# Creating a Jira Board from an SRSD Document

I can certainly help you plan how to create a Jira board based on a Software Requirements Specification and Design (SRSD) document, though I can't directly create a Jira board for you since I don't have direct access to Jira instances.

How to Create a Jira Board from SRSD

1. **Analyze the SRSD document**:
   * Identify all functional requirements
   * Note non-functional requirements
   * Review design components and architecture
2. **Break down into Jira issues**:
   * Create Epics for major feature areas
   * Create User Stories for specific requirements
   * Create Tasks for implementation details
   * Create Sub-tasks for granular work items
3. **Set up your Jira board**:
   * Create a new Scrum or Kanban board in Jira
   * Define columns that match your workflow (e.g., To Do, In Progress, Done)
   * Configure swimlanes if needed (by priority, team, etc.)

What I Can Provide

If you share your SRSD document (or key sections), I can help:

* Suggest an appropriate board structure
* Recommend how to break down requirements into Jira issues
* Provide sample issue titles and descriptions
* Suggest workflow configurations
* Recommend labels, components or custom fields

# Jira Board Setup for Fixithub

Based on the SRD document, I'll outline how to structure your Jira board with appropriate epics, user stories, and tasks. This will help your team track development progress effectively.

Recommended Board Structure

**Board Type:** Scrum (for iterative development)  
**Issue Types:** Epics → User Stories → Tasks/Sub-tasks  
**Columns:** Backlog → To Do → In Progress → Review → Done  
**Swimlanes:** By priority (Critical, High, Medium, Low)

Epics Breakdown

1. User Authentication & Profiles

text

- Citizen Registration & Login

- Admin Authentication System

- Problem Solver Profile Creation

- Role-Based Access Control

2. Issue Reporting & Management

text

- Core Reporting Functionality

- Issue Tracking System

- Status Updates & Notifications

- Community Engagement Features

3. Admin Dashboard & Tools

text

- Comprehensive Monitoring Dashboard

- Issue Assignment System

- Reporting & Analytics

- System Configuration

4. Infrastructure & Non-Functional

text

- Performance Optimization

- Security Implementation

- Scalability Setup

- Accessibility Compliance

Sample User Stories with Acceptance Criteria

Epic: Issue Reporting & Management

**US-1: As a citizen, I want to report issues with images and location**

text

- Fields: Title, Description, Image upload, Auto-location tagging

- Generates Unique Issue ID

- Saves to database with timestamp

- Notifies relevant admins

AC:

- Image compression to <2MB

- Location accuracy within 50m

- Submission confirmation shown

**US-2: As an admin, I want to assign issues to problem solvers**

text

- Dropdown of verified problem solvers

- Assignment notification sent

- Issue status changes to "Assigned"

AC:

- Assignment history tracked

- Email/SMS confirmation

- Dashboard updates in real-time

Epic: Admin Dashboard & Tools

**US-3: As a management admin, I need visual analytics**

text

- Charts for issue volume by category/location

- Resolution time metrics

- Export to PDF/CSV

AC:

- Loads in <3 seconds with 10k records

- Date range filtering

- Mobile-responsive

Technical Tasks Examples

For US-1 (Issue Reporting):

text

1. Design image upload component (React)

2. Implement location capture (Google Maps API)

3. Create API endpoint for issue submission (Node.js)

4. Write unit tests for submission flow

5. Implement input validation

For Infrastructure:

text

1. Configure JWT authentication

2. Set up PostgreSQL with PostGIS

3. Implement database encryption

4. Configure AWS auto-scaling

5. WCAG compliance audit

Recommended Custom Fields

1. **Module:** (Authentication, Reporting, Dashboard, etc.)
2. **User Role:** (Citizen, Admin, Problem Solver)
3. **Priority:** (Critical, High, Medium, Low)
4. **Tech Stack:** (Frontend, Backend, Database, etc.)

Labels to Consider

text

#security #location #notifications #ui #api #scalability #accessibility

Suggested Workflow

1. Requirements review (link SRD document to each epic)
2. Technical design
3. Implementation
4. Peer review
5. QA testing
6. Deployment

Would you like me to provide more detailed breakdowns for any specific section or suggest specific Jira configurations for your team's workflow?