



**REMODEL - Robotic tEchnologies
for the Manipulation of cOmplex
Deformable Linear objects**

Deliverable 3.2 – User Interface

Project acronym: REMODEL

Project title: Robotic tEchnologies for the Manipulation of cOmplex Deformable Linear objects

Grant Agreement No.: 870133

ObjectsTopic: DT-FOF-12-2019

Call Identifier: H2020-NMBP-TR-IND-2018-2020

Type of Action: RIA

Project duration: 48 months

Project start date: 01/11/2019

Work Package: WP3

Lead Beneficiary: TAU

Authors: TAU, UNIBO, TECNALIA

Dissemination level: Public

Contractual delivery date: 30/06/2023

Actual delivery date: 28/06/2023

Project website address: <https://remodel-project.eu/>

Table of contents

1.	Task description: T3.2 User interface	2
2.	Introduction	3
3.	Partners surveys	3
4.	User Interface design and structure	4
5.	User Interface features	8
5.1.	Multirole security clearance	8
5.2.	Dynamic modules launching and monitoring	9
5.3.	Sensor data visualization	10
5.4.	REMODEL Safety Manage (RSM) integration	10
5.5.	CAD Platform files update	11
5.6.	Trajectory/routine recording and execution	12
5.7.	Robot manual motion control	13
5.8.	Robot production process control and monitoring	14
5.9.	System and process configuration	15
5.10.	Simulation stream visualization	15
6.	User Interface implementation	16
7.	Conclusion	16

1. TASK DESCRIPTION: T3.2 USER INTERFACE

Leader: **TAU**; Participants: **UNIBO, IEMA, TECNALIA, ELIMCO, PUT, ELVEZ, VW, ENKI**

The User Interface system will be composed of a touch screen multimedia device and will be designed to enable the operator to teach the robot about the task to be performed by composing macro-actions that can also be taught to the robot when new procedures enter in the manufacturing process. Another helpful element will be the possibility to monitor the advance of the assembly process. In this way, at the end of the assembly, an activity report can be produced. Furthermore, at system runtime, the precise knowledge of the state of the assembly process allows the project designer to decide about possible process modifications and the impact on the product manufacturing in response to a customer request. Wearable solutions will be investigated, including the use of a 10-inch tablet, that the user can wear attached to the forearm to allow a simple interaction and usage without creating any obstacle to the other worker activities. The possibility of editing which part of the manufacturing can be executed by the robot from the task plan generated during the post-processing of the product design (T3.1) will be considered.



To TRL 4: Laboratory tests will be executed to evaluate the usability by involving specialized workers in suitable surveys. How to specialize the user interface for the different use cases will also be evaluated.

To TRL 5: Integration with the other system components and the design chain carried out by the companies and validation against their design procedures. Execution of communication tests with the CAD platform and the robotic system in the industrial environment.

To TRL 6: Demonstration of the User Interface in the design procedure carried out by the companies. Execution of tests considering timing and constraints of the real production environment. Specialized workers will be involved in suitable surveys to evaluate the interface usability. The control by the user of the robot task sequence will be tested.

2. INTRODUCTION

Task 3.2 aims to develop an interface that allows the users to interact with the REMODEL system in an easy and intuitive manner. For the development of this User Interface (UI), comprehensive surveys were conducted among the project partners to analyze and gather the necessary requirements. Based on this analysis, a versatile and scalable UI framework was designed, which can be used for all the use cases, and a wide range of features and functionalities were developed, which can be incorporated into the generic framework to fulfill the specific requirements of every use case. Some of these functionalities are: process execution control, real-time process monitoring, sensor value tracking, safety status monitoring, system configuration, dynamic launching of system modules, and the ability to record new robot trajectories. Moreover, the UI was developed as a web-based interface, which offers widespread accessibility without the need for specific software installation, enabling users to conveniently interact with the system using standard web browsers.

The deliverable aims to summarize in a concise manner, the development of the REMODEL User Interface. It begins by analyzing the UI requirements for the different use cases, followed by the presentation of the design and structure of the generic UI framework. Subsequently, the document introduces all the developed UI features, which can be incorporated into the UI framework. Finally, the implementation of the UI in the different use cases of the project is documented. The above-mentioned information is sequentially documented in the subsequent sections of the report.

3. PARTNERS SURVEYS

Three surveys were conducted among the REMODEL partners at different stages of the project, to understand the requirements for the UI in each of the use cases. The first survey was carried out in May 2020, at the start of Task 3.2. The purpose of this initial survey was to determine the functionalities required for the UI. All the partners participated in this survey and obtained results are summarized in Figure 1. Based on these responses, an initial UI was developed with a basic implementation of some of the most popular features (see Figure 3).

In October 2020, a second survey was carried out to gather feedback on the initial UI, with active participation from all partners. The results of this survey closely aligned with those obtained in the first survey. Moreover, after the analysis of the responses, it became evident that each use case had distinct requirements for the UI. Therefore, the decision was to develop a generic and scalable framework for the UI, which incorporates some of the functionalities that are common for all the use cases (e.g., multirole access or page navigation menu), and a list of specific features which can be easily integrated within the generic framework for every use case implementation.

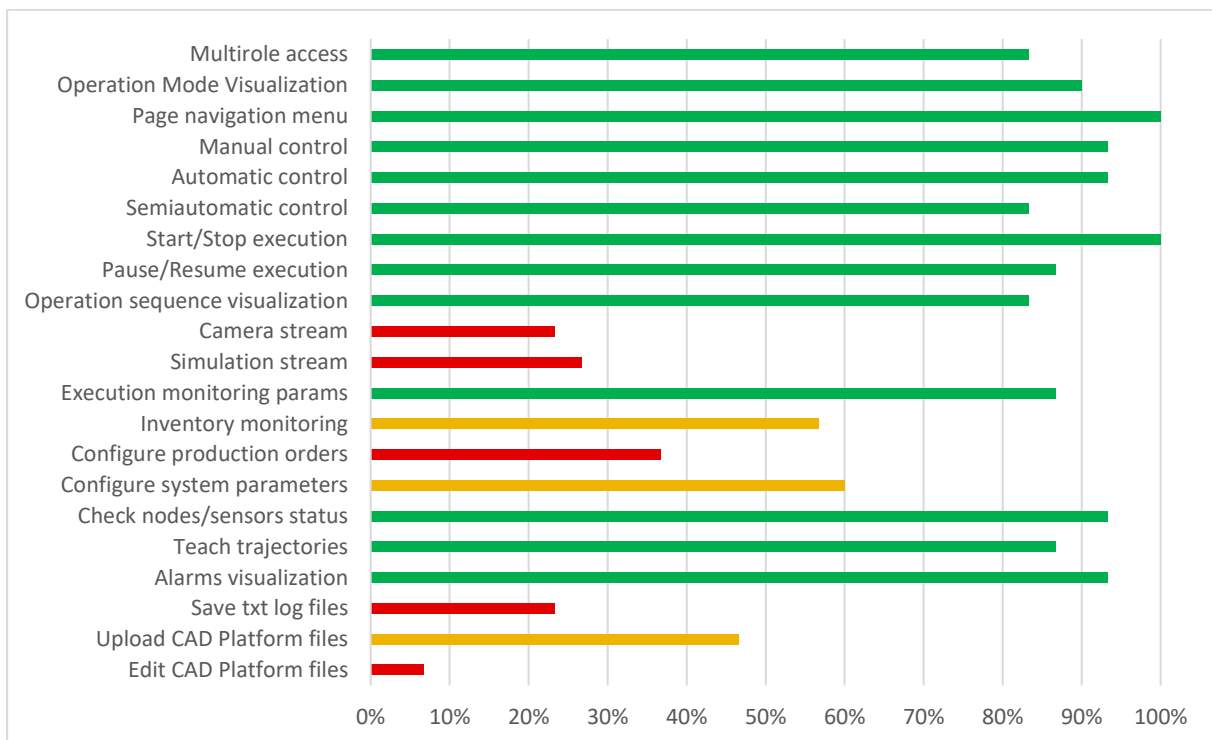


Figure 1. Results of the first survey to determine the required functionalities for the UI. The results show the percentage of votes received by each functionality. Different colors are used to highlight the popularity of each feature (Green: $\geq 70\%$, Yellow: $>40\%$ and $<70\%$, Red: $\leq 40\%$).

Finally, the last survey was conducted in January 2023 with the objective of gathering feedback on a mockup of the final user interface prior its development. This survey allowed to obtain input and insights from the project partners regarding the proposed design and layout of the UI. Based on the received responses, the final UI design was refined and subsequently initiated for development.

4. USER INTERFACE DESIGN AND STRUCTURE

As introduced in Section 3, the REMODEL UI is composed of a scalable framework that is common for its implementation in the different use cases, and a list of features that can be integrated into this generic framework depending on the specific requirements of each use case. This section documents the common UI framework, while the following (Section 5) introduces all the developed features.

The REMODEL UI has been designed as a web-based platform, allowing users to conveniently access and interact with the system through standard web browsers. By adopting a web-based approach, users are not constrained by specific software installations or device limitations, enabling them to seamlessly access the UI from a variety of devices and operating systems. To achieve this, the UI has been developed utilizing the core web technologies of HTML, CSS, and JavaScript. HTML provides the structural framework for organizing the UI elements, CSS enhances the visual presentation and styling, while JavaScript adds interactivity and dynamic functionality, empowering users with a rich and engaging experience.

To allow communication between ROS and JavaScript, ROSBridge and ROSJS were employed. ROSBridge serves as a bridge between the ROS environment and non-ROS applications by utilizing a WebSocket interface. This enables the establishment of a bidirectional communication channel between ROS and JavaScript. To achieve this, the ROSBridge server needs to be running on the ROS side, while on the JavaScript side, the ROSJS library is utilized. ROSJS offers JavaScript client libraries that provide a set of functions and APIs to interact with ROS. These libraries allow JavaScript to subscribe to ROS topics, publish messages, call ROS services, and receive responses. An example of the communication between the web-based UI and the REMODEL ROS system is represented in Figure 2.

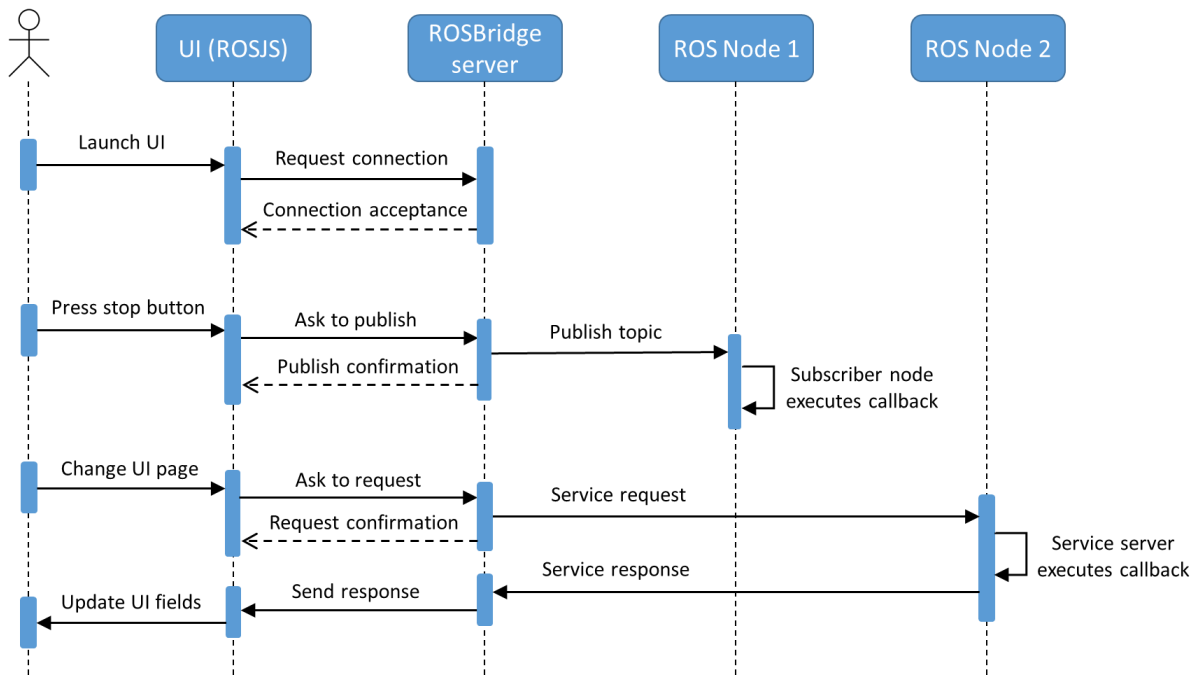


Figure 2. UML sequence diagram of the communication between the web-based UI and the REMODEL ROS system using ROSBridge and ROSJS. To emphasize the communication between the UI and ROS, the communication between the ROS nodes and the ROS Master is not represented in the diagram.

Vue.js (Vue) was chosen as the framework to build the REMODEL UI due to its simplicity, flexibility, and performance. Vue can be used in two different ways, to create and control entire websites (which are known as Single-Page Applications (SPAs)) or to create

standalone widgets that control certain parts of the webpage. The first option is more powerful and versatile but it also requires more setup and configuration. Therefore, the initial UI was built adding different Vue components as widgets. This prototype was developed to test the communication between ROS and JavaScript and to evaluate the UI structure (receiving feedback from the partners in the second UI survey, as explained in Section 3). This initial implementation included some basic features, such as the control and monitoring of the robot process execution and the communication with the REMODEL Safety Manager (RSM), as can be seen in Figure 3.

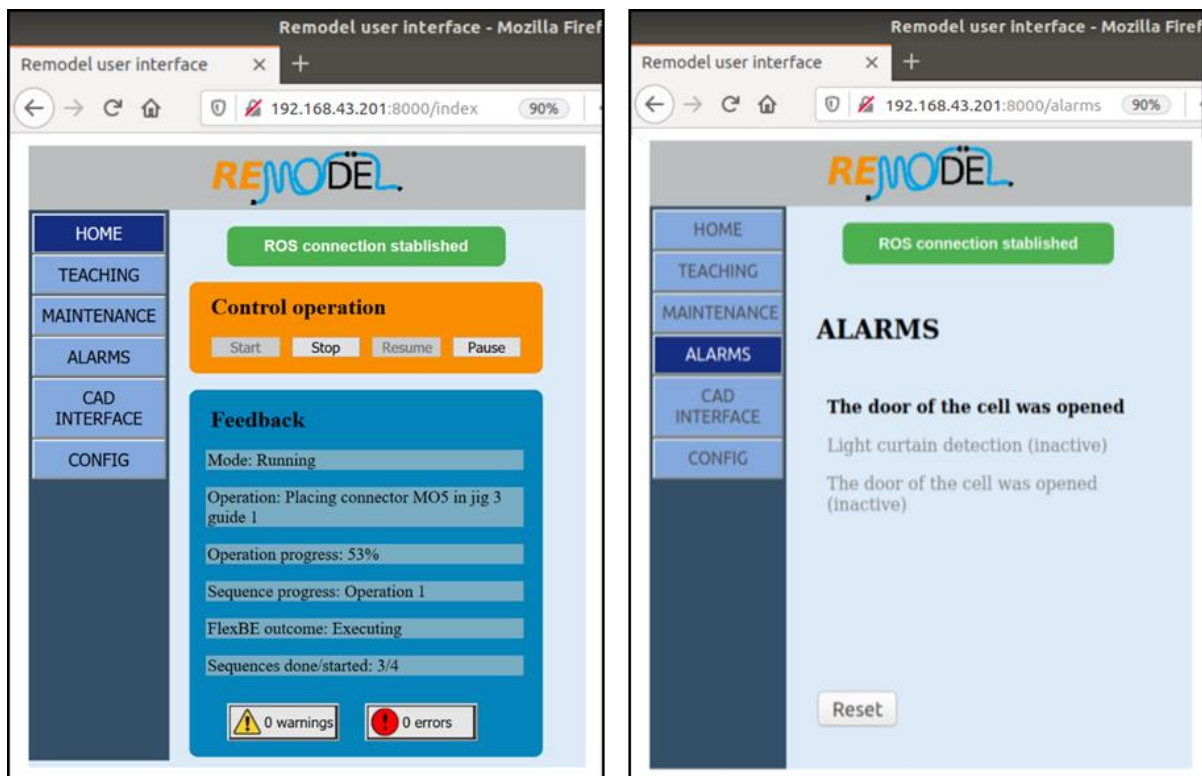


Figure 3. Initial REMODEL UI, which implemented some basic features, such as the control and monitoring of the robot process execution (left) and the communication with the RSM (right).

However, the final REMODEL UI was built as an SPA using Vue CLI (Vue Command Line Interface). An SPA, or Single-Page Application, is a web application that operates within a single web page by dynamically updating the content without the need for full page reloads. Unlike traditional multi-page applications, where each interaction typically involves a server request, SPAs provide a more fluid and interactive experience. In particular, in the final UI implementation, the browser just sends an initial request to the server, which sends back a single HTML page. After this, Vue takes control of the whole application, dynamically updating its contents whenever is required. This reduces significantly the loading times and improves the UI performance, providing a more seamless user experience.

Another notable advantage of SPAs is their modularity, as they follow a component-based development approach. This allows for code reusability, as components can be developed independently and reused across different parts of the application, resulting in faster development and easier maintenance. The structure of the REMODEL UI leverages this

methodology, reutilizing some components for all its pages. This structure can be seen in Figure 4 and it is composed of the following elements:

- **Side menu:** Expandable Vue component composed of three elements. The first one is the a navigation menu that facilitates page switching wihtin the UI. As introduced before, the UI is built as a Single-Page Application, therefore, there is just one HTML page. However, when a new page is selected in the side menu, Vue dynamically updates this HTML page by injecting a new view component in the “Page content area”. The second element is the UI and system status monitoring block. This block provides information on the status of the ROSBridge socket communication between the UI and ROS, indicating if ROS is currently connected or not. Additionally, it displays the current operation mode of the robotic platform, which can be: idle, running, alarm, or teaching. Finally, the third element user menu, which showcases the profile picture, username and role of the current user. It also contains a logout button for the user convenience.
- **Top bar:** Vue component situated on the top of the UI, it indicates the name of the page that is currently displayed.
- **Page content area:** Area where the current page contents are displayed. The view component displayed in this area is updated by Vue when the user switches the UI page. This is done using the Vue Router library, which also updates the UI URL. As a result, to be incorporated into de UI, every page must be defined as a Vue view component.

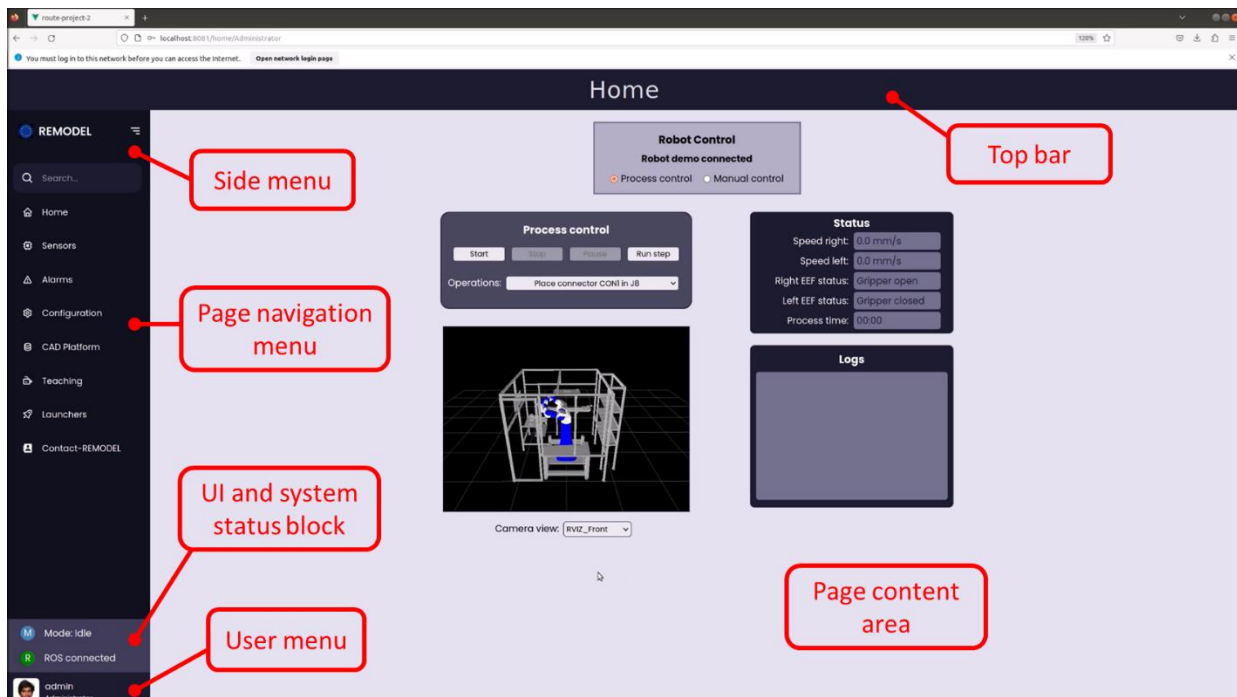


Figure 4. REMODEL UI structure and common components

5. USER INTERFACE FEATURES

This section presents the list of features developed for the UI. However, it's important to note that the UI is just the front-end component of the REMODEL system, and the implementation of these features often involved back-end development in the ROS system. Consequently, the subsequent subsections don't only cover the UI aspects but also delve into the necessary developments of ROS to enable the desired functionality. The UI demonstrator video¹ showcases the functionality of all these features, and it serves as a reference for illustrating each feature as it is described.

5.1. Multirole security clearance

The UI incorporates a robust multirole security clearance feature, enabling users to assume either an administrator or operator role, each offering distinct permissions and visibility within the system. Every user is assigned a specific role, ensuring proper access control and segregation of duties. In order to gain access to the UI, users are required to authenticate themselves via the login menu (shown in Figure 5), where they must provide their unique username and password. This login information is securely encrypted and stored in an immutable text file, guaranteeing the confidentiality and integrity of user credentials. In order to add a new user to the system, the immutable attribute of this file has to be unset (for which root privileges are required), add the user, and make the file immutable again.

In the demonstrator video (00:05-00:38), this feature is depicted, showcasing the distinct access privileges granted to the operator and administrator roles. The operator user is limited to accessing four pages, while the administrator has access to nine pages. Furthermore, the operator is only able to launch six out of the eleven available system modules, whereas the administrator has the capability to launch all of them. This clearly separates the roles of operator and administrator. The operator can only control and monitor the robot process execution and visualize the safety status of the robotic platform, while the administrator is also able to configure the system through the UI, teach the robot new trajectories, operate the robot manually, and visualize the sensor values for debugging or maintenance purposes.

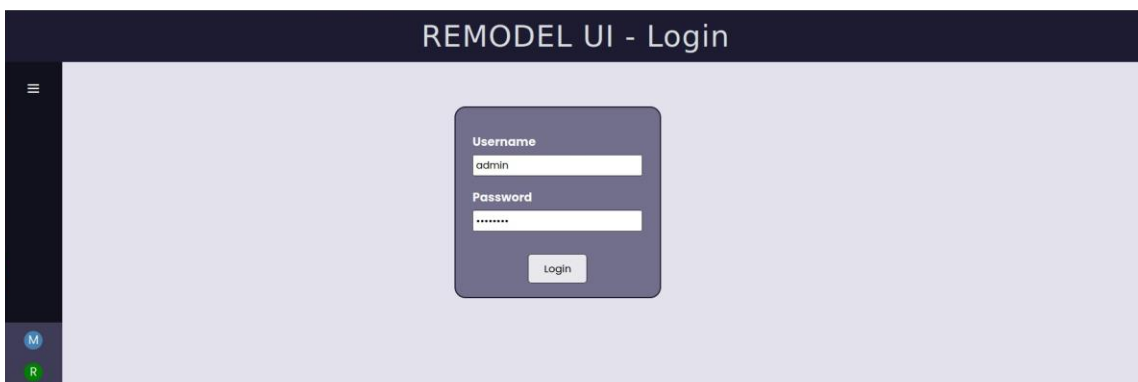


Figure 5. UI login menu

¹ <https://drive.google.com/file/d/1WU4nGZVwAfFTU2NWd4-taugbxnt6E0eL/view?usp=sharing>

5.2. Dynamic modules launching and monitoring

The REMODEL system is composed of a significant number of modules, which add different functionalities to the system. Although the system can work with all the modules being constantly active, this is not efficient, as it is consuming unnecessary resources and can cause slowdowns. To avoid this, the UI allows the user to dynamically launch and stop the different system modules, depending on the functionalities required at each time. However, the user is still able to launch and kill all the system modules at once with the “All” option at the bottom of the page. Additionally, this feature allows monitoring the status of each module with a status indicator at the side of the launch and stop buttons. The color of this indicator indicates the module status, and it can be:

- **Green:** Active module. All its associated nodes are active.
- **Gray:** Inactive module. None of its associated nodes are active.
- **Orange:** Launching/stopping module. The module is in a transitory state. Either it has been launched but not all its associated nodes are active yet, or it has been stopped but not all its associated nodes are inactive yet.
- **Red:** Failed/incomplete module. Some of its associated nodes haven’t been launched or have died after the module was launched.

To achieve this functionality, a ROS service to launch and kill launch files has been created, which is called when clicking the correspondent UI buttons. Moreover, the “Launchers” page of the UI is checking every second the list of active ROS nodes, and it uses this information to update the status of all the modules. Finally, the design of this feature has prioritized reconfigurability, enabling the list of modules to be easily updated by making modifications to a single list.

In the demonstrator video (00:38-01:06 and 04:07-04:17) this feature is shown, launching and stopping some of the system modules depending on the required system functionalities. Additionally, the visual appearance of the “Launchers” page is shown in Figure 6.

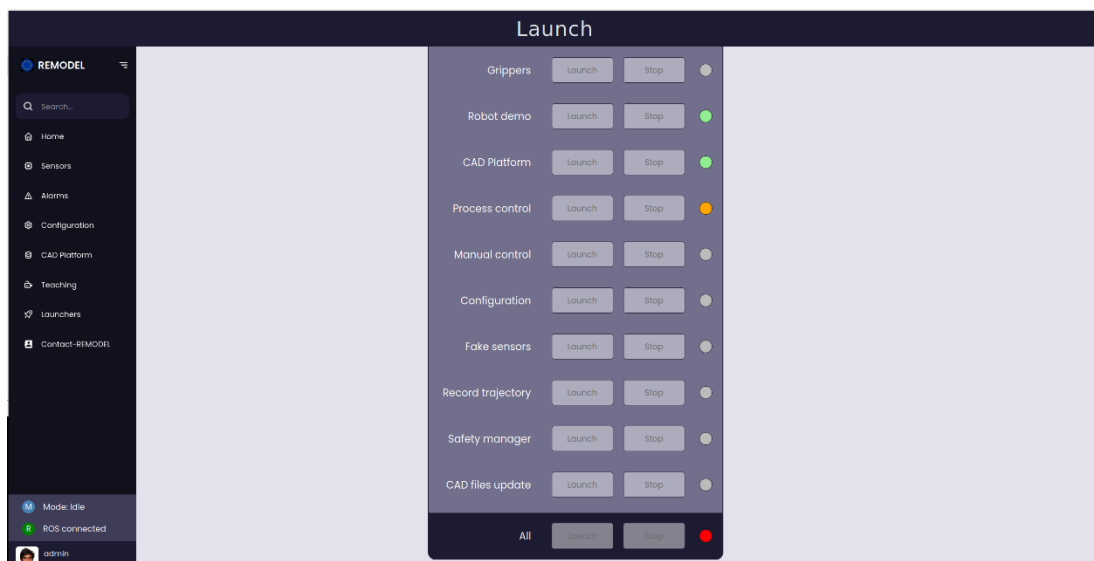


Figure 6. UI “Launchers” page. All the Launch and Stop buttons are disabled because the “Process control” node is being launched.

5.3. Sensor data visualization

The data captured by the various system sensors is visually presented on the “Sensors” UI page, utilizing the Chart.js JavaScript library. An example implementation has been developed for the force and tactile sensors. In the case of the force sensors, the detected forces and torques are displayed over time, while for the tactile sensors, the estimated shape of the grasped cable is updated with a frequency of 10 Hz. This feature is easily reconfigurable enabling the creation of new sensor representations with minor modifications to the existing examples. Furthermore, the UI gives provides flexibility in terms of displayed data, allowing users to choose the sensors they wish to visualize.

For testing purposes, a ROS node that publishes fake sensor data with a frequency of 10 Hz has been developed. The visualization of this fake data in the UI can be seen in the demonstrator video (05:17-05:43) and in Figure 7.



Figure 7. UI sensor data visualization.

5.4. REMODEL Safety Manage (RSM) integration

The RSM is the ROS node in charge of managing the safety of the REMODEL robotic platforms, and it allows the bidirectional communication between the physical safety devices and the rest of the modules of the ROS system (For more details, please refer to Deliverable 2.6). The feature presented in this section enables bidirectional communication between the RSM and the UI, empowering the UI to showcase the platform’s safety status, display active and inactive alarms, and provide the capability to reset alarms directly from the UI. This communication happens through ROS topics. The functionality of this feature is shown in the demonstrator video (09:32-10:02 and 11:55-12:08) and in Figure 8.

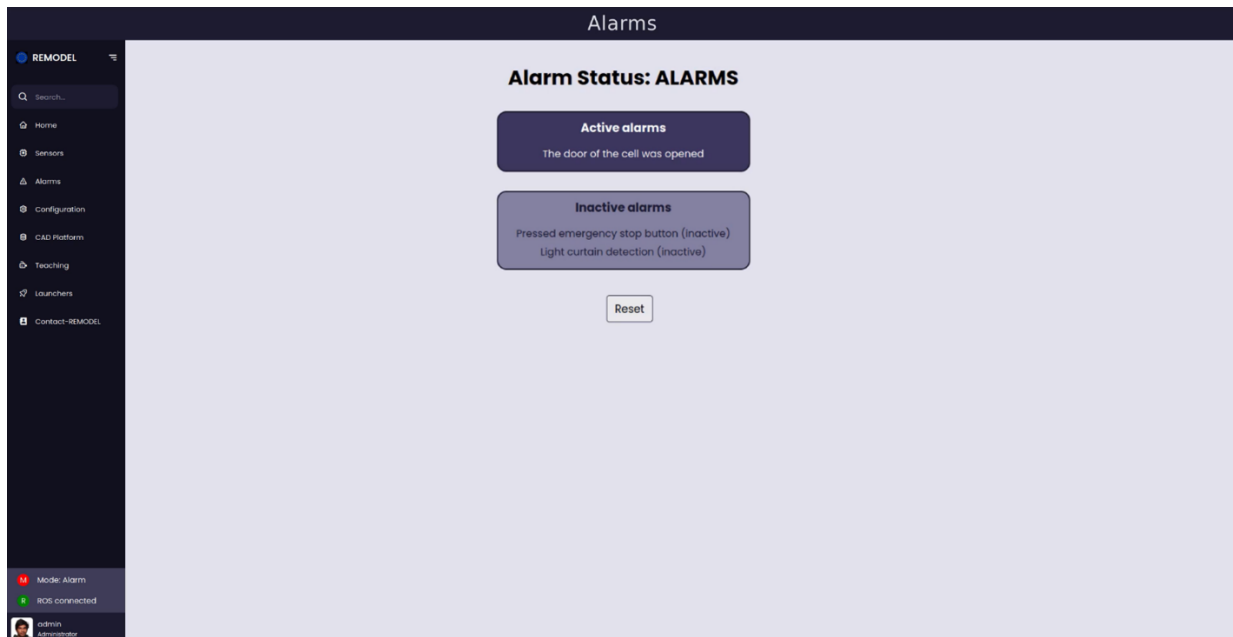


Figure 8. UI Alarms visualization.

5.5. CAD Platform files update

The REMODEL project aims to develop highly reconfigurable robotic systems for the manipulation of Deformable Linear Objects (DLOs). To achieve this, all the robot trajectories and actions are configured based on the physical hardware layout and geometry, the properties of the handled objects, and the list of operations to perform. All this information is defined with a set of input files of different formats and natures, including CAD 3D files, 2D assembly drawings, and text files. The CAD Platform is the REMODEL module that extracts useful information from these input files and makes it available for the rest of modules. As a consequence, the whole system can be reconfigured just by updating these input files.

The feature presented in this section makes the system reconfiguration even more intuitive, allowing users to directly load new files to the CAD Platform module from a USB flash drive via the UI. This is done using three dropdown menus, as depicted in Figure 9. The first menu is used to select the USB device containing the new files, the second menu enables the selection of the specific file (or folder), and the last menu is utilized to designate the CAD Platform input file (or folder) that will be replaced, displaying only files of the same format as the new file. Additionally, the UI contains a message panel, which informs users about the outcome of their file update actions. The operation and implementation of this feature can be observed in the demonstrator video (04:02-05:05).

To accomplish this functionality, a dedicated ROS node was created, incorporating several service servers to interact with the connected devices and the computer storage system. The UI calls these services to populate and update the information displayed in the three dropdown menus, and to replace the specified CAD Platform files with the desired from the USB drive.

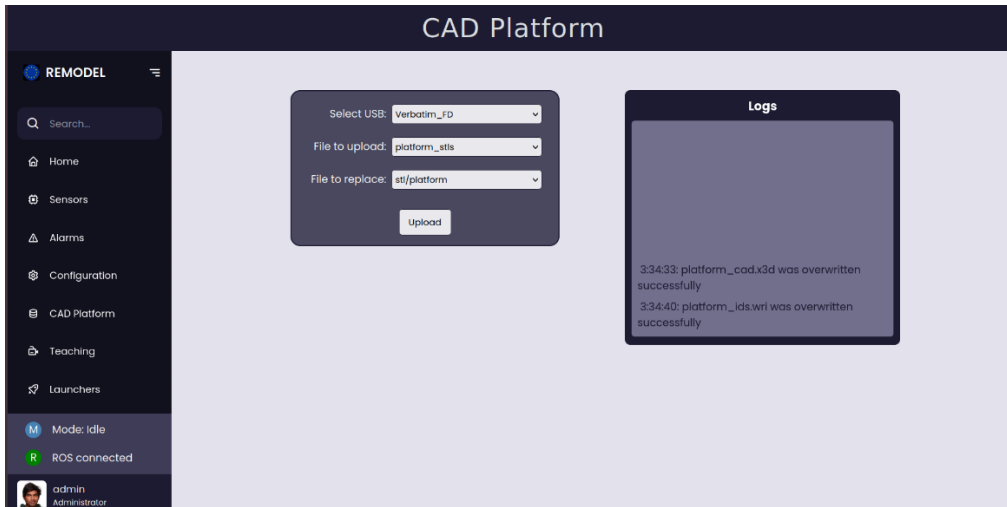


Figure 9. CAD platform files update from the UI

5.6. Trajectory/routine recording and execution

This feature allows to teach the robot new routines from the UI. These routines are defined as a sequence of waypoints that the robot must follow, and commands for the end effector. The UI assists the user during this process making it very intuitive. As can be observed in Figure 10, this functionality is provided by three elements: the “Record trajectory” menu, which manages the recording of new robot trajectories and routines; the “Execute trajectory” menu, which can be used to execute or delete existing routines; and a message panel, which displays feedback about the actions performed. A demonstration of this feature can be observed in the demonstrator video (07:35-09:12).

A ROS node has been developed to provide the UI with this functionality. This node communicates with the UI through topics, to save a sequence of robot poses and end effector actions in a file. This feature is agnostic to the specific method used to move the robot to new poses, whether it be through gravity compensation or teleoperation. Moreover, this node offer some services to get the existing trajectories, and to execute or delete a certain trajectory.

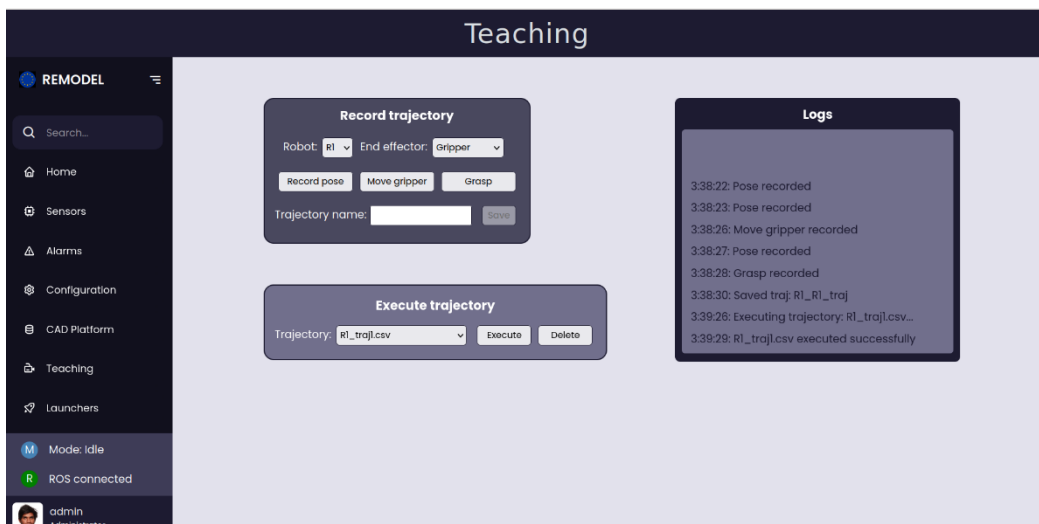


Figure 10. Robot's trajectory/routine recording from the UI

5.7. Robot manual motion control

This feature allows for manual operation or movement of the robot through the use of the UI. It provides three options for controlling the robot, which are facilitated by the interaction between the UI and a dedicated ROS node. The first option allows the user to move the robot to a predefined configuration. As can be seen in Figure 11 (right), to do this the user must select the robot group and its target configuration from two dropdown menus. These menus are populated automatically when the page is loaded, requesting the data to the manual control ROS node through a service. This service extracts the information from the SRDF robot configuration file and sends it to the UI.

The second option, depicted in Figure 11 (left), allows for relative movements in the Cartesian axes, specifying the displacements in millimeters and the rotations in degrees. Finally, the third option enables the user to move the robots' end effectors to absolute poses in the Cartesian space. The current position and orientation of each end effector are displayed, with the information obtained from the manual control ROS node through a service. Furthermore, for all three options, users can specify the maximum speed and acceleration limits. When the "Move" button is clicked, a service of the manual control ROS node is called, which utilizes MoveIt functions to move the robot to the specified target pose or configuration.

In addition to robot movement, this feature also enables users to actuate the robot's end effectors and change the robot's tools. Figure 11 shown an example of this functionality for two different tools: parallel grippers and a taping gun. With the parallel grippers, users can open, close, or adjust the finger distance to a specific value, all through the UI. Similarly, the taping gun can be actuated directly from the UI. This functionality is achieved by interacting with the ROS system using services. Both the robot manual motion and the end effector's actuation through the UI are performed in the demonstrator video (10:58-12:47).

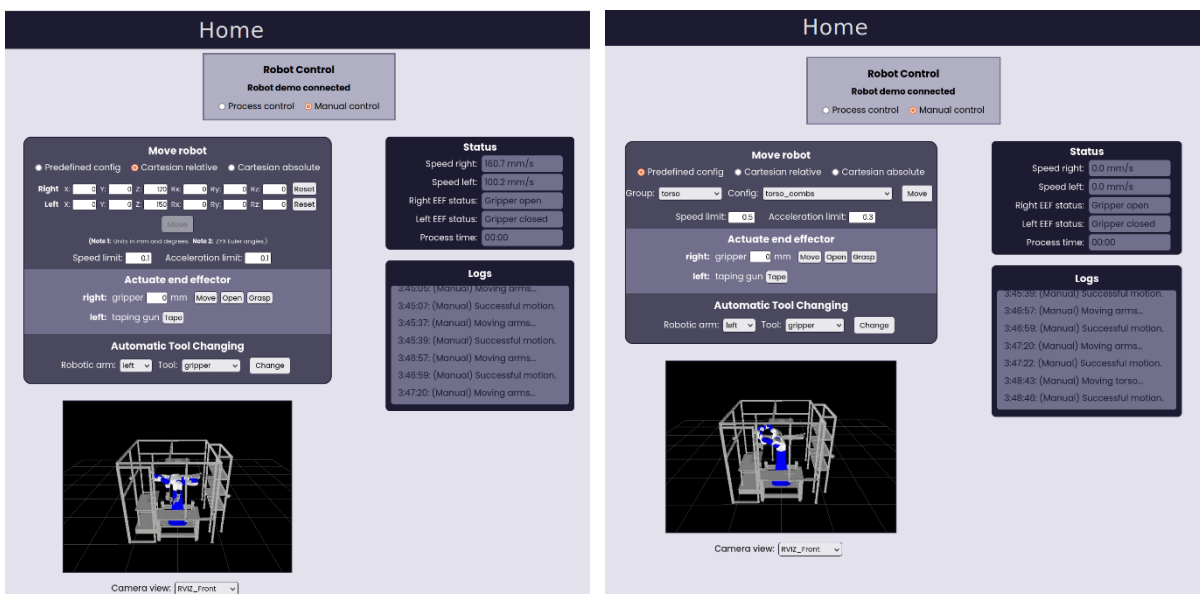


Figure 11. Manual control of the robot through the UI, executing cartesian relative movements (left) or moving to predefined robot configurations (right).

5.8. Robot production process control and monitoring

This feature enables the user to control and monitor the execution of the robot production process. This process can be executed either in automatic or semi-automatic mode. The automatic mode is initiated with the “Start” button, which can be seen in Figure 12, and the robot starts executing the entire sequence of operations. Two other buttons can be used to control the robot execution within this mode, “Stop” and “Pause/Resume”. Both “Stop” and “Pause” buttons stops the robot movement, however, “Stop” resets the process (i.e., it has to be re-started from the initial operation), while “Pause” can be resumed from the point where the process was stopped (clicking on the “Resume” button). To achieve this, the ROS node that manages the process execution must have flag indicators after each movement to register the last executed movement before the process was paused. On the other hand, the semi-automatic mode is initiated with the “Run Step” button, executing only the operation selected in the dropdown menu and stopping after that. Both execution modes can be observed in action in the demonstrator video (01:17-02:48).

Two ROS nodes were defined for managing the robot process execution. The first node contains an action server, which generates and executes the robot trajectories, and the second contains an action client, that either calls the action server with different goals, or cancels it. This second node is the one that interacts directly with the UI through different topics, acting as an intermediary between the UI and the action server.

Furthermore, the UI receives feedback from both of these nodes, which is then displayed in two separate message panels. The first panel shows the status of the system, updating the values of different fields in real-time, such as the robot cartesian speed, or the total process execution time. On the other hand, the second panel spawns informative messages continuously, such as the success or failure of the executed movements.

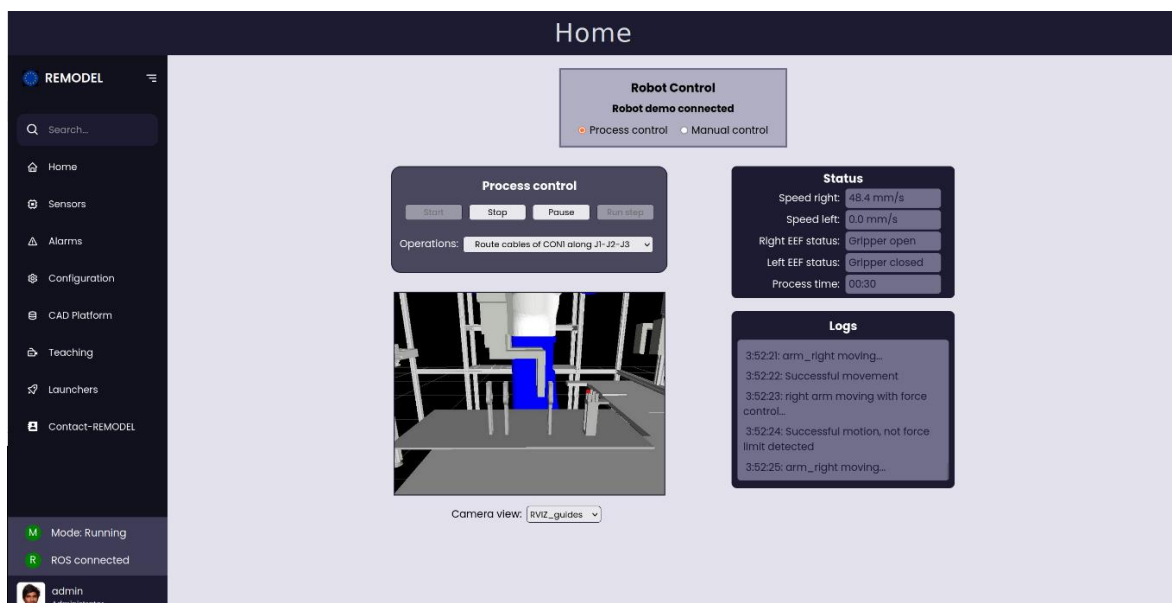


Figure 12. Robot production process control and monitoring through the UI

5.9. System and process configuration

Upon initialization of the ROS node that manages the robot production process execution, it retrieves crucial information from a configuration file. This includes parameters such as the robot speed, offset distances, or different threshold sensor values. This information is used to generate the robot trajectories and adjust its actions. As a result, the process execution can be easily configured by simply updating the values within this file. The feature presented in this section enables users to directly visualize and edit these parameters from the UI (as depicted in Figure 13), significantly simplifying the process configuration for the user. The demonstrator video (05:50-06:12) provides a visual demonstration of the utilization of this feature.

The design of this feature has emphasized its reconfigurability, allowing for easy modification of the displayed configuration fields by updating in a single dictionary variable. This approach enables the seamless implementation of the feature in different system, regardless of variations in the configuration fields.



Figure 13. Configuration of the robot production process through the UI

5.10. Simulation stream visualization

This feature allows to visualize in the UI the robotic platform environment within the RVIZ simulation environment from different view points. To achieve this, different CameraPubs must be added to the RVIZ environment in the desired locations. Each of these cameras publish the captured images in a ROS topic with a certain frequency. By subscribing to these topics, the UI is capable of streaming the robot simulation. Two different points of view of the simulation stream can be seen in Figure 11 (a global view of the robotic cell) and Figure 12 (a view focused on the workbench).

6. USER INTERFACE IMPLEMENTATION

As introduced in Section 3, the REMODEL UI, is composed of a versatile and adaptable framework that is common for all the use cases (UCs), and a list of features that can be integrated into this framework for its implementation in each use case. This allows the UI to have a common structure while still being customizable to meet the unique requirements of each use case. Table 1 provides an overview of the features that have been implemented in the UI for UC1, 2.1, 2.2, and 4. The REMODEL UI was not implemented in UC3 due to the internal policies of Volkswagen, which require to use a uniform and standardized UI design in all the robotic cells of the factory.

Table 1. Features incorporated in the UI for its implementation in the different use cases of the project.

Feature	UC1 (IEMA)	UC2.1 (ELIMCO)	UC2.2 (ELVEZ)	UC4 (ENKI)
Multirole security clearance	X	X	X	X
Dynamic modules launching and monitoring	X		X	X
Sensor data visualization	X		X	
RSM integration (alarms visualization)	X	X	X	X
CAD Platform files update	X		X	
Trajectory/routine recording and execution				X
Robot manual motion control		X	X	X
Robot production process control	X	X	X	X
System and process configuration	X	X	X	X
Simulation stream visualization	X	X	X	X

7. CONCLUSION

This deliverable documents the User Interface developed for the REMODEL project. The UI has been designed as a web-based platform that allows the intuitive interaction between users and the REMODEL ROS system. Moreover, special emphasis has been placed on making the UI reconfigurable to facilitate its integration with the different use cases. To enable this, the REMODEL UI consists of a generic and scalable framework to which different features can be added depending on the specific requirements of each use case.

The UI has been successfully integrated into UC1, 2.1, 2.2, and 4, requiring only minor modifications. However, integration with UC3 was not possible due to internal Volkswagen policies. Additionally, this deliverable includes a link to a demonstrator video, which shows all the developed UI features and its integration into different use cases.