export templates for demonstration.



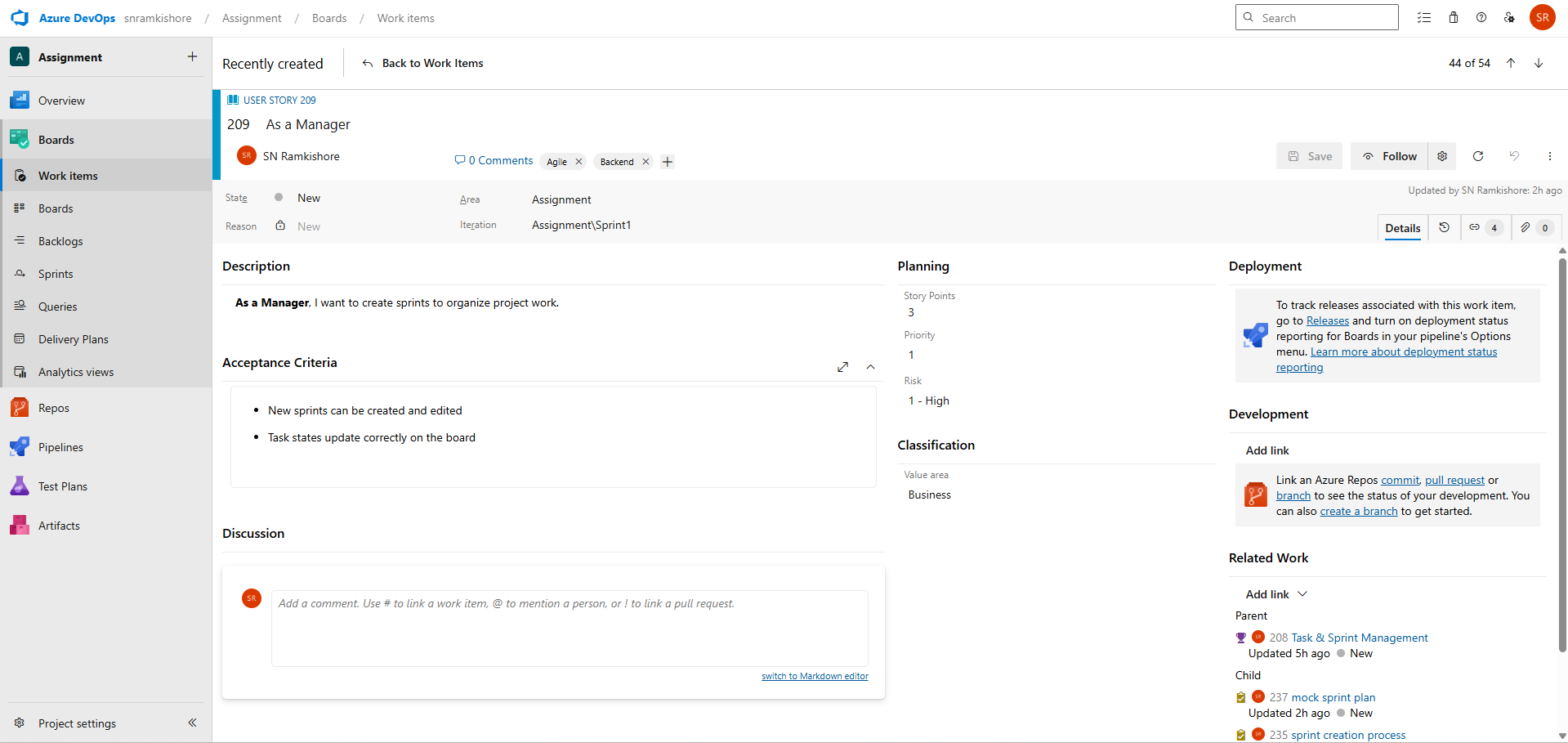
**Feature 3: Task & Sprint Management**

* Allow teams to plan sprints, assign tasks, and track progress visually.



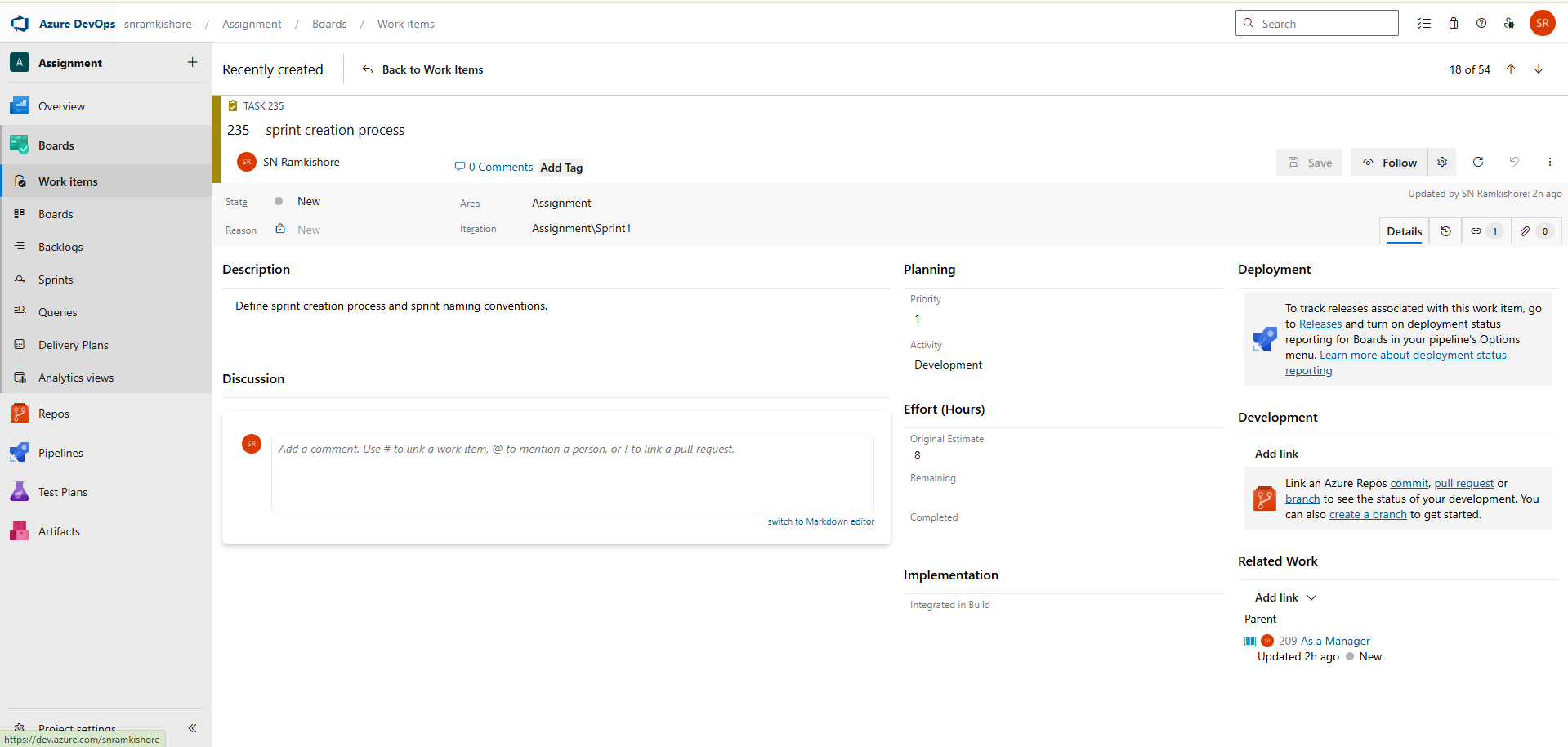
**User Story 1:**

**As a Manager,** I want to create sprints to organize project work.

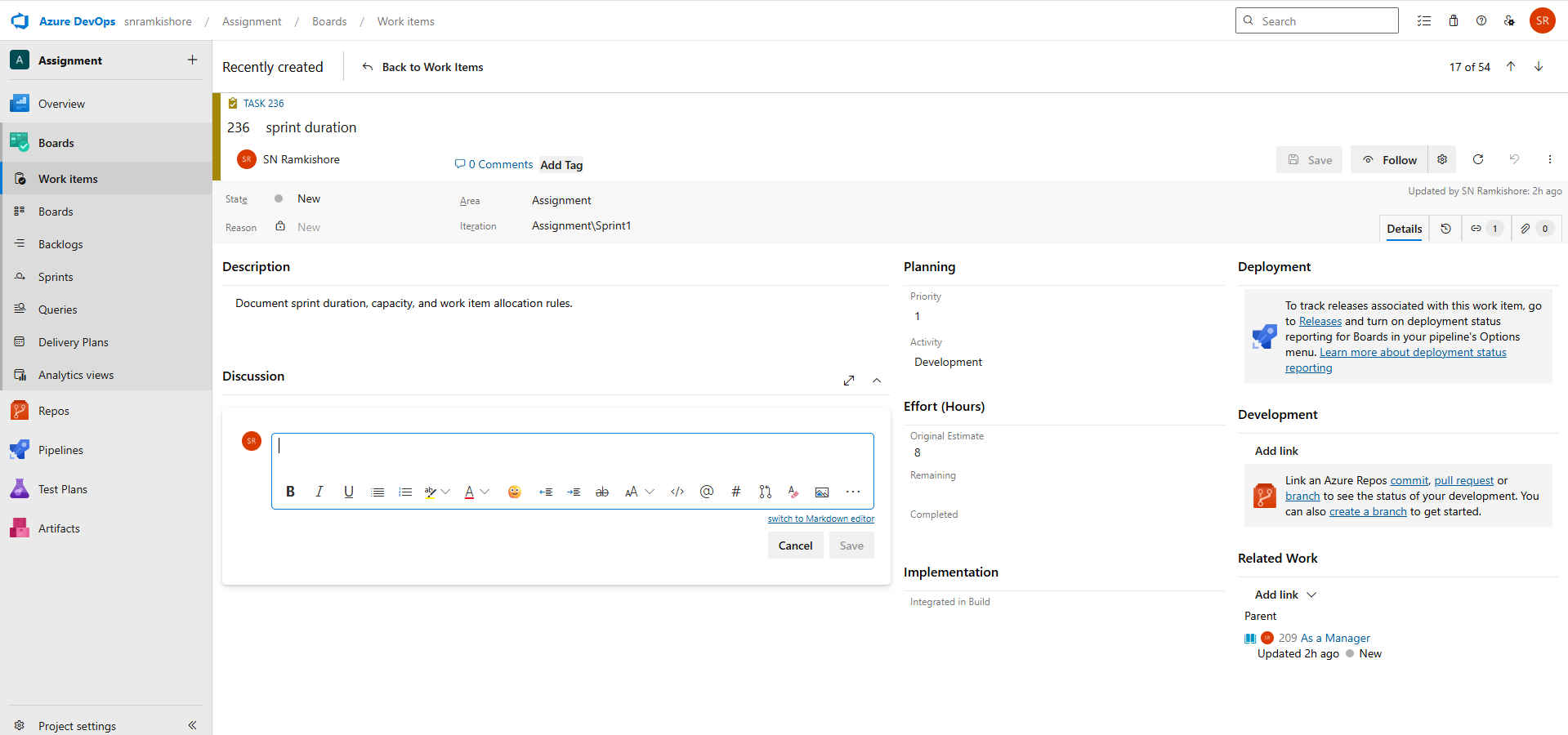


**Tasks:**

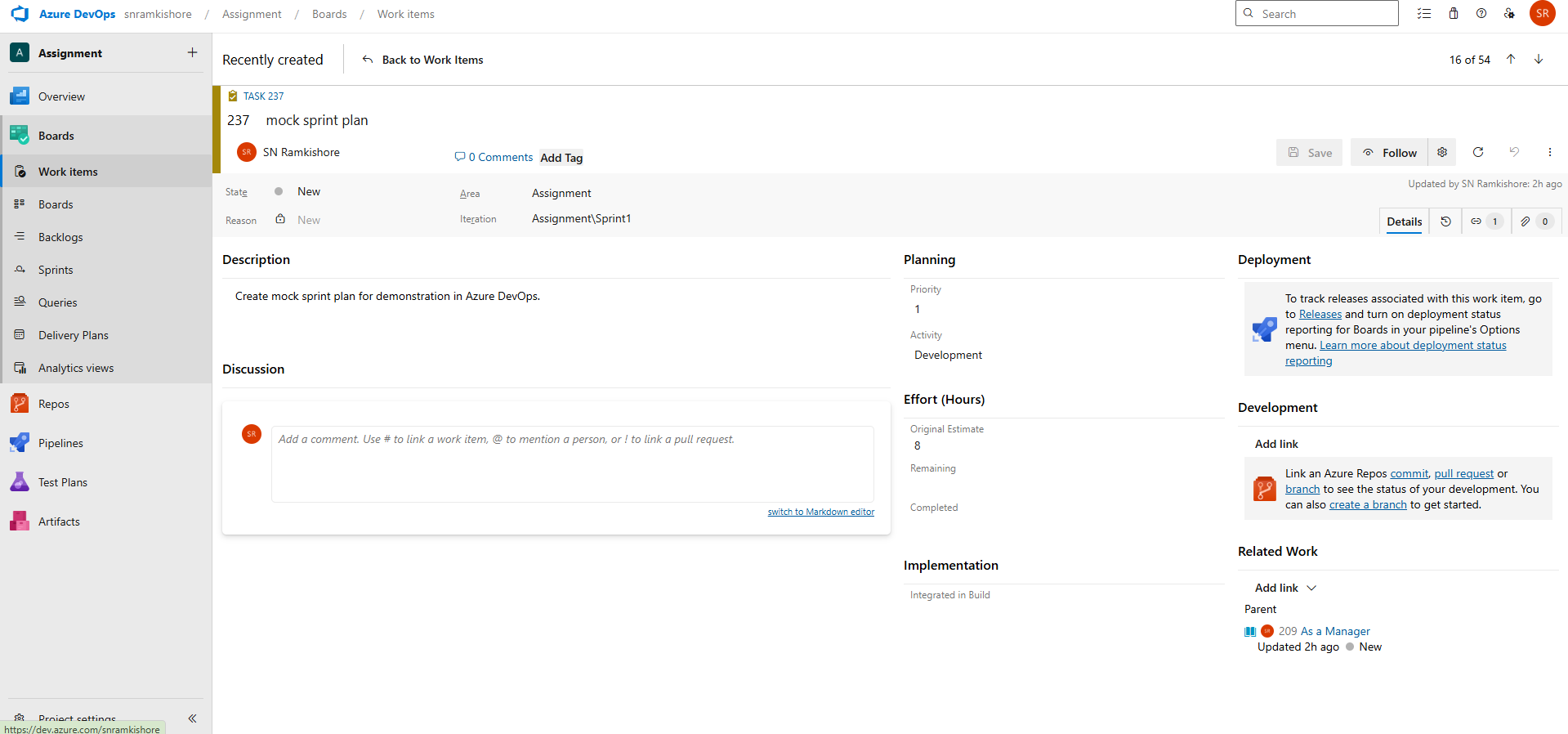
1. Define sprint creation process and sprint naming conventions.



2.Document sprint duration, capacity, and work item allocation rules.

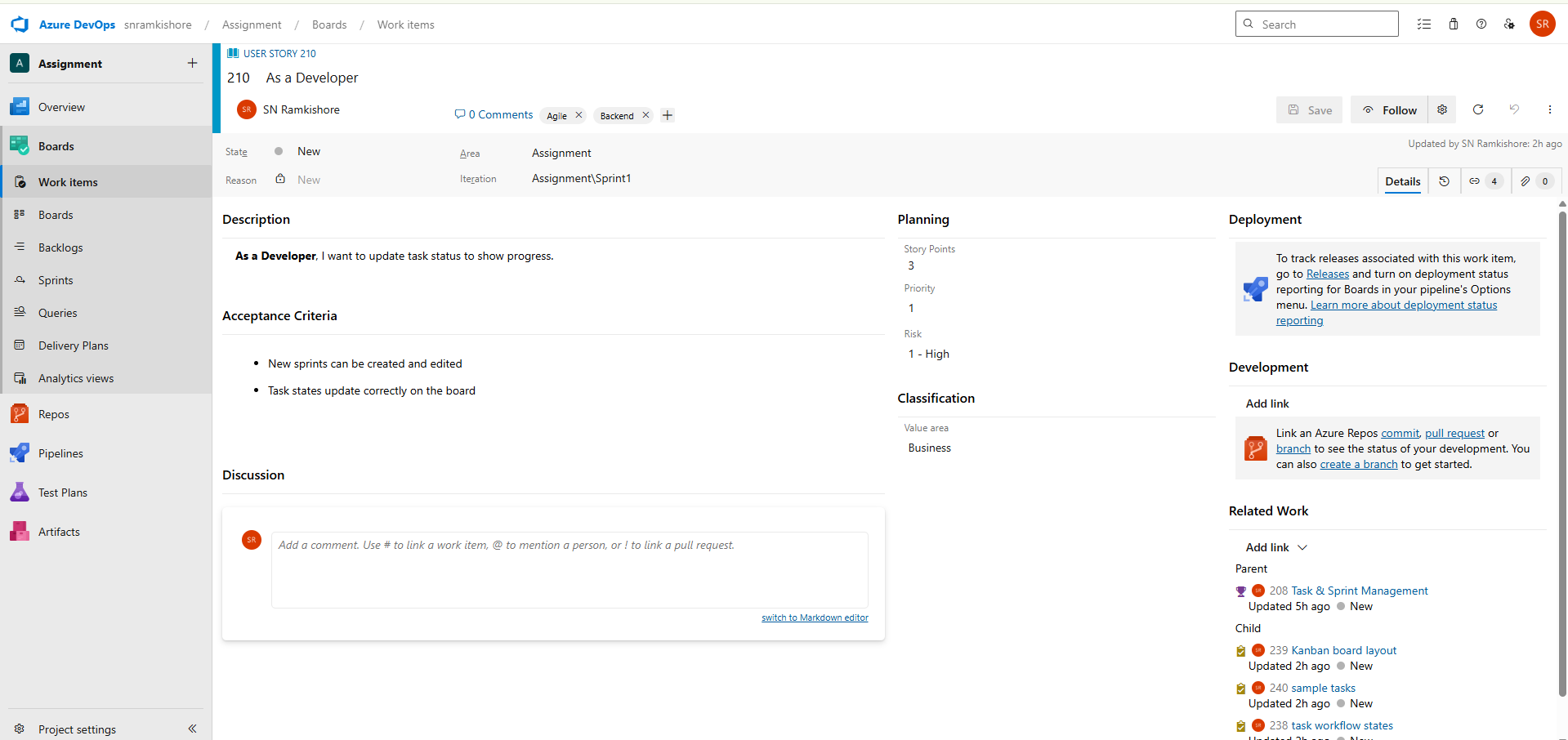


3.Create mock sprint plan for demonstration in Azure DevOps.

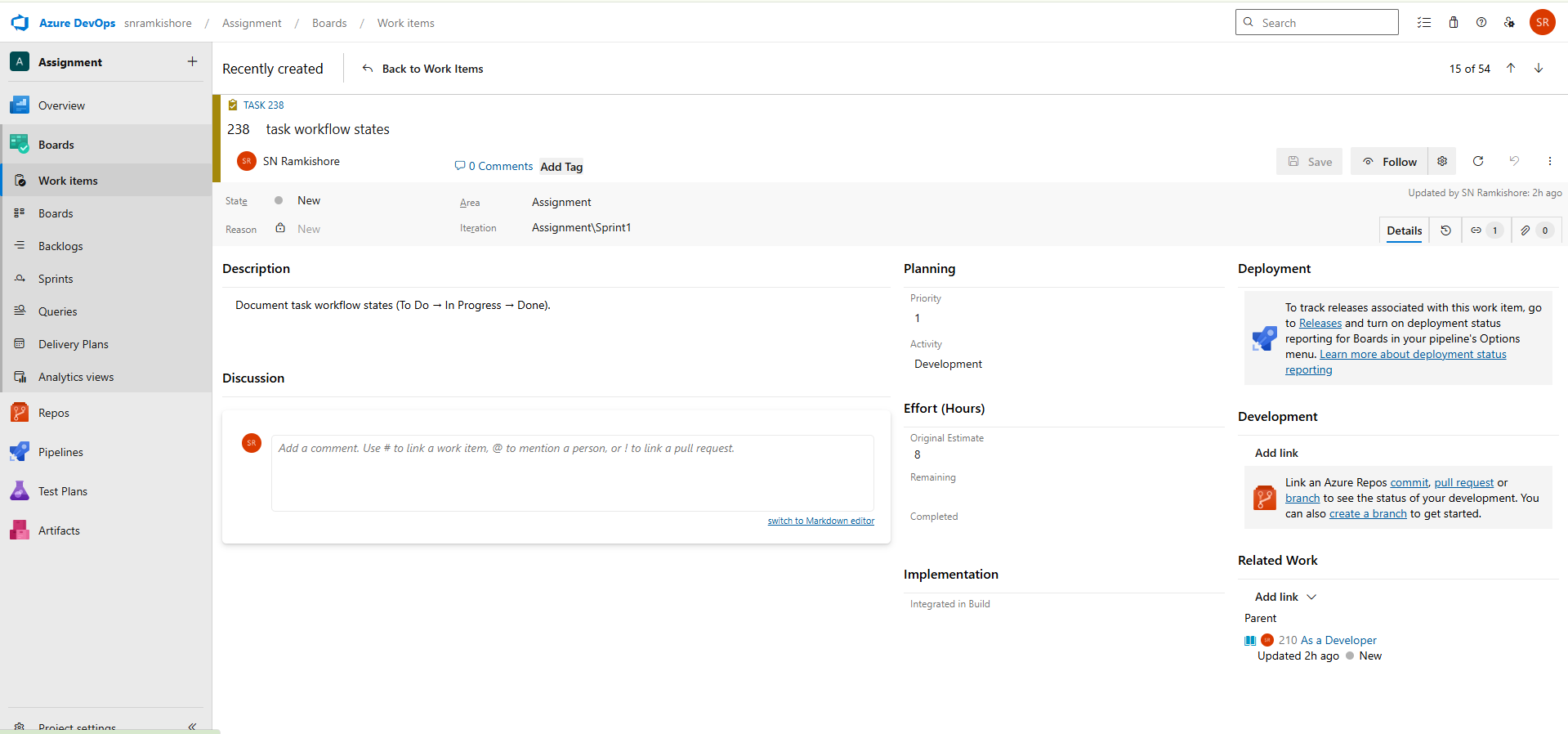


**User Story 2:**

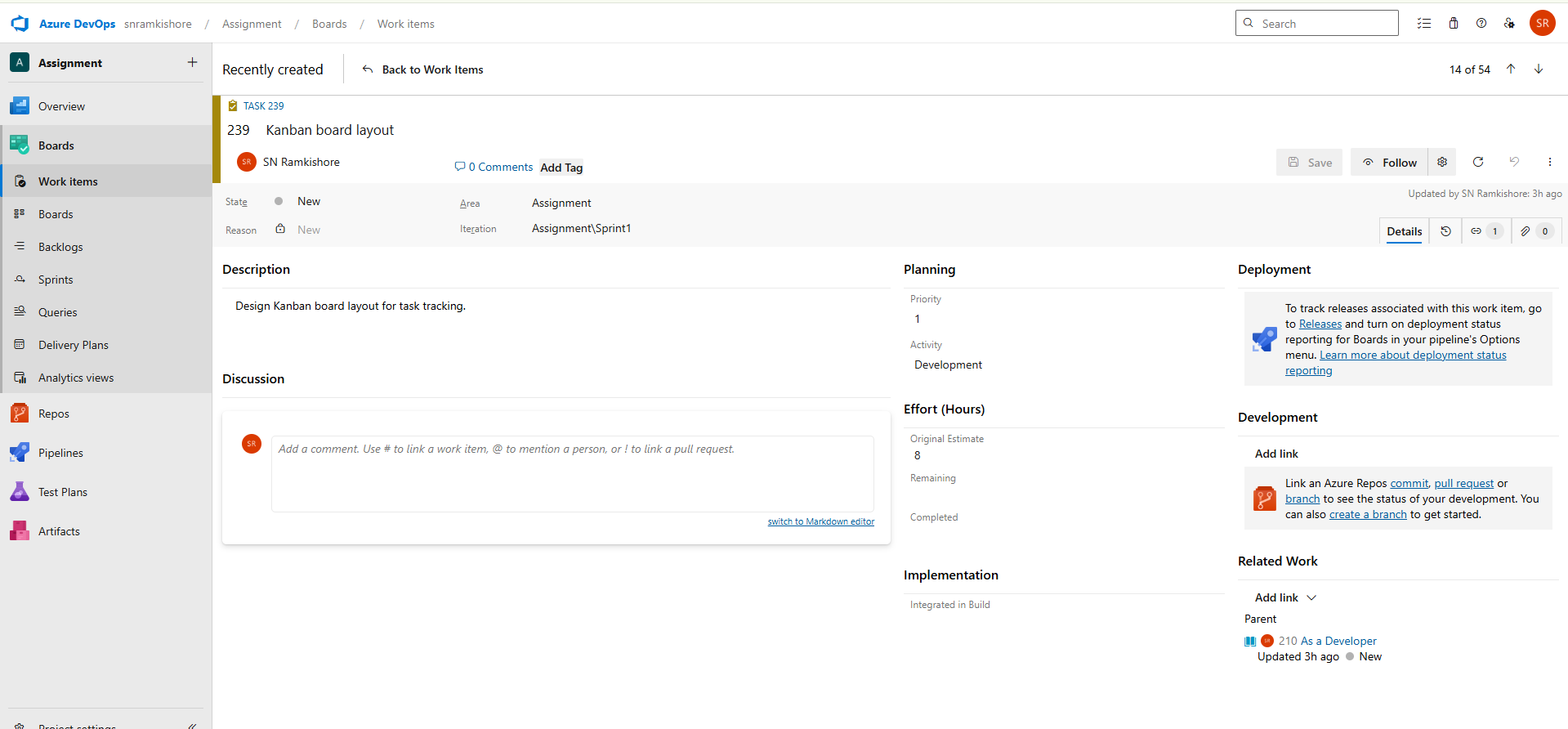
**As a Developer,** I want to update task status to show progress.

  
**Tasks:**

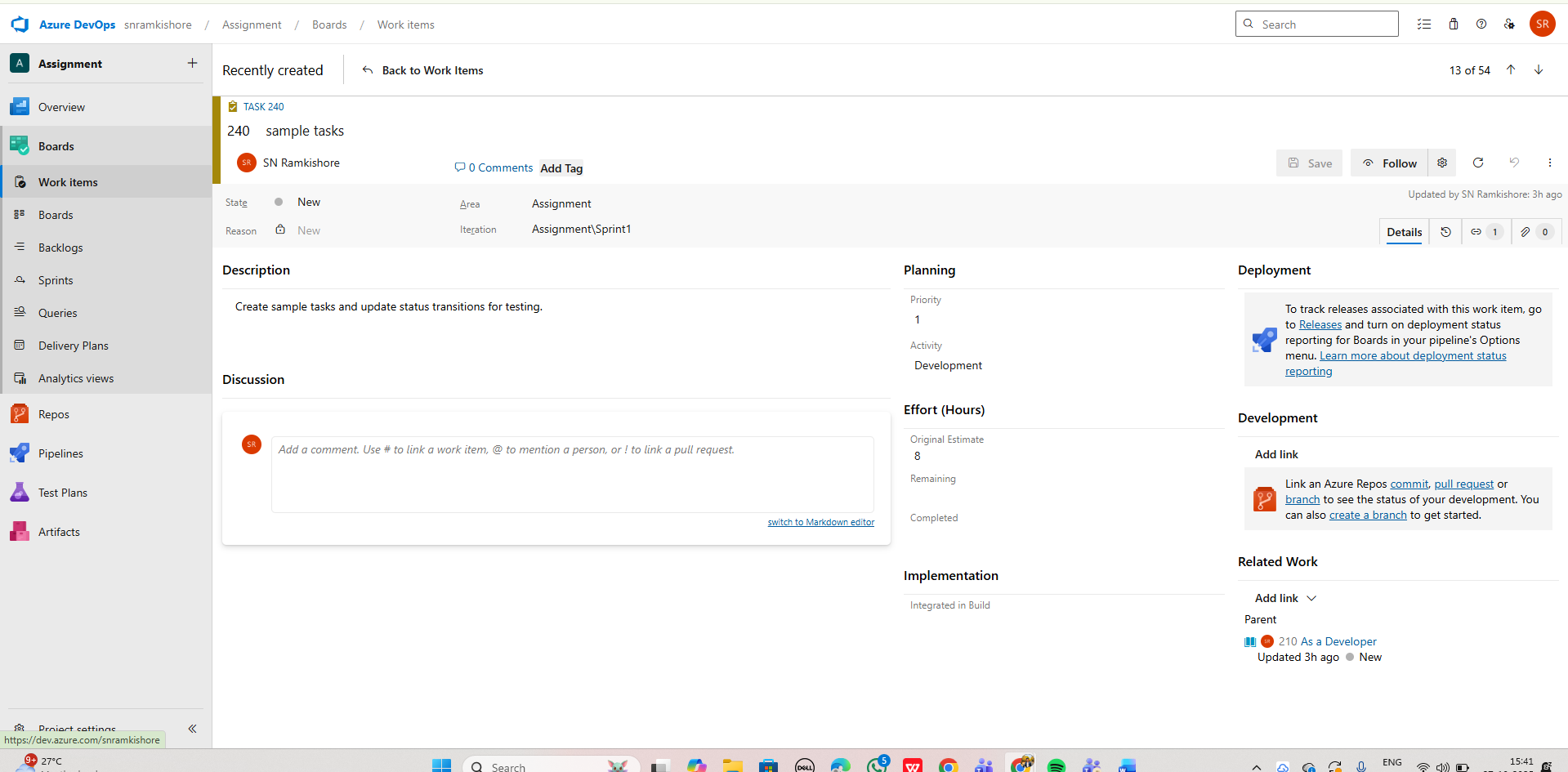
1. Document task workflow states (To Do → In Progress → Done).



2.Design Kanban board layout for task tracking.

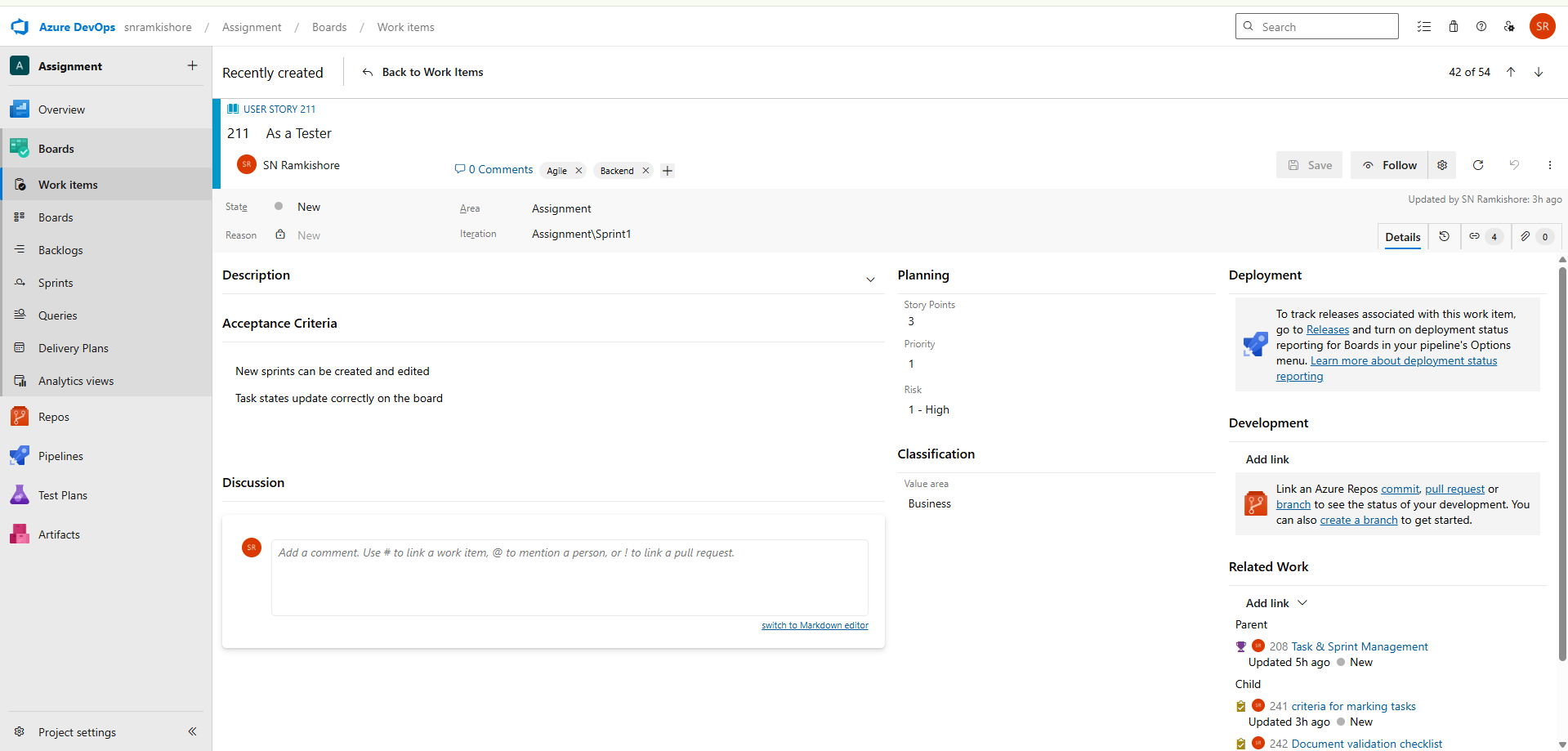


3.Create sample tasks and update status transitions for testing.

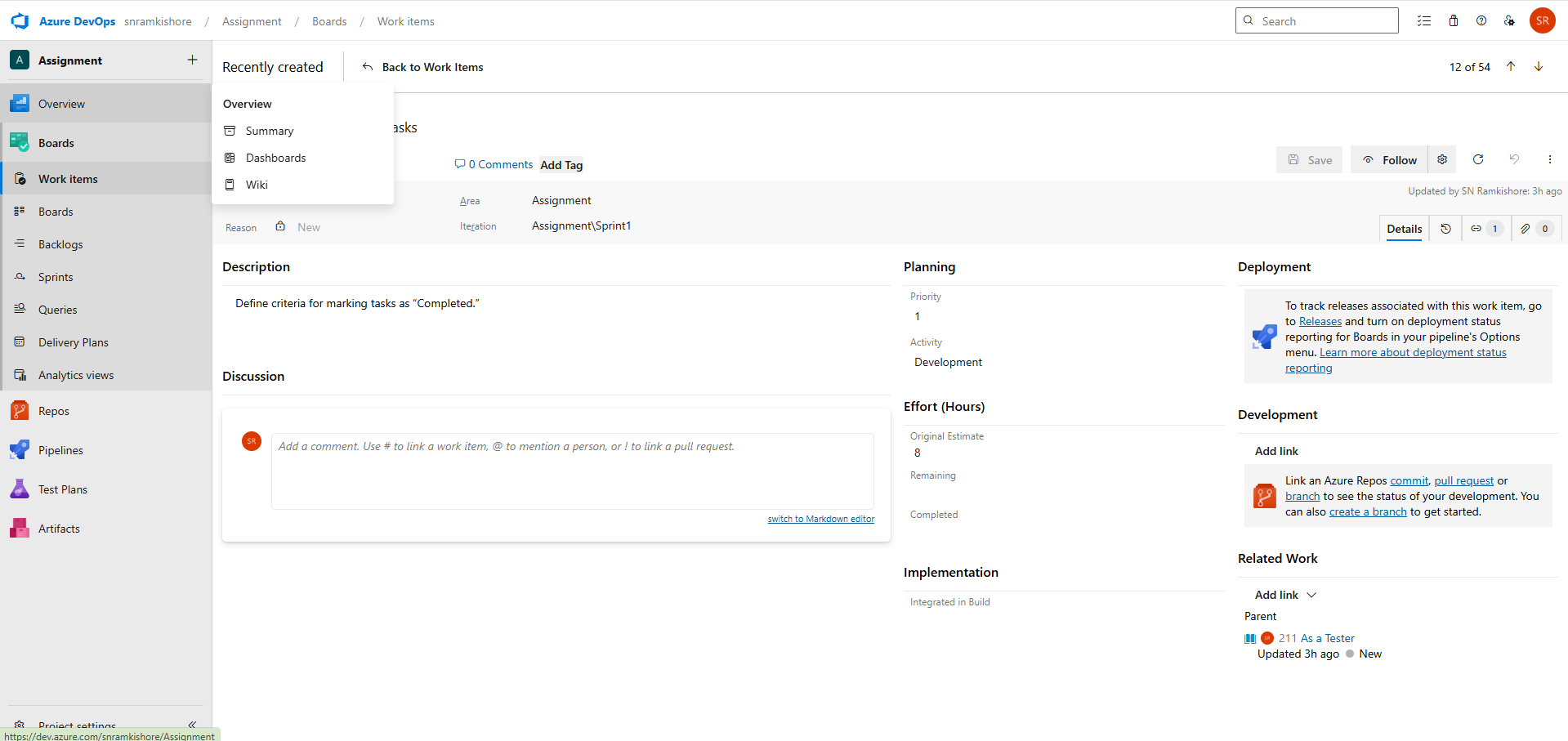


**User Story 3:**

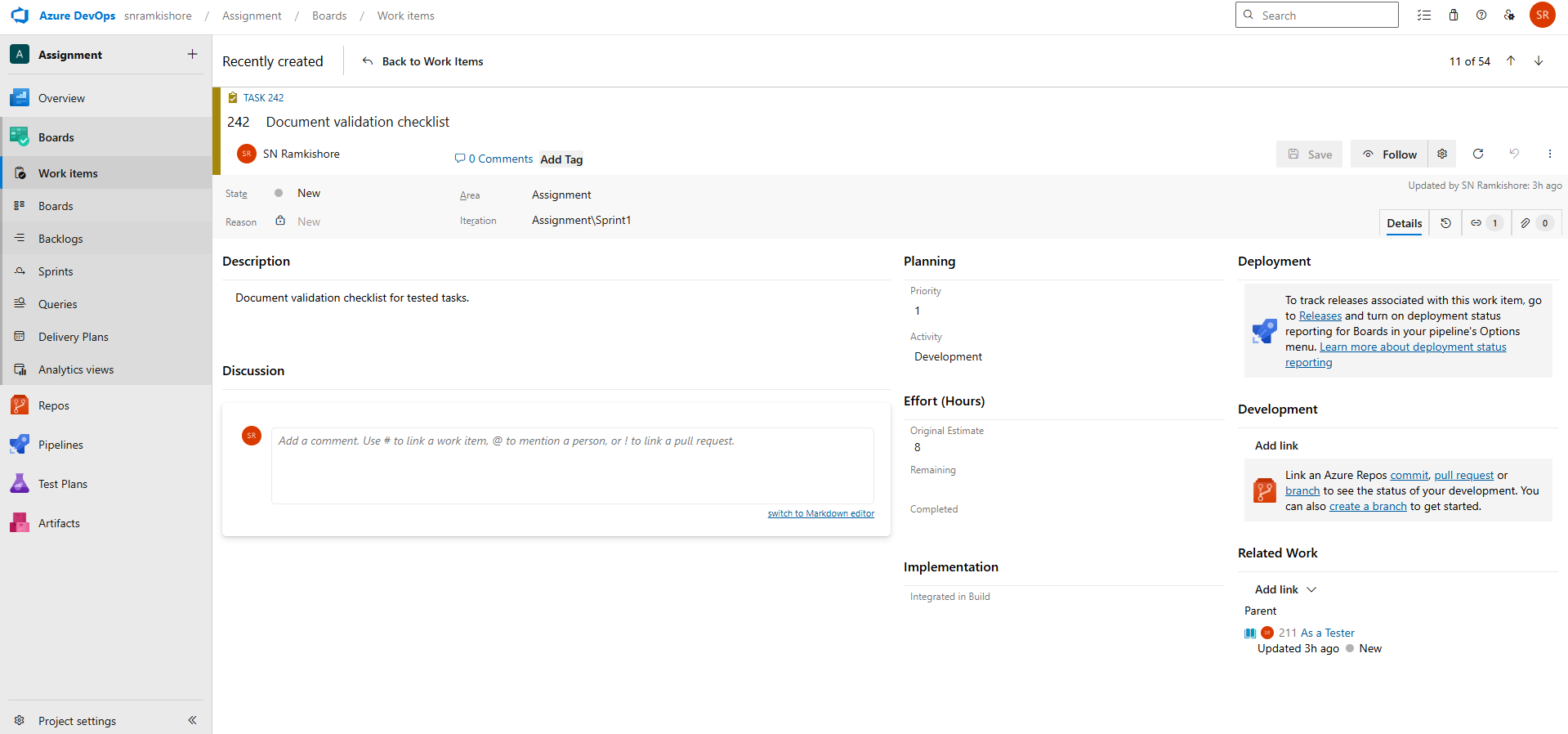
**As a Tester**, I want to mark tasks as completed after validation.

  
**Tasks:**

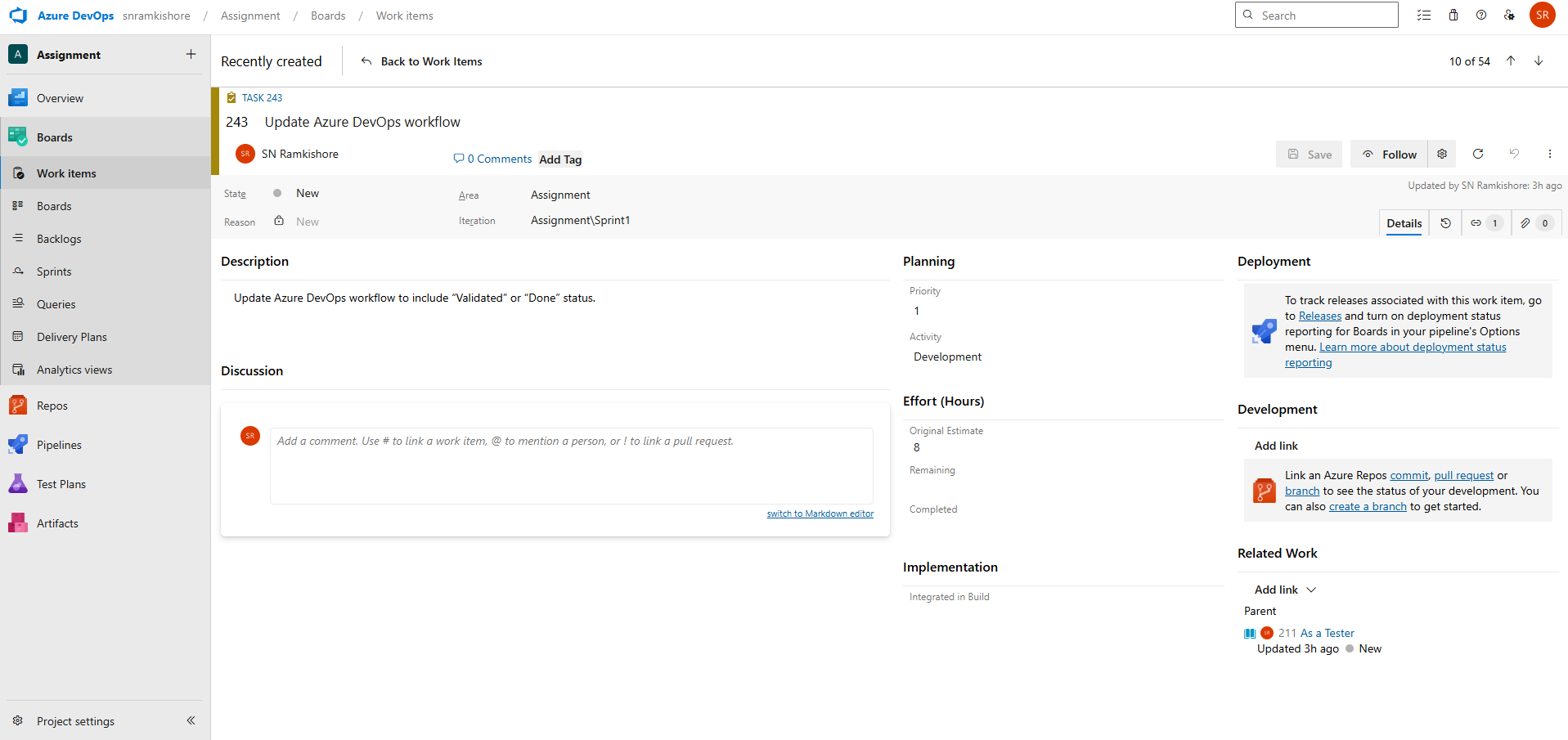
1. Define criteria for marking tasks as “Completed.”



2.Document validation checklist for tested tasks.

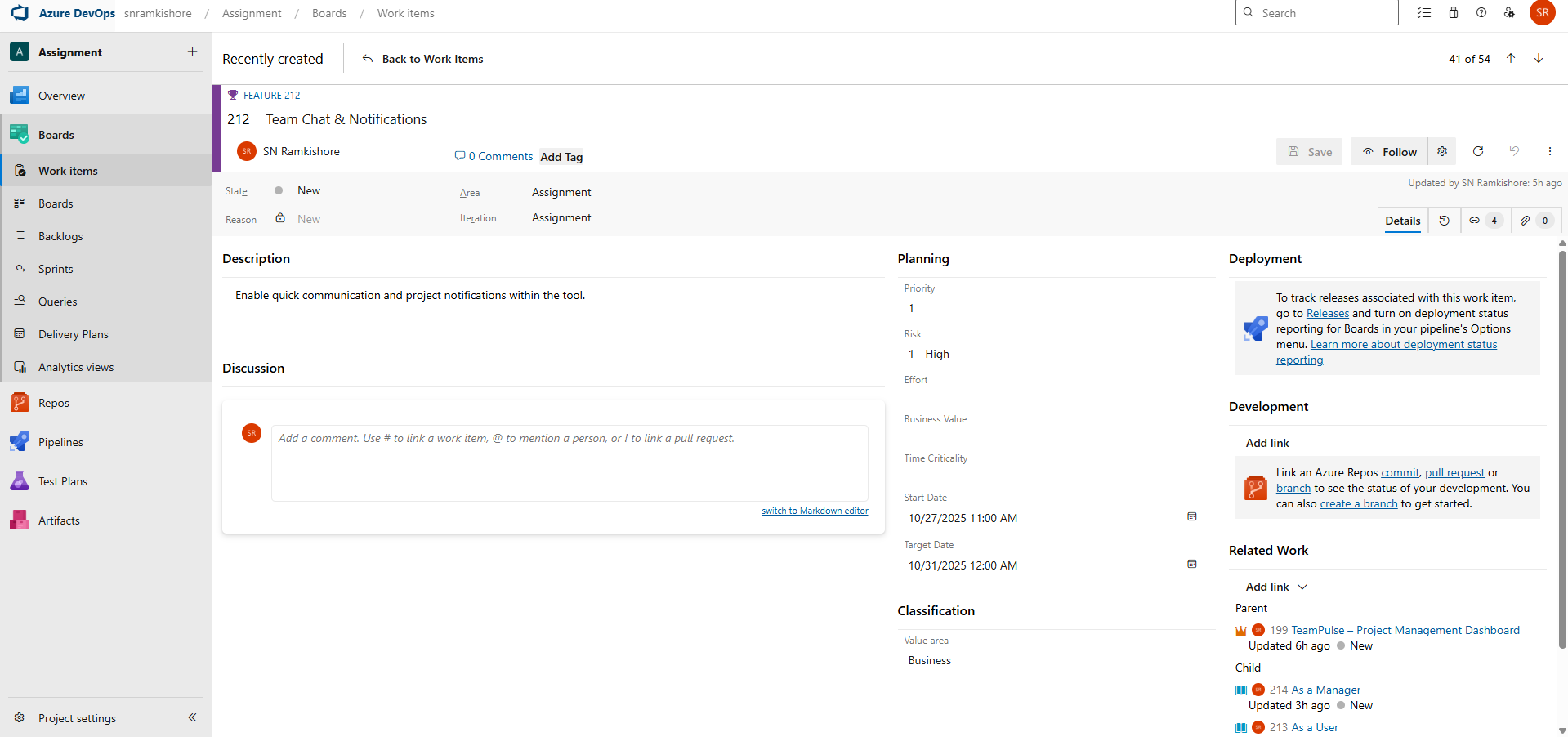


3.Update Azure DevOps workflow to include “Validated” or “Done” status.

****

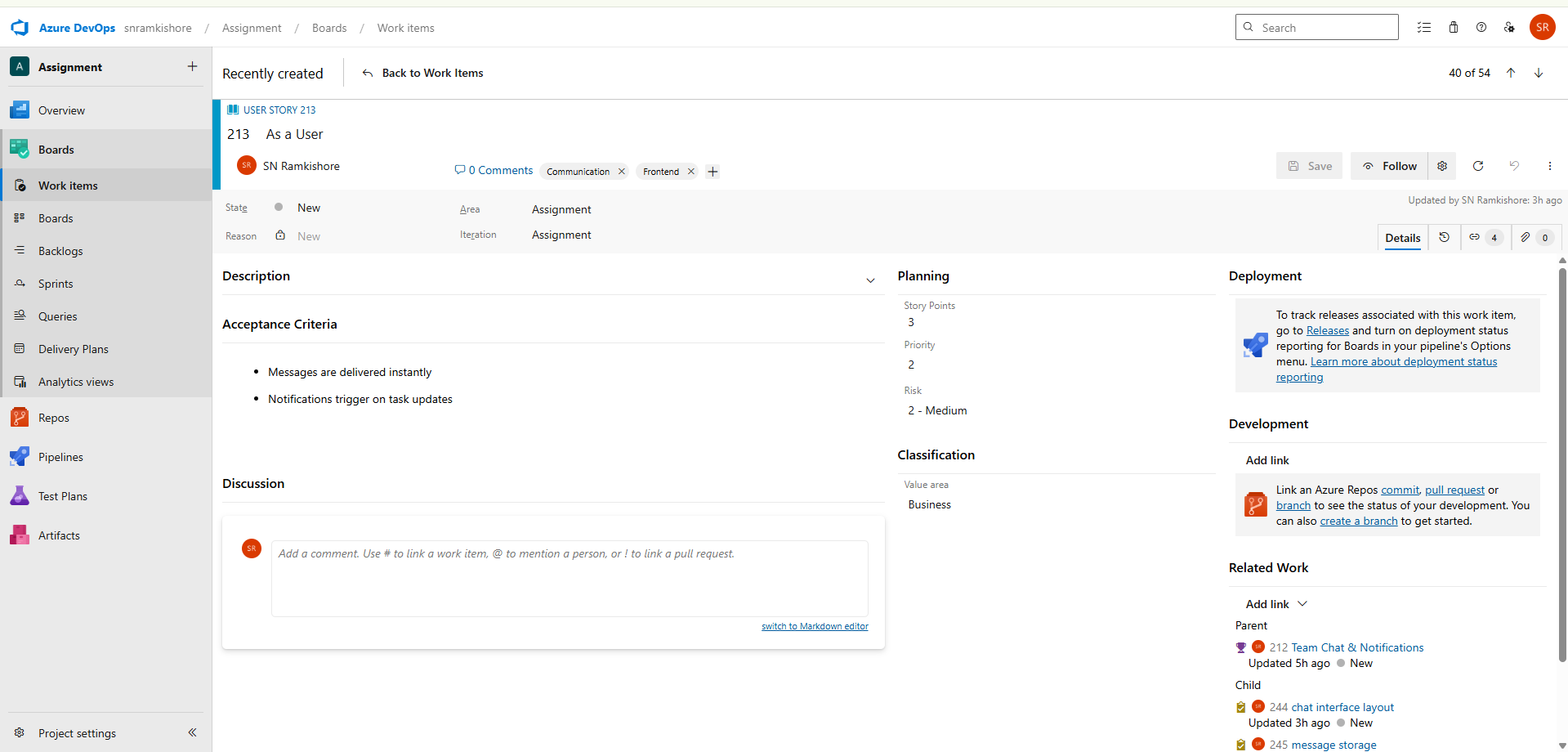
**Feature 4: Team Chat & Notifications**

* Enable quick communication and project notifications within the tool.



**User Story 1:**

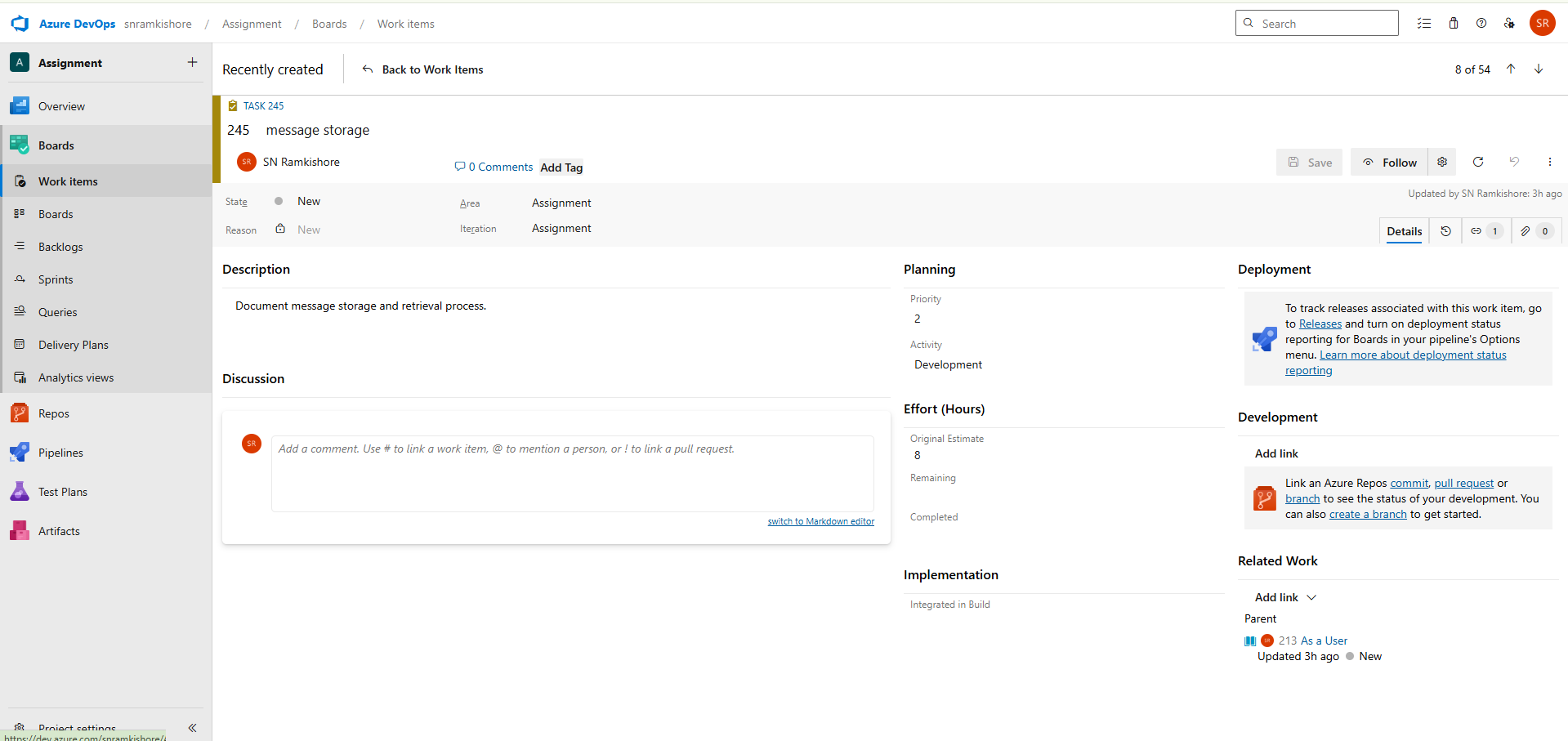
**As a User,** I want to send and receive messages to collaborate with team members.

 **Tasks:**

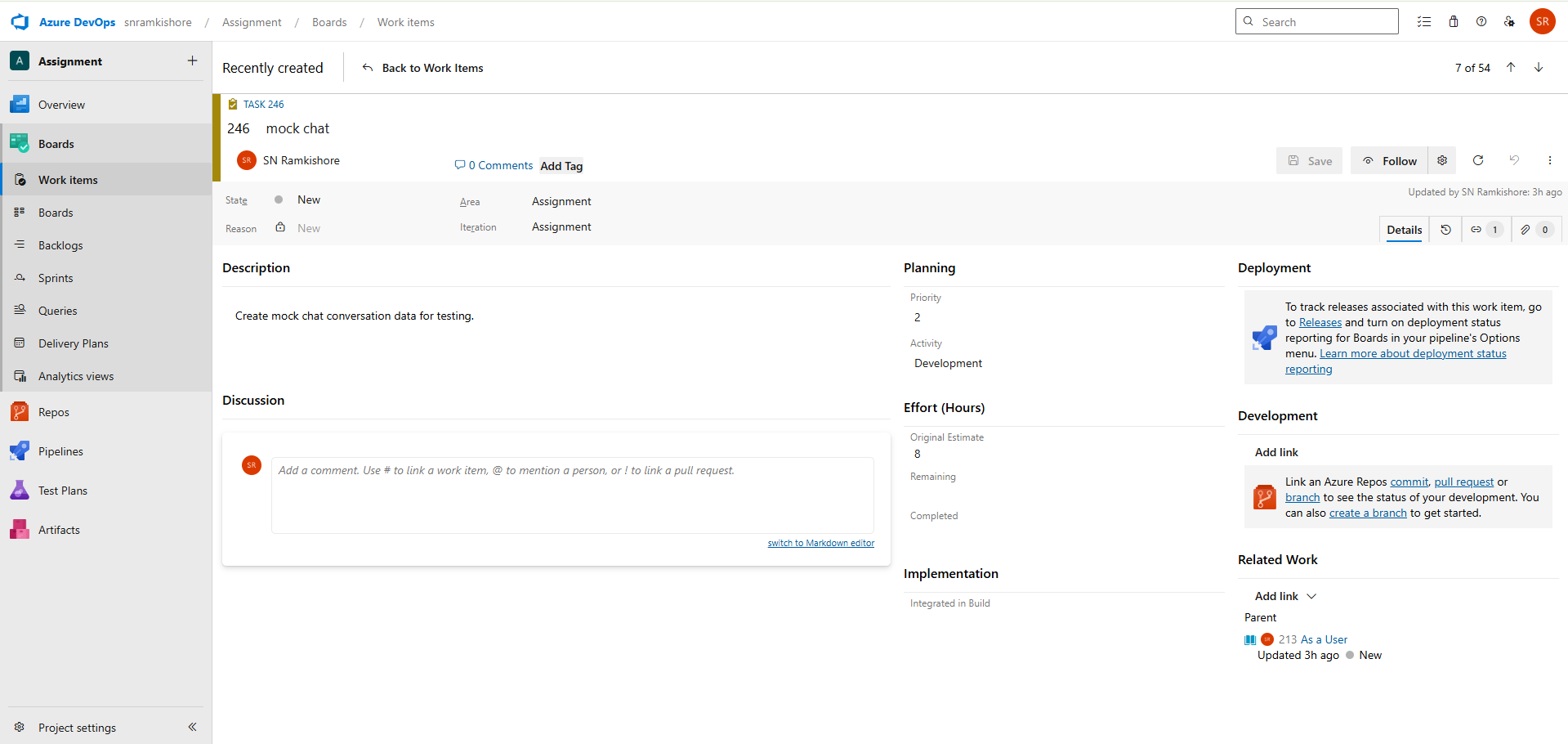
1. Design chat interface layout and message flow diagram.



2.Document message storage and retrieval process.

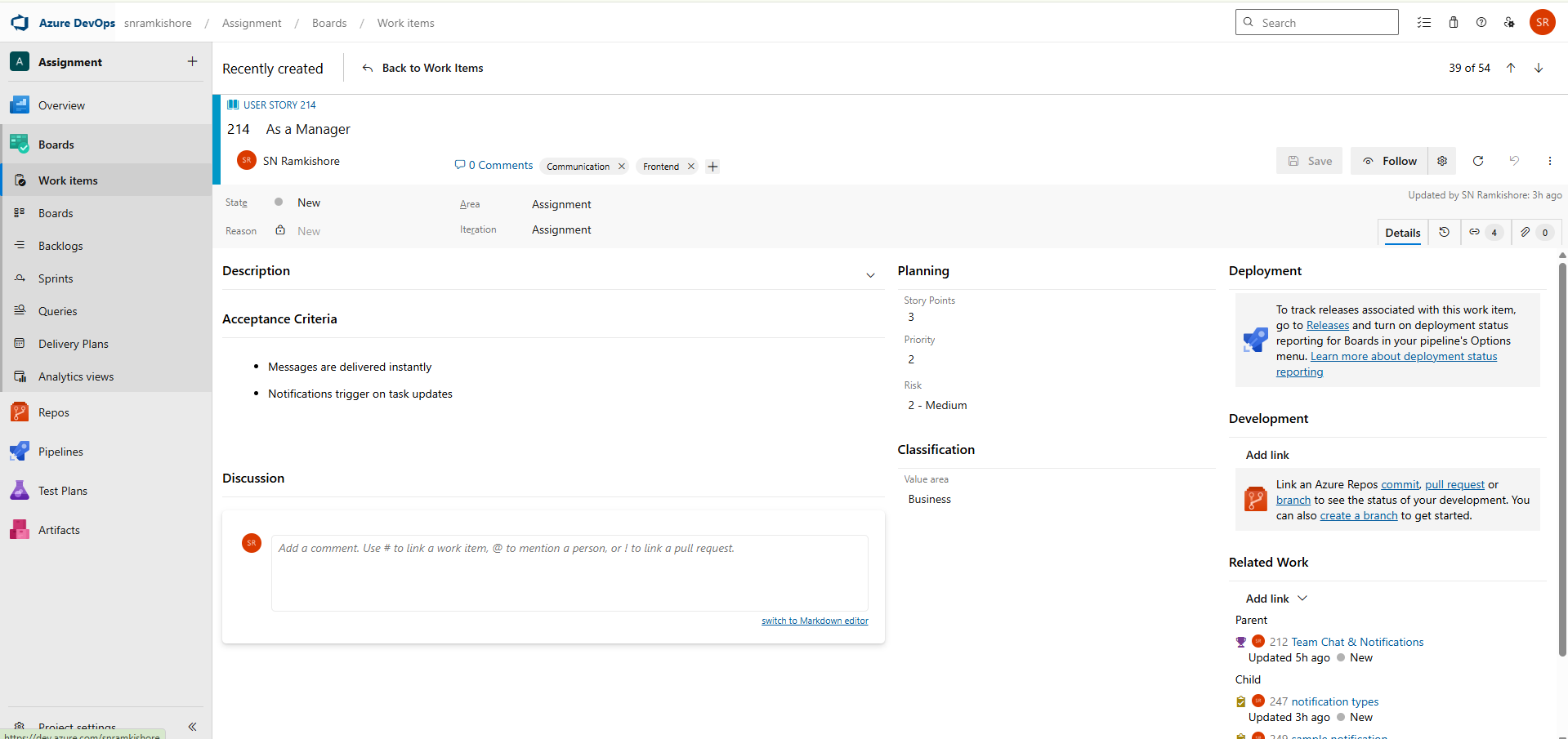


3.Create mock chat conversation data for testing.

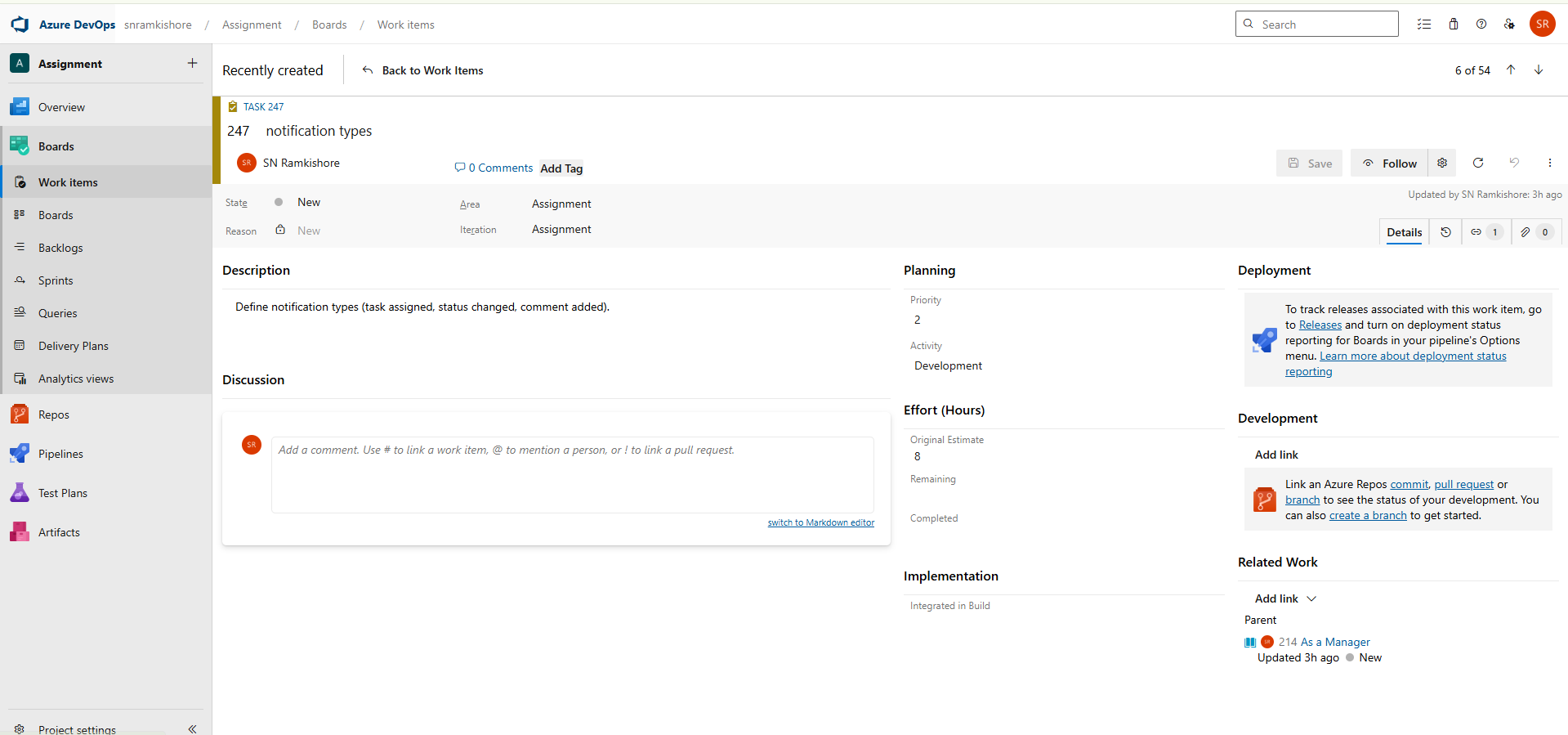


**User Story 2:**

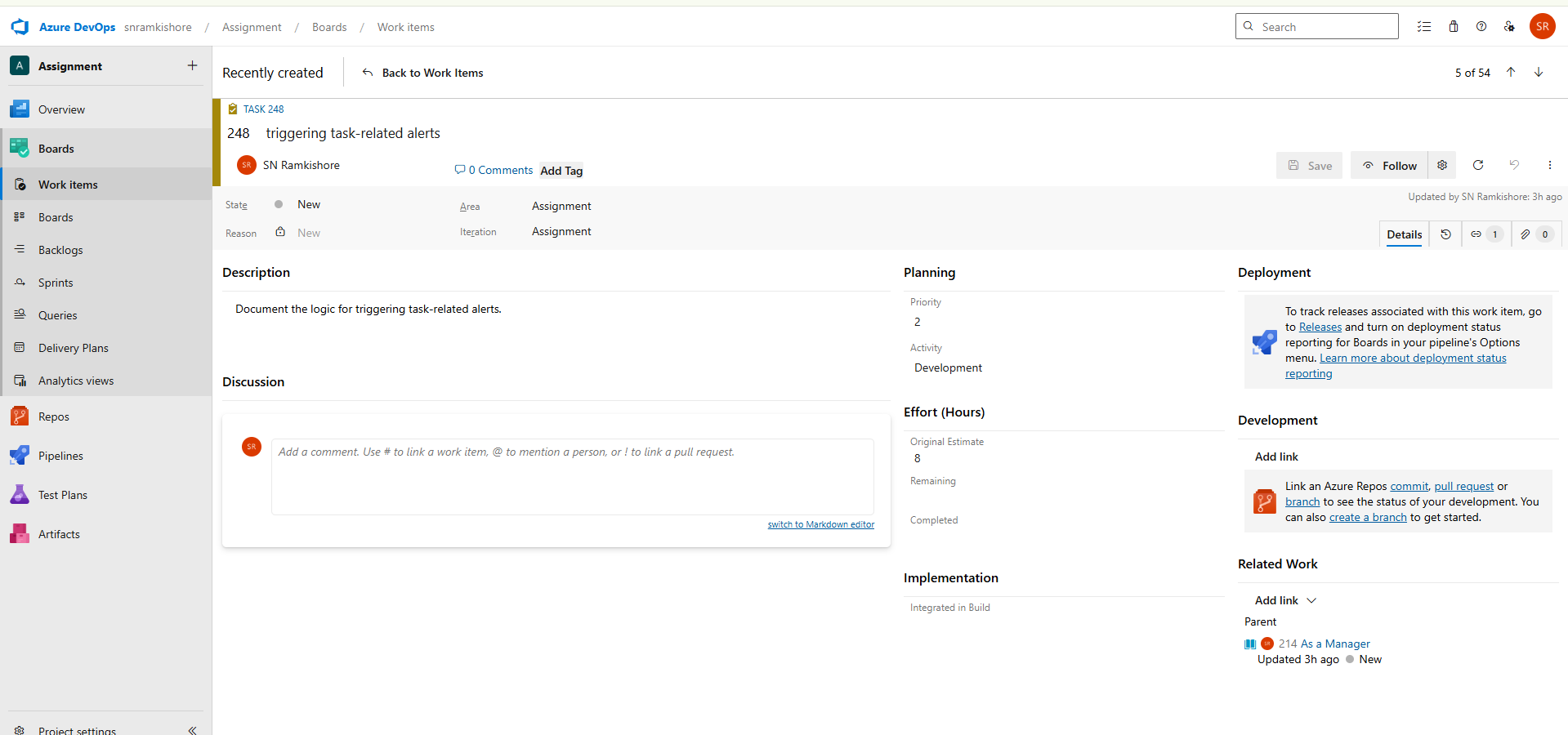
**As a Manager,** I want to receive alerts for task updates.

  
**Tasks:**

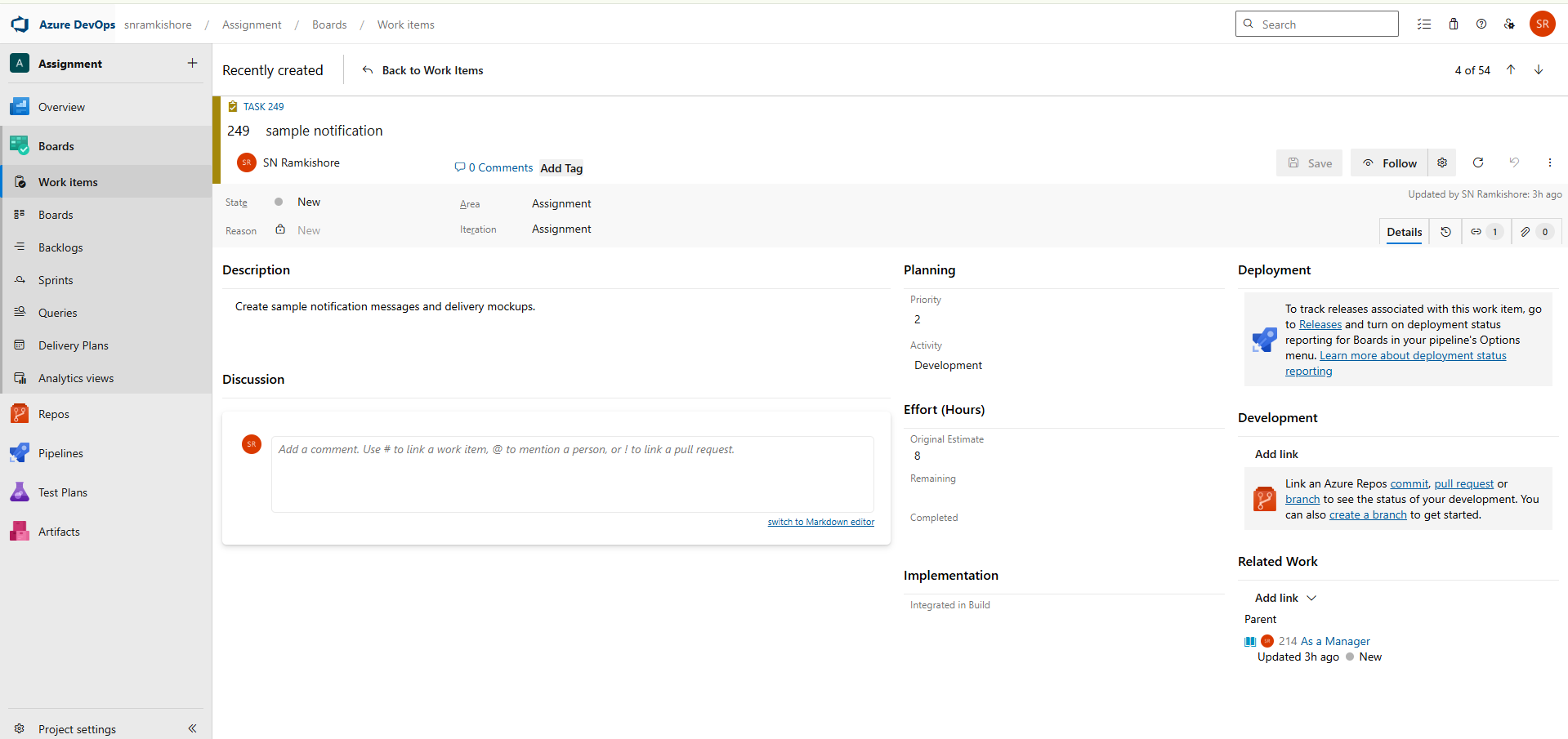
1. Define notification types (task assigned, status changed, comment added).



1. Document the logic for triggering task-related alerts.

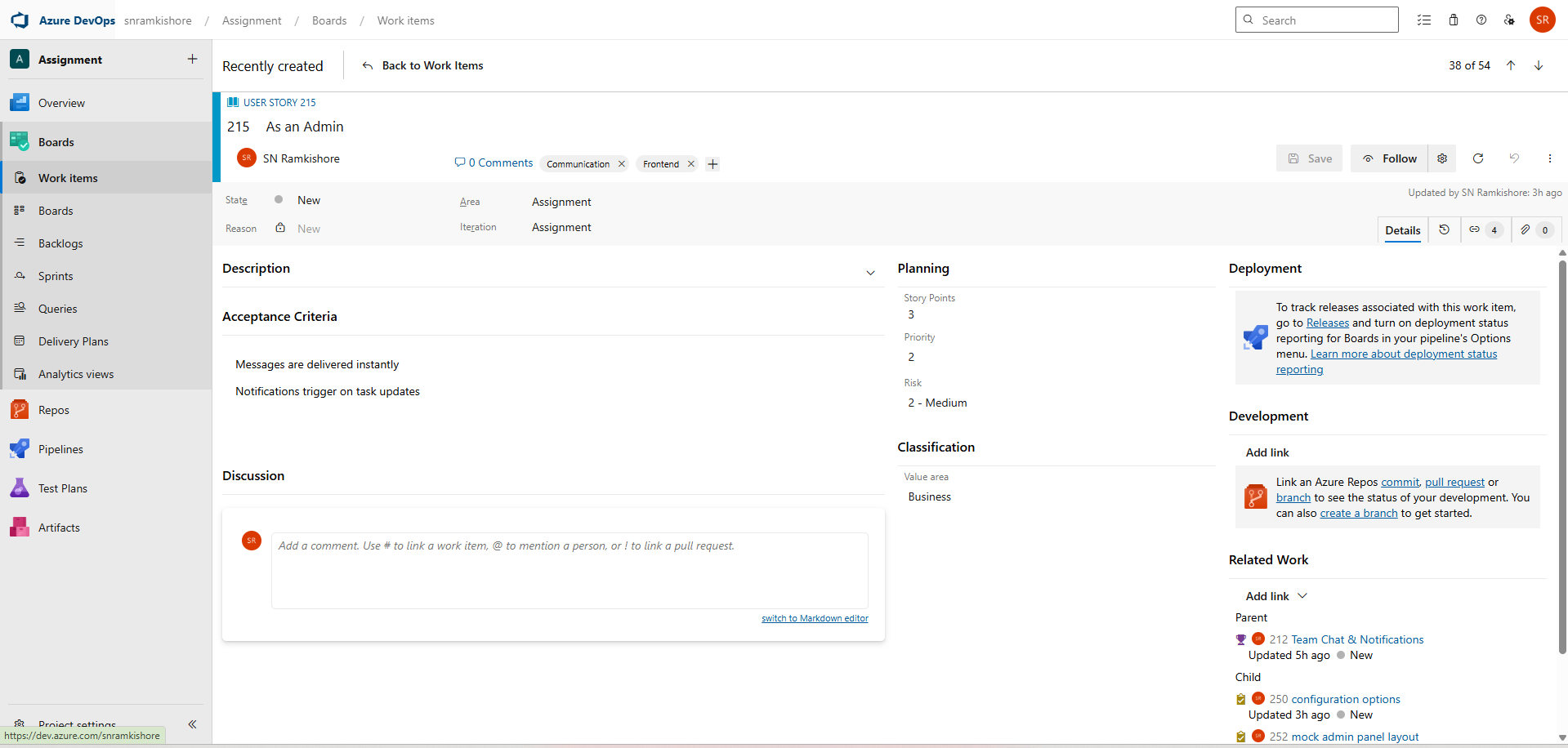


1. Create sample notification messages and delivery mockups.

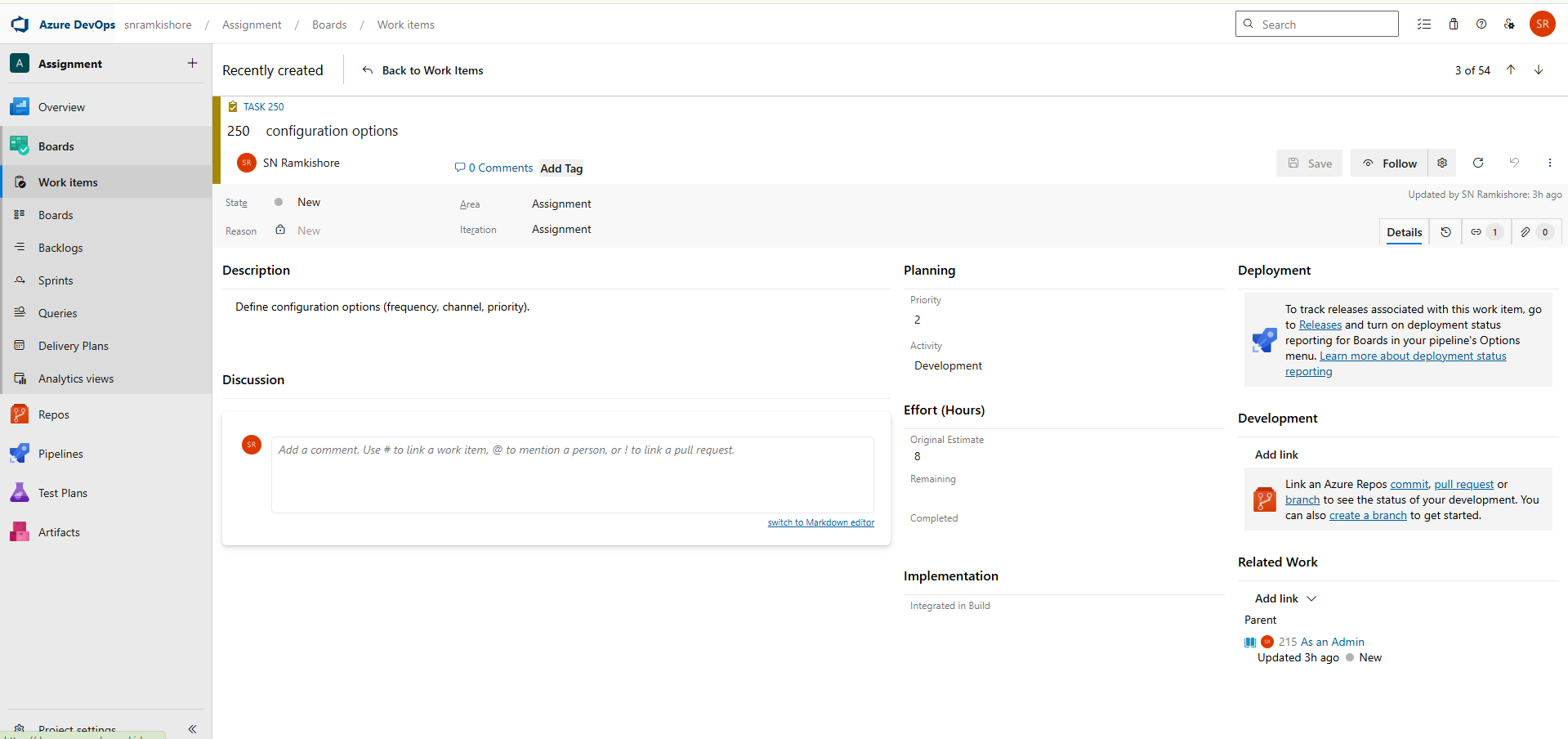


**User Story 3:**

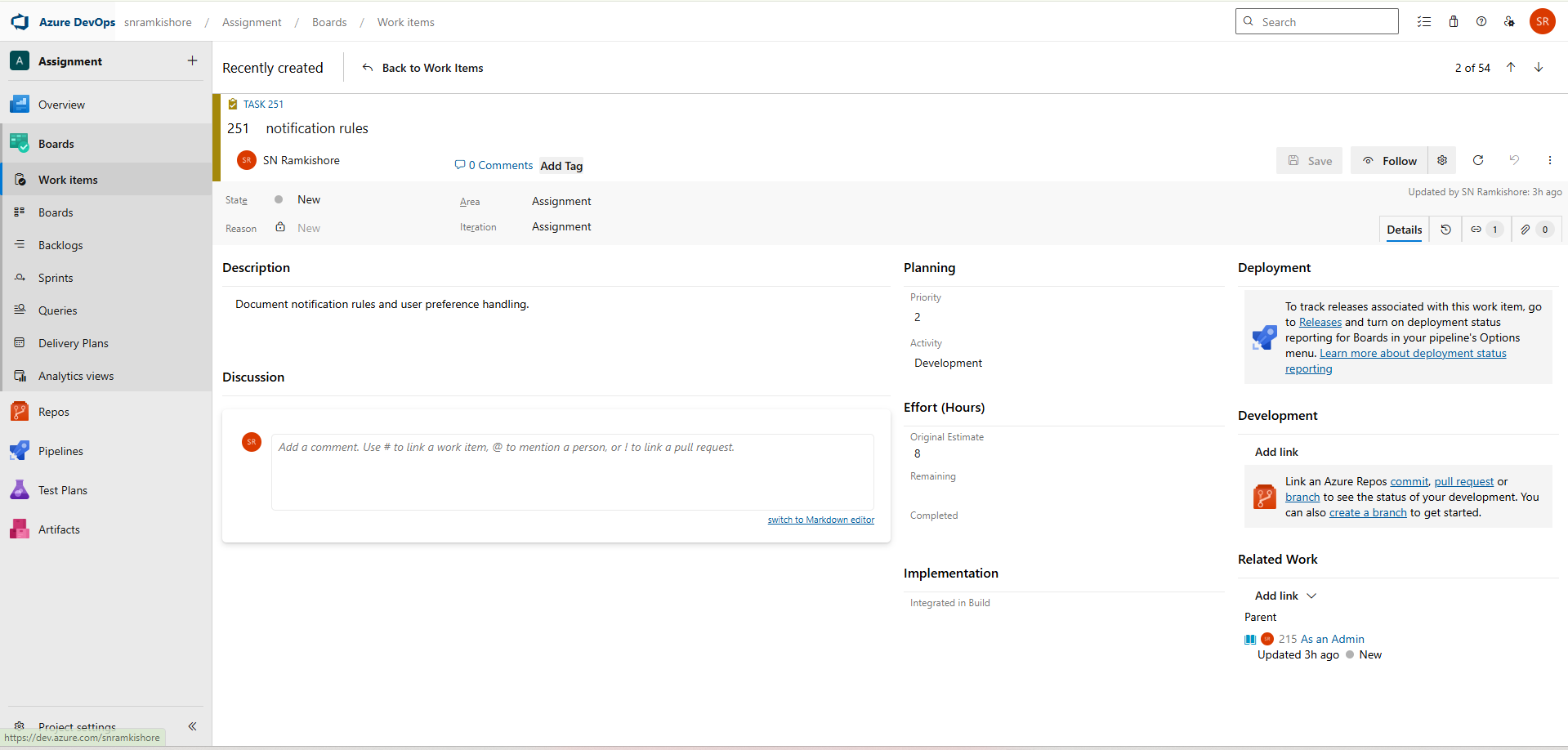
**As an Admin**, I want to configure notification settings.

  
**Tasks:**

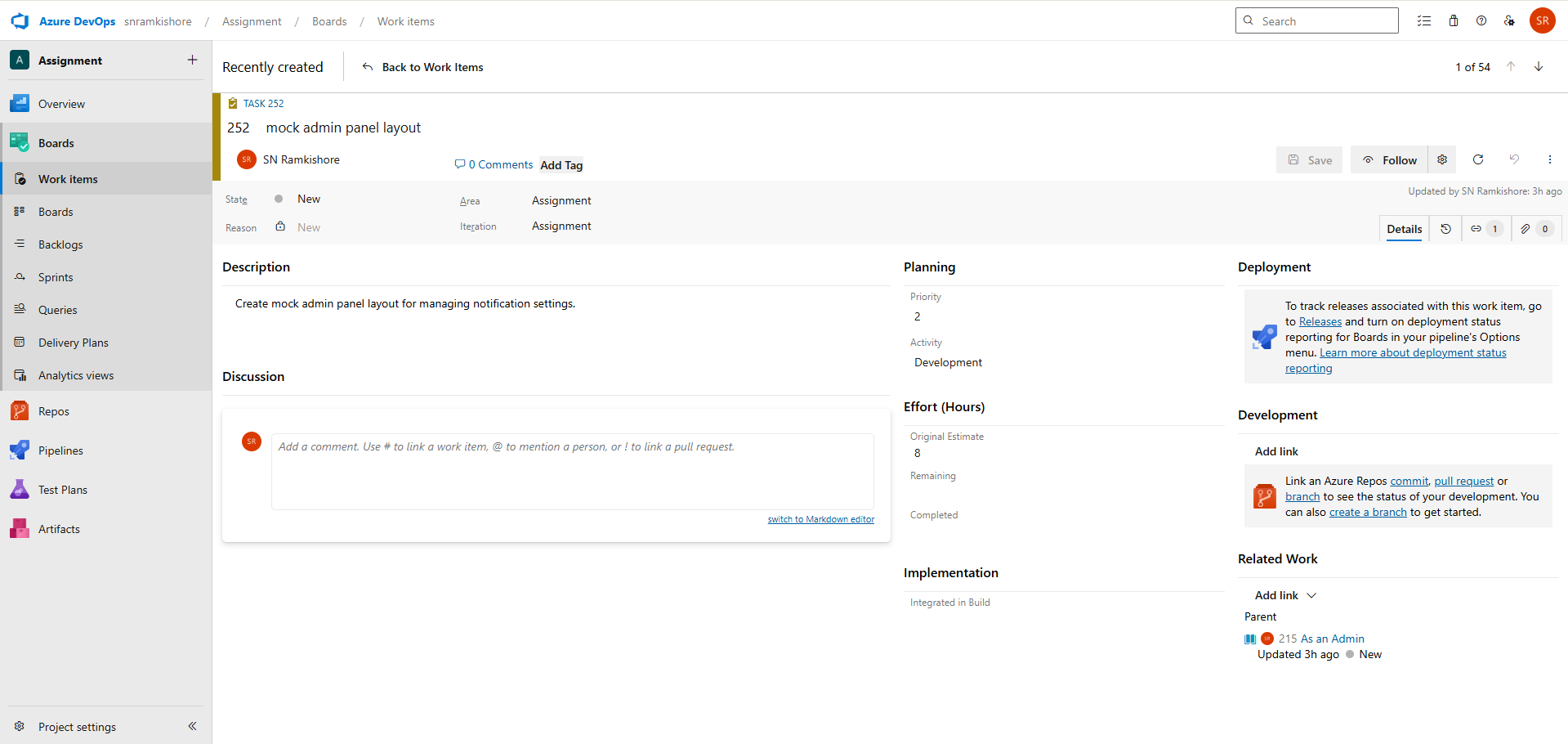
1. Define configuration options (frequency, channel, priority).



1. Document notification rules and user preference handling.



1. Create mock admin panel layout for managing notification settings.



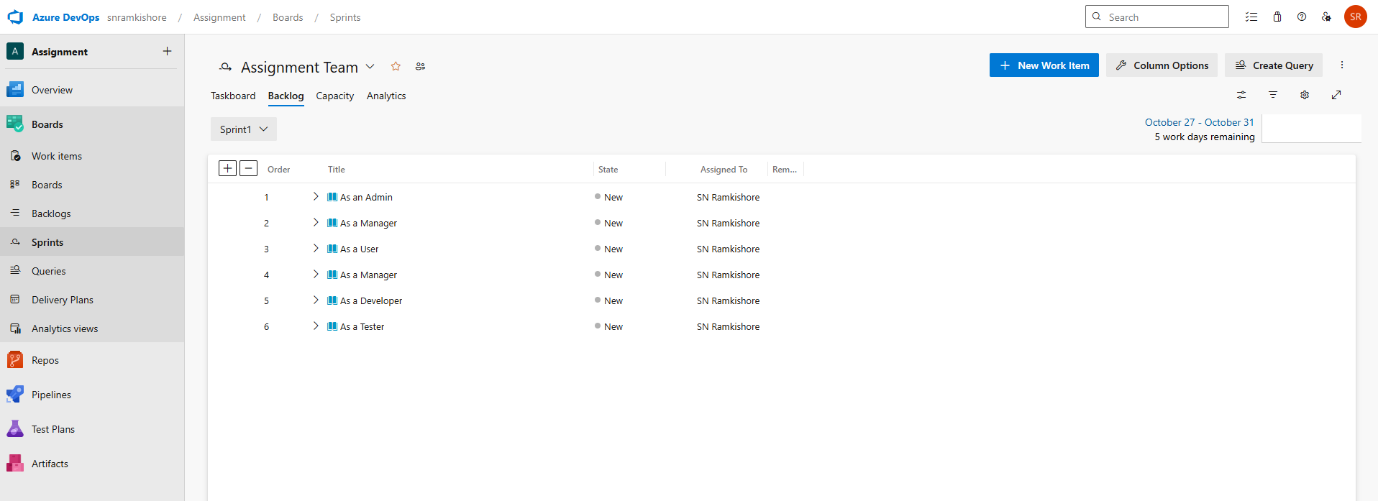
**1.Iteration / Sprint Planning**

**Sprint 1**

1Go to Project Settings → Project Configuration → Iterations.

2.Click New child → Name it “Sprint 1” and set dates (e.g., 01-Nov to 14-Nov).

3.Go to Boards → Backlogs, select a few User Stories or Tasks → Right-click → Move to Iteration → Sprint 1.

****

**2. Tags and Priorities**

Tags and priorities help classify work items by functionality and urgency, improving coordination across teams.

Example for TeamPulse:  
• Frontend – UI elements like dashboards and chat (P2)  
• Backend – Authentication, logic, APIs (P1)  
• Database – Schema and data design (P1)  
• Analytics – Reports, KPIs, charts (P3)

In Azure DevOps, tags are added directly to work items, while priorities (P1, P2, P3) indicate importance. Filtering by tags or priority ensures efficient tracking and balanced workloads.

**3. Acceptance Criteria**

Acceptance criteria define what success looks like for each User Story. They ensure a shared understanding between developers, testers, and stakeholders.

Examples for TeamPulse:  
• Login Authentication: Must reject invalid credentials, auto-expire sessions, and redirect after login.  
• KPI Visualization: Dashboard must display accurate sprint data and refresh periodically.  
• Task Assignment: Tasks should link to a sprint and contain mandatory fields.  
• Team Chat: Only project members can send messages; unread notifications must appear.

**4. Definition of Done (DoD)**

The Definition of Done ensures consistent completion standards for all work items. A task or story is considered Done when:  
• All acceptance criteria are satisfied.  
• Documentation is complete and peer-reviewed.  
• Work is ready for testing or sprint closure.  
• Dependencies are closed and linked properly.

**5. Burndown Chart (Optional – Bonus)**

A Burndown Chart visualizes sprint progress, showing the remaining work over time. It helps track completion trends and forecast sprint success.

Implementation in Azure DevOps:  
• Go to Boards → Sprints → Analytics tab.  
• Select the Burndown Chart widget.  
• Assign all Sprint 1 items to visualize progress.  
The chart’s X-axis represents time, and the Y-axis shows remaining tasks or story points.

This visualization helps teams identify bottlenecks early and maintain a sustainable development pace.

