

# SSAD Assignment 2

---

**Deadline: 02 Sept. 2015, 11:55**

Technologies : Ruby on Rails, AngularJS

## Overview

---

Issues are a great way to keep track of tasks, enhancements, and bugs for your projects. They're kind of like emails—except they can be shared and discussed with the rest of your team. Most industry and otherwise software projects have a bug tracker of some sort.

As a part of this assignment, you will have to create a **Web-Based Issue Tracker** using Ruby on Rails & AngularJS. This is a **team assignment (at most 2 members)**.

### Key points:

- All the back-end work should be completely in Ruby on Rails, which will involve exposing different APIs.
- On the front-end, use of AngularJS is required, which will consume APIs exposed above.

## Requirements

---

### Projects

- Any new user can create a new project as an owner.
- Only the owner of a project, can send e-mail invites to other users, to become project members.
- Each project can have **any number of members**.
- Each project can have **any number of issues**.
- A project's **issues can be public/private** depending on the settings.
- Owner of project can **revoke access** for any team member, at anytime.
- Project settings can be changed by **only owner** of the project.

### Users

- A user can have **any number of projects**.
- New user can **register through e-mail or oath service**.
- Every user **can create any number of issues**, only in projects he/she has access to.
- A user can **comment on issues**.

### Issues

- A **title and description** describe what the issue is about.
- Issues can have **multiple tags** to which they can be linked to. (An issue can exist without a tag.)
- A user, depending on the access (*described above*), can **vote** for issues. **One vote for one issue.**
- Users can **assign** an issue to themselves, or to any other team member(read **assignee**).
- Users can **mention** other project members in comments.
- An issue can **only be deleted** by the owner of the project.
- There should be a separate view where users can view issues they have been assigned or mentioned in.
- All the history changes viz., tag changes, assignee changes, etc should be visible.

## API (suggested pattern)

### Users

HTTP Verb	CRUD	Entire Collection (e.g. /users)	Specific Item (e.g. /users/{id})
POST	Create	201 (Created), 'Location' header with link to /users/{id} containing new ID.	404 (Not Found)
GET	Read	200 (OK), list of users. Sorting and filtering optional parameters.	200 (OK), single user. 404 (Not Found), if ID not found or invalid.
PUT	Update	404 (Not Found)	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
DELETE	Delete	404 (Not Found)	200 (OK). 404 (Not Found), if ID not found or invalid.

### Projects

HTTP Verb	CRUD	Entire Collection (e.g. /projects)	Specific Item (e.g. /projects/{id})
POST	Create	201 (Created), 'Location' header with link to /projects/{id} containing new ID.	404 (Not Found)
GET	Read	200 (OK), list of projects. Sorting and filtering optional parameters.	200 (OK), single project details. 404 (Not Found), if ID not found or invalid.
PUT	Update	404 (Not Found)	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.

DELETE	Delete	404 (Not Found)	200 (OK). 404 (Not Found), if ID not found or invalid.
--------	--------	-----------------	--

## Issues

HTTP Verb	CRUD	Entire Collection (e.g. /issues)	Specific Item (e.g. /issues/{id})
POST	Create	201 (Created), 'Location' header with link to /issues/{id} containing new ID.	404 (Not Found)
GET	Read	200 (OK), list of issues. Sorting and filtering (name, tag, etc) optional parameters.	200 (OK), single issue details. 404 (Not Found), if ID not found or invalid.
PUT	Update	404 (Not Found)	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid. (opened/closed issue change through this only)
DELETE	Delete	404 (Not Found)	200 (OK). 404 (Not Found), if ID not found or invalid.

## Comments

HTTP Verb	CRUD	Entire Collection (e.g. /issues/{id}/comments)	Specific Item (e.g. /issues/{id}/comments/{cid})
POST	Create	201 (Created), 'Location' header with link to /issues/{id}/comments/{id} containing new ID.	404 (Not Found)
GET	Read	200 (OK), list of comments.	200 (OK), single comment details. 404 (Not Found), if ID not found or invalid. <i>(optional)</i>
DELETE	Delete	404 (Not Found)	200 (OK). 404 (Not Found), if ID not found or invalid.

## Other

HTTP Verb	Path	Response

POST	/issues/{id}/vote	200 (OK). Toggle between adding vote and removing vote
POST	/projects/{id}/members	201 (Created). Add a member in a project
PUT/POST	/projects/{id}/settings	200 (OK). Settings tweak of the project (only owner of {id} can access

These are the *suggested* APIs, you can add as many more APIs you wish to achieve overall functionalities.

## Database

You are allowed to use any DBMS.

## Evaluation Criteria

- Users CRUD - 5 Marks
- Projects CRUD - 10 Marks
  - Create/Edit project
  - Add members in project (atleast direct addition reqd., invitation email optional)
  - Remove project member (by owner)
  - Settings (Public/Private, etc)
- Issues CRUD - 10 Marks
  - Create/Edit issues
  - Close an issue
  - Add/Remove Tags
- APIs quality (Token-based implementation) - 15 Marks
  - Token-based implementation - 8 Marks
  - Restrictions - 7 Marks
    - eg, issues of projects which are private, should not be visible without project member login.
- User interface - 10 Marks
- Misc. functionalities - 50 Marks
  - Issue searching with name/description - 5 Marks
  - Multi-tag searching/filtering (tagA,tagA and tagB, etc) - 10 Marks
  - Filter all tags assigned to personX, or issues where personX is mentioned - 5 Marks
  - Mark an issue as a duplicate. - 5 Marks
  - Refer an issue.(reference should be visible in both the issues) - 5 Marks
  - Comments - 5 Marks
  - User mentions in comments. - 5 Marks
  - Vote - 5 Marks
  - History changes - 5 Marks

**TOTAL : 100 Marks**

# Bonus

Bonus of 50 marks is available.

Division:

- Notification system - 20 Marks
  - Email notifications. - 4 Marks
  - Site wide notifications list(eg Quora/Facebook) - 16 Marks (4 Marks each)
    - "You have been mentioned in issueX."
    - "You have been assigned for issueX."
    - "A new issue has been created in projectX."
    - "A new comment on issueX."
- Public/Private projects - 10 Marks
- Milestone assignment (refer GitHub) - 10 Marks
- Issue create and comments editor - 10 Marks
  - File upload - 7 Marks
  - WYSIWYG / Markdown support - 3 Marks