# Issue Tacker – 01.03.2017

Issue Tracker is a system that reports bugs. Your task is to build one. You have to use **TomEE**, **Hibernate, JSP** and the **custom MVC Framework** for that purpose. Your application needs to have **six** pages.

## Views

* **Home** 
  + Entry point of the application
* **Issues**
  + All the issues are reported here
  + You can search, add and edit them
* **Log in** 
  + Log in with an existing user
* **Create User**
  + Creates a user in the database
* **Add Issue**
  + Add and issue in the database
* **Edit Issue**
  + Edit an existing issue

## Data Models

Create the required **entities**. Use the appropriate **data types**.

* **User**
  + Username
  + Full Name
  + Password
  + Role
* **Role**
  + User or Admin
* **Issue**
  + Name
  + Priority
  + Status
  + Creation Date
  + Author – the user that creates the issue
* **Priority** 
  + Low, Medium, High
* **Status**
  + New, Solved

## Functionality

* **Guests** (anonymous users) can register an account with their own username, email and password.
* **Guests** can login by username and password.
* **Logged-in users** can logout.
* **Guests** can view the home page.
* **Logged-in users** can create new issues.
* **Logged-in users** can edit or delete their own issues.
* **Administrators** can add, edit or delete all issues
* When guest user tries to access a page that is allowed only for logged in user he should be redirected to the login page

## Design the Database

Design **entity classes** and create a **database** to hold the **users** and the **issues**. Use **Hibernate** implementation of **JPA**.

## Use DTO

Use appropriate **binding** and **view** models.

## Use Repositories

Create repositories for your **entities**.

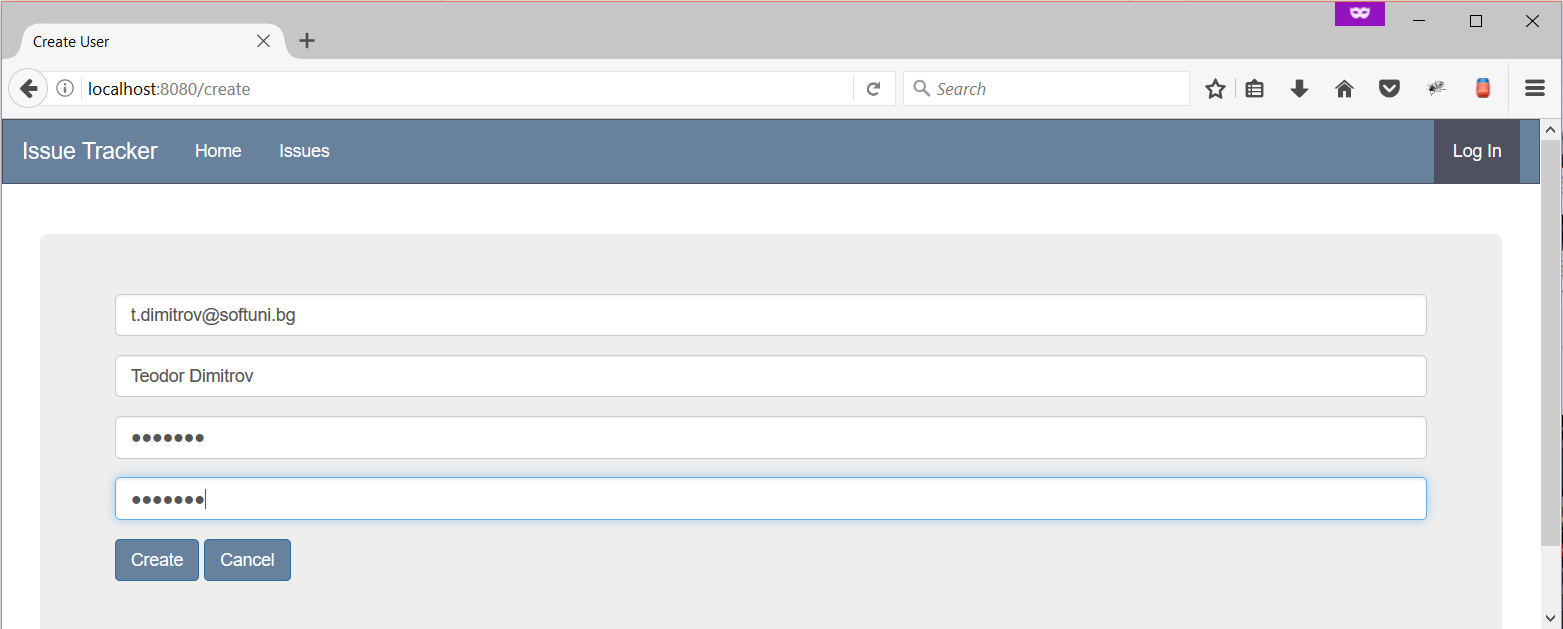
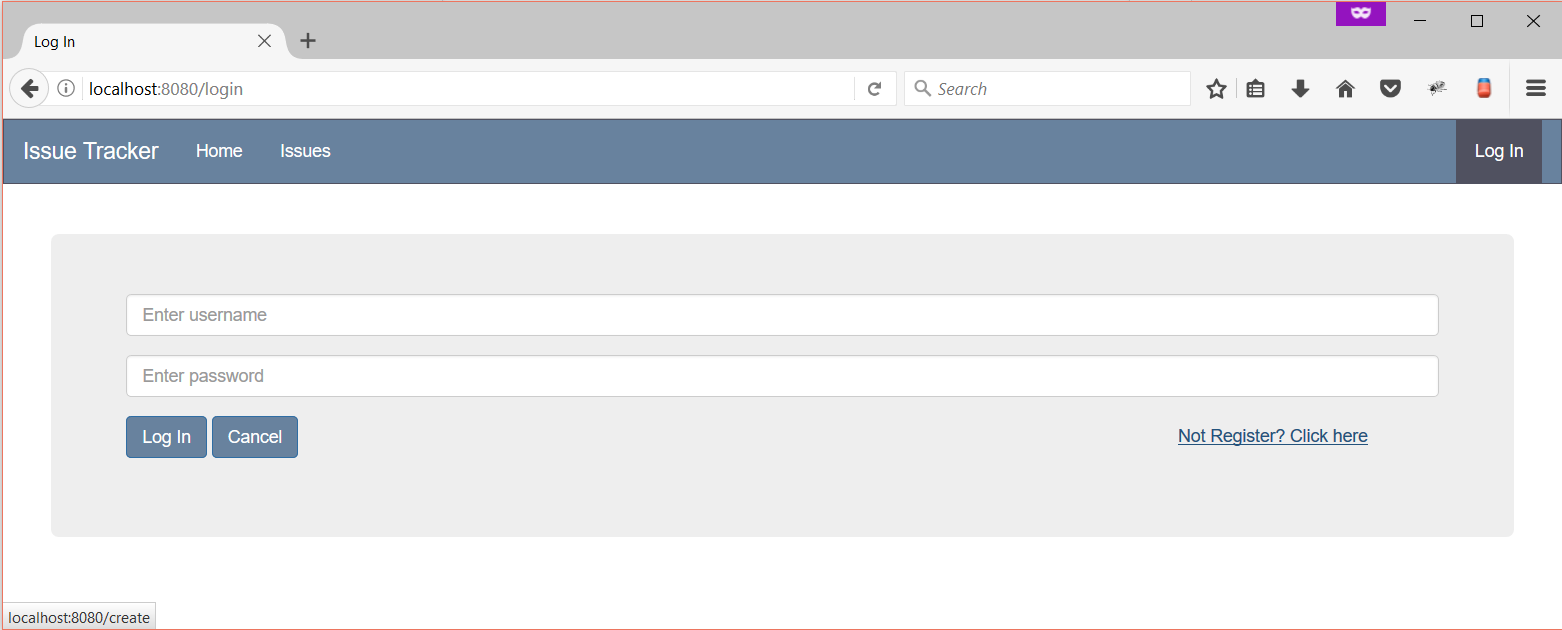
## Use Services

Create services to hold your **business** **logic**.

## Implement User Registration, Login and Logout

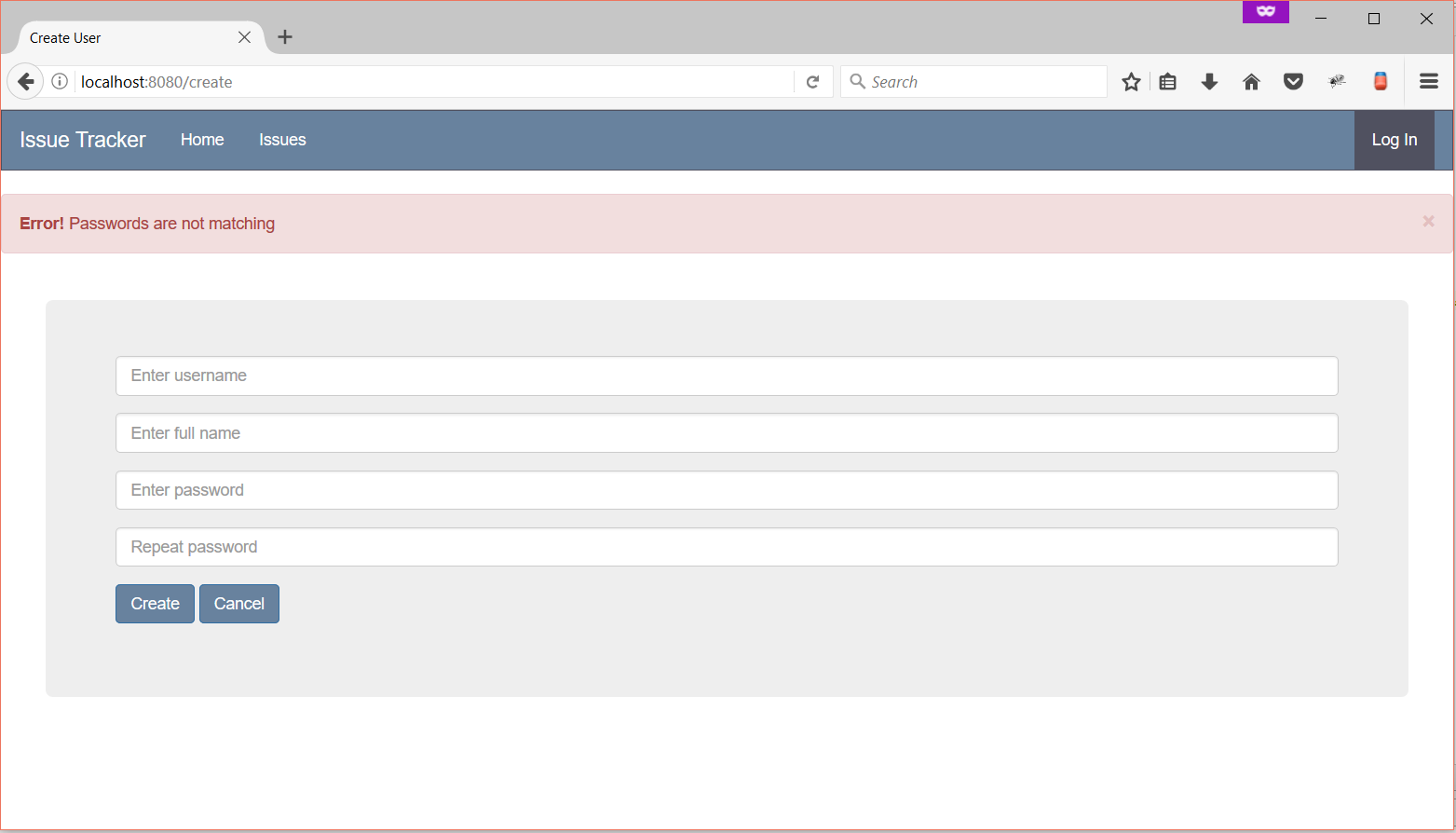
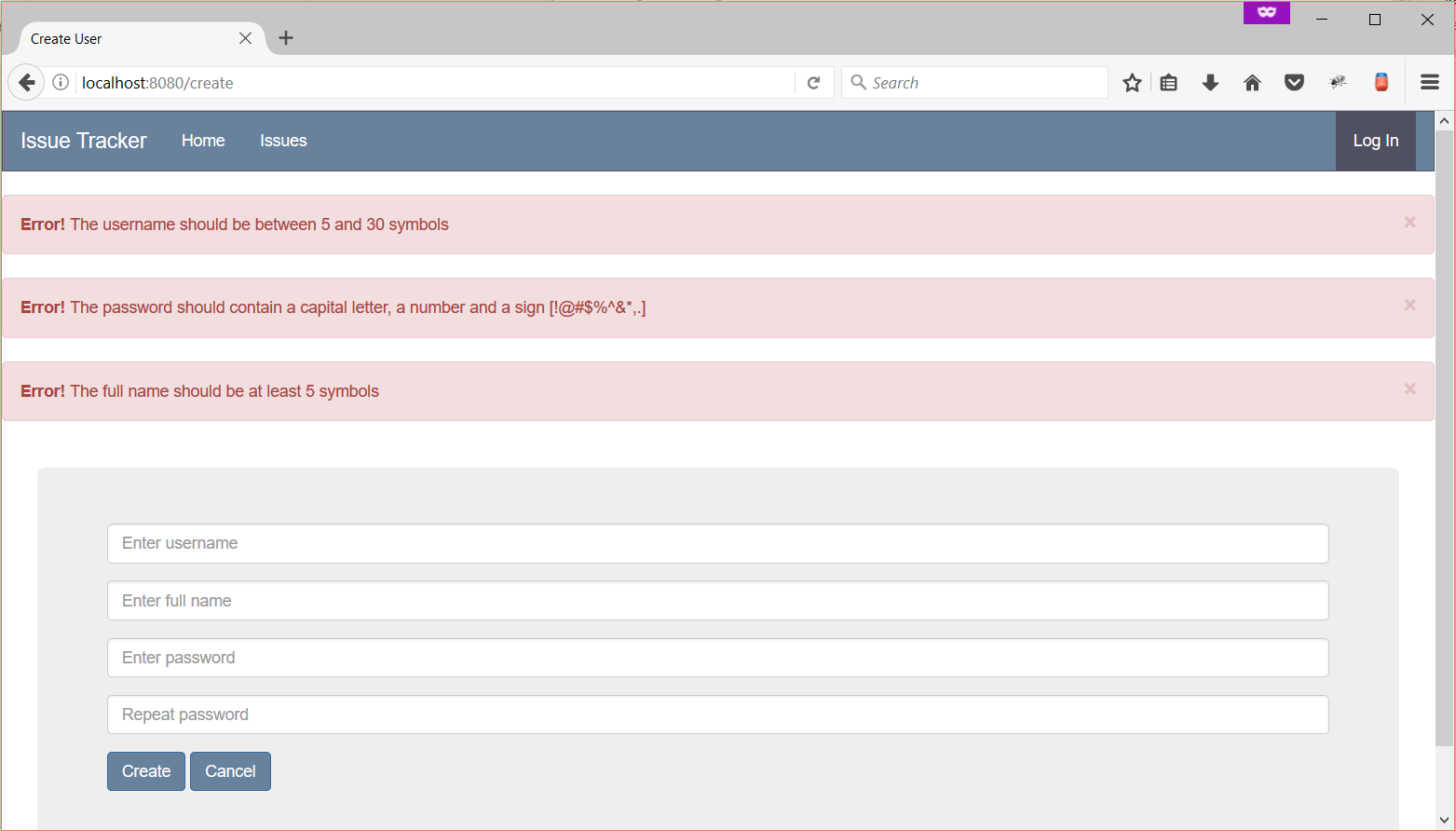
The guest users can register and log in the forum. He can provide to:

* **Register**
  + **Username** – must be between **5** and **30** symbols
  + **Full Name** - must be at least **5** symbols
  + **Password** – at least **8** symbols. It should contain a **capital letter**, a **number** and one of the following **signs: [!@#$%^&\*,.]**
  + **Confirm Password** – must **match** the provided password
* **Login**
  + User can log in with **username and password**
* **Logout**
  + When logged in the user should have option to **log out**



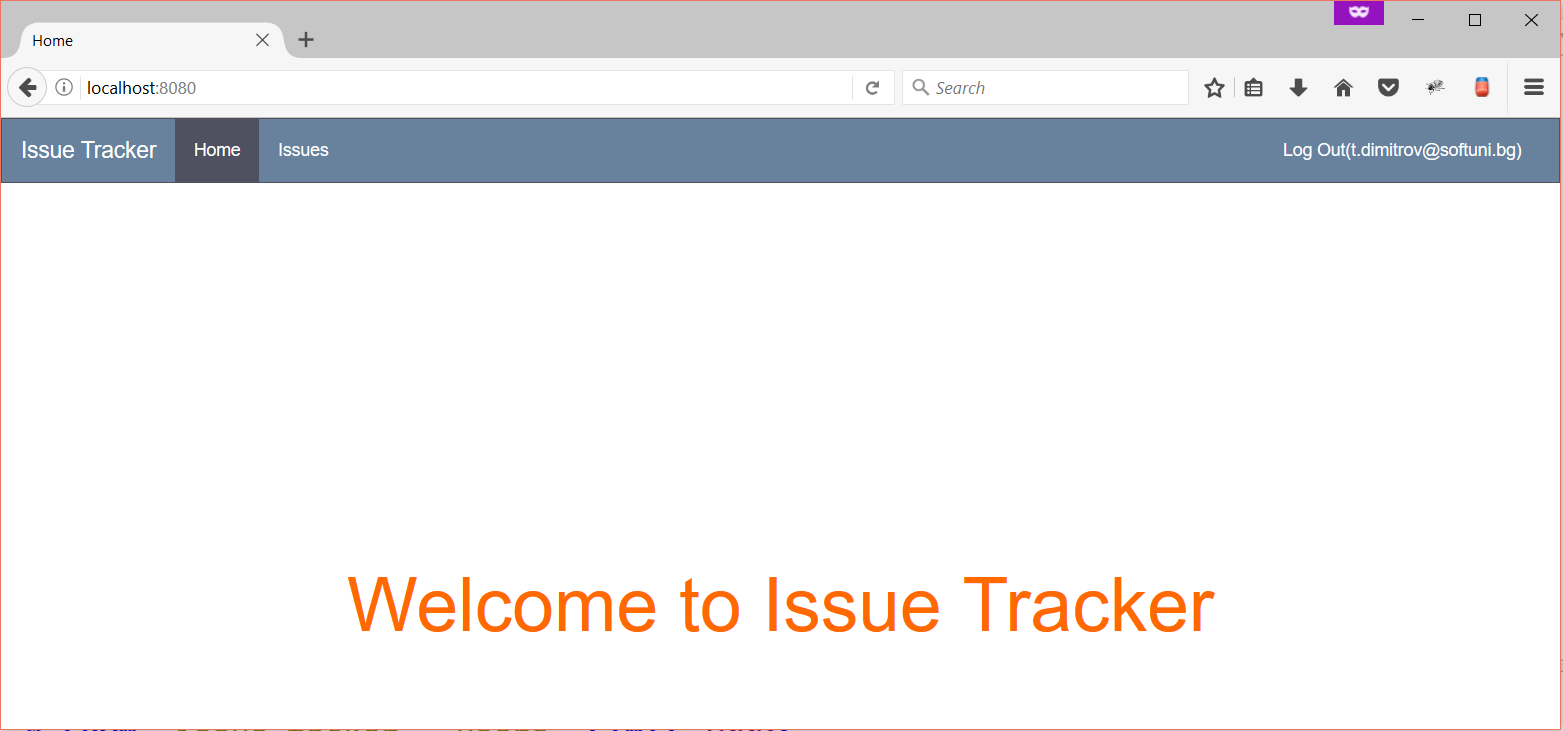
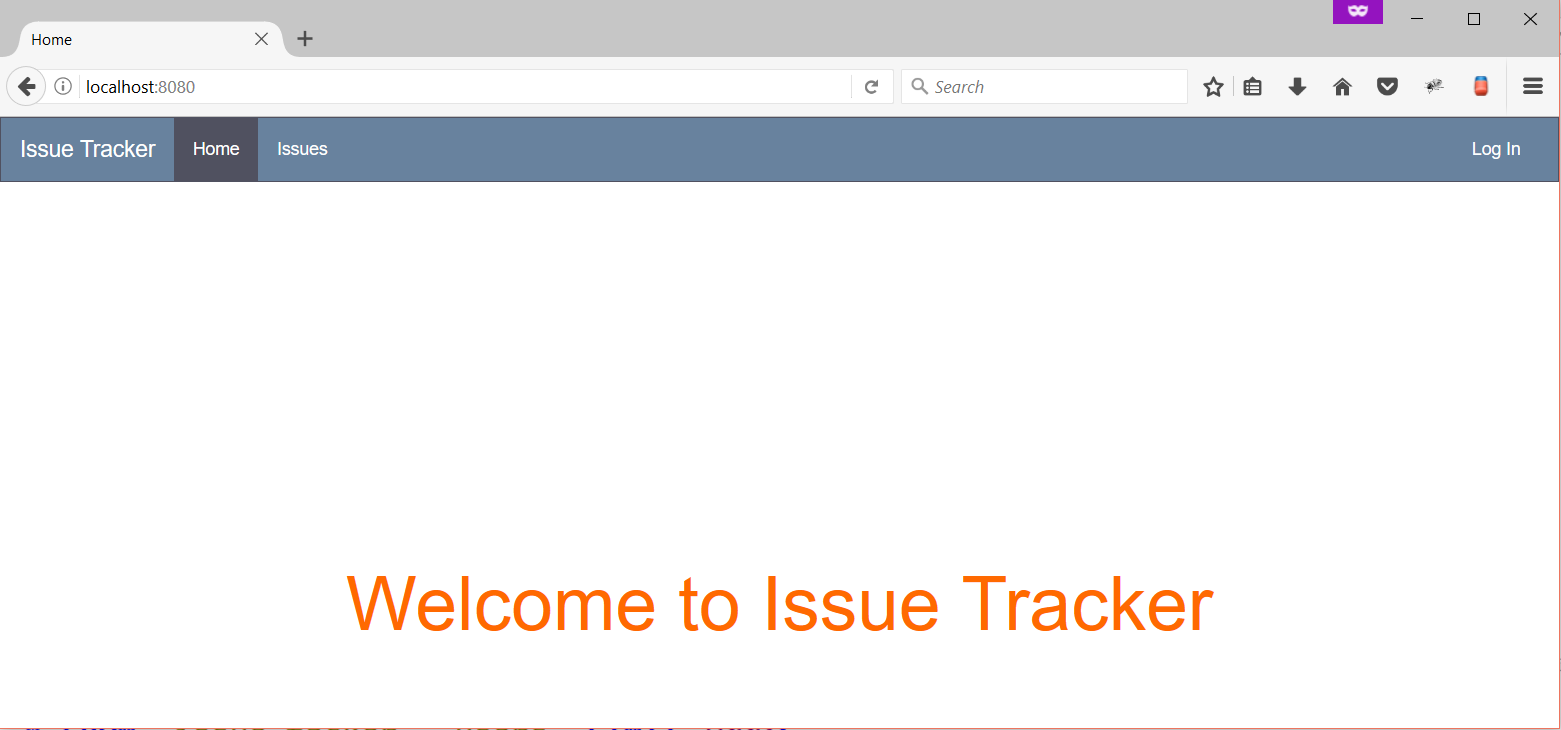
## Implement Validations

If any of the constraints fails **alert** a message.



## Implement Home Page

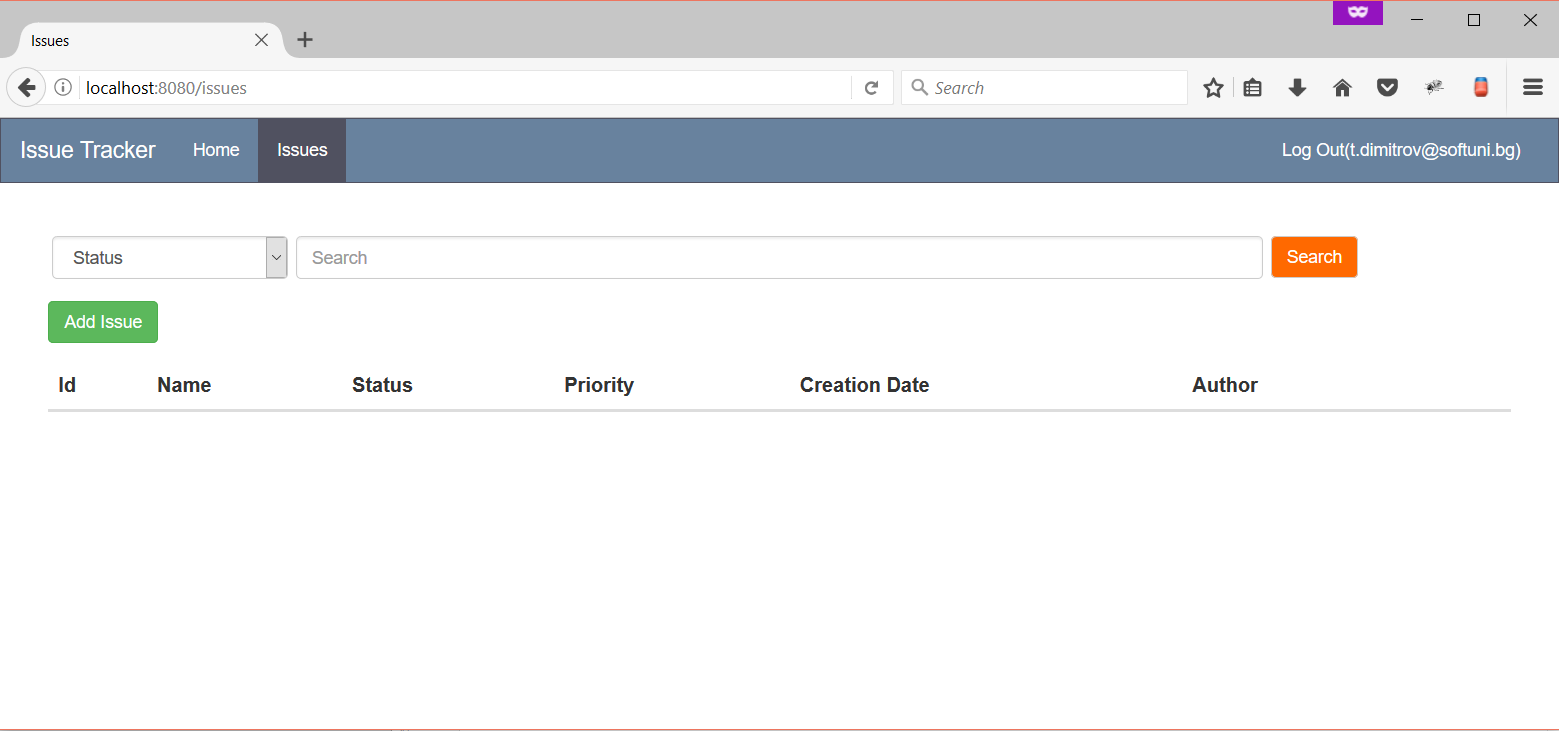
The home page is available to **guests** and should have a small **menu**. If the user is logged in a **logout** option should appear. **Edit menu.jsp** to achieve this.



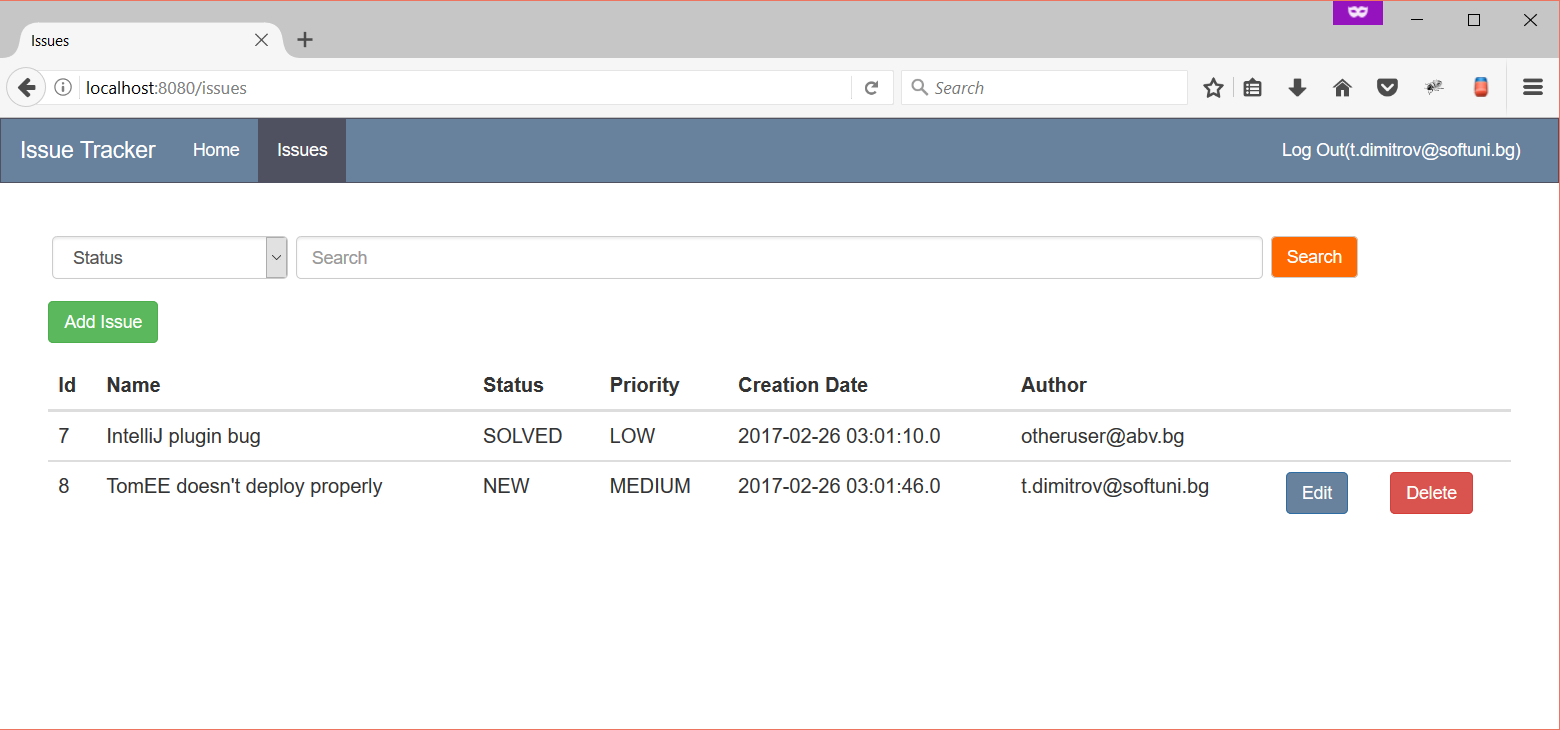
## Implement Issues Page

You should access this page **only if you are logged in**. Otherwise, a redirect to **/login** should a happen. Here all **the issues** are listed. If you are the **owner of the issue** you can edit and delete your own issues. If you are **admin** you can edit and delete any issue. Logged in users can create issues. **Edit issues.jsp**. Implement the **search functionality**.

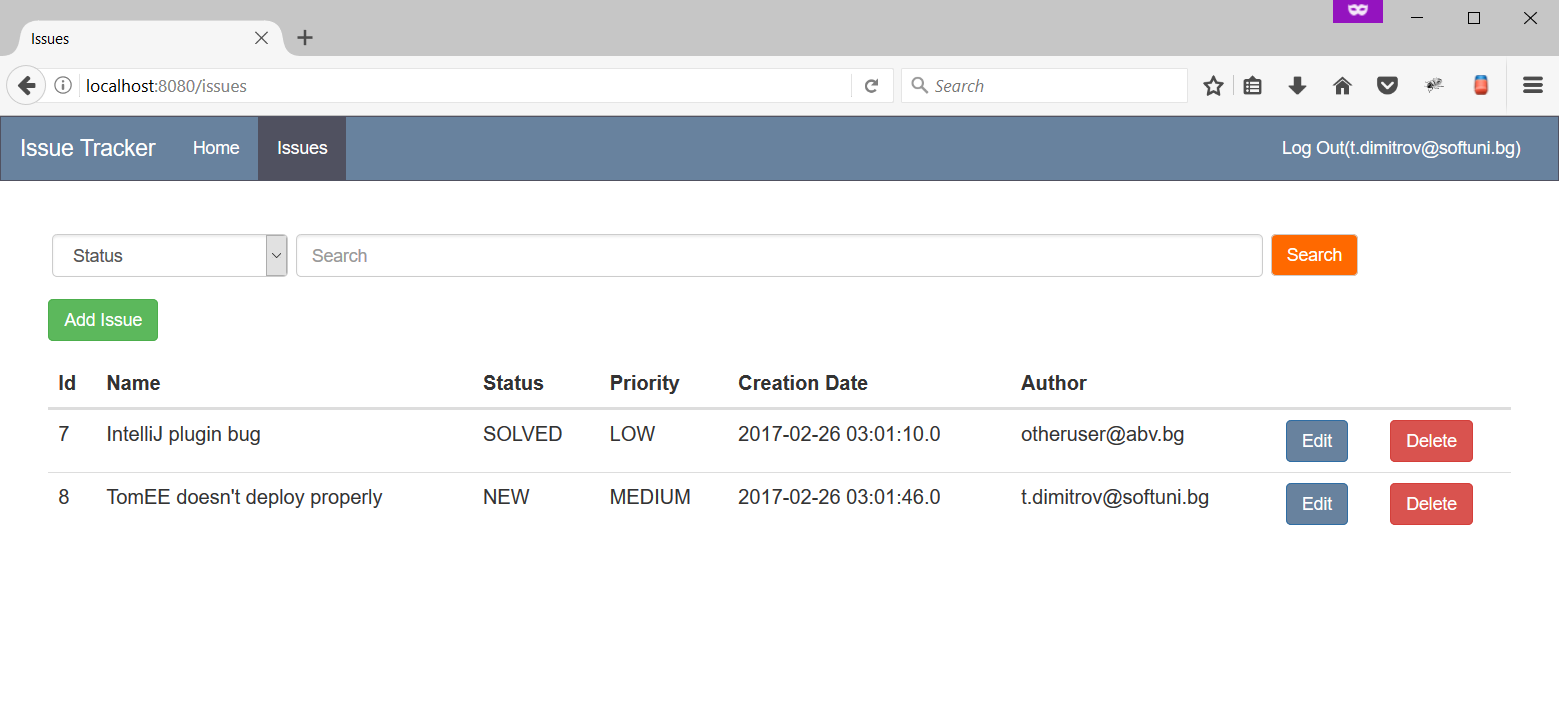
**Default view**



**Logged In View**

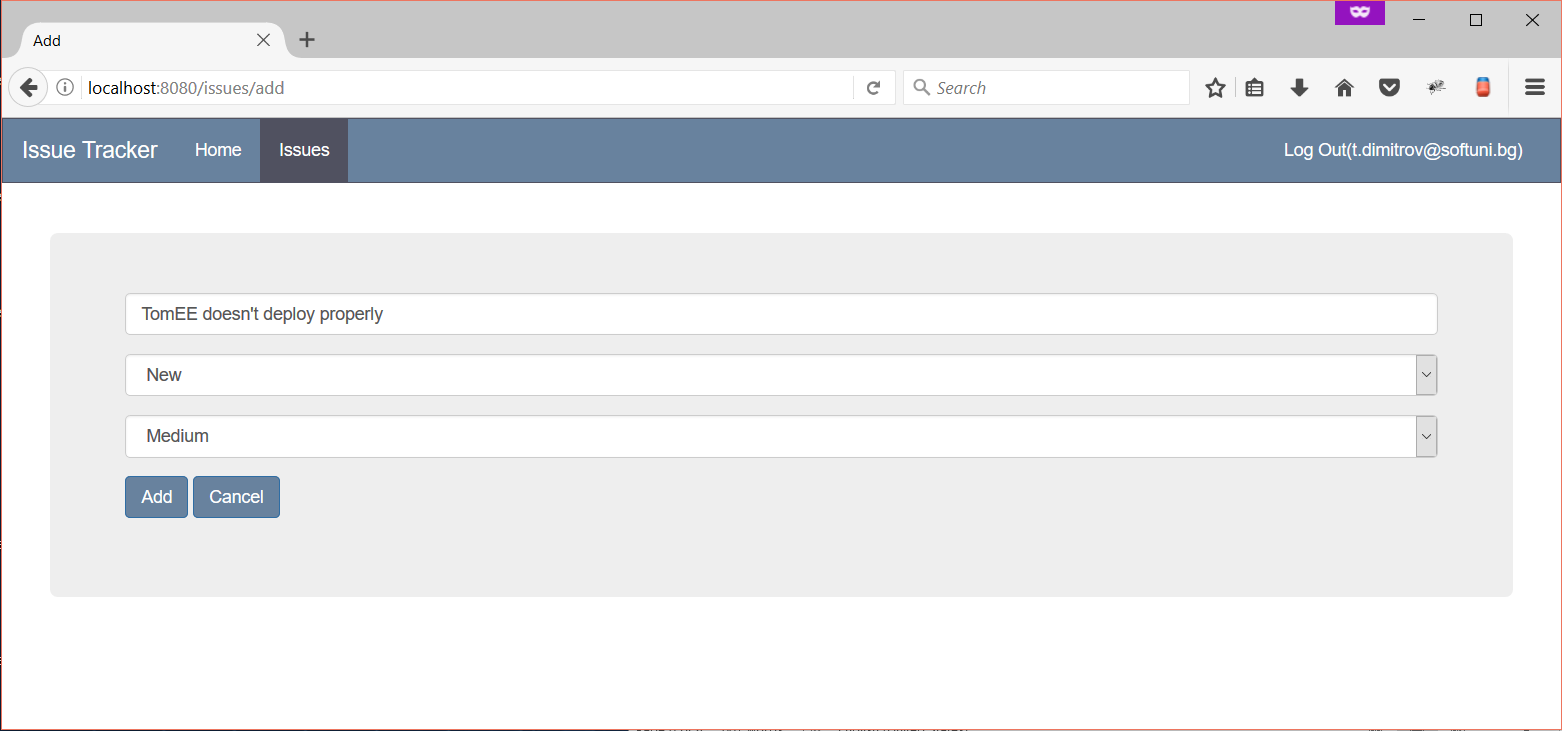


**Admin View**



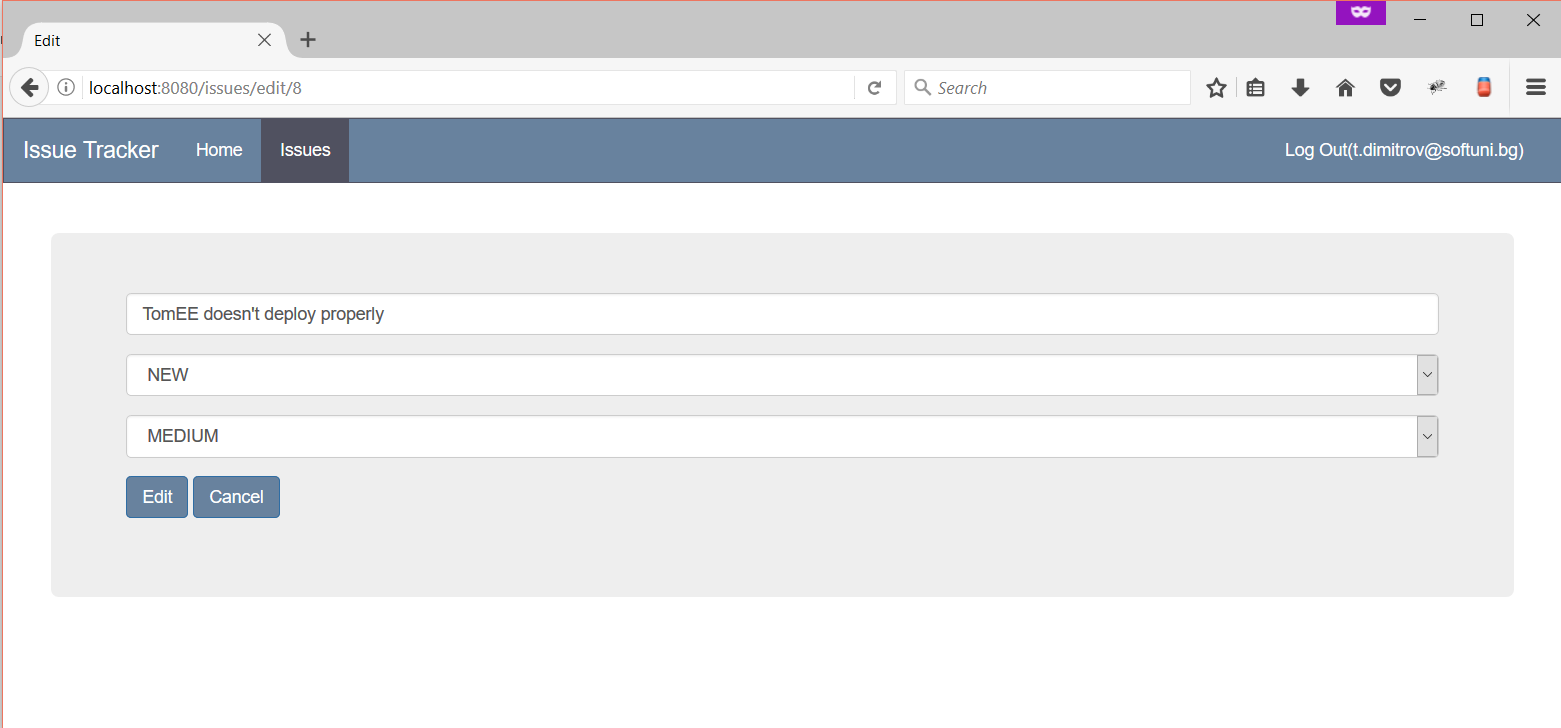
## Implement Add Issue Page

On that page **a new issue** should be added. **Edit add-issue.jsp**



## Implement Edit Issue Page

On that page **any current issue** should be edited. **Edit edit-issue.jsp**



## Implement Delete Issue Functionality

You should be able to **delete** an issue if that is **permitted**.

## Project Infrastructure Bonus

Bonus points code quality / good application structure / additional effort.

* Bonus points for using **Web Filter**.
* Bonus points for using **ModelMapper**.
* Bonus points for **quality cutom extension** of the application.