# Workshop: Manual Testing

This document defines the exercises and homework assignments for the [**"QA Basics" Course @SoftUni**](https://softuni.bg/opencourses/qa-basics).

Complete the provided **MS Word document template**. Put the **solution** for each exercise in the document. Name your document Workshop-FirstName-LastName.docx. Submit this document as your homework.

## Get Familiar with the “Eventures” Web App

You are given a **Web app**. To **run the app,** go to:

* [**http://softuni-qa-loadbalancer-2137572849.eu-north-1.elb.amazonaws.com:81**](http://softuni-qa-loadbalancer-2137572849.eu-north-1.elb.amazonaws.com:81/)

Go ahead and **look around the app**. Look at the different menus, links, forms, and fields you can interact with. Then, you can **look below** to understand **how the app is expected to work**.

Graphical user interface, text, application, email

Description automatically generated

The **Web app** holds a list of **events**, created by different **users** (after register / login).

### Users

* Users can **register** on the site with their own provided username, email, password, first name and last name.
* Users can **login** by username + password.
* After either **register / login** users can **view** existing events, **edit** / **delete** owned events, and **create** new events.

### Events

Each **event** has a name, place, start date, end date, total tickets, price per ticket and owner.

The **event owner** is the user who created a certain event.

* Anyone can view all events but editing and deleting events is limited to their respective owners.
* By requirement the **event owner can edit and delete his own events** but **cannot edit and delete other user’s events**.

## Graphical user interface, application, Teams Description automatically generatedCreate a Jira Project

Create a new Jira project to log the issues related to the “**Eventures**” project. You know how to do this!

## Test Scenarios and Test Cases

Design **test scenarios** and **test cases** for each of the functionalities of this Web application.

Functionalities are as follows: **Register**, **Log-in**, **Home page**, **Create new event**, **All events page, Edit event, Delete event**.

This is a sample of how you can describe your test scenarios.

|  |  |
| --- | --- |
| **Test scenarios for Register form** | **Test cases** |
| Test username field | 1. Test with empty username 2. Test with valid username 3. Test with invalid username 4. Test with numbers 5. … |
| Test email field | 1. Test with valid e-mail 2. Test with invalid e-mail 3. Test with empty e-mail 4. … |
| Test password field | 1. … 2. … |
| … | 1. … |

## Find the Bugs

**Execute your test cases** and **report all bugs found** in your Jira project.

* Use appropriate **title**.
* **Describe** the problem.
* Write the **steps to reproduce** the problem.
* Optionally, you may add summary for the bugs in **Manual-Testing-Homework-FirstName-LastName.docx**. and **attach** **screenshots**.

### Graphical user interface, application Description automatically generatedRegister

Users can **register** on the site with their own provided **username**, **email**, **password**, **first name** and **last name**:

**Requirements:**

* **Username** must be **at least 5 characters long** and consists of **only letters and digits**.
* **Email** consists of something **@ (at)** something **. (dot)** something. Email is required.
* The **Password** must be **at least 6** and at **max 20** characters long.
* **Confirm password** should **match password**.
* **First name** is required and consists **only letters, at least 2**.
* **Last name** is required and consists **only letters, at least 2**.

### Graphical user interface, text, application, email Description automatically generatedLogin

Users can **login** by username + password:

**Requirements:**

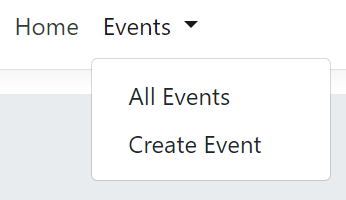
* User **can’t log in** without either **username** or **password**.
* You can log in with **user: guest, password: guest**.
* When clicked **Login button redirects** to **Home page**.

### Homepage

**After** either **register / login**, users are taken to **Home page**.Graphical user interface, text, application, email

Description automatically generated

**Requirements:**

* User should have **two ways to navigate** to **all events** page: clicking **“all events” link** and from the **drop-down menu** under “Events”.
* User should have **two ways to navigate** to **Create event**: clicking **“new event”** link and from the **drop-down menu** under “Events”.
* User should be able to **log out**, by clicking the **Logout button**.
* At the home page the **number of total events** in the system are shown
* At the home page the **number of user’s owned events** are shown.

### Create New Event

**Logged users** can **create new** **events**.

Graphical user interface, text, application, email

Description automatically generated

**Requirements:**

* **Each event** should **have a name**.
* **Name** should be **at least** **3 characters** long and no more than 50 characters.
* Each **event** should **have a place**.
* **Place** should be at **least 3 characters** long and no more than 70 characters.
* Each **event** should have a **Start** and **End date**.
* **Start date** should be **before** the **End date**.
* **Start date** should be in the **future**.
* The **year** of the **start date** should be **before 2100**.
* **End date** should be in the **future**.
* The **year** of the **end date** should be **before 2100**.
* **Total Tickets** should be a **positive number** and **> 0**.
* **Total Tickets** maximum should be **1000**.
* **Price Per Ticket** should be **>= 0** and **<= 1000**
* When clicked **Create event button, creates** the **new event** and **redirects** user to **All events page,** with the newly created event.
* **Back to list** button**,** when clicked, redirects to **All events page**.

### All Events

Users can **view** all existing events but can **edit** and **delete** only events that **they created**.

**Graphical user interface, text, application

Description automatically generated**

**Requirements:**

* There **are Delete | Edit** links for the event that the **current user owns**.
* There **are not Delete | Edit** links for the event that is **owned by another user**.
* There is a **Create new link,** which leads to creation of new event form.

### Edit Event

**Logged user** can **edit** the **name**, **place**, **start date**, **end date**, **total tickets**, **and price** per ticket **for his own events**.

**Graphical user interface, text, application, email

Description automatically generated**

**Requirements:**

* All fields are **editable**.
* When clicked **Edit button**, redirects to All events page, where edited **event is saved**.
* **Back to List button** leads to All events page, **without saving the changes** made to the event.

### Delete Event

From the All events page**, logged user** can delete his own events. When clicked, the [Delete] link leads to **Delete Existing event form**.

**Graphical user interface, application

Description automatically generated**

**Requirements:**

* **Name field** is **not editable** and is populated with the name of the event for deletion.
* **Place field** is **not editable** and is populated with the place of the event for deletion.
* **Delete button**, when clicked, deletes the event, and redirects to **All events page**, where the **deleted event is not present.**
* **Back to List button**, when clicked, redirects to **All events page**, **without deleting the event**.