Source code:

I have shared the source code on my GitHub repository in the following address:
https://github.com/AfshinParhizkari/issuetracker

Build and run the project:

1. In the main directory of project, you can build the project and run the product by following commands:
   a. sudo mvn package -Dmaven.test.skip
   b. java -jar ./target/issuetracker.jar
2. Alternatively, in Intellij IDEA you can build the project via Maven Lifecycle (click on package) and run.
3. On the other hand, run via docker private registry:
   a. First of all, deactivate https(insecure-registries) via add daemon.json file in docker path:
      cp ./src/main/resources/daemon.json /etc/docker insecure-registries
   b. If you have private Docker registry on your machine just run the following command to push image:
      sudo mvn clean compile package -Dmaven.test.skip
   c. pull the image from private registry:
      docker pull localhost:5000/issuetracker:latest
   d. Run the container:
      docker container run -itd --hostname issuetracker --name issue-tracker -p 8080:8080 localhost:5000/issuetracker:latest
   e. issuetracker container:
      docker exec -it issue-tracker bash

The main URL of product:
http://localhost:8080/issuetracker/

The web services are in the following link:
http://localhost:8080/issuetracker/api-ui

Connect to h2 database via:
http://localhost:8080/issuetracker/h2
datasource.url= jdbc:h2:mem:issuedb
User=sa
Password=No password(empty)

Technical review:

OS: Ubuntu 20: I have chosen Ubuntu because of stability and performance. Based on my background in operation environment, I prefer to work on Linux family and command line.

Database: H2: As you describe in Backend-Test-project document, h2 db is a fast and easy db to test new ideas and create product quickly.

IDE: Intellij IDEA: I like Eclipse and Intellij and work with both of them.

Build tool and library repository: Maven3.6.3: there are some build tools but I prefer Maven based on my experience.

Source Controller: GitHub: The free, stable source controller that I used to share my source codes and documents.

Framework: J2EE 11 and spring boot 2.5.6: the easiest way to control versioning and many useful ability that developer can enjoy it.

Test: Junit4: to test data and service layer I have chosen Junit4.

Log: Logback because of slf4j.

ORM: Spring data JPA repository

Servlet Container: TomEE

Rest api Documentation and test: spring doc-openapi3 (swagger)

UI technology: JSP, JSTL

UI Programming: javascript, JQuery

UI Design: Bootstap5

Methodology: Scrum: I have tried to complete project on three to half a day based on quick delivery:

Day1: deliver Developer Rest web service, h2 db with test data, Analysis.

Day2: deliver Plan algorithm and other web services, Documentation.

Day3: a sample UI for Stories, developer tests, Dockerize.

More time:

If I have more time, I will work on:

1. Security: I will chose spring security for Authentication and JJWT for token authentication.
2. Complete UI: I will complete Bug and Developer UI.
3. I will enrich service package and move some business codes from controller package to service package.
4. I will add Controller Advice to handle exception better.
5. I will add some reports via Jasper Report to product.
6. I will focus on data validation on UI and web services.

Data Model:

story
- issueid VARCHAR(36)
- title VARCHAR(45)
- description VARCHAR(100)
- creationdate DATETIME
- estimatedpoint INT(11)
- status VARCHAR(10)
- assignedev INT(11)

Indexes

developer
- devid INT(11)
- devname VARCHAR(15)

Indexes

bug
- issueid VARCHAR(36)
- title VARCHAR(45)
- description VARCHAR(100)
- creationdate DATETIME
- priority VARCHAR(10)
- status VARCHAR(10)
- assignedev INT(11)

Indexes