"LIETUVOS SPAUDOS FOTOGRAFIJA" competition system development

Test Plan

Table of Contents

[1 Project Description - 5 -](#_Toc161833775)

[2 Scope of testing - 6 -](#_Toc161833776)

[2.1 Preconditions and Outer Dependencies - 6 -](#_Toc161833777)

[2.2 Limitations - 6 -](#_Toc161833778)

[2.3 Suspension criteria and resumption requirements - 6 -](#_Toc161833779)

[2.4 Release/ Acceptance Criteria - 7 -](#_Toc161833780)

[2.4.1 Register - 7 -](#_Toc161833781)

[2.4.2 First system page - 7 -](#_Toc161833782)

[2.4.3 Log In - 7 -](#_Toc161833783)

[2.4.4 Forgot password - 7 -](#_Toc161833784)

[2.4.5 Edit personal information - 8 -](#_Toc161833785)

[2.4.6 Explore Active Photo Competitions - 8 -](#_Toc161833786)

[2.4.7 Photo submission - 9 -](#_Toc161833787)

[2.4.8 Photo submission editing - 9 -](#_Toc161833788)

[2.4.9 Check active photo competition - 9 -](#_Toc161833789)

[2.4.10 Competition category - 9 -](#_Toc161833790)

[2.4.11 Thumbnails to full-screen photo - 10 -](#_Toc161833791)

[2.4.12 Like & unlike - 10 -](#_Toc161833792)

[2.4.13 Sorting functionality - 10 -](#_Toc161833793)

[2.4.14 Manage users - 10 -](#_Toc161833794)

[2.4.15 Download of photos by category - 11 -](#_Toc161833795)

[2.4.16 Add a new system user - 11 -](#_Toc161833796)

[2.4.17 Pending participant - 11 -](#_Toc161833797)

[2.4.18 Past competition records - 12 -](#_Toc161833798)

[2.4.19 Manage competition stages - 12 -](#_Toc161833799)

[2.4.20 System language - 12 -](#_Toc161833800)

[2.4.21 Change password - 12 -](#_Toc161833801)

[2.4.22 Create new competition - 13 -](#_Toc161833802)

[2.4.23 Edit competition information - 13 -](#_Toc161833803)

[3 Test Items - 13 -](#_Toc161833804)

[Software to be tested : - 13 -](#_Toc161833805)

[Interfaces to be tested: - 13 -](#_Toc161833806)

[3.1 Features to be tested: - 14 -](#_Toc161833807)

[3.2 Features Not to Be Tested - 14 -](#_Toc161833808)

[4 Testing Approach - 14 -](#_Toc161833809)

[Project objectives - 14 -](#_Toc161833810)

[4.1 Prioritization - 14 -](#_Toc161833811)

[4.2 Entry/Exit Criteria - 15 -](#_Toc161833812)

[Entry criteria for each testing level are: - 15 -](#_Toc161833813)

[Exit criteria for each testing level include: - 15 -](#_Toc161833814)

[4.3 Item Pass/Fail Criteria - 15 -](#_Toc161833815)

[4.3.1 Item passes if it is proved that: - 15 -](#_Toc161833816)

[4.3.2 Item fails if there are: - 15 -](#_Toc161833817)

[4.4 Classification of Defects - 15 -](#_Toc161833818)

[4.5 Resumption and Suspension Criteria - 16 -](#_Toc161833819)

[5 Testing Levels - 17 -](#_Toc161833820)

[5.1 Unit Testing - 17 -](#_Toc161833821)

[5.1.1 Test Responsibilities - 17 -](#_Toc161833822)

[5.1.2 Test Methods - 17 -](#_Toc161833823)

[5.1.3 Entry Criteria - 17 -](#_Toc161833824)

[5.1.4 Exit Criteria - 17 -](#_Toc161833825)

[5.2 System Testing - 17 -](#_Toc161833826)

[5.2.1 Test Responsibilities - 18 -](#_Toc161833827)

[5.2.2 Test Methods - 18 -](#_Toc161833828)

[5.2.3 Entry Criteria - 18 -](#_Toc161833829)

[5.2.4 Exit Criteria - 18 -](#_Toc161833830)

[5.3 Other Testing Types - 18 -](#_Toc161833831)

[5.4 Smoke Testing - 18 -](#_Toc161833832)

[5.4.1 Regression Testing - 18 -](#_Toc161833833)

[5.4.2 Performance Testing - 18 -](#_Toc161833834)

[5.4.3 Confirmation Testing (Retesting) - 19 -](#_Toc161833835)

[5.4.4 API Testing - 19 -](#_Toc161833836)

[6 Test Schedule, Organization, and Responsibilities - 19 -](#_Toc161833837)

[6.1 Test Project Organization & Resources - 19 -](#_Toc161833838)

[6.2 Training Needs - 20 -](#_Toc161833839)

[7 Tools for Test Design and Execution - 20 -](#_Toc161833840)

[7.1 Test Management Tool - 20 -](#_Toc161833841)

[7.2 Testing Tools - 21 -](#_Toc161833842)

[8 Test Environments - 21 -](#_Toc161833843)

[8.1 Development Environment - 21 -](#_Toc161833844)

[Actions Required for Development Environment - 21 -](#_Toc161833845)

[8.2 Test Environment - 21 -](#_Toc161833846)

[Actions Required for Development Environment - 21 -](#_Toc161833847)

[8.3 Production Environment - 22 -](#_Toc161833848)

# Project description

The " Lietuvos spaudos fotografija " competition is organized annually. This competition provides an opportunity for Lithuanian and foreign photographers to publish their photos in different categories. Although the competition has a long tradition, there is no convenient system in which the members of the commission can evaluate photos, and the participants raise them. The system currently in use is outdated and creates usage challenges. The creation of a modern system would facilitate the work of the competition commission, and also the participants would have the opportunity to use an intuitive system.

The organizers of " Lietuvos spaudos fotografija " expressed their desire to create a new system that will be dedicated to the administration of the competition. The functionality will have to be created in the system that will allow uploading photos to the contestants, evaluators to log in and vote for the photos they like, moderators and the administrator to manage the system users, create contests, see evaluated photos, etc.

# Scope of testing

## Preconditions and Outer Dependencies

At the initial stage of the project, local testing is provided.

This requires tools/programs (if not available, must be additionally installed on the computer): IntelliJ IDEA, GitBash, Node.js, pgAdmin4, Google Chrome, Mozilla Firefox, and Microsoft Edge.

Once/if we have a domain, the test will be redirected to the web page and tested in Google Chrome, Mozilla Firefox, and Microsoft Edge.

## Limitations

It is planned to study in detail the main parts of the system:

* First system page
* Registration form
* Log In form
* Roles: Administrator, Moderator, Jury, User.
* The form of creating and editing a competition
* Competition categories
* Photo submission and editing
* Jury voting opportunity
* Manage competition stage

Other parts of the system will also be tested:

* Forgot password form and email sending
* System language
* Administrate and edit personal information

## Suspension criteria and resumption requirements

The test project manager has the right to suspend all or part of the test if any of the following occurs:

* One “fatal fault”
* More than three “serious faults” occurring in the same function. This will be regarded as equivalent to one “fatal fault”

Project management will be informed immediately and will determine the appropriate action.

## Release/ Acceptance Criteria

### Register

* When I navigate to the registration page, I should see a clear and user-friendly registration form.
* The form includes fields for Name, Surname, Birth Year, Telephone Number, Email, and choose between "Atstovaujama žiniasklaidos priemonė" or "Laisvai samdomas.", Password.
* All fields are mandatory.
* If "Atstovaujama žiniasklaidos priemonė" is chosen, an additional field for entering the media entity's name should appear.
* If the provided email address already exists in the system, the user should receive an error message indicating that the account already exists.

### First system page

* When I navigate to the system's first page, I should see a clear and user-friendly interface welcoming me to the system.
* There should be options or buttons to either login or create a new account.
* There should be a section or link to view the list of active competitions, visible even before logging in.
* There should be the ability to change language LT/EN.

### Log In

* When navigating to the login page, I should see a clear and user-friendly login form.
* The login form should include fields for entering my email address and password.
* If I forget my password, there should be an option to initiate the password recovery process ("Forgot Password").
* After entering valid credentials, clicking the "Login" button should authenticate me into the system.
* If I enter invalid credentials, I should receive a clear error message
* Once logged in, there should be an option to log out, terminating the current session.

### Forgot password

* On the login page, there should be a clearly visible and accessible "Forgot Password" link.
* Clicking the "Forgot Password" link should lead me to a page with a field to input my registered email address.
* After entering my email address, there should be a button to initiate the password recovery process.
* Upon submission, an email should be sent to the provided email address with instructions and a secure link for resetting the password.
* Requirements for email can be found here: *UC2 Password setting*

### Edit personal information

* When logged into the system, I should be able to navigate to an "Edit Personal Information" page.
* All fields should be editable.
* All existing personal information data (Name, Surname, Birth Year, Telephone Number, Email, Role, and Media Entity's Name) should be prefilled.
* The page should include a dropdown menu to choose between "Atstovaujama žiniasklaidos priemonė" or "Laisvai samdomas."
* If "Atstovaujama žiniasklaidos priemonė" is chosen from the dropdown or if it was already entered, an additional field for entering or changing the media entity's name should appear.
* If "Laisvai samdomas" was previously selected, changing the dropdown to "Atstovaujama žiniasklaidos priemonė" should make the media entity's name field appear.
* There should be a "Save Changes" button to apply the modifications.
* Validation messages should be displayed if any of the edited fields have errors or missing information.
* Users should have the option to cancel the editing process, reverting any changes made.
* Upon successfully saving the changes, a feedback message should confirm the update.
* All fields are mandatory.

### Explore Active Photo Competitions

* On the first page, participants should see a dedicated section displaying a list of currently active photo competitions.
* Each competition entry in the list should display relevant details, including:
  + - Competition Name
    - Description
    - Photo amount
    - Categories
    - Deadlines (start and end dates)
    - For each active competition entry, there should be a "Participate" button.

If a user logged in to the system:

* After clicking the "Participate" button, a confirmation message should be displayed, confirming that the participant has expressed interest in the competition.
* The participant should receive an email confirmation for participation.
* The email confirmation should contain a brief message confirming that the participant's intent to participate will be reviewed as soon as possible.
* If the user is not logged into the system:
* After clicking the "Participate" button, should be opened log in.

### Photo submission

* When submitting photos, I can choose a category for each photo from a predefined list (e.g., sports, portraits, landscapes).
* The submission process allows me to upload one or more photos for the selected primary category.
* Should be visible the total amount of loaded photos and the possible amount of photos.
* For each photo, a description should be loaded from the metadata.
* Participants should have the option to edit the auto-loaded descriptions.
* After submitting photos for one category, I have the option to choose another category and submit additional photos.
* The system ensures that submitted photos adhere to specified file format and size requirements for each category.

### Photo submission editing

* The system should provide an option to withdraw a submission before the competition deadline.
* The system should provide an option to edit a submission before the competition deadline.
* Each photo entry should display the selected category, the auto-loaded description, and the uploaded image.
* Participants should be able to edit the category selection and the description for each submitted photo.
* The system should update the submission details in real time as participants make edits
* Participants can add new photos for additional categories or replace existing photos.
* The system should validate edits and additions, ensuring they adhere to specified file format and size requirements for each category.

### Check active photo competition

* The system should display a list of competitions with relevant details (e.g., name, description, start date, end date).
* Only competitions that are currently active should be visible.

### Competition category

* The system must present a comprehensive list of categories for the active competition, easily accessible to the jury member.
* Each category should prominently display the total number of photos submitted, providing a quick overview of the competition's scale.
* The jury member should be able to enter each category seamlessly to review the submitted photos.
* For the Individual photo category, the photos must be shuffled, ensuring a randomized presentation order.
* In the case of photo series, each series must be clearly distinguished from others.
* Individual photos must be shuffled and presented in a different order than originally uploaded into the system.
* The original file names of photos should be replaced with random identifiers to protect author anonymity.
* The commission member should not have access to information revealing the identity of the photo authors during the review process.
* The system must display small, clear thumbnails of the photos within each category.
* Thumbnails should accurately represent the content of the photos, enabling the jury member to make informed assessments.
* Thumbnails can be opened as a full-screen photo US11

### Thumbnails to full-screen photo

* Clicking on a thumbnail should open a full-screen view of the photo.
* The full-screen view should include the photo, its description, and navigation controls.
* The full-screen view should allow the jury member to navigate between photos using the next and previous buttons.

### Like & unlike

* When viewing a photo, I should see a clearly identifiable "Like" button.
* Clicking on the "Like" button should immediately register my like for the photo.
* The "Like" button should visually indicate that I have liked the photo
* If I have already liked a photo, I should be able to undo my action by clicking the "Unlike" button.
* The "Unlike" button should visually indicate that the photo is no longer liked by me.
* Like and unlike should be for full-screen photos and thumbnails as well.

### Sorting functionality

* On the photo viewing page, there should be a "Sort by Liked" option.
* When I select "Sort by Liked," the photos should be rearranged to display first those that I have liked.
* There should be a second sorting option labeled "Sort by Without Like to Liked."
* Choosing this option should organize the photos so that those without likes are displayed first, followed by photos I have liked.

### Manage users

* The admin can view a list of all users.
* The list should display key user information like name surname email PLUS
* Near participant name should be entered max photo number. By default <50 at all. <30 for 'alone' photos and <
* The admin, can click on a user in the list to access a detailed user information page.
* The detailed page should display additional information such as date of birth, phone number, and email (fields can't be edited by admin).
* Admin can edit user roles in the system. Participant, jury, moderator, admin.
* Admin can block users.

### Download of photos by category

* This section is easily navigable and provides a clear overview of available categories.
* Near each section should be visible the total amount of photos.
* Admin and Moderator can choose to download all photos from the category.

### Add a new system user

* As an admin, I should have the option to select the type of user to add, which includes participant, jury, moderator, or admin.
* When adding a participant or moderator:
  + - Admin must provide details such as Name, Surname, Birth Year, Telephone Number, and Email.
    - Admin should choose between "Atstovaujama žiniasklaidos priemonė" or "Laisvai samdomas."
    - If "Atstovaujama žiniasklaidos priemonė" is selected, an additional field for the media entity's name should appear.
* Admin has to set a password for the new user and confirm it by entering it twice.
* When adding a jury member:
  + - Admin should input Email, Name, and Surname.
    - Admin should set the password for the jury member and confirm it by entering it twice.

### Pending participant

* As an Admin and Moderator, I want to access a list of pending users who want to participate in a competition.
* Acceptance criteria:
* The Admin and Moderator can access a page displaying users who have submitted competition entry requests.
* In the pending list should be visible full participant information (such as name, surname, etc. this information comes from user registration).
* On the detailed page, the Admin and Moderator have clear options to either approve or reject the user's competition entry.
* After approving or rejecting a user's competition entry, the system sends a notification or message to the user with the status update.???

### Past competition records

* The Admin and Moderator can access a dedicated section that lists all past competitions.
* Should be visible all information which was entered before.
* Admin and Moderator can check all photos which were summited for a competition.

### Manage competition stages

* The system should provide a clear and accessible option for the administrator and moderators to manually reset photos that have not received any likes for the next stage of the competition.
* Upon initiating the reset action, the administrator or moderator should be asked to confirm action before the reset is executed.
* Admin and moderator should be able to view the total number of likes received by each photo in the competition.
* This display should be visible alongside each photo and should accurately reflect the number of likes, ranging from 0 to any positive integer.
* After the reset action is performed, only photos that have received likes should be visible to the jury members.
* Photos without any likes should be hidden from the jury's view.
* Admin should see hidden photos as well.

### System language

* There should be a clearly visible option or button within the system interface to switch languages.

### Change password

* There should be a clearly visible option or button within the user account settings to change the password.
* When I click on the password change option, I should be directed to a secure page or modal where I can enter my current password and then enter and confirm my new password.
* All fields (current password, new password, confirm new password) should be mandatory.
* Password strength requirements should be clearly communicated to the user (e.g., minimum length, required characters).
* If I enter the current password incorrectly, I should receive an error message indicating that the password is incorrect.
* If the new password and confirm new password fields do not match, I should receive an error message indicating that they do not match.
* After successfully changing the password, I should receive a confirmation message indicating that the password has been updated.

### Create new competition

* The admin should have access to a user-friendly interface where they can input the following details for the new competition:
  + - name
    - description
    - total number of photo submissions allowed
    - categories
    - start and end dates.
* Admins should be able to select existing categories from a predefined list or add new categories as needed for the competition.
* When adding a new category, the admin must provide the category name, description, and specify whether it allows only one photo submission or a series of up to 15 (default) photos. Default number of photos should be editable.
* Created category should be saved for future and added to category list. (Related with "Admins should be able to select existing categories from a predefined list or add new categories as needed for the competition").
* If a category allows only one photo submission, participants should be restricted to submitting only one photo for that category.
* If a category allows a series, participants should be able to submit up to 15 (can be another number) photos for that category.

### Edit competition information

* The admin should have access to a specific section or interface where they can view a list of all created competitions and select the one they wish to edit.
* Admin should be able to modify the following details of the competition:
  + - Name
    - Description
    - Total number of photo submissions allowed
    - Categories (add, remove, or modify existing categories)
    - Date (start and end dates)
* Admins should have the ability to add new categories, edit existing ones, or remove categories entirely from the competition.
* Before finalizing any changes, the system should ask the admin to confirm their edits to prevent accidental modifications.

# Test Items

Software to be tested :

Web applications, Database systems, security software

Interfaces to be tested:

UI, API, Database interface, network interface, web browser interface.

## Features to be tested:

Functionalities are going to be tested:  
**User Interface (UI):** This includes the design and layout of the software, how users interact with it, and its overall usability.    
**Automation:** Capability to automate repetitive tasks or workflows, improving efficiency and productivity.    
**Security:** Features to protect user data and privacy, including encryption, authentication, and access controls.

Test cases for functionalities are going to be defined:  
 **Requirement Analysis:** Understand the functional requirements of the software feature. This involves reviewing specifications, user stories, or any documentation that describes what the feature is supposed to do.    
**Create Test Cases:** For each test scenario, create one or more test cases.

The functionalities of the system will be tested to the acceptance testing level.

## Features Not to Be Tested

Functionalities that are out of testing scope, e.g. standard functionalities of some applications if only small configurations are done.

Environmental-specific configurations and performance testing are out of scope.

# Testing Approach

## Project objectives

1. To create a convenient platform that allows members of the commission to view and evaluate the submitted work.

2. Provide participants with an easy way to upload their photos to the system.

3. Ensure a smooth registration and participation process.

4. To allow the administrator to manage the content of the system, to control the registration of participants.

5. To create a system in which it is possible to organize competitions with several stages of selection.

## Prioritization

Implemented functionalities

Common new interfaces

Basic use cases or end-to-end scenarios

Abnormal usage of the system

## Entry/Exit Criteria

### Entry criteria for each testing level are:

The system part is up and running:

Created in a programmer's environment

All needed configurations have been completed

Tested by Unit tests

The system part has been fully allocated for testing:

A code review must be performed

In the GitHub repository, the code must be transferred to the

Development branch

In Jira, the task must be moved to the Ready for QA section.

Test cases are created based on acceptance criteria from User stories in Jira

Test cases are reviewed and approved

Earlier testing-level exit criteria have been fulfilled

### Exit criteria for each testing level include:

Test Cases related to testing level are agreed to cover enough of the requirements

All urgent and high-severity defects related are fixed and closed

All Test Cases related to the testing level in question are executed

All entry and exit criteria are fulfilled

## Item Pass/Fail Criteria

### Item passes if it is proved that:

The material used as the base for testing is approved

The test case is based on the requirement test base

The test case is reviewed and approved

All test cases related to the part of the system are successfully run

The coverage of the test cases is seen and approved to cover enough

The part of the system is seen as functional from integrated parts as well

### Item fails if there are:

Inconsistencies with acceptance criteria detected during testing

Items are not working properly

The code is not correct

Human error may cause an item to fail in the testing phase

If the item is not created at all

## Classification of Defects

The table below shows the classification of the defect during the implementation project.

| **Severity** | **Description** | **Impact to testing** | **Actions** | **Analysis & Fix schedule** |
| --- | --- | --- | --- | --- |
| **Urgent Defect** | Defect causes that important functionalities are not achievable, that data disappears, performance deteriorates or parts of the system crash | showstopper for testing | Example: Needs actions immediately | Example: Defect analysis should be started immediately, fix delivery time ASAP |
| **High Defect** | Defect means that a functionality does not work according to specifications or does not work at all | Essential function not functional | Example: Needs actions soon | Example: Defect analysis should be started latest in one to two days, fix delivery in one week |
| **Medium Defect** | With Medium severity defect it is meant that some part of the system does not work normally but needs some special actions in order to get all the normal functionalities | Some part of function is not working | Example: Can be accepted, actions to be decided in the Steering group | Example: Defect analysis should be started in one week, fix delivery in two weeks |
| **Low Defect** | With a Low severity defect it is meant a defect that is mainly cosmetic and does not have any effect on functionality | Cosmetic error, e.g. wrong error message | Example: Can be accepted | Example: Defect analysis should be started in one week if there is time and higher severity defects are fixed, fix delivery in one month |

## Resumption and Suspension Criteria

The system is considered complete when it meets these functional and non-functional requirements and is fully tested

**Functional:**

* Submit a registration request (valid for the contestant).
* Log in to the system (valid for members of the commission, contestants, moderators, administrators).
* Manage your data (valid for members of the commission, contestants, moderators).
* Create an account for internal users (i.e. jury, moderator).
* Create contests (valid for administrator).
* View the submitted photos for each category.
* Vote for the photos you like (valid for the jury).
* Filter photos that have collected at least one "like" (valid for admin and moderator).
* Submit photos to the competition.
* Manage system users.
* Edit your submitted photos.

**Non-functional:**

* Java, Spring, React.js technologies, relational database are used to create the program.
* Any request must be answered by the system in less than 2 seconds (web page loading).
* If passwords are used in the system - system passwords cannot be stored in open text.
* At the time of delivery of the system, the system must work correctly with the latest versions of Google Chrome, Mozilla FireFox and Microsoft Edge browsers of that time.

# Testing Levels

In this project, testing will be conducted at these levels: Unit testing and System testing. Additionally, API (if needed), Performance, Regression, Confirmation testing will be performed. The test activities will be performed based on requirements.

## Unit Testing

The software is tested on the level of source code. Components of code are tested.

### Test Responsibilities

The developers are responsible for this testing level.

### Test Methods

Each developer will test his code.

### Entry Criteria

All planned functionality and bug fixes have been implemented.

### Exit Criteria

The coverage of the testing is at least 50%.

100% of unit tests pass.

Unit-tested software is available for the next test phase.

There are no outstanding medium or high-severity issues.

## System Testing

System Testing verify that a system meets its specified requirements. The system is tested as a whole under the assumption that the main elements, especially the software part, have already been independently tested. System test focuses on functions.

Test cases will be based on requirements. The most risky test scenarios will be tested first. The least risky area will be tested as time permits.

### Test Responsibilities

The test team is responsible for this testing level. The members of this team will define and create test cases.

### Test Methods

The test team will create test cases based on requirements and experience. Then manual and automated tests will be executed.

### Entry Criteria

The graphical user interface is functional.

Most of the developed code is unit-tested.

Test cases have been prepared and peer-reviewed.

Smoke Testing has been executed and confirmed that the system is ready for System Testing.

### Exit Criteria

100% of test cases are executed.

100% of tests pass.

All high and medium-priority errors from System Testing are fixed and tested.

## Other Testing Types

The other testing types are taken into consideration. These testing types are Smoke, Regression, Performance, and Confirmation. Additionally, API testing will be included if necessary. The test team is responsible for this testing.

## Smoke Testing

When a new version of code will be delivered, Smoke Testing will be executed to ensure that basic operations are still functioning properly in the system.

### Regression Testing

Regression Testing will be done to verify that a system has not been affected by updates to a new version or defect fixed. A regression test set will be executed every time a new version is installed on a system or a related defect is fixed.

Regression tests will be automated.

### Performance Testing

Based on requirements, the system has to respond within 2 seconds to any query. So Performance Testing will be performed to determine how a system performs in terms of responsiveness.

Additionally, the system has to function correctly with the latest versions of Google Chrome, Mozilla Firefox, and Microsoft Edge browsers. Tests will be performed in the mentioned browsers.

### Confirmation Testing (Retesting)

After a defect is detected and fixed, the system will be re-tested to confirm that the original defect has been successfully removed.

### API Testing

API Testing is under consideration. It will help to test core functionality until the user interface is not ready. Verify if the API returns the expected status code.

# Test Schedule, Organization, and Responsibilities

The high-level testing schedule for the project is acceptance testing.

The main parts involved in the tests at different stages are:

* Tester
* QA
* Development

Effective collaboration and communication between these parts and organizations are essential for successful testing outcomes, ensuring that the software meets quality standards and fulfills user requirements.

## Test Project Organization & Resources

Programmers in charge of Unit tests.

QA/Test team is responsible for executing test cases, identifying defects, and verifying that the software meets specified requirements.

Each tester will take available tasks and notify other team members.

**The main tasks:**

* Test Plan
* Test Execution
* Requirement Analysis
* Test Project organization.

**Duties and responsibilities of individuals:**

* PO – Armine
* PM – Gytis
* SM – Žydre
* Programers – Gedvinas , Dimitrij , Rūta , Valdemaras , Vitalij
* Testers – Edgaras , Žydrė , Austė

**Responsibility and authority**

All responsibilities are split amongst the development team.

Authorization has been granted to Žydre in jira as PM.

Other development team members have the same Authorization.

## Training Needs

Training for the testing team and test management is essential to ensure that testing activities are conducted effectively and efficiently throughout the software development.

A team of testers will need training to familiarize themselves with the principles of program or manual operation:

* Database management and use: pgAdmin4

Testing tools that require consultations to master:

* Testing tools: Zephyr in Jira

Tools that will be used for testing purposes (the possibilities of their use are known to the team):

* Programing tools: IntelliJ IDEA
* Git Bash
* Git Hub
* Internet browsers: Firefox, Edge, Chrome

The necessary consultations for the testing team with relevant training are provided by:

* QA Coach – Maksim Trojan
* Zephyr and Jira – Vaclav Zelenkevič
* Installation and commissioning of the system being developed – development team

# Tools for Test Design and Execution

In this project, the following test tools will be used to support the test process for this Test Plan.

## Test Management Tool

Jira Software will be used for issue tracking, project management, and test case management.

## Testing Tools

Selenium WebDriver will be used for automating web browsers. It’s an open-source tool that meets W3C standards.

Postman will be used for testing APIs. It allows sending various types of HTTP requests to a server and receive responses.

RestAssured Java library will be used for RESTful API testing.

# Test Environments

Each environment serves a distinct purpose in the software development lifecycle, facilitating collaboration, testing, and validation activities at different stages of development. By leveraging these environments effectively, organizations can ensure the quality, reliability, and successful deployment of software products to end-users.

The system will be tested on these browsers: Chrome, Edge, Firefox

Wireframes were made by – Edgaras

The design was selected from other example pages.

When the product has been reviewed by the product owner.

## Development Environment

Specific environments necessary for the operation of the system:

* Git
* IntelliJ IDEA
* PgAdmin4(PostgreSQL 16.1)

Actions Required for Development Environment

Currently, there is no server so the environment is running from a local host.  
The definition and setup of software environments are typically a collaborative effort involving various stakeholders within the organization.

## Test Environment

Currently, the software will be tested on a local host (described in 8.1)

Specific environments necessary for the operation of the system:

* PostgreSQL 16.1
* IntelliJ IDEA
* React

Actions Required for Development Environment

Specified requirements for the environment to be working:

* Sofware for local hosting (described in 8.1)

Environments are typically a collaborative effort involving various stakeholders within the organization.

## Production Environment

Specified requirements for the environment to be working:

* Described in (8.1, 8.1)

Setting up a specific environment involves configuring hardware, software, and other resources to meet the needs of a particular stage

**Hardware:**

* Servers (Physical or Virtual)
* Routers
* Firewalls
* Storage Arrays
* Workstations (for developers and testers)
* Desktops/Laptops (for administrative purposes)
* Backup Devices

**Software:**

* Database Management Systems (e.g., MySQL, PostgreSQL, MongoDB)
* Development Tools (e.g., Integrated Development Environments - IDEs, Git)
* Testing Tools (e.g., Selenium, JUnit, Postman)
* Virtualization Software (e.g., VMware, VirtualBox)
* Security Software (e.g., Antivirus, Intrusion Detection Systems - IDS)
* Monitoring and Logging Hardware (e.g., monitoring servers, logging servers)