# Issue Tracking System – AngularJS Practical Project

You are assigned to design and implement an **issue tracker system**. The system holds **projects**, which have **issues**. Each **project** has a **leader** (the person responsible for the project), **name**, **description**, a **project key**, a set of **priorities** which the project’s issues can use and some **labels**.

Each **Issue** has a **title**, **description**, **issue key**, an **assignee** (the person responsible to resolve the issue), a **status** which must follow a specific **status transition** (more on that below), a **due date**, a **priority** (which is some of the available priorities for the project), some **labels** and user-made **comments**.

A **status transition scheme** describes the workflow of an issue. See the image below for an example scheme:



A logged in user can view the **dashboard**, which depending of his role, will consist of different elements. All users can view the assigned **Open issues** to them, ordered by **due date** in descending order. **Administrators** can also create or edit issues and projects. Clicking on **[Create Issue]** opens a modal popup screen. Clicking on **[Create Project]** opens another modal popup screen.

The **project leader** can edit the project, add new issues to it and change the status of the current ones. The **issue assignee** has privileges to edit his issue and change the status of it.

Anonymous site visitors can only **login** and **register**. Logged in users can **logout**. You are given the server-side REST services to be called by your app with AJAX requests so you do not need to develop the back-end.

## Important: Commit Every Day in GitHub

Please **use GitHub** for your project development!

* Commit several times a day.
  + You need to show many small commits that indicate your constant work on the project.
  + Avoid committing large blocks of code at once.
* You should prove that you have worked at least 3-4 days over your project!

Note that Git does not require Internet connection in order to commit changes. You can commit locally and push your changes to GitHub at once. Please commit many times to show your work progress step by step.

Up to 100 score

## Issue Tracking System REST Services

You are given the following **REST services** for your Issue Tracking System SPA application.

* Services base URL (in the Windows Azure cloud): <http://softuni-issue-tracker.azurewebsites.com/api>.
* Services help page (online documentation): <http://softuni-issue-tracker.azurewebsites.com/Help>.

The documentation below provide additional details for the Issue Tracking System REST services.

## Issue Tracking System Web Design

You аre given the **Web design** and **UI prototype** of the Issue Tracking System SPA application as PDF document.

You need to convert the design to HTML + CSS. Pixel-perfect layout is not required. You do not need to match exactly the sizes, fonts and colors of the elements. A responsive design is highly desirable. You can use responsive CSS frameworks like **Bootstrap**.

10 score

## Issue Tracking System SPA Application

Design and implement a client-side SPA application based on **AngularJS**.

### AngularJS Project Structure

* Prepare an AngularJS project structure following the industry best practices.
* You should have separate folders for controllers, directives, filters, services, views, etc.

30 score

# Project Requirements

### Public Screens

Public screens are accessible for site visitors without login.

* **Login Screen**
  + Route: **#/**
  + **Logins an existing user**. Shows notification for success or error message.
  + After login, the user is automatically redirected to the dashboard.

5 score

* **Register User Screen**
  + Route: **#/**
  + **Registers a new user**. Shows notification for success or error message.
  + After registration, the user is automatically logged in and is redirected to the dashboard.

10 score

### User Screens

User screens are accessible for authorized users only (after login).

* **User Dashboard**
  + Route: #/
  + Includes the user’s open issues, ordered by due date in descending order

5 score

* **Project page**
  + Route: #/projects/:id
  + Includes all the project info and all of it’s issues. If the user is the project’s leader he can add new issues.

5 score

* **Issue page**
  + Route: #/issues/:id
  + Opening an issue displays all information regarding the issue. The data is not available for edition unless the user is its respective owner or administrator. If the user is the assignee, they can change the status for the available status transition (e.g. Open -> Closed).

10 score

* **Edit User Profile**
  + Route: #/profile
  + Users should be able to **edit their profile data** (**name**, **email**, **gender**,upload **profile picture**,upload **cover picture**). Show notification for success or error message.

10 score

* **Change User Password**
  + Route: #/profile/password
  + Users should be able to **change their password** from form (contains **old password**, **new password** and **confirm new password**). Show notification for success or error message.

5 score

* **Logout**
  + Route: #/logout
  + Successfully logged in users should be able to **logout** from the app.
  + Logout shows a notification message and redirects to the Home screen.

5 score

* **Guest Authorization Checks**
  + Anonymous site visitors (without login) should be able to access only Login and Register screens.
  + An attempt to access anonymously these screens should redirect the user to the Home screen.

5 score

### Administrator Screens

User screens are accessible for authorized users only (after login).

* **User Dashboard**
  + Route: #/
  + Includes the user’s open issues, ordered by due date in descending order. In addition holds buttons for creating new projects and issues.

5 score

* **Create Project** 
  + Route: #/projects
  + A modal dialog with a form for creating a new project. The form consists of:
    - Leader (Drop-down with all available users)
    - Project Key (Text-box)
    - Priorities (Text-box)
    - Label (Text-box which suggests already created labels by typing a substring. If the label does not exist – creates it)
    - Name (Text-box)
    - Description (Text area)
    - Create Project (Button)

5 score

* **Create Issue** 
  + Route: #/issues
  + A modal dialog with a form for creating a new issue. The form consists of:
    - Assignee (Drop-down with all available users)
    - Issue Key (Text-box)
    - Project (Drop-down with all available projects)
    - Due (Calendar)
    - Priority (Drop-down with available priorities for this project)
    - Label (Text-box which suggests already created labels by typing a substring. If the label does not exist – creates it)
    - Title (Text-box)
    - Description (Text area)
    - Create Issue (Button)

5 score

Total 300 score

# Project Bonuses

Total 60 score