

Project Specification: Lira (Jira-Lite)

Course: CS-UY 4513 – Software Engineering

Professor: Dr. DePasquale

Project Title: Lira (Jira-Lite)

Date: 9/27/2025

1.0 Project Overview

SprintBoard is a Rails-based SaaS for planning and tracking software work. It provides:

- **Issues** (bug, task/feature, subtask), statuses, assignees, and comments. (Modeled after Jira's issue types and workflows.)
 - For every feature, there will be a progress bar with the number of tasks completed divided by the total tasks.
 - Subtasks
 - Bugs
- **Time-oriented work:** Admins/developers can add deadlines to tasks and view them on a calendar
- **Roles & permissions** at the project level (Admin, Developer, Customer) to control who can create, move, and manage work.

Each module must expose a well-defined API for inter-module communication.

2.0 Core Requirements

The project must adhere to the following mandatory characteristics:

2.1 User-Based System

The application must be a user-based system, requiring a secure user authentication and authorization framework. User accounts will be managed via standard email/password registration and login. Passwords and all other sensitive information will be stored as hashes and operations relating to them will only involve hashes.

2.2 User Roles (Minimum 3)

- **Admin:** Full access to the system and can manage tasks, subtasks, and users
- **Developer:** Can manage tasks, subtasks
- **Customer:** Can add to tasks only (feature requests)
 - Follows AGILE methodologies

2.3 Persistent Storage (\leq 10 tables)

The application must use a relational database to persist all data. The database schema must be well-designed and contain no more than 10 tables.

1. **users** – credentials, profile
2. **projects** – name, key, description, settings.
3. **project_memberships** – user_id, project_id, **role** (enum: customer, admin, developer).
4. **issues** – project_id, assignee_id, **issue_type** (enum: bug/task/subtask), **status** (enum: not started, in progress, done, cancelled), title, description, start_date, end_date, eta
5. **dependencies**: issue_id, dependent_id
6. OPEN TO ADDING MORE TABLES AS NEEDED :)

2.4 Modular Architecture (3–5 Modules)

1. Identity & Access

- Registration/login, profile, project membership, role checks.
- **Dependency:** none; foundational for all others.

2. Projects & Boards

- Create projects; configure
- Board views: column WIP visualization, filters.
- **Dependency:** Identity & Access.

3. Issues & Workflow

- CRUD for issues; transitions between statuses per board mapping; comments; attachments; basic search/filter.
- **Dependency:** Projects & Boards (to validate allowed transitions), Identity & Access.

2.5 API Interfaces (REST)

(Addresses rubric 2.5: RESTful endpoints w/ role-based access.)

Project Specification Template ...

Identity & Access

- `POST /api/users/register`
- `POST /api/users/login`
- `GET /api/users/:id`
- `GET /api/projects/:project_id/memberships` (manager/admin only)

Projects

- `POST /api/projects` (any user)
- `GET /api/projects/:project_id`
- `DELETE /api/projects/:project_id`
- `PUT /api/projects/:project_id`

Issues & Workflow

- `POST /api/projects/:project_id/issues`
- `GET /api/projects/:project_id/issues/:issue_id`
- `GET /api/projects/:project_id/issues/?issue_type=bug&task&subtask`
(allow queries for issues)

Calendar

- `GET /api/projects/:project_id/calendar`

3.0 Technical Stack

- **Language/Framework:** Ruby 3.x, Rails 7.x (ActiveRecord).
- **DB:** PostgreSQL
- **Auth:** Devise. **AuthZ:** Pundit or CanCanCan.
- **Testing:** RSpec, Mockaroo, Cucumber, Postman

4.0 Deliverables

The final submission must include:

1. **Complete Source Code:** A well-structured and commented codebase, including all necessary configuration files.
2. **Project Documentation:** A README.md file that explains the system's architecture, setup instructions, and how to run the application.
3. **API Documentation:** A separate document or file detailing all API endpoints, their expected parameters, and response formats.
4. **Presentation:** A brief presentation and demonstration of the application's core features.