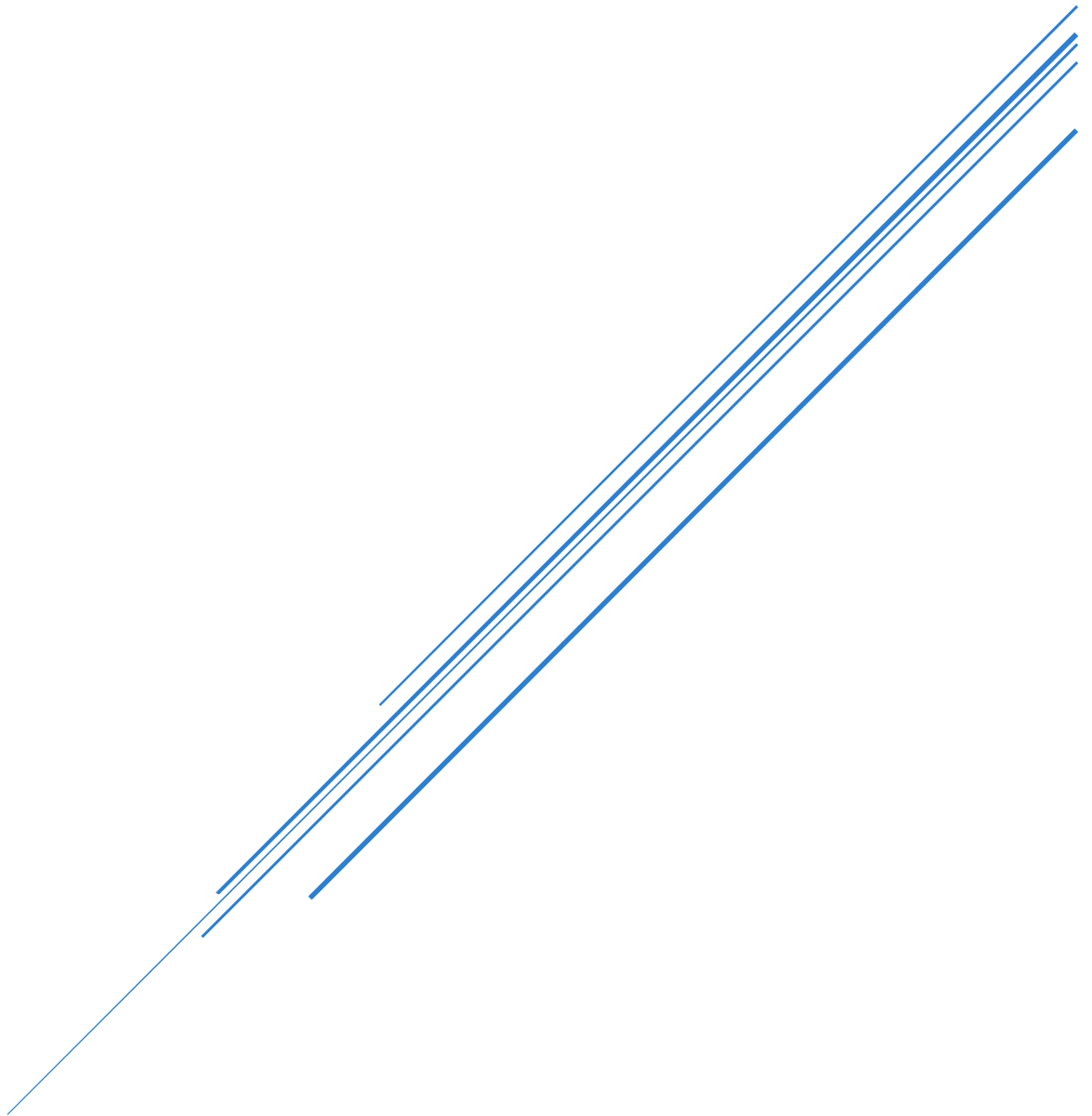


PORTFOLIO

Sigrid Laas-Hurt, Junior Software Tester



2024

Table of Contents

Project#1 Restaurant Food Ordering application	2
Project overview	2
My responsibilities include	2
Project#2 Company Collaboration Platform	3
Project overview	3
My responsibilities include	3
Project#3 CRM Platform	6
Project overview	6
My responsibilities include	6
Project#4 Registration Forms	9
Project overview	9
My responsibilities include	9
Project#5 Task Management Platform.....	10
Project overview	10
My responsibilities include	10
Project#6 Work Management Tool	11
Project overview	11
My responsibilities include	11

Project#1 Restaurant Food Ordering application

Project overview

Our customer Bella Pizzeria has decided to build a food ordering application that will allow contactless ordering in the restaurant (replacing the physical menu). The app must provide a customer with an overview of all the food and drink available in the restaurant, accept orders, and securely store customer data to comply with GDPR and data privacy laws.

The main functionalities of this app are:

- Contactless ordering for customers to keep staff safe
- No download is required, just scan a QR code or enter a web address
- Browse the menu, and order food and drinks
- Simple product management
- Contact Tracing securely records customer data and meets privacy laws.

My responsibilities include

1. Worked in an agile team and followed Scrum routines.
2. Acted as the coordinator for Spint#1.
3. Utilized various testing techniques such as functional testing, use case testing, checklist-based testing, and cross-browser testing.
4. Wrote test documentation in **Confluence**, including checklists for use cases and final report.
5. Reported bugs in **Jira**. Our team has found 35 issues.

Example of the reported bug:

The screenshot displays a Jira issue page for a bug report. The issue title is "Submitting order - all mandatory fields are not marked" with the ID "RFOPTB-367". The description states: "Submitting order - all mandatory fields are not marked (information about possible allergies and telephone number)". The environment is listed as "Windows 11, Google Chrome 116.0.5845.97, Firefox 116.0.3 (64-bit), Apple Ipad, Safari, IOS ver. 14.7.1". The steps to reproduce are: 1. Go to the menu page, 2. Add some items in the basket, 3. View my order, 4. Please note fields "Do you have any allergies or information we should know?" and "Telephone number". The actual result shows two red X icons indicating that mandatory field notation is missing from the specified fields. The expected result states that the notation should be present. The right sidebar shows the issue details, including the assignee "Sigrid Laas-Hurt", the team "Team8", and the reporter "Sigrid Laas-Hurt". The automation section shows the issue was created on August 23, 2023, and updated on September 3, 2023.

Project#2 Company Collaboration Platform

Project overview

The large energy company decided to build an application, that will increase employee productivity and internal communication. The company has more than 3000 employees, that are distributed between tens of departments and several offices. The pandemic highlighted the need for an up to date, cross-device solution for reaching the entire organization. In the increasingly hybrid world, employees need to be empowered to make decisions and communicate openly with peers.

The company decided to build an application that will allow employees to:

1. Have visibility over the company structure with an up to date organization chart
2. Build stronger connections with colleagues by having easy access to their interests, experience, skills and more
3. Be more productive by enabling them to quickly browse through facility and meeting room availability and office locations
4. Collaborate efficiently by creating custom contact lists for projects, events, social activities
5. Create and update their user status, such as sick leave, vacation, maternity leave

My responsibilities include

1. Discovered existing functionalities and improved testing documentation
2. Developed new functionalities to the system that extended system capabilities
3. Worked in an agile team and followed Scrum routines.
4. Wrote test documentation in **Confluence**, including test cases and final report.
5. Utilized various testing techniques such as exploratory testing, functional testing, checklist-based testing, cross-browser testing, and regression testing.
6. Reported bugs in **Jira**. Our team has found 26 issues.
7. Tracked time in **Jira**.

Example of the Test cases:

Test case ID	9.1.3 Departments Module – Search bar
Description	Search bar is visible and usable
Precondition	1. The application is accessible and functioning properly. 2. Admin or User is logged in with appropriate permissions and on the departments page.
Test data	-
Test steps	1. Verify that the searching bar is visible 2. Verify that the user can enter a text "Search" into the searching bar
Expected result	1. The searching bar is visible on the left corner above the filter by section 2. Text "Search" can be typed into the search bar

Test case ID	9.1.4 Departments Module – Search by full match
Description	The user can find a department using text search by full match
Precondition	1. The application is accessible and functioning properly. 2. Admin or User is logged in with appropriate permissions and on the departments page.
Test data	-
Test steps	1. Enter the department name "Product Management" into the search bar and click on "Enter" on the keyboard or Search button 2. Click on "x" next to the searched name
Expected result	1. The searched departments are displayed in a way so only departments with "Product Management" as department name are presented. Above the list of the departments appears "Searching for" text with applied search - "Product Management" 2. Searching the departments reset to default view - by the department name (ascending). "Searching for" text with applied search - "Product Management" are NO longer present.

Example of the Final report:

Confluence

HomeRecentSpacesTeamsAppsTemplatesCreate

Search

TB. Company Collaborati...

Final testing report

Squad Jira board: TB. Collaboration Platform - Agile board - Jira (atlassian.net)

The version of application: 4.0.3

Sprint testing report last update date: Oct 6, 2023

TC-1 Login system module

TC-2 Home page module

TC-3 Directory module

TC-4 User profile module

TC-5 Log out module

Final testing report

Squad Jira board: TB. Collaboration Platform - Agile board - Jira (atlassian.net)

The version of application: 4.0.3

Sprint testing report last update date: Oct 6, 2023

TC-1 Login system module

TC-2 Home page module

TC-3 Directory module

TC-4 User profile module

TC-5 Log out module

Test case ID and name	Description	Team member name	Team member name	Team member name	Team member @Sigrid Laas-Hurt
		Date of retesting: Oct 3, 2023 Mac OS: Ventura 13.6 (22G109) Browser: Mozilla Firefox Version 118.8.1 (64-bit)	Date of retesting: Oct 3, 2023 Windows 10 Pro: 19045.3324, browsers: Chrome Version 116.0.5845.188	Date of retesting: iPhone 14Pro, browser: Safari 16.6.1	Date of retesting: Oct 2, 2023 Android 13, browser: Chrome 117.0.5938.61
Log out module					
Test case ID 5.0 "Log out" button is visible	Check if the "Log out" button is visible and accessible on the web page when the user is logged in.	Passed	Passed	Passed	Passed
Test case ID 5.1 Log out functionality	Check that clicking the "Log out" button successfully logs the user out of the system.	Passed	Passed	Failed TBCCP-157: Log out - Impossible to log out TEST FAILED	Failed TBCCP-157: Log out - Impossible to log out TEST FAILED
Test case ID 5.2 "Log out" directs user to "Log in" page	Check that after successful login out, the user is redirected to the "Log in" page.	Passed	Passed	Unable to test TBCCP-157: Log out - Impossible to log out TEST FAILED	Unable to test TBCCP-157: Log out - Impossible to log out TEST FAILED
Test case ID 5.3 User is not logged out when the system remains logged out when	Check that a user who is logged out of the system remains logged out when	Failed	Failed	Unable to test	Unable to test

Quickstart

Project#3 CRM Platform

Project overview

Elevate Digital is a digital agency offering innovative marketing and web design solutions. They use custom strategies, eye-catching visuals, and data-driven optimization to make online experiences that are fun and increase the visibility of a company's brand.

Elevate Digital organizes its sales and project management with a simple CRM platform. There are 3 main business processes that are supported by the CRM:

- Collection of potential customers from different marketing channels and organising the marketing activities.
- Preparing offers, negotiating terms with potential customers and signing contracts.
- Managing projects with the signed contracts and tracking the progress of projects.

The marketing and sales teams at Elevate Digital currently use the CRM system on a daily basis. Because the number of manual tasks is continuously going up, our customer wants to start automating some parts of the processes and integrating them with other systems used in the company.

The main goal of this project is to make an API that covers all of the main functions of the CRM platform. The CRM platform lets our customers manage their work and different sales processes, like getting new customers, handling customer requests, making proposals, and closing them. The CRM platform is also used as a database of all customers and their data. The first step for this initiative is to build a comprehensive API for the CRM, that will allow for integrations and implementation automation procedures. All of the system's main tasks must be able to be done through the web services, which means that the API must allow this.

My responsibilities include

1. Worked in an agile team and followed Scrum routines.
2. Acted as the coordinator for Spint#2.
3. Wrote test documentation in **Confluence**, including API testing test cases and final report.
4. Built a stable and well-tested API for the following functionalities of the CRM system:
 - a. *Leads management (creation and management of the customer's database);*
 - b. *Invitation of new users;*
 - c. *Funnels management (creation and management of CRM funnels);*
 - d. *Deals management (creation and management of deals in different funnels).*
5. Utilized various testing techniques such as exploratory testing, API testing, browser testing, and end-to-end testing (example is in separate file Team 8 End-to-end test cases_Sigrid_LH).
6. Created **Postman** collections for testing of API.
7. Reported bugs in **Jira**. Our team has found 65 issues.
8. Tracked time in **Jira**

Example of the Test cases:

6.3 Test cases to Create a new note - Sigrid

Test case ID	6.3.1 Create a new note and check the created object
Description	The creation of a new note and verification
Precondition	-
Test data	<pre>1 { 2 "content_text": "STeam8" 3 }</pre>
Test steps	1. Send POST {baseUrl}/obj/note request using body from test data 2. Copy id of the created note from the response 3. Send GET {baseUrl}/obj/note/{uniqueID}, set correct Unique ID (taken from step 2)
Expected result	Step 1: Successful code: 201 Created, a new note is created Step 3: Successful code: 200 OK, correct note is found, data corresponds to input.

Test case ID	6.3.2 Create a new note without authorization
Description	It is NOT possible to create a new note without authorization
Precondition	Authorization type is No Auth
Test data	<pre>1 { 2 "content_text": "STeam8_no_auth" 3 }</pre>
Test steps	1. Send POST {baseUrl}/obj/note
Expected result	<ul style="list-style-type: none">Error code: 401 UnauthorizedObject is not created, error message is shown: "message": null, "translation": "Permission denied: cannot create this object"

Test case ID	6.3.3 Create a note with incorrect "content text" value type
Description	It is NOT possible to create a new note with "content text" of incorrect type: Boolean
Precondition	-
Test data	<pre>1 { 2 "content_text": true 3 }</pre>
Test steps	1. Send POST {baseUrl}/obj/note
Expected result	<ul style="list-style-type: none">Error code: 400 Bad RequestObject is not created, error message is shown: "message": "Invalid data for field content_text: Expected a string, but got a boolean (original data: true)"

Test case ID	6.3.4 Create a note with empty body
Description	It is NOT possible to create a new note with empty body
Precondition	-
Test data	<pre>1 { 2 }</pre>
Test steps	1. Send POST {baseUrl}/obj/note
Expected result	<ul style="list-style-type: none">Error code: 400 Bad RequestObject is not created, error message is shown: "message": "NO_CONTENT_TEXT", "translation": "Please include a content text"

6.6 Test cases to Delete note - Sigrid

Test case ID	6.6.1a Create a new note
Description	The creation of a new note
Precondition	-
Test data	<pre>1 { 2 "content_text": "STeam8_testDELETE" 3 }</pre>
Test steps	1. Send POST {baseUrl}/obj/note request using body from test data
Expected result	<ul style="list-style-type: none">Successful code: 201 CreatedObject is created without any errors

Test case ID	6.6.1b Delete note and check data
Description	It is possible to delete existing note
Precondition	6.6.1a is successfully completed
Test data	-
Test steps	1. Send DELETE {baseUrl}/obj/note/{uniqueID}, set correct Unique ID (taken from 6.6.1a) 2. Send GET {baseUrl}/obj/note/{uniqueID}, set correct Unique ID (taken from previous step)
Expected result	<ul style="list-style-type: none">Step 1: Successful code: 204 No Content, deletion was successfulStep 2: Error code: 404 Not Found, "status": "MISSING_DATA", "message": "Missing object of type note: object with id {(uniqueID)} does not exist"

Example of the reported bug:

T8CA-159

Assign to me

Environment

Windows 11 Home; Postman: Desktop platform version 10.19.0

Priority

Medium

Labels

Team8

Description

Stages API: GET list of all stages - it is possible to set Boolean for limit

Steps:

1. Set limit **Boolean**

2. Send **GET** request `{{baseurl}}/obj/stage?limit=false`

Actual result:

Status 200 OK is presented and the user can see list of all stages.

Expected result:

Error message should displayed that only integer is possible to set.

Reporter

Sigrid Laas-Hurt

Attachments (1)

Add a comment...

Pro tip: press **M** to comment

To Do

Actions

Details

Releases

Automation

Rule executions

Configure

Created October 19, 2023 at 8:50 PM

Updated November 3, 2023 at 10:58 AM

T8CA-157

Assign to me

Environment

Windows 11 Home; Postman: Desktop platform version 10.19.0

Priority

High

Labels

Team8

Description

Notes API: POST create a new note with empty body

Steps:

1. Set "Body" - raw and JSON and leave "Body" empty

2. Send **POST** `{{baseurl}}/obj/note` request

Actual result:

Status 201 Created is presented and unique ID is available.

Expected result:

Error code: 400 Bad Request is presented

Object is not created, error message is shown: "message": "NO_CONTENT_TEXT", "translation": "Please include a content text"

Reporter

Sigrid Laas-Hurt

Attachments (1)

Image-20231026-180705.png

346 KB

26 Oct 2023 9:08pm

Confluence pages

mentioned on

Team 8 S. Final testing report

Updated on Nov 3, 2023

Activity

Show

All

Comments

History

Add a comment...

Pro tip: press **M** to comment

To Do

Actions

Details

Releases

Automation

Rule executions

Configure

Created October 19, 2023 at 8:35 PM

Updated November 3, 2023 at 10:58 AM

Project#4 Registration Forms

Project overview

We have a project with different registration forms (HTML, CSS, and JavaScript stack is used). Using those registration forms we will not only cover the basic functionality of Cypress but also make an introduction to the main parts of front-end architecture. Registration forms allow people to enter their data for registering them as users on the Web page.

The registration forms are partially covered by the automation tests, which ensure that forms are working and all workflows and elements on the forms are working, though some parts are not tested at

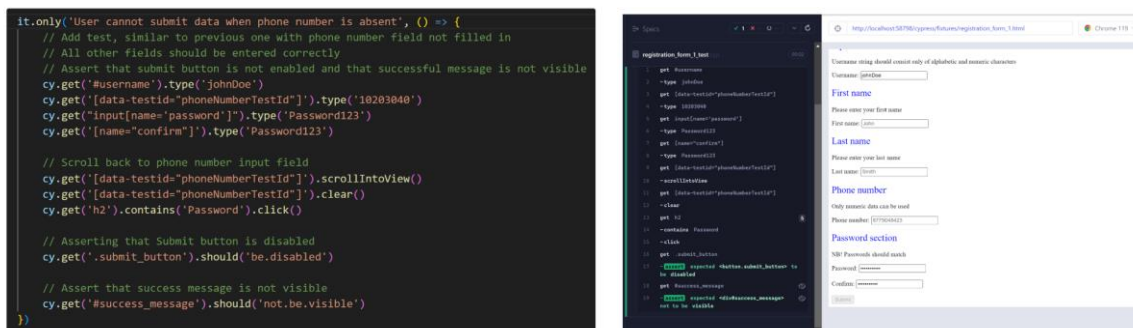
Also, some parts of the automated tests for the second form intentionally contain some bugs, so automated tests are not working properly and need to be fixed.

My responsibilities include

1. Worked individually.
2. Created UI automation tests using **Cypress**.
3. Modified the code using the **Visual Studio Code**
4. **Git** workflow. Code is committed to the **Github** repository
5. Tracked time in **Jira**.

Example of the test:

Add a new test when the phone number is not filled in: Add content to the empty test 'User cannot submit data when phone number is absent'



Summary of results is presented in file Project4_outcomes-Sigrid Laas-Hurt

Github repository: [Automation-Project-1/e2e at master · SigridLH/Automation-Project-1 \(github.com\)](https://github.com/SigridLH/Automation-Project-1)

Project#5 Task Management Platform

Project overview

The goal of the project is to automate UI testing of the Jira Clone application. The Jira Clone is a lightweight ReactJS application that copies functionalities from the popular Jira task management system. The Jira Clone allows users to manage tasks for the project using a simple Kanban board. The goal of this project will be to cover as many functionalities as possible in the UI tests using the Cypress.

My responsibilities include

1. Worked individually.
2. Created UI automation tests using **Cypress** for the following functionalities
 - a. Creation, editing and deleting of tasks;
 - b. Creation, editing and deleting of task comments;
 - c. Time tracking functionality;
 - d. Search functionality
3. Used the tool **Visual Studio Code**
4. Page Object Model approach
5. **Git** workflow. Code is committed to the **Github** repository
6. Tracked time in **Jira**.

Summary of results is presented in file Project5_outcomes-Sigrid Laas-Hurt

Github repository: [Automation-Project-2/cypress/e2e at master · SigridLH/Automation-Project-2 \(github.com\)](https://github.com/SigridLH/Automation-Project-2)

Project#6 Work Management Tool

Project overview

The goal of the project is to automate API testing of the Jira Clone application.

My responsibilities include

1. Worked individually.
2. Created Postman collections and wrote backend automation tests with **Postman** for the following functionalities
 - a. Creating and updating tasks
 - b. Deleting tasks
 - c. Time tracking functionality – fir logging time and adding time estimation
 - d. Creating, editing and deleting task comments
3. Tracked time in **Jira**.

Example:

