UMinho

Master's in Software Engineering
Information Technology Project

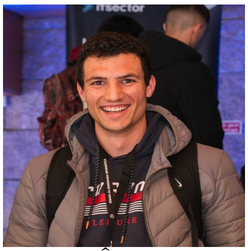
Group E07 - DTx
Maintenance Worker App (MWA)

Pedro Miguel Castilho Martins
PG54146

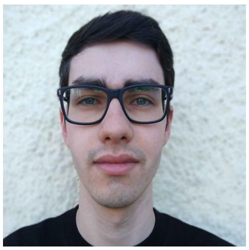
Ana Rita Santos Poças
PG53645

João Bernardo Teixeira Escudeiro
A96075

Orlando José da Cunha Palmeira
PG54123



Miguel Ângelo Silva Senra
PG54093



Miguel Silva Pinto
PG54105

Simão Oliveira Alvim Barroso
PG54236

Oriented By Manuel Alves and Ricardo Rodrigues from DTx

Contents

1. Use Cases Description

This section presents a tabular specification of each use case considered, in order to facilitate the implementation process of each functionality in our system. In this way, we believe that the sequential flow of the actor's interaction with the system is quite clear.

1.1 Authentication

The "Authentication" use case involves the Maintenance Worker obtaining access to the system. Depending on the provided email domain, the user will be directed to the appropriate login page to input their password and if the credentials are valid, it will be granted access to the system.

Use case	1	
Primary Actor	Maintenance Worker	
Secondary Actor		
Pre-conditions	-	
Post-conditions	The Maintenance Worker successfully authenticates on the platform.	
Normal Scenario	**Actor Input**	**System Response**
	1 - The Maintenance Worker selects the Authentication section.	
		2 - The System prompts the user for an email.
	3 - The Maintenance Worker provides his email.	
		4 - Based on the entered email domain, the user is redirected to the corresponding login page to enter the password.
	5 - The Maintenance Worker enters the password.	
		6 - The System validates the credentials.
		7 - The System grants access to the application.
Exception 2 [Incorrect credentials] (Step 6)		6.1 - The System informs that the entered credentials are incorrect. 6.2 - Return to step 2.

Table 1.1: Specification of the "Authentication" use case

1.2 View Ticket

The "View Ticket" use case ensures that, after successfully authenticating, the Maintenance Worker has access to a list of Tickets with Tasks that were assigned to them, not showing Tickets that don't have Tasks assigned to the Maintenance Worker. From this list, the Maintenance Worker can select a specific Ticket to view its details, including the tasks associated with that Ticket.

Use case	2	
Primary Actor	Maintenance Worker	
Secondary Actor		
Pre-conditions	The Maintenance Worker is authenticated.	
Post-conditions	The Maintenance Worker views the desired Ticket.	
Normal Scenario	**Actor Input**	**System Response**
		1 - The System displays a list of Tickets where the Maintenance Worker has Tasks assigned to them, sorted by priority.
	2 - The Maintenance Worker selects the Ticket they wish to view.	
		3 - The System displays the tasks associated with the selected Ticket.

Table 1.2: Specification of the "View Ticket" use case

1.3 View Task

The "View Task" use case ensures that the Maintenance Worker has access to a list of Tasks associated to the selected Ticket. From this list, the Maintenance Worker can select a specific task to view its details and what needs to be done to complete the task.

Use case	3	
Primary Actor	Maintenance Worker	
Secondary Actor		
Pre-conditions	The Maintenance Worker is authenticated and has selected a Ticket.	
Post-conditions	The Maintenance Worker views the desired task.	
Normal Scenario	**Actor Input**	**System Response**
		1 - The System displays a list of tasks that are associated with the selected Ticket.
	2 - The Maintenance Worker selects the task they wish to view.	
		3 - The System displays the task's details, and a stopwatch for the worker to start working and manage the working hours needed to complete the task.

Table 1.3: Specification of the "View Task" use case

1.4 Participate in Ticket Chat

The "Participate in Ticket Chat" use case ensures that the Maintenance Worker has access to a Ticket's Chat. This chat intends to provide a communication channel between the Maintenance Workers assigned to the Tasks in the Ticket and other actors related to the ticket, such as its creator and approver.

Use case	4	
Primary Actor	Maintenance Worker	
Secondary Actor		
Pre-conditions	The Maintenance Worker is authenticated.	
Post-conditions	The Maintenance Worker accesses the Chat related to the Ticket.	
Normal Scenario	**Actor Input**	**System Response**
	1 - The Maintenance Worker selects the pretended Ticket.	
		2 - The System presents a view of the Ticket information and options.
	3 - The Maintenance Worker selects the option that shows the Chat related to the Ticket.	
		4 - The System presents the chat history between the users that have access to the chat.
	5 - The user is able to send text messages or files through the chat, and receive new messages from the other users.	

Table 1.4: Specification of the “Participate in Ticket Chat” use case

1.5 Edit Task

The “Edit Task” use case permits the user to modify the details of the task such as materials used, devices related to the task and photos of the problem to resolve. Additionally the user can change the time slots worked in this task.

Use case	5	
Primary Actor	Maintenance Worker	
Secondary Actor	-	
Pre-conditions	The Maintenance Worker is authenticated and is assigned to the selected task.	
Post-conditions	The Maintenance Worker successfully edits the task details.	
Normal Scenario	**Actor Input**	**System Response**
		1 - The System displays the task page.
	2 - The Maintenance Worker selects the details option or the time slots edit option.	
		3 - The System saves the changes made by the Maintenance Worker.
Alternative 1 [Alternative] (Step 2)	2.1 - The Maintenance Worker selected the details option	
		2.2 - The System displays 3 tabs for the user to edit the photos, materials or devices related to the task.
	2.3 - The Maintenance Worker selects which tab wants to edit.	

Alternative 2 [Alternative] (Step 2)	2.1 - The Maintenance Worker selects the time slots edit option. 2.3 - The Maintenance Worker adds, deletes or edits his time slots related to that task.	2.2 - The System displays the time slots that the users has for that specific task. 2.4 Jump to step 3.
Alternative 1.1 [Alternative] (Step 2.3)	2.3.1 - The Maintenance Worker selects the photos tab. 2.3.3 - The Maintenance Worker grants permission to the app to use the camera. 2.3.4 - The Maintenance Worker selects the camera to take a photo or selects a photo from the phone gallery.	2.3.2 - The System asks the user the permission to use the phone camera. 2.3.5 - Jump to step 3.
Alternative 1.2 [Alternative] (Step 2.3)	2.3.1 - The Maintenance Worker selects the materials tab. 2.3.3 - The Maintenance Worker adds, removes or edits any materials to the task.	2.3.2 - The System presents the materials related to the task. 2.3.4 - Jump to step 3.
Alternative 1.3 [Alternative] (Step 2.3)	2.3.1 - The Maintenance Worker selects the devices tab. 2.3.3 - The Maintenance Worker adds or removes any devices to the task.	2.3.2 - The System presents the devices related to the task. 2.3.4 - Jump to step 3.

Table 1.5: Specification of the “Edit Task” use case

1.6 Close Task

The use case “Close Task” allows the Maintenance Worker to declare a Task as completed.

Use case	6	
Primary Actor	Maintenance Worker	
Secondary Actor	-	
Pre-conditions	The Maintenance Worker is authenticated.	
Post-conditions	The task is marked as closed.	
Normal Scenario	**Actor Input**	**System Response**
	1 - The Maintenance Worker selects the Ticket he wants to handle.	
		2 - The System displays the list of tasks associated with the selected Ticket.
	3 - The Maintenance Worker selects the task he wants to resolve.	
		4 - The System displays the tasks page.
	5 - The Maintenance Worker starts the task by clicking the "Start" button, which triggers the task timer that will track the time spent on the task.	
	6 - The Maintenance Worker, after finishing the task, clicks the "Stop" button, stopping the task's timer.	
		7 - The System displays the time spent on the task through the stopped timer.
	8 - The Maintenance Worker closes the task by clicking the "Close Task" button.	
		9 - The System closes the task prompting the task was closed.
Exception 1 [Exception] (Step 5)		5.1 - The System verifies the worker doesn´t have any temporal record. 5.2 - The System disables the option to close task
Alternative 1 [Alternative] (Step 5)	5.1 - The Maintenance Worker chooses the option to edit time. 5.2 - The Maintenance Worker selects a temporal period. 5.3 - Continue on Step 8	

Table 1.6: Specification of the "Close Task" use case

1.7 Request the Opening of a new ticket

The "Request the Opening of a new ticket" section explains how an user can request a new ticket within the system.

Use case	7	
Primary Actor	Maintenance Worker	
Secondary Actor		
Pre-conditions	The Maintenance Worker is authenticated.	
Post-conditions	A new Ticket is requested.	
Normal Scenario	**Actor Input**	**System Response**
	1 - The Maintenance Worker selects the option of “Issue Request”.	
		2 - The System redirects the Maintenance Worker to the first page of the “Issue Request”.
	3 - The Maintenance Worker selects the “Tenant” related to the new Issue. 4 - The Maintenance Worker selects the “Asset Group” related to the new Issue. 5 - The Maintenance Worker inputs the “Issue Description” related to the new Issue. 6 - The Maintenance Worker selects the “Next” button.	
		7 - The System redirects the Maintenance Worker to the next page where the Maintenance Worker can Add Devices from the Asset Group selected in the previous page.
	8 - The Maintenance Worker selects which device to add to the issue request. 9 - The Maintenance Worker selects the “Next” button.	
		10 - The System redirects the Maintenance Worker to the next page where the Maintenance Worker can Add Photos.
	11 - The Maintenance Worker takes a photo with his camera or selects from the phone gallery to add to the issue request. 12 - The Maintenance Worker selects the “Submit” button.	
		13 - The System validates the information provided by the Maintenance Worker. 14 - The System notifies the user via a message pop up on the screen that the issue request was created successfully.
Exception 1 [Exception] (Step 13)		13.1 - The System notifies the Maintenance Worker that the information inserted were not valid. 13.2 - Return to step 2.

Table 1.7: Specification for the “Request a New Ticket” use case.

1.8 Read notifications

The “Read notifications” section explains how an user can view its notifications within the system.

Use case	8	
Primary Actor	Maintenance Worker	
Secondary Actor		
Pre-conditions	The Maintenance Worker is authenticated.	
Post-conditions	The Maintenance Worker is able to read notifications addressed to him.	
Normal Scenario	**Actor Input**	**System Response**
	1 - The Maintenance Worker selects the “Notifications” icon in the navigation bar at the bottom of the home page.	
		2 - The System redirects the Maintenance Worker to the “Notifications” page.
	3 - The Maintenance Worker views notifications addressed to him in the system.	
	4 - The Maintenance Worker selects a Notification.	
		5 - The System redirects the Maintenance Worker to the page related to the Notification.

Table 1.8: Specification for the “Read Notifications” use case.

2. Requirements

In this chapter, we will present and describe the base requirements needed in order to proceed to the development of this project. This requirements were defined in collaboration with our client and other entities, with the main goal of assuring that the system we will develop responds to the functional and non functional needs expected. Each requirement will be presented with its details, indicating their main functionalities, as well as technical restrictions.

2.1 Functional Requirements

Requirement #: **FR1**	Type: Functional	*Use cases* #: 1
Description	**The System must have an authentication mechanism.**	
Rationale	To ensure proper identification of Users and management of access levels, ensuring data protection and functionalities according to the User's profile.	
Origin	Client	
Fit criterion	The System must be able to present an authentication System where it is possible to enter an email and a password.	
Priority	Must	

Table 2.1: Functional requirement regarding System authentication.

Requirement #: **FR2**	Type: Functional	*Use cases* #: 2
Description	**The User must be able to access the list of Tickets associated to him, where he has Tasks.**	
Rationale	Providing access to a list of associated Tickets ensures that Users can easily locate and manage their responsibilities within the System.	
Origin	Client	
Fit criterion	The System must display a list of Tickets associated with the User, ensuring visibility of all related Tasks.	
Priority	Must	

Table 2.2: Functional requirement regarding the accessibility of User-associated Tickets.

Requirement #: **FR3** Type: Functional *Use cases* #: 2

Description	**The System must display the Tickets to the User with a color coding that reflects their priority level.**
Rationale	Color-coding issues based on priority ensures that Users can quickly identify and address the most critical Tasks, improving efficiency and focus.
Origin	Client
Fit criterion	The System must use distinct colors to represent different priority levels in the issue list.
Priority	Must

Table 2.3: Functional requirement regarding the visual representation of issues based on their priority.

Requirement #: **FR4** Type: Functional *Use cases* #: 3

Description	**Each Ticket contains at least one Task.**
Rationale	Each Ticket must have at least one Task because Tasks are the only means to assign work.
Origin	Client
Fit criterion	Each Ticket must contain at least one Task.
Priority	Must

Table 2.4: Functional requirement regarding the organization of Tickets and Tasks.

Requirement #: **FR5** Type: Functional *Use cases* #: 3

Description	**After the User selects a Ticket, the System must list the Tasks associated with the Ticket, ordered by their due date in ascending order (earliest first).**
Rationale	Ordering Tasks by due date helps Users prioritize their work and focus on the most urgent Tasks first.
Origin	Client
Fit criterion	The System must ensure that all Tasks associated with a Ticket are displayed in ascending order of their due dates, with Tasks without due dates appearing at the end of the list.
Priority	Must

Table 2.5: Functional requirement regarding the ordering of Tasks by due date.

Requirement #: **FR6** Type: Functional *Use cases #*: 7

Description	**The User must be able to create a request for a Ticket in the System.**
Rationale	Allowing the User to create Ticket requests ensures that all problems, enhancements, or other needs are logged systematically and can be addressed efficiently.
Origin	Client
Fit criterion	The System must provide a feature to create a new issue request, including mandatory fields such as description, asset group, tenant related to the issue.
Priority	Must

Table 2.6: Functional requirement regarding the ability to create an issue request in the System.

Requirement #: **FR7** Type: Functional *Use cases #*: 7

Description	**When creating a Ticket request, the User must be able to associate devices and photos relevant to the Ticket.**
Rationale	Providing the ability to attach devices and photos ensures better contextual information for the issue, facilitating accurate assessment and resolution.
Origin	Client
Fit criterion	The System must allow Users to select associated devices from the list of the asset group selected and upload photos related to the issue when creating the request.
Priority	Must

Table 2.7: Functional requirement ensuring Users can provide additional context by attaching devices and photos to an issue request.

Requirement #: **FR8** Type: Functional *Use cases #*: 5

Description	**The User can view all Tasks associated with a Ticket, but can only edit the Tasks assigned to them.**
Rationale	This restriction ensures that Tasks are managed properly, allowing Users to modify only the Tasks they are responsible for.
Origin	Client
Fit criterion	The System must ensure that the User can only edit the Tasks assigned to them, preventing changes to Tasks that are not their responsibility.
Priority	Must

Table 2.8: Functional requirement regarding the viewing and editing of Tasks associated with a Ticket.

Requirement #: **FR9**	Type: Functional	*Use cases* #: 2

Description	**The System must provide filters to allow Users to view only their Tasks or all Tasks, as well as to filter by open or closed Tasks.**
Rationale	Providing these filters enables Users to easily navigate and manage Tasks by displaying only relevant items, improving efficiency and user experience.
Origin	Client
Fit criterion	The System must include filtering capabilities for Tasks, allowing Users to toggle between viewing their own Tasks and all Tasks. Additionally, the System must allow filtering Tasks based on their status (open or closed).
Priority	Must

Table 2.9: Functional requirement regarding Task filtering capabilities in the System.

Requirement #: **FR10**	Type: Functional	*Use cases* #: 3

Description	**A Task consists in a Facility Manager description, Maintenance Worker description, time slots, list of materials, list of devices, and optionally, photos.**
Rationale	Defining the composition of a Task ensures it includes all necessary information for effective communication, planning, and execution of maintenance activities.
Origin	Client
Fit criterion	The System must ensure that each Task includes fields for a Facility Manager description, Maintenance Worker description, time slots, list of materials, list of devices, and an optional field for photos.
Priority	Must

Table 2.10: Functional requirement regarding the standard composition of Tasks in the System.

Requirement #: **FR11** Type: Functional *Use cases* #: 3

Description	**The User must be able to view the descriptions, time slots, list of materials, list of devices, photos, start/stop the timer, and add or remove time slots.**
Rationale	Allowing the User to access and manage Task details, including the ability to control time slots and the timer, enhances the flexibility and usability of the System for efficient Task management.
Origin	Client
Fit criterion	The System must enable the User to view all Task details (descriptions, time slots, list of materials, list of devices, and photos). It must also allow the User to start/stop the timer and add or remove time slots as needed.
Priority	Must

Table 2.11: Functional requirement regarding the detailed interaction and management of Tasks by the User.

Requirement #: **FR12** Type: Functional *Use cases* #: 5

Description	**The User must be able to edit the Task's Maintenance Worker description.**
Rationale	Providing the ability to edit the Maintenance Worker description ensures that Task details remain accurate and up-to-date, improving communication and task management.
Origin	Client
Fit criterion	The System must allow the User to modify the Maintenance Worker description field within a Task to ensure the information reflects current requirements or changes.
Priority	Must

Table 2.12: Functional requirement regarding the editability of specific Task details by the User.

Requirement #: **FR13** Type: Functional *Use cases* #: 5

Description	**The system must enable users to manage materials by providing features for selecting predefined materials from a service (SAP), manually adding unlisted materials, and editing associated materials for tasks.**
Rationale	To ensure a usage of known materials and to give the possibility of adding materails not available on the external service.
Origin	Client
Fit criterion	The System must allow the User to add materials to a task via a external service.
Priority	Must

Table 2.13: Functional requirement regarding the adding of materials to a task.

Requirement #: **FR14** Type: Functional *Use cases* #: 5

Description	**The System must present a list of available devices in the Issue's Asset Group, provided by the IoT platform, allowing the User to select devices to associate with a Task and edit the devices associated with their Tasks.**
Rationale	To ensure that the devices presented to the user exists in the selected asset group.
Origin	Client
Fit criterion	The System must allow the User to add or remove devices from the task.
Priority	Must

Table 2.14: Functional requirement regarding the editing of devices to a task.

Requirement #: **FR15** Type: Functional *Use cases* #: 6

Description	**The User must be able to mark their Tasks as completed.**
Rationale	For the user to mark their work as completed to signal de Facility Manager to review the completion of the task.
Origin	Client
Fit criterion	The System must allow the User to mark the task as completed.
Priority	Must

Table 2.15: Functional requirement regarding the completion of a task.

Requirement #: **FR16**	Type: Functional *Use cases* #: 3
Description	**The System must define and support the following possible states for Tasks: Pending, Assigned, In Progress, Pending Signatures, Resolution Validation, Failed Closed, and Success Closed.**
Rationale	Defining these distinct Task states ensures a clear workflow and allows Users to track the progress and current status of each Task accurately.
Origin	Client
Fit criterion	The System must implement and display Tasks with the following states: Pending: Task is created but not yet assigned; Assigned: Task is assigned to a User but not started; InProgress: Task is actively being worked on; PendingSignatures: Task is waiting for required signatures; ResolutionValidation: Task is under final validation to confirm resolution; FailedClosed: Task was closed unsuccessfully and SuccessClosed: Task was completed successfully.
Priority	Must

Table 2.16: Functional requirement regarding the definition and tracking of Task states in the System.

Requirement #: **FR17**	Type: Functional *Use cases* #: 6
Description	**Before completing a Task, the User must provide a description of the problem resolution.**
Rationale	By providing a description of the problem resolution before completing a Task serves multiple purposes, all of which contribute to operational efficiency, accountability, and quality assurance.
Origin	Client
Fit criterion	The System must ensure that the Maintenance Worker provides a description of the problem resolution before marking a Task as complete.
Priority	Must

Table 2.17: Functional requirement regarding ensuring that the Maintenance Worker provides a description of the problem resolution.

Requirement #: **FR18**	Type: Functional	*Use cases* #: 5
Description	**The User shall not be allowed to edit a Task once it has been marked as "closed".**	
Rationale	Restricting edits to closed Tasks ensures data integrity and prevents unauthorized changes to completed or finalized information.	
Origin	Client	
Fit criterion	The System must disable editing capabilities for any Task that has been marked as "closed", ensuring that such Tasks remain immutable.	
Priority	Must	

Table 2.18: Functional requirement regarding the restriction of editing capabilities for closed Tasks.

Requirement #: **FR19**	Type: Functional	*Use cases* #: 6
Description	**Tasks that require a client signature must be signed by the client after the User completes and closes the Task.**	
Rationale	Requiring a client signature for specific Tasks ensures accountability and serves as confirmation that the Task has been satisfactorily completed.	
Origin	Client	
Fit criterion	The System must prompt for and record the client's signature for designated Tasks after the User marks the Task as completed and closed. The Task cannot be finalized without the client's signature when required.	
Priority	Must	

Table 2.19: Functional requirement to ensure the inclusion of client authorization for certain completed Tasks.

Requirement #: **FR20**	Type: Functional	*Use cases* #: 4
Description	**A User assigned to a Ticket's Tasks must have the capability to communicate with all users who have access to that ticket through the chat feature.**	
Rationale	This allows the Maintenance Workers to collaborate, share information, and coordinate their efforts efficiently.	
Origin	Client	
Fit criterion	The System must enable real-time and asynchronous communication between Maintenance Workers assigned to Tasks within an Issue, ensuring that messages are delivered reliably and accessible to all relevant Maintenance Workers at any time.	
Priority	Must	

Table 2.20: Functional requirement regarding the communication in issues.

Requirement #: **FR21** Type: Functional *Use cases* #: 4

Description	**The User must be able to upload and share files in the chat associated with a Ticket.**
Rationale	The ability to share files in the chat enhances collaboration and provides a means to share additional context, such as documents, images, or other relevant materials.
Origin	Client
Fit criterion	The System must allow Users to attach and send files in the chat, supporting file types such as PDFs, images, and documents, with a maximum file size of 10 MB.
Priority	Must

Table 2.21: Functional requirement regarding file sharing in the chat feature.

Requirement #: **FR22** Type: Functional *Use cases* #: 8

Description	**The User must be able to receive notifications when he is assigned to a Task.**
Rationale	Providing notifications for Task assignments ensures that Users are promptly informed about their responsibilities, improving task awareness and response time.
Origin	Client
Fit criterion	The System must generate and deliver a notification to the User whenever they are assigned to a Task, ensuring the notification includes relevant Task details.
Priority	Must

Table 2.22: Functional requirement to enhance communication and awareness through notifications for Task assignments.

2.2 Non Functional Requirements

Requirement #: **NFR1** Type: Operational

Description	**The MWA must be developed as a cross-plataform app using PWA.**
Rationale	To be functional across multiple platforms and devices.
Origin	Client
Fit criterion	The application must work on any platform or device.
Priority	Must

Table 2.23: Non-functional requirement regarding the client's request for the application's architecture.

Requirement #: **NFR2**	Type: Maintenance and Support
Description	**The System’s frontend must be developed using the Nuxt.js framework built on Vue.js.**
Rationale	The Nuxt framework is used in other applications already developed by DTx.
Origin	Client
Fit criterion	The application must have a frontend developed in Nuxt.
Priority	Must

Table 2.24: Non-functional requirement regarding the client’s request for the framework used in the frontend.

Requirement #: **NFR3**	Type: Maintenance and Support
Description	**The System’s database must be built using PostgreSQL.**
Rationale	PostgreSQL DBMS is used in other applications already developed by DTx.
Origin	Client
Fit criterion	PostgreSQL DBMS must be used in the development of the application’s logic.
Priority	Must

Table 2.25: Non-functional requirement regarding the client’s request for the DBMS used in System’s database

Requirement #: **NFR4**	Type: Maintenance and Support
Description	**The System’s logic must be developed using .NET 8.0.**
Rationale	.NET 8.0 technology is used in other applications already developed by DTx.
Origin	Client
Fit criterion	.NET 8.0 technology must be used in the development of the application’s logic.
Priority	Must

Table 2.26: Non-functional requirement regarding the client’s request for the technology used in the development of the application’s logic.

Requirement #: **NFR5**	Type: *Performance*
Description	**The application must be available 99% of the time.**
Rationale	To ensure the System's integrity, fairness, reputation, and efficiency.
Origin	Team
Fit criterion	The System must be reliable, minimizing failures and downtime.
Priority	Must

Table 2.27: Non-functional requirement regarding the application's availability.

Requirement #: **NFR6**	Type: *Maintenance and Support*
Description	**The System architecture must implement the clean code architecture principles.**
Rationale	To ensure the System's maintainability, testability, scalability and Flexibility
Origin	Client
Fit criterion	The Clean Code Principles must be applied whenever is possible.
Priority	Must

Table 2.28: Non-functional requirement regarding the application's architecture.

Requirement #: **NFR7**	Type: Maintenance and Support
Description	**The System must have an interface allowing for easy and intuitive navigation across all functionalities.**
Rationale	The
Origin	Client
Fit criterion	The System ensures that Users of all backgrounds, including those who are unfamiliar with the System, can interact with it effectively, therefore eliminating the need for previous training, minimizing the learning curve and reducing on-boarding time and associated costs.
Priority	Must

Table 2.29: Non-functional requirement regarding the client's request for the application's interface.