

# Acceptance Test Report

*Production Optimiser*

Politecnico di Milano, Information Engineering School, Italy

University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia

Distributed Software Development

6<sup>th</sup> of January 2024

# Overview

This document shows the acceptance testing report to determine and verify which requirements are met. Features and components functionalities are tested with unit/integration tests through code so the primary idea of our acceptance testing is to involve the customer and manually ensure his requirements are fulfilled.

Acceptance testing was supposed to also provide an opportunity to gather final feedback on the frontend part of the application, allowing last minute refinements to be made based on the customer's preferences.

During the planned acceptance testing period, we encountered scheduling challenges due to the Christmas and New Year holidays, which made it difficult to arrange meetings with customers. As a result, the customer involvement we aimed for was not achieved. Our next meeting with them is scheduled for January 9th.

Additionally, the frontend of the application was not ready in time for final testing. Consequently, several tests did not meet the defined Pass/Fail criteria. However, this period allowed us to identify and address bugs, ensuring a more polished product for the final presentation.

Each test report is structured in given format:

<b>Requirement ID(s)</b>	[FR1, FR2, NFR4]
<b>Test ID</b>	AT1
<b>Test Name</b>	User needs platform access
<b>Test description</b>	Unregistered users don't have accounts and want to use the platform. They can request access with contact form.
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. User visit registration request page</li><li>2. Fills email and company details</li><li>3. Submits access request</li><li>4. ....</li></ol>
<b>Pass/Fail criteria</b>	User successfully created registration request that admin can see and approve/deny.
<b>Report</b>	Passing / Failing – additional information

# Acceptance Test Cases

[AT1]      User login, logout, account changes

<b>Requirement ID(s)</b>	[FR1, FR2, FR4, NFR1, NFR2]
<b>Test ID</b>	AT1
<b>Test Name</b>	User login, logout, account changes
<b>Test description</b>	Existing user performs actions with his credentials. Login, logout, and account modifications like email/password change or account deletion.
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. User accesses platform for the first time (account already created)</li><li>2. Inputs email + password and logs in</li><li>3. Performs credential change</li><li>4. User logs out</li><li>5. Performs login with new credentials</li><li>6. User doesn't want to use the platform anymore and performs account deletion</li></ol>
<b>Pass/Fail criteria</b>	Test is passing if all steps are performed without errors and after account deletion user can no longer log in. All sensitive data should be encoded in the database.
<b>Report</b>	Failing – Missing tab for credentials change, only available through admins

[AT2] User session is kept across tabs

<b>Requirement ID(s)</b>	[FR1, FR4]
<b>Test ID</b>	AT2
<b>Test Name</b>	User session is kept across tabs
<b>Test description</b>	User accesses the platform from another tab while already logged in, session is kept if timeout is not met
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. User logs into the platform</li><li>2. Opens the new tab and accesses the platform</li><li>3. Immediately logs in with existing session</li></ol>
<b>Pass/Fail criteria</b>	Test is passing if user doesn't need to input credentials in the new tab, if specific timeout is met(24h) the session is invalidated and he will need to input credentials again.
<b>Report</b>	Passing

[AT3] New user wants to join the platform

<b>Requirement ID(s)</b>	[FR2, FR3]
<b>Test ID</b>	AT3
<b>Test Name</b>	New user wants to join the platform
<b>Test description</b>	New user requests access to the platform and when accepted can login.
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. User fills contact form for account creation</li><li>2. Admin accepts the user and sets initial password</li><li>3. User gets the email and logs into the platform</li></ol>
<b>Pass/Fail criteria</b>	Test is passing if admin successfully approves the request, user gets sent an email and user can log in.
<b>Report</b>	Passing

[AT4]      Admin sees existing tools and has ability to modify them

<b>Requirement ID(s)</b>	[FR5, FR6]
<b>Test ID</b>	AT4
<b>Test Name</b>	Admin sees existing tools and has ability to modify them
<b>Test description</b>	Admin can see all existing tools, has ability to change their name and service URL.
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. Admin logs in</li><li>2. Goes to Service Tool Management tab</li><li>3. Edits some info for some existing service</li></ol>
<b>Pass/Fail criteria</b>	Test is passing if admin successfully sees all tools and can modify them.
<b>Report</b>	Failing – Admins can see the tools but some work still needs to be done for editing

[AT5]      Admin can give/revoke access to specific tool to each user

<b>Requirement ID(s)</b>	[FR7, FR8]
<b>Test ID</b>	AT5
<b>Test Name</b>	Admin can give access to specific tool to each user
<b>Test description</b>	Admin can check which tools user has access to, and can modify that list
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. Admin logs in</li><li>2. Goes to Tool Assignment</li><li>3. Inputs specific user email</li><li>4. Modifies permissions to tools</li></ol>
<b>Pass/Fail criteria</b>	Test is passing if admin successfully sees all models assigned to a user and can modify that list, give access to new models and revoke access to existing ones. User can call specific tool only if access is granted to him else he gets unauthorized error
<b>Report</b>	Passing

[AT6]      Uploading files and text to service tools

<b>Requirement ID(s)</b>	[FR9, FR8]
<b>Test ID</b>	AT6
<b>Test Name</b>	Uploading files and text to service tools
<b>Test description</b>	Both users and admins can call specific service tools with shown input, either image, file or text
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. User selects the tool in top left of the screen</li><li>2. Starts new call with selected tool</li><li>3. Provides required input to the tool</li></ol>
<b>Pass/Fail criteria</b>	Test is passing if service tool is correctly called without any validation errors. Prompt will give users information about which file extensions are supported for specific tool.
<b>Report</b>	Passing

[AT7]      User sees his historical data

<b>Requirement ID(s)</b>	[FR12]
<b>Test ID</b>	AT7
<b>Test Name</b>	User sees his historical data
<b>Test description</b>	User can see all his past calls in the left sidebar
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. User can see his history of calls</li><li>2. User can click on specific one to check inputs and outputs</li></ol>
<b>Pass/Fail criteria</b>	Test is passing if user can successfully retrieve his historical data and sees each specific one
<b>Report</b>	Passing

[AT8] User successfully calls service tool and sees the response

<b>Requirement ID(s)</b>	[FR9, FR10, FR11]
<b>Test ID</b>	AT8
<b>Test Name</b>	User successfully calls service tool and sees the response
<b>Test description</b>	User can call the tool with given input, and receive its output on the platform
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. User selects service tool</li><li>2. Provides input that passes validation</li><li>3. Calls the service and receives the response in raw and formatted</li></ol>
<b>Pass/Fail criteria</b>	Test is passing if user can successfully call the service, then see the provided input, be able to download it and see the output
<b>Report</b>	<b>Failing – More work needed on result showing and input/output downloading</b>

[AT9] Admins can see tool and user statistics

<b>Requirement ID(s)</b>	[FR13, FR14]
<b>Test ID</b>	AT9
<b>Test Name</b>	Admins can see tool and user statistics
<b>Test description</b>	Admins see stats for each tool and see stats for each user
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. Admins goes to statistics tab</li><li>2. Sees dashboard with graphs</li><li>3. Can click specific info for tool or for user and receive more data</li></ol>
<b>Pass/Fail criteria</b>	Test is passing if admin can successfully retrieve info about usage statistics of models and users
<b>Report</b>	<b>Passing</b>



[AT10] Multiple request handling for one tool

<b>Requirement ID(s)</b>	[FR9, NFR5, NFR6, NFR8]
<b>Test ID</b>	AT10
<b>Test Name</b>	Multiple request handling for one tool
<b>Test description</b>	Multiple users can access the tools at the same time and when deployed each tool is easily scalable
<b>Test steps</b>	<ol style="list-style-type: none"><li>1. 2 separate users log in in 2 separate browsers</li><li>2. Call the same tool at the same time</li><li>3. Both requests should pass without any significant delays</li></ol>
<b>Pass/Fail criteria</b>	Test is passing if both calls pass without any added delays compared to single request
<b>Report</b>	Passing

# Conclusion

Out of the 10 acceptance test cases, 3 are still failing at this stage. Despite tests failing at the current stage, we have addressed all the requirements outlined by the customer, ensuring that no requirement has been overlooked or abandoned. During the testing process, we identified 15 small and significant bugs, which provided valuable opportunities to improve the overall quality of the application.

While additional time is needed to polish the implementation and resolve these outstanding issues, we are confident in our ability to achieve 100% success in the test cases. This phase has been really valuable in refining and finishing the application on time.