# QA-Fundamentals Retake Exam (March 2016) – Issue Tracker

Long time ago, when the quality assurance was not so popular profession, you were hired as a tester in a growing startup. Everything you do not like in the projects, you do report to your manager by shouting from one corner of the room to the other. As the time passes, the team realized it was not so good practice, however they were not providing a solution for ages. Your testing expertise grows, you do get familiar with JIRA and once you have suggested it to your manager, but it has been rejected due to the cost.

Once there was a free week, where no projects were in progress, so you’ve dared the manager and the team to develop an issue tracker for 3 days. By some remarkable circumstances, they’ve done it for 4 days! It was there! A fully functional issue tracker or so they’ve thought. And because it was only a week that is free and they have done it by the end of Thursday’s day, you were assigned at Friday’s noon to test it. Because you have limited time, they allowed you to use another issue tracker to report some issues. Tracker’s trial (if any) will not expire before the problems been fixed.

Your manager has written a specification for 30 minutes, right after you’ve dared him. You need to use your knowledge for existing issue trackers; for reading specifications; and for reaching corner-cases in order to make the application polished and to meet the standards in your team.

You are given a web application which is an online issue tracker, also a specification regarding the requirements. Your task is to find the misconceptions in the specification, the broken functionalities, user experience problems, security issues and do some automation for the repetitive work.

As you are the tester in the team and the very perspective future QA Lead, you can pick tools by your choice for test automation, test planning, test execution and bug reporting. But it’s very important to document all your work in corresponding issues in the issue tracker you have chosen (different from the one you are testing ☺) and to give access to the issues tracker to your employer. By the end of the work day (six hours), you need to report all the problems to make the tracker useful for your later work.

# Project Team

The Development lead is **Douran Koulakov**, the lead of Design and Art is **Ivan Gog** and the Project Manager is **Atlass Iankov**. Communicate well with them in the issue tracker and address issues to the relevant department.

# Software Requirements

1. Introduction
   1. Purpose

The purpose of this document is to present a detailed description of Issue Tracker application (hereinafter called Tracker). The Tracker is a web application used by the internal team. The team members can create, edit, assign and transit issues to other team members. Only tracker administrators can create new members (Developers). This document will explain some of the key features of the Issue Tracker at very high level.

* 1. Scope

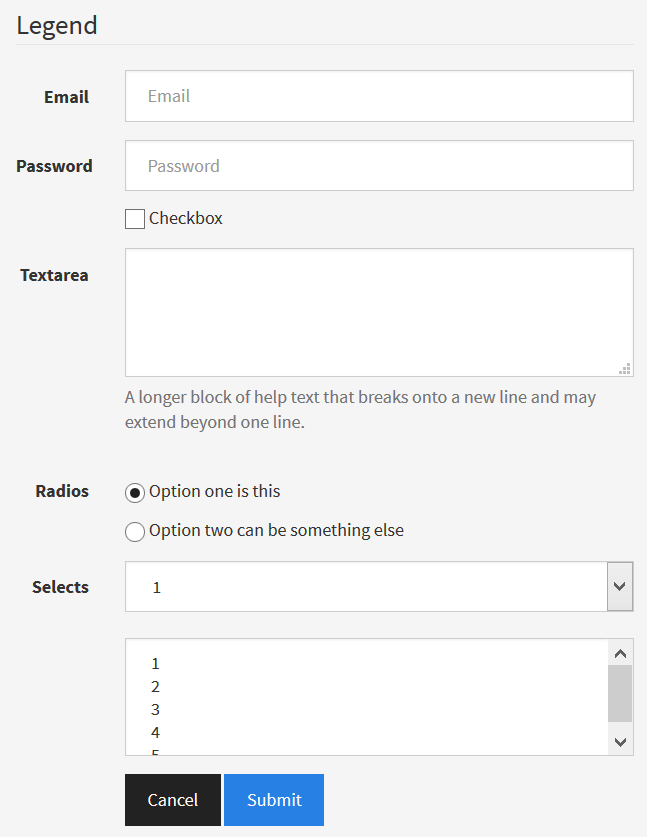
The document will cover the basic functionality on high level regarding the Issue Tracker.

1. Overall description
   1. System environment

The Tracker has three active actors and one cooperating system. All of them are accessing their parts from the internet. The Unregistered user can access only the login page of the project. The Registered user (+role USER) (hereinafter called: Developer) can access some of the basic operations (create, edit, delete) over the issues. The Administrator (+role ADMIN) is the only one who can register and manage the developers. Only users with the certain role can access the menus under that role. Combining roles gives more power to the user.

* 1. Vocabulary
     1. Form

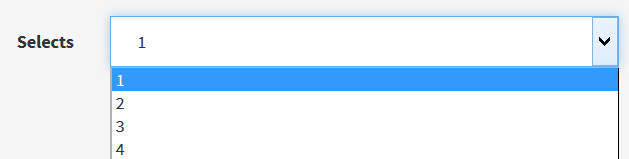
A “Form” is a set of fields and corresponding buttons with attached events. It looks like:



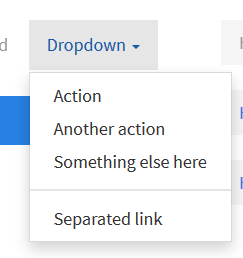
Some of the fields may or may not exist in different forms. Use the fields in the mock-up as vocabulary. Only the last two fields will be explained in different bullets.

* + 1. Select or Dropdown

A “Select” or a “Dropdown” may refer to a form element for choosing one option from a set of options. It looks like

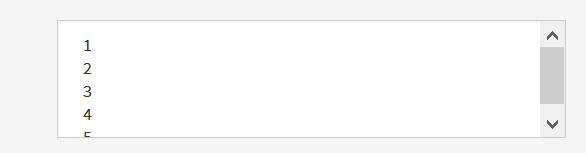


If it is part of a navigation bar or tab, it looks like:



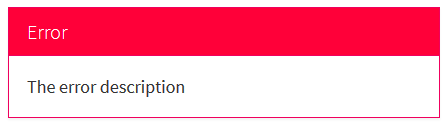
* + 1. Multiple choice or Multi-choice

A “Multiple choice” or “Multi-choice” may refer to a form element for choosing multiple options from a set of options. In order to choose more than one option, the user needs to press the multiselect key, it’s usually bound to “CTRL”. The element looks like:



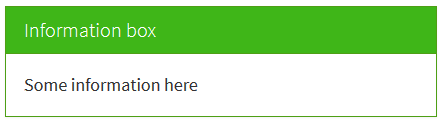
* + 1. Error-box

An “Error-box” (“Error box”, “Errorbox”) is an element in the page that informs the user that something wrong has happened. It’s usually displayed on the same page where the error happened and looks like:



* + 1. Inform-box

An “Inform-box” (“Info-box”, “Info box”, “Information box”) is an element in the page that informs the user some application notification. It’s usually displayed on the page where the notification should happen and looks like:



* 1. Functional requirement specification
     1. Use Case 1 (Accessing the system)

The system is accessed through the internet from its initial URL. It loads navigation bar, login form and footer.

* + 1. Use Case 2 (Navigation bar)

The navigation bar is different for each role:

* Unregistered users:
  + Contains the Project Logo (I-Tissues); A hyperlink “Login” leading to the login page and an informational text “Secured Area”
* Developers:
  + Contains the Project Logo; Hyperlinks [Create Issue, My Issues, Search Issue] and a Logout link
* Administrators:
  + Contains the same as Developers and additionally hyperlinks for [Filter Developer, Create Developer, All Developers]

The content under these hyperlinks is also filtered to the certain roles.

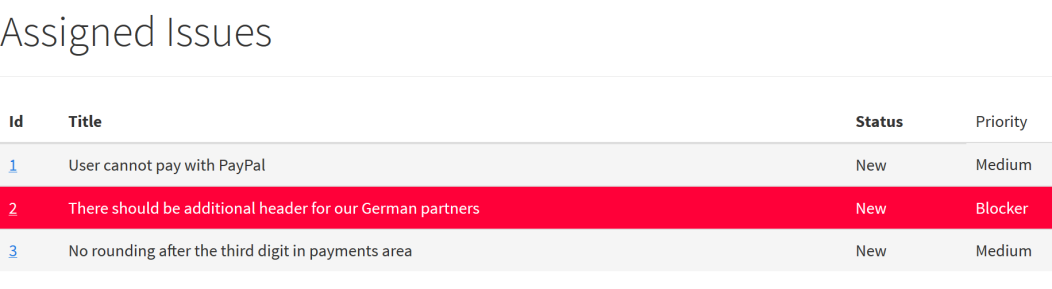
* + 1. Use Case 3 (Login page)

The login page contains login form with two input fields (username and password) and two buttons: [Cancel, Login]. When “Cancel” is clicked, all the data written in the text fields is erased. When “Login” is clicked, the process verifies the username and password pair; if they are wrong, an error box is shown; if they are verified, a redirect to the Home page happens and the user is logged in.

If there was a previous page and the session was invalidated by some reasons (expired or the user was never logged in, while trying to access that page) the successful login redirects to this page, instead of the “Home” page.

* + 1. Use Case 4 (Home page)

The home page is actually “My Issues” page. It shows all non-resolved issues that are assigned to the currently logged user in a table view. They are sorted in insertion order. Blocker issues are highlighted in red color.



By clicking the Issue ID, a new page is opened containing the issue information, which is covered in Use Case 7.

If there are no assigned issues to the current user, an info box is displayed with text “You have no assigned issues. Enjoy your day”

* + 1. Use Case 5 (Create Issue)

Create issue page can be accessed via the navigation bar. It opens a form where the developer can input a required “Title” text (up to 40 symbols inclusive); An optional “Description” textarea with 10 rows (without limitations); A dropdown “Assignee” where an assignee could be chosen among the developers registered (including the author of the issue); A dropdown “Priority” where a priority can be chosen among “Low”, “Medium”, “High” and “Blocker”; A dropdown “Type”, where an issue type can be chosen among “Bug” and “Task”.

There are two buttons [Cancel, Create]. Clicking on “Cancel” clears all fields back to their initial values. Clicking on “Create” validates the input data. If it’s not valid, the user is redirected back to the issue create page. If it’s valid, the new issue is created and the user is redirected to the issue information page, covered in Use Case 7.

* + 1. Use Case 6 (Search Issue)

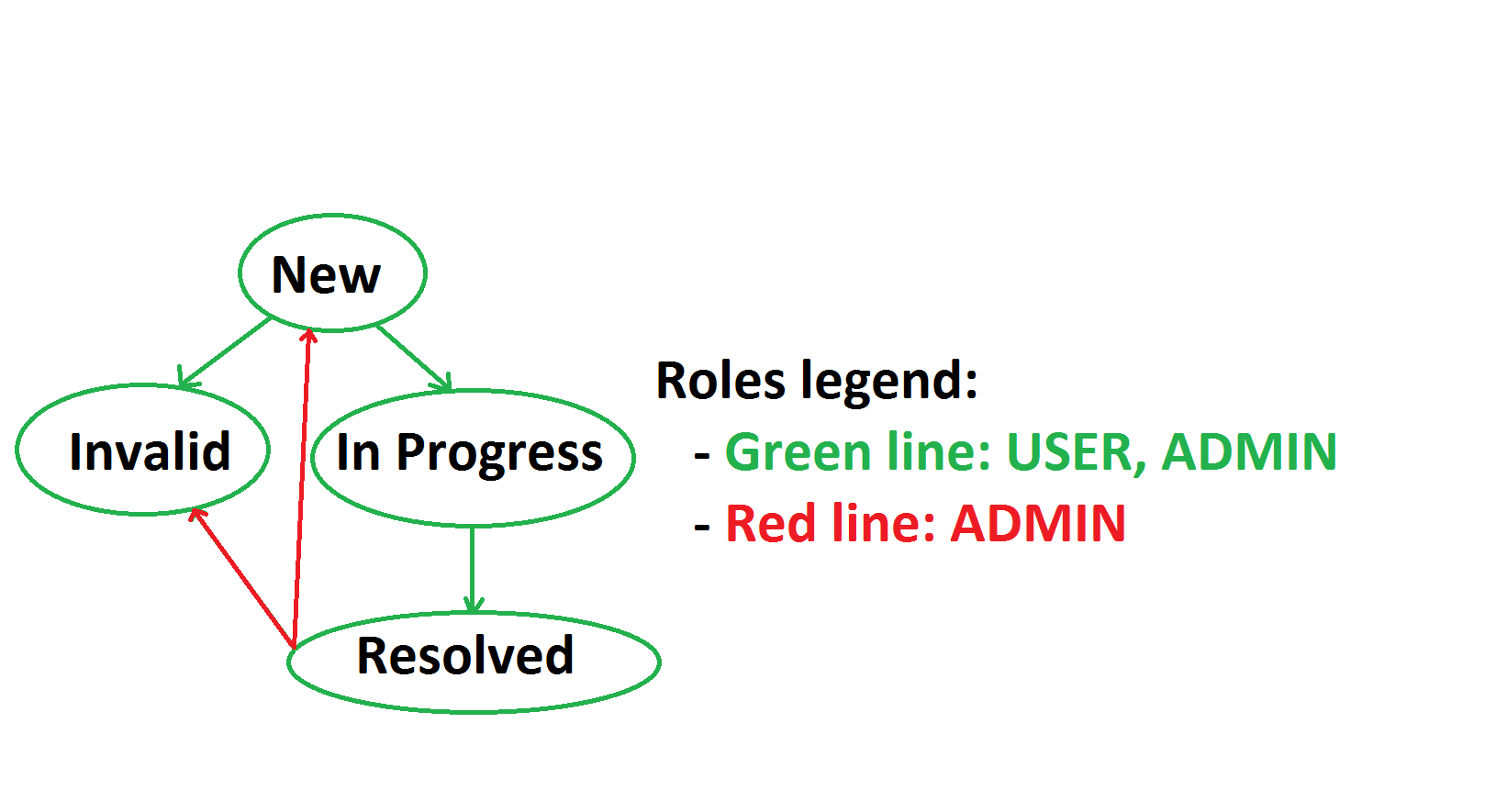
Search Issue page can be accessed via the navigation bar. It opens a form where the user can filter issues by a criterion. The “Search by” field is a dropdown which allows the user to choose the criterion among the issue priority, status, date of creation, type, creator, part of the description, id, part of the title or its assignee; The “For” field is a text field containing the value that corresponds to the desired criterion.

There are two buttons [Cancel, Search]. Clicking on “Cancel” clears all fields back to their initial values. Clicking on “Create” performs a search. If there is no issue that corresponds to the desired search criteria, an error box is shown under the form with text “No results found”. Otherwise the user is redirected to the found issue information page, covered in Use Case 7.

* + 1. Use Case 7 (Issue Information)

There is no direct link to that page. It happens via navigation from another page or a redirect. Usually corresponds to information of a single issue got by id. The id’s in our tracker are public and their exposition is not considered as a security flaw. The page displays as a centered header the issue title. Then a mini navigation bar which gives information who is the author of the issue (its username). By clicking the username, the user is redirected to the User Information page which is covered in Use Case 9. Then there is information regarding the issue type; priority and status. Additionally there is a dropdown “Workflow” which expands the available statuses from this one. If the current status is a leaf node, “No Transitions” appears instead.

The Issue Transitions Graph is:



The date of creation in ISO8601 is shown. Then a panel holding the description is shown.

After the description there is a panel that holds the assignee. It’s shown its username. Clicking on the username leads to the user information page. A dropdown after the username consisting of all users in the tracker points the user to change the assignee by clicking the black button “Reassign”.

If the currently logged user is the issue owner, or is an administrator, there are two additional blue buttons – [Edit, Delete]

Clicking on “Delete” button opens a browser-native confirm box “Are you sure you want to delete this issue”. If the box is canceled – nothing happens. If it’s confirmed the issue is deleted and the user is redirected to the home page. Clicking on “Edit” redirects the user to Edit Issue page covered in Use Case 8.

* + 1. Use Case 8 (Issue Edit)

The header of the page is “Edit Issue” and the issue title.

This page renders a form similar to the Issue Create form, with additional field for Creator, which is a dropdown, such as the Assignee. All the information in the fields is prefilled with the information taken from the issue to be edited. This is no direct page, it’s related to a certain issue.

If the editor is an administrator, an additional dropdown field is rendered containing all statuses. The administrator can jump to an arbitrary status without restriction, but should be warned that they may break the standard workflow.

There are two buttons in the bottom – [Cancel, Edit]. Clicking on “Cancel” clears the fields to their initial value. Clicking on “Edit” performs a persistent edit upon the issue and redirects the user to the Issue Information page

* + 1. Use Case 9 (User Information)

This page behaves three different ways depending on the current user role

* Current user is neither the opened user, nor administrator:
  + A centered title with the opened user’s username
  + A navigation panel with the username and it’s roles separated by a comma
* Current user is the opened user and is not an administrator:
  + A centered title with the opened user’s username
  + A navigation panel with the username and it’s roles separated by a comma
  + A form with three text fields – “Username”, “Name” and “Password”. The first two are prefilled with the information regarding the current user.
    - Username and Name are required and cannot be more than 20 symbols
    - Password is initially empty. If it’s submitted empty, it does not change.
  + Two buttons – [Cancel, Edit Profile]. Clicking on “Cancel” clears the fields to the initial values. Clicking on “Edit Profile” performs a persistent edit on the current user.
* Current user is an administrator:
  + A title with the opened user’s username
  + A form is opened with 4 fields. All fields except “Password” are prefilled with the information regarding the user to be edited.
    - Username and Name are required and cannot be more than 20 symbols
    - Password is initially empty. If it’s submitted empty, it does not change.
    - Roles is a multiple choice and can have arbitrary combination of roles.
  + Four buttons – [Cancel, Edit, Deactivate/Activate, Delete]. Clicking on “Cancel” clears the fields to the initial values. Clicking on “Edit” performs a persistent edit on the current user. If the user is active – “Deactivate” performs a persistent deactivation of the user, so they cannot login anymore, and the button becomes “Activate”. Clicking on “Activate” performs a persistent activation of the user, so they can login from now on, and the button becomes “Deactivate”. Clicking on “Delete” opens a browser-native confirmation box “Are you sure you want to delete this user?”. Canceling the dialog – nothing happens. Confirming the dialog performs a persistent deletion of the user and the user is redirected to All Developers page covered in Use Case 10
    1. Use Case 10 (All Developers)

The page is accessed via the navigation bar available for administrators.

This page contains an information grid about all developers in the issue tracker. The header of the page is “Developers”. There are navigational pagers before the grid and after the grid, which points the current page. Additionally if there is a previous page, a “Previous” hyperlink on the left side is displayed and if there is a next page, a “Next” hyperlink on the right side is displayed.

There are four columns in the grid – the user Id (Id); whether it’s active or not (Is Active); its username (Username); its name (Name)

The Id is clickable and leads to the User Information page.

Clicking on “Next” or “Previous” slides to the next chunk of developers.

There are five developers per page.

If there are no developers, there is an info box with text “There are no developers yet.”

* + 1. Use Case 11 (Create Developer)

The page is accessed via the navigation bar available for administrators.

This page renders a form with header “Create a Developer” and four empty fields. Username and Name are required and could be no more than 20 symbols. Password is required. Roles is a multiple choice field where user can select an arbitrary set of roles.

There are two buttons – [Cancel, Create]. Clicking on “Cancel” clears the fields to their initial value. Clicking on “Create” performs a persistent creation of an user and redirects the user to the User Information page for the newly created user.

* + 1. Use Case 12 (Filter Developer)

The page is accessed via the navigation bar available for administrators.

This page renders a form with header “Search for a developer” and two fields. The field “Search by” is a dropdown field holding the searching criterion among user’s role, name or username. “For” is a text field – the value corresponding to the criterion.

There are two buttons – [Cancel, Search]. Clicking on “Cancel” clears the fields to their initial value. Clicking on “Search” performs a search by the given criterion and value pair. If there is no developer that matches the criteria and error box is shown with text “No Results Found”. If there is a developer – a redirect is performed to the respective result’s User Information page.

* + 1. Use Case 13 (Logout)

Clicking on “Logout” invalidates the current session and redirects to the login page

# Tasks

1. You know your developers well for all that years. They usually make certain count of mistakes for certain development time. For these 4 days of rapid development you know you need to find **at least 5 user-acceptance bugs**, **8 functional bugs, 4 design bugs and 5 security bugs all ordered by severity.** Provide reproducing steps for each bug report.
2. Write automation scripts in any programming language by your choice, (if you are familiar with Selenese’s conditions and assertions you can use Selenium IDE as well) that tests the following functionalities
   1. Issue’s workflow

Constraints:

* Use fresh data (create one when necessary) so tests will always be valid and will not depend on data which could disappear
* Assert as many times as you want in a test method but assert only things related to the scenario e.g. if you expect the test to provide error messages, assert the error message presence.
* Avoid code repetition. Extract in methods the repetitive code. If you have shared configuration of all your methods, extract it in a initialization method (constructor, test initializer, or whatever your test runner supports)
* Each tested component should be one class. Each tested condition should be one method.
* Naming convention for test method. Each test method should follow the pattern **testedFunctionality\_testedConditions\_expectedResult()** e.g. (**testLogin\_testValidUserNamePassword\_expectRedirectToProfile()**). Also use appropriate method naming for the language you have chosen (e.g. Java says methods should **be camelCase())**
* Test should pass if everything works by specification. If there are conditions from the specification that are not met, the test should fail expecting the conditions to be those in the specification, rather than actual ones.

# Resources

The project is located at: <http://192.168.111.66:8080/>

When you access the site, click the “Create Tracker” button. After ~1 minute you will receive a blank page with new URL. Use that URL from now on. Click refresh and you will be redirected to the login page. **You do not need and must not test the tracker creation. Please avoid creating multiple libraries as it may slow down the whole examination process!**

The first administrator’s credentials are “admin” and “admin”. You cannot edit/delete this user (**do not test it and do not report bugs in this functionality if there are any. This user is only for your comfort to test the exam exercise better**).

# Reporting

Once you are ready, archive all your local work (without compiled binaries) and upload it as a Homework for the respective exam row in the course instance. Add a file to the archive that is named “IssueTracker.txt” which contains an URL to the respective issue tracker and credentials if needed in order to visit it.

IT IS PROHIBITED TO DO ANY ACTIVITIES TO THE ISSUE TRACKER AFTER THE EXAM IS FINISHED. VIOLATING THE CONSTRAINT WILL RESULT IN ZERO POINTS RECEIVED.

## Exam Information

You are allowed to use any resources you have like Internet, software, existing code.

You are not allowed to get help from other people: Skype, ICQ, FB, email, talks, phone calls, etc. are forbidden.

Exam time: **6 hours**.