IN2DREAMS

Contract No. 777596

INtelligent solutions 2ward the Development of Railway Energy and Asset Management Systems in Europe

D4.3: Proof of Concept of a Smart Contract

Report of Technical Demonstrator related to the Proof of Concept
Task T4.3 Development and testing of smart contracts solution

DUE DATE OF DELIVERABLE: M21 - 31/05/2019

ACTUAL SUBMISSION DATE: 27/05/2019

Leader/Responsible of this Deliverable: Nadia Fabrizio - Cefriel

Reviewed: Y

Document status		
Revision	Date	Description
1	30/01/2019	First Issue of the TOC
2	1/04/2019	Cefriel added Sections 1,2,3,4 and 5
3	15/04/2019	Cefriel Concluded Sections 1,2,3,4,5 and added 6
4	05/05/2019	Cefriel Added sections 7,8 and concluded 6
5	14/05/2019	UNIGE and UKON revised the report.
6	15/05/2019	Cefriel updated after UNIGE and UKON revision and added conclusion
7	27/05/2019	Final Version after TMT approval and Quality Check

Project funded from the European Union's Horizon 2020 research and innovation programme		
Dissemination Level		
PU	Public	X
CO	Confidential, restricted under conditions set out in Model Grant Agreement	
CI	Classified, information as referred to in Commission Decision 2001/844/EC	

Start date of project: 01/09/2017 - Duration: 24 Months

Executive Summary

The present report **Proof of Concept of a Smart Contract** is the presentation of the technical demonstrator (TD) on blockchain, and includes: the technical description of the selected scenario (asset maintenance) in term of actors, data and workflows; the component and functional description of the TD; the installation and user guidelines.
The presented solution is the one of the two outputs of Task *4.3 Development and testing of smart contracts*, the second will be a further document containing technical testing, on-site testing and analysis of results (strengths and weakness) as impacting the Intelligent Asset Management Strategies (IAMS)' (TD 3.8).[1]
The statement of work for the TD is:
Manage the maintenance jobs workflow through the employment of smart contracts, automatically enforcing the rules and clauses (like SLA) of the maintenance contracts between the IM and the Contractor.
The selected scenario resulted as outcome of previous activities of WS2 (in WP2) and is described (included selection criteria) in D2.1 IN2DREAMS Services, Use Cases and Requirements. The presented TD is a TRL 3 level ([5]. It will be then integrated at demonstrative level, in order to be able to show an end-to-end scenario, with the other components developed in WP 4 and WP5. The impact of the TD is to enable a new approach to Asset Maintenance, rethinking the infrastructure and developing a cross-supplier and cross vertical system. In the TD all nodes of the IM network are synchronized with the same data, the system is duplicated and fault tolerant and SC logic can control consistency of state changes with preset rules. The TD has been developed over Fabric Hyperledger Project and the Smart Contracts Decentralized Applications have been developed in Go. The selected technology, included the data exchange model, have already been explained in Deliverable D4.1 The Data Transactions model in railways ecosystems as result of other activities conducted in WP4 for WS2. The TD is related to other TD of WP5 (explained in D5.1 Data Analytics Scenarios and D5.3 Visual Analytic of Railway Data and Models): all the TD developed in WS2 are based on the same scenario selected in WP2.

During the 4.3 Task, related to TD, the following activities have been conducted:

1. Description of IMs workflows with interviews on the field;
2. Mapping of the workflow into the functional requirements;
3. Installation of a permissioned blockchain open source network based on Hyperledger Fabric;
4. Description of the TD in term of functional requirements (integrating the business with the technical, also related to the ability of deploy testing);
5. Developing the TD (blockchain back bone + smart contracts).

.

[1] IAMS is whole-system approach for the asset management that will employ, collect and process the data sources provided by the TD3.6 and the TD3.7 in Shift2RAIL H2020 program.

Abbreviations and Acronyms

Abbreviation	Description
AMS	Asset Management System
CEFRIEL	Cefriel (IN2DREAMS WS2 Partner)
CCC	Command and Control Centre
EU	European Union
SC	Smart Contract
EVOLUTION ENERGIE	Evolution Energie (IN2DREAMS Partner)
IM	Infrastructure Manager
IP	Innovation Programme
KPI	Key Performance Indicator
KUL	Katholieke Universiteit Leuven (IN2DREAMS WS2 Partner)
POC	Proof of Concept
RFI	Rete Ferroviaria Italiana (IN2DREAMS WS2 Partner)
TMS	Traffic Management System
TO	Train Operator
TRL	Technology readiness level
UKON	University of Konstanz (IN2DREAMS WS2 Partner)
UNIGE	University of Genoa (IN2DREAMS WS2 Partner)
VA	Visual Analytics
WP	Work Package
WS	Work Stream
TD	Technical Demontrator
TRL	Technology Readiness Level
SC	Smart Contract
DLT	Distributed Ledger Technology

Table of Contents

List of Figures

List of Tables

1 Background

This document contains the guidelines of the TD ”D4.3: Proof of Concept of a smart contract” as results of Task 4.3. Previous activities in WP4 conducted to the analysis of scenario[2] Asset Maintenance (selected in WP2) and the definition of the technical aspects and platform selection (retrievable in D4.1).
The **objective** is to develop a TRL 3 -technical demonstrator (from now, the **TD**) about a decentralized system about the Asset Maintenance-Manage maintenance jobs workflow through the employment of smart contracts, automatically enforcing the rules and clauses (like SLA) of the maintenance contracts between the Infrastructural Manager (IM) and the Contractors. Other TD are developed in WP5, related to the same scenario of Asset Maintenance (for details see other deliverables of WS2, 5.1 and 5.3).
As-is in the asset maintenance context has been analyzed through interviews on the field conducted on RFI premises. The resulting frameworks is made up by several processes, actually characterized by being based on heterogeneous systems (mixing digital and paper based), requiring a lot of human effort during the authorization steps, or the reconciliation phase. The entire process is really time consuming, but the time and effort reduction are critical due to the need to maintain an elevated level of safety, error-free environment with optimal risk control, while optimizing and automating processes. As recognized also in other sector and experiences ([3], the adoption of decentralized approaches like DLT and blockchain way may be a good improvement for Cost reduction, Increase process transparency and accountability of maintenance costs, while assuring a certain level of safety and security (the workflow cannot be modified, the steps are checked with no single point of failure, and so on). Anyway a detailed discussion will be conducted in the deliverable of 4.3 task (next of this report). The nature of railway networks is to be large, and split into regions of different sizes, which make it complex to manage in a centralised (and secure) way. In general, applying a decentralisation concept may help to simplify the process and help its digital transformation, while maintaining a certain level of security and control in the process.
In our case, IM management system has been modelled into two main levels of accountability, accordingly to the evidence collected on the field (mainly RFI premises):

- management of the railway infrastructure;
- relationship between IM and all the suppliers involved in the maintenance.

The TD has been developed considering these two separate levels. Ultimately, TD should help both main aspect of IM, on one side being a single point for tracking the process of maintenance, audits, reconciliation and financial budgeting, on the other to track the suppliers works and internet overcoming the heterogeneous patchwork of region-specific systems. After release of the TD, lab test benches will be launched for i) collecting feedback on the site and for ii) testing performance. Related to i), on RFI premises, a supervised demo will be conducted with operators and an interview to collect feedback and results for giving a quantitative evaluation of the solution. Related to ii), since the adopted technology is recent, it will be tested the TD in term of potential transaction speed, data limits, verification processes.

2 IMs actors and workflows for asset maintenance

This section summarizes the results of the work conducted to define an high-level picture of the Asset Maintenance derived from norms, rules, practices and administrative processes. Our references have been primarily

[2] Asset Maintenance is the business high level scenario selected in WP2. For uniformity with the other deliverable, we have continued to call it scenario, even though withing the scope of the description of the TD, it is better to refer to Asset Maintenance as the high level use case grouping a series of ”scenarios” that is the exact user stories related to the TD

the procedures ([8], [6] and [7]) actually governing the planning, execution and audit of the maintenance jobs normally carried out as part of the Infrastructure Manager (IM) mission to maintain and improve the service level of the Railway Infrastructure.
This analysis has defined at top level (high features) the SW logic that the the Proof of Concept (PoC) on the asset maintenance shall implement through the employment of smart contracts. Due to the technical aspect of the blockchain, and since a process has to be defined before being translated into a smart contracts, only programmed maintenance has been considered. The maintenance jobs carried out to quickly respond to emergency situations are not part of the analysis as they generally don't follow a standard procedure and are not in the scope of the TD. The following elements are derived from several face-to-face interviews with Rete Ferroviaria Italiana S.p.A. (RFI) personnel and with an analysis of the already cited norms, rules and procedures ([1], [2], [3]) employed in RFI, which is the IM member of IN2DREAMS. RFI is a relevant stakeholder in Europe and we can assume that the evidences and results of the conduct analysis on the field can be considered relevant at European level. In fact RFI is responsible for more that 80 % of Italian Railway Infrastructures. Considering that every European country is subjected to a series of European Directives aiming at harmonizing and homogenizing the Railway sector in all of Europe, the validity of the claims related to the workflows described in this document may be considered valid inside the Shift2RAIL context. In subsections it is presented the overview of the actors involved, and the administrative workflows Activity Diagram of normally employed when dealing with maintenance jobs, as derived from the analysis and interviews.

2.1 Actors and Roles in the IM

The asset maintenance ecosystem is IM-centric. The IM is the main participant of the ecosystem and the one that dictates the rules, according to the current legislation: all the Contractors, indeed, have to accept and operate by the rules in order to enter and remain in the ecosystem. Obviously, while the IM is a single entity, the Contractor category is populated by multiple participants. Each of these entities, or categories, is composed of several actors that collaborate with each other to execute the procedures that permit to carry out the maintenance works ensuring the safety of the people involved and of the railway traffic, and the quality of the work itself. In the following tables, all the actors are introduced, and their role is described (see [8], [6], [7] and [9]).

IM's Role	Description
Project Representative (Referente di Progetto)	Physical person that represents RFI by all legal means, and sign the Framework Agreement with the Contractor.
Project Manager (PM)	Person in charge of the framework agreement from RFI side, that prepares all the needed "Works Specification" documents for each maintenance job executed inside the framework agreement.
Works Manager (Direttore Lavori)	Person responsible of supervising the execution of the works from RFI side Head of Maintenance Unit
Maintenance operator	RFI person that physically reach the place where the work has to be carried out, checks that all the conditions are met before letting the contractor's workers in, check that all the safety measures are taken, and that all the needed safety is in place after the work is done, before letting the train circulation restart.
CCC operator	Person responsible of rescheduling the trains affected by the works and stop/restart the circulation on the affected areas when the works are taking place.

Table 1: The IM's Roles involved in the related to Asset Maintenance and to the next Activity Diagram.

Also for contractor on site, there are different roles to consider.

Contractor's Role	Description
Contractor Representative	Physical person that represents the Contractor by all legal means.
Technical Director (Direttore Tecnico)	Counterpart of the Works Director on the Contractor side. It has all the authority to organize the works on behalf of the contractor.
Workers	Workers Who actually do the job.

Table 2: The Contractors 's Roles involved in the related to Asset Maintenance and to the next Activity Diagram.

2.2 Workflows of the asset maintenance management

As an assumption, it is considered that the scenario under analysis does not cover the procedures and workflows employed to set up calls for tenders to assign ordinary and extraordinary maintenance jobs to external companies. Nevertheless, it is important to know that, once a public tender is concluded and the external company selected, a framework agreement, built on the example shown in [2], is put in place between the IM and the Contractor. Such an agreement is the foundation for the execution of all the maintenance jobs that will be required during its validity. Maintenance jobs are periodically scheduled by the IM keeping into account a lot of different factors, like the resource availability, the effects of the train circulations interruptions, etc. Also, the Contractor itself may influence the scheduling by asking to delay or anticipate some jobs. Once a specific job has been detailed and scheduled by the PM on the IM side, a specific document called "Works specification", that must adhere to the parameters and boundaries agreed in the framework agreement, is created. For our use case purpose, limited to the life-span of the single maintenance job, not considering all the planning activities beforehand, this is the starting point of the workflow.

2.3 The maintenance job's execution

Once the procedure has been started with the issuing of a work specifications document, the job enters the execution phase. Several actors interact to carry on the process. On the IM side, the main role is represented by the Works Director: a technical role responsible of allowing the job execution (i.e., give permission to access the job area), supervising its progress and evaluating the results. Works Director's counterpart on the Contractor side is represented by the Technical Director: a figure appointed by the Contractor's governing body for managing the execution of the required works. Both the Works Director and the Technical Director collaborate to ensure that the required works are executed as intended. In the following figure, an activity diagram explains the main steps of the workflow typically followed when carrying on a maintenance job that has an impact on the train circulation. The swim lanes indicate which actor is responsible of which action. The white boxes put between the swimlanes, represent the input (or output) of a specified action and, in the current procedure, they correspond to a recorded physical document.
Each framework agreement should explicit which are the **penalties for delays** that the Contractor will suffer, usually in terms of a payment reduction, for a delay in the completion of the jobs. A delay is certified calculating the total amount of time elapsed between the certification of the permission to start the work (when the IM maintenance operator communicate officially that the contractor's workers may enter the area and start working) and the report of work completion. Of course, such a rule is valid in case there were no suspensions of the works requested by the Works Directors, in which case the works deadline shall be adjusted accordingly. Also, the IM may be held responsible of delays by the Contractor in case it does not permit the execution of the works or in case it suspends it for more than an agreed amount of time. The terms the IM has to respect are detailed in [8].
The Activity Diagram in Figure 1 is translated into a visual representation of the interactions as below (see 2)

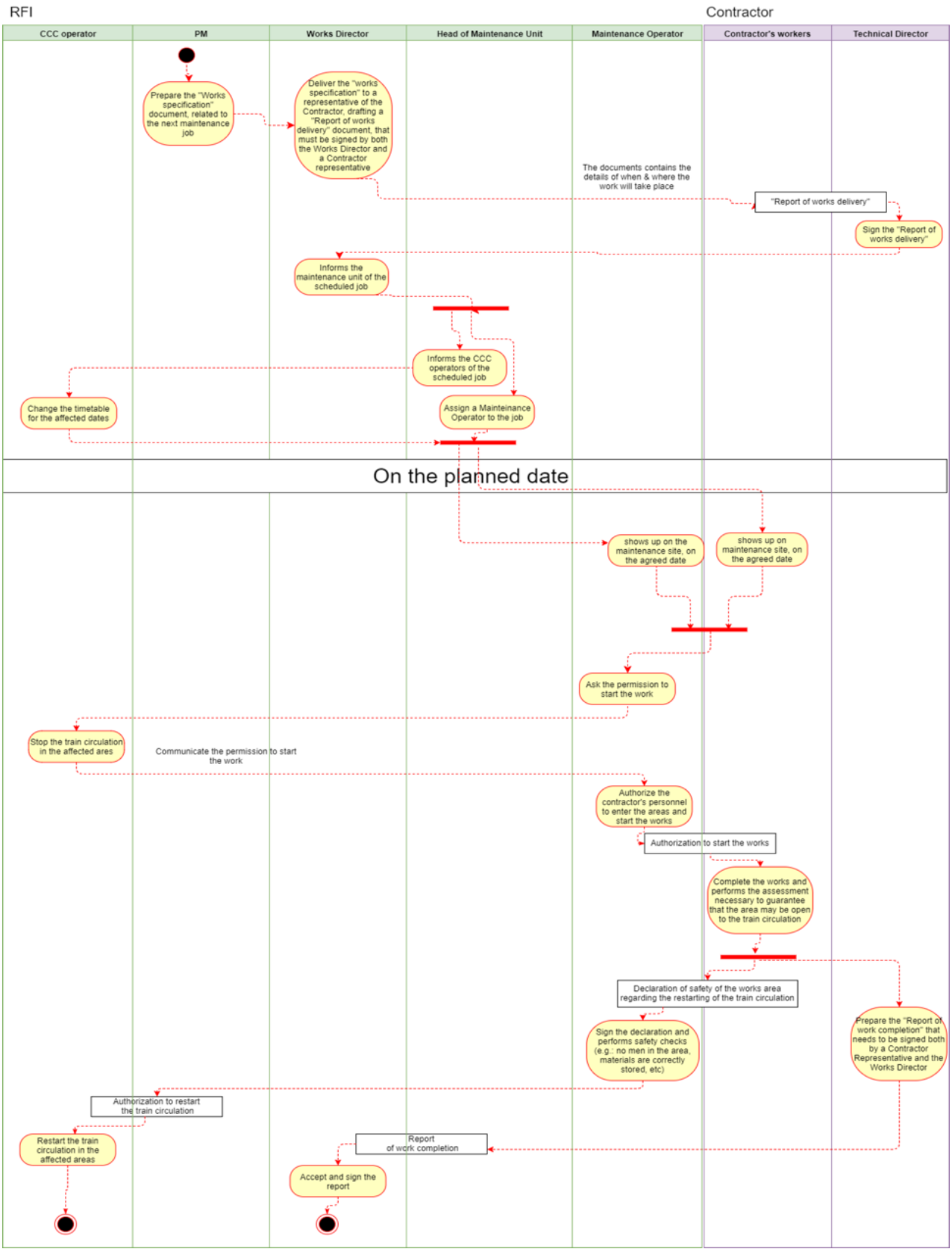


Figure 1: Activity Diagram Workflow of the Asset Programmed Maintenance

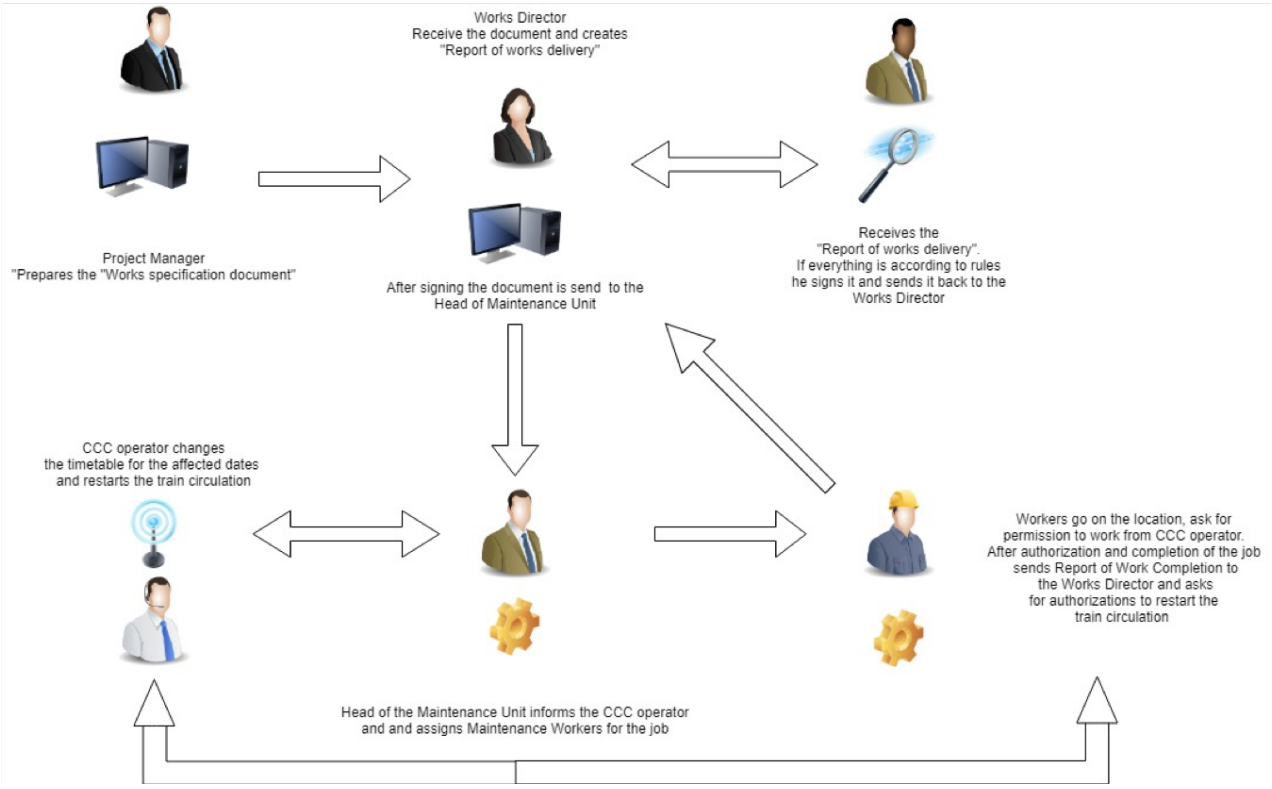


Figure 2: Simplified maintenance

The figure shows in a simplified way the interactions and processes related to Asset Maintenance and previously explained in detail in the Activity Diagram 1.

3 Architectural Design

In this chapter, it is given a synthetic explanation of the TD for the PoC in term of: components view, runtime and deployment views (included interactions), and user-system interactions with sequence diagrams. For the purpose of technical description of the TD, from now and further on, standard glossary, tools and methodologies are used (see Computer Science IEEE Standard Glossary and Methodologies ([4]) and related).

3.1 Component view and interactions

In this section, TD system is illustrated by components and their mutual interactions. All of the components are connected intro one cohesive system. The graph below will explain and illustrate the structure of the arbitrary complex system. All of the connected components are providing services that are needed by some other component. The main component of the TD is the **blockchain core ecosystem** smart contract demonstrator and it is made by the following parts:

- The Smart Contract for Work Delivery & The Smart Contract for The Work Specification Document;
- The Core Ledger;
- The Identity Card.

In order to be integrated with the machine learning and visual analytic parts, other components have been added:

- The Data Filtering;
- The Maintenance Ops Chaincode;
- The Front End APIs.

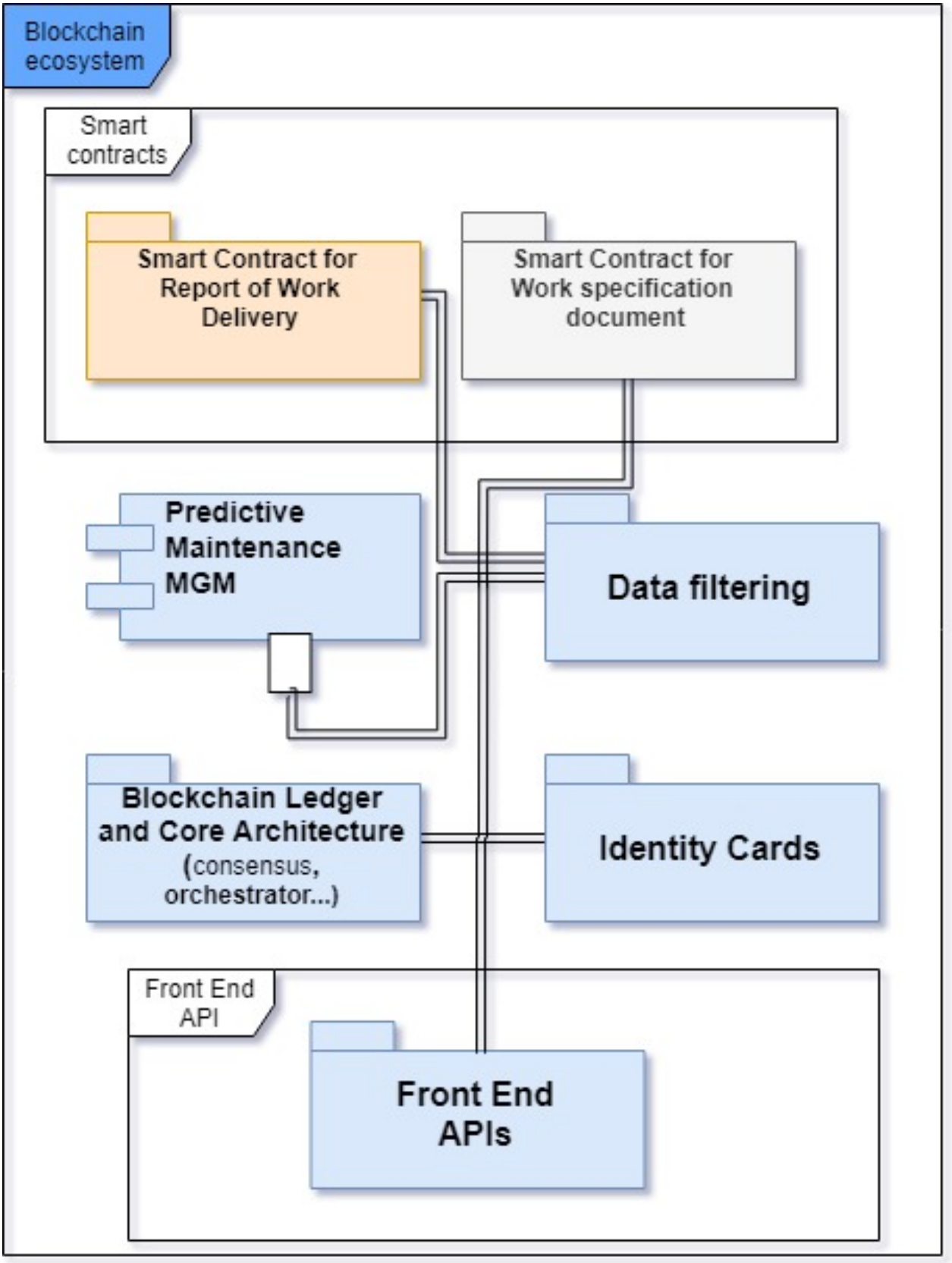


Figure 3: Components view of the TD system for the PoC

The blockchain system is represented into encapsulated, cohesive, reusable, encapsulated component units where each component has its own interface. In figure 3, it is possible to see a visual representation of the system- Each component of the system can be described as following:

- This system has two types of smart contracts one for each document. The smart contract for "Report of Work Delivery" (orange component in 3) is enforcing rules regarding the consistency and correctness of the filled data. This smart contract has the following transactions:
  -CheckROWD- function that checks if everything is consistent (correct start date, end date etc...)
  -AssignWorker- function that assigns worker that is available for specific job
  -Sign- is a function that is signing the document and time stamp is created
  -UpdateStatus- function that changes the status of the document. Possible document status: Open, Closed, Assigned, Reported.
  The other smart contract is enforcing rules for the "Work specification document" with the following transactions:
  -SafetyCheck- function that checks if safety check was performed
  -UpdateStatus-function that changes the status of the document. Possible document status: Open, Closed, Assigned, Reported.

- Predictive Maintenance management - is interacting with the prediction engine. This component is responsible for getting predicted data and showing to the workers and pushing data back to the engine for retraining the model.

- Data filtering- this component is cleaning and processing the data gathered from the prediction engine and from the smart contracts. After this is done this component sends the data to the visualization engine.

- Blockchain Ledger and Core Architecture- this is the backbone of the Hyperledger fabric.

- Identity Cards- this component includes various information's about the participants in the network. Also this component diagram is able to issue, delete or edit participant instances in the network.

- Front end APIs- This component holds all of the APIs that are needed for connecting the back end with the front end.

- Workers front end- front end dedicated for the workers that will interact with the smart contracts.

In the interactions with the frontend, any generic blockchain user-client (it may be representing any of the roles described above in 1) is able to see the front endonly and it interacts with it while every other interaction else (like membership service, certificate authority, consensus etc..) is not visible and it is done in the back end (see 5 for the runtime view).

3.2 Deployment view

This part will show the execution architecture of the blockchain system. It will show the configuration of run time processing nodes and the components that live on them.

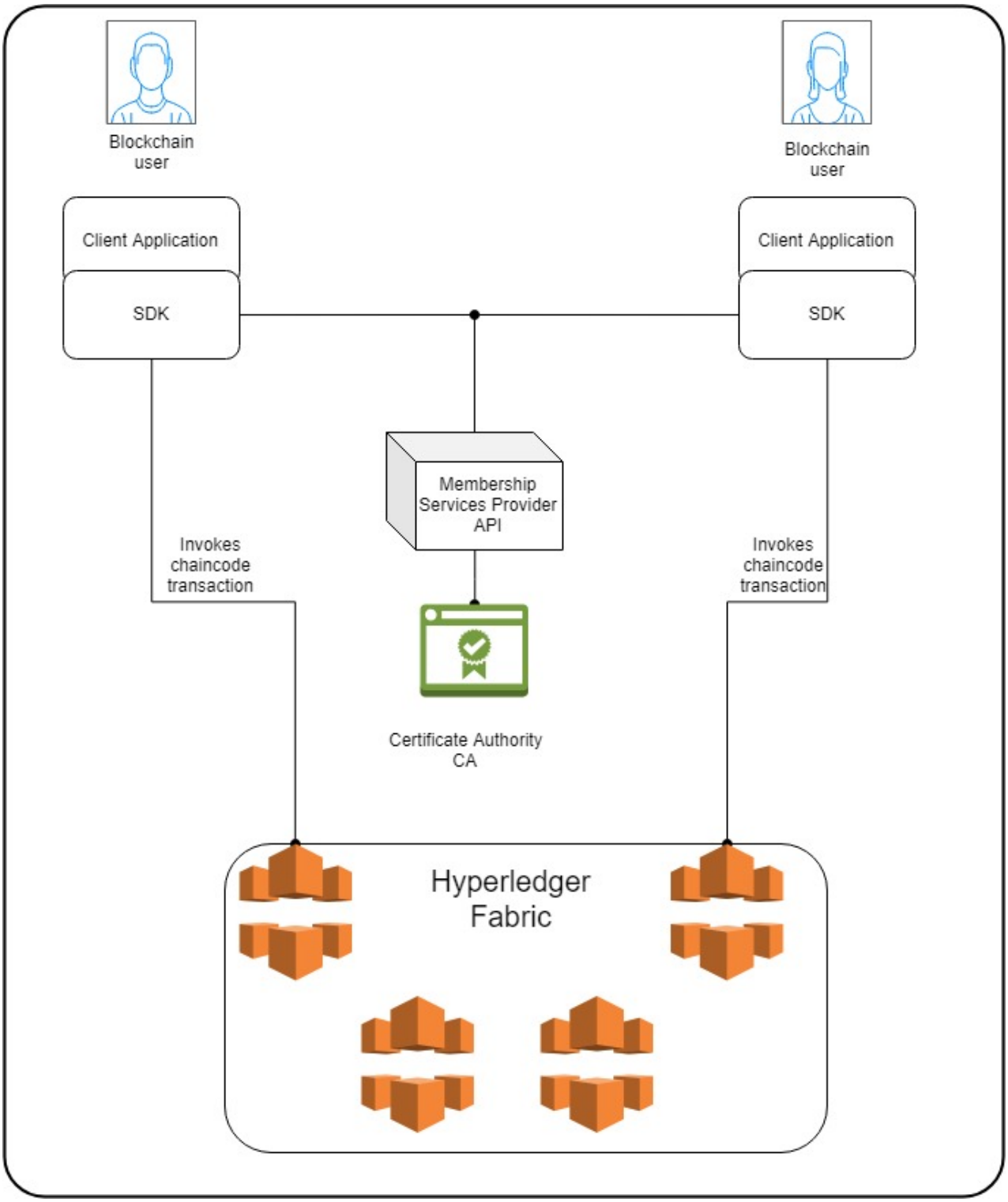


Figure 4: The runtime view of the system TD and the main interactions between frontend and generic client

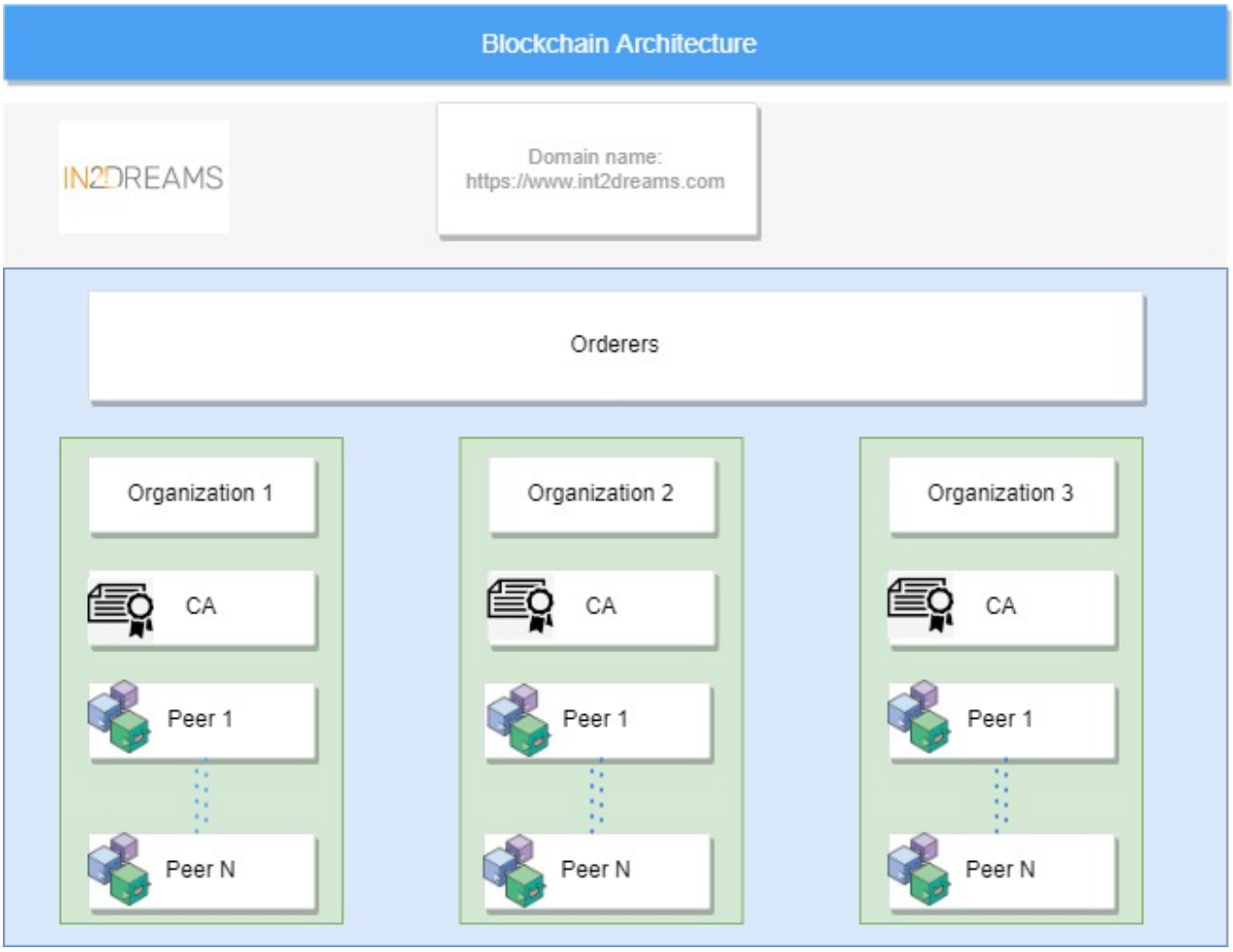


Figure 5: Blockchain Architecture

This figure explains how the deployed network of the blockchain architecture, which is the back bone of the TD running the smart contracts for the Asset Maintenance.

3.3 Sequence Diagrams

In this part runtime view of the PoC is shown. It shows concrete behavior and interactions of the systems building blocks in form of possible scenarios.

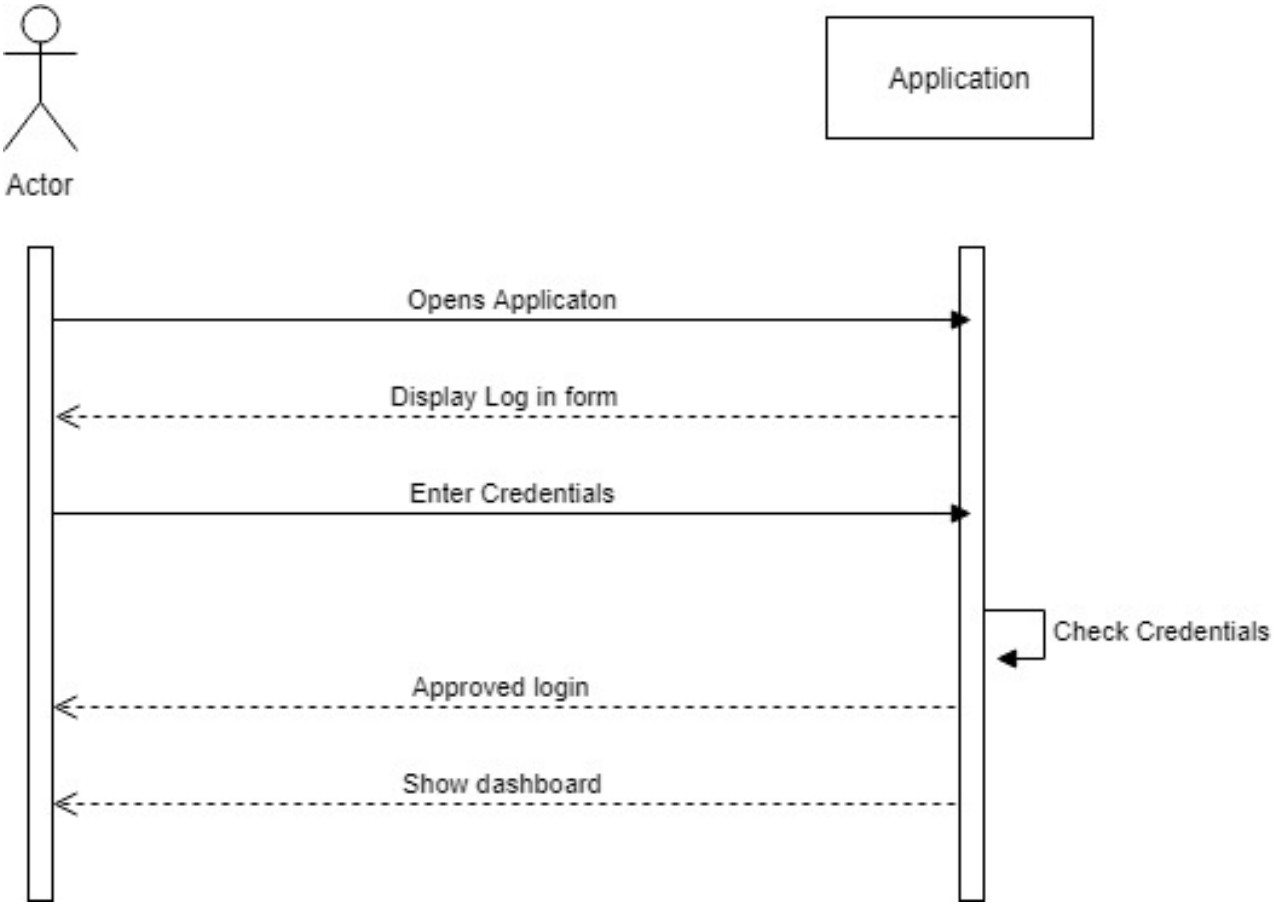


Figure 6: Login

Here is shown the procedure of login in the system. All of the actors will be provided credentials and if they provide the correct credentials they can see the appropriate dashboard.

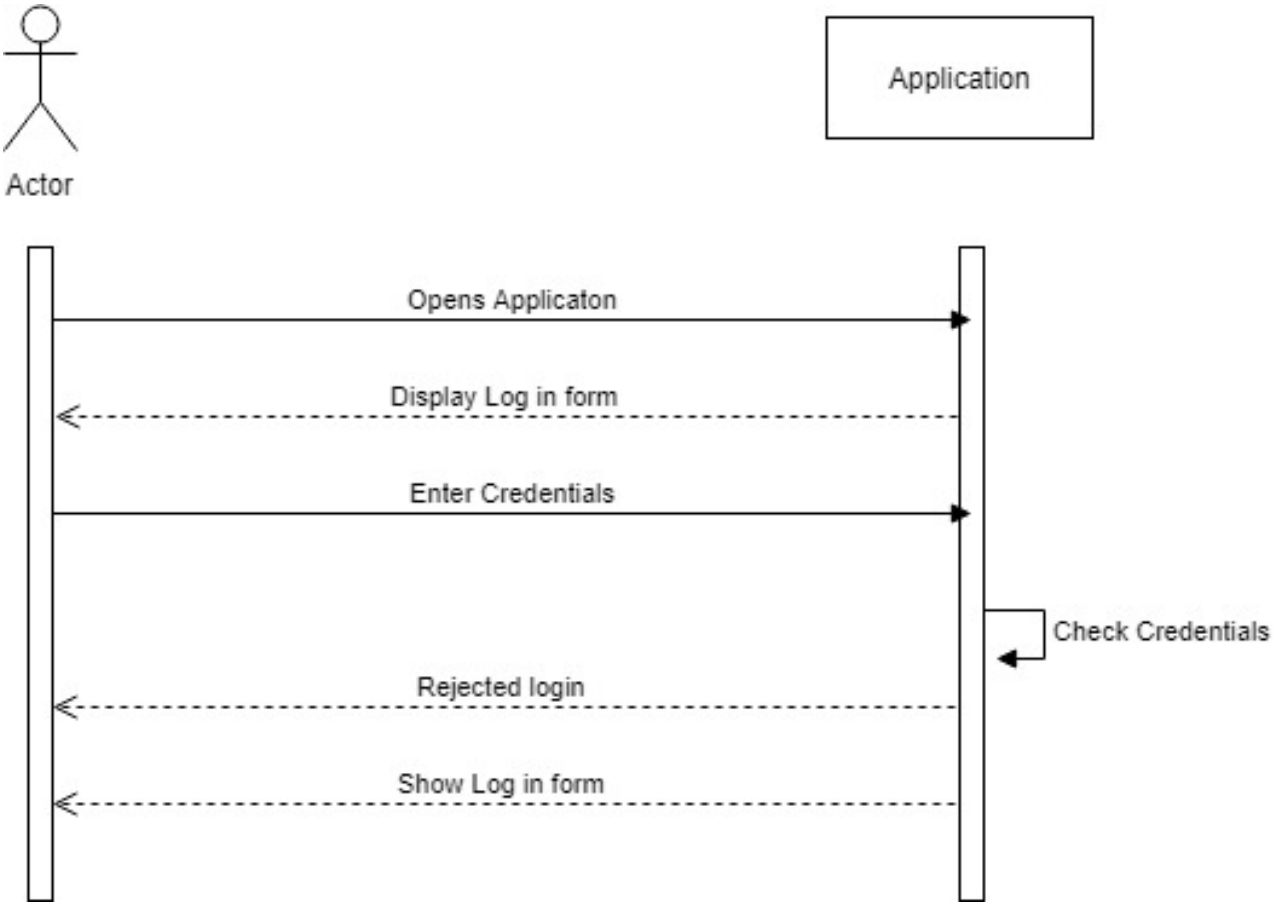


Figure 7: Rejected Login

In case if the participant does not provide correct credentials the system will reject his/her login. This step will provide security features also against outside malicious users.

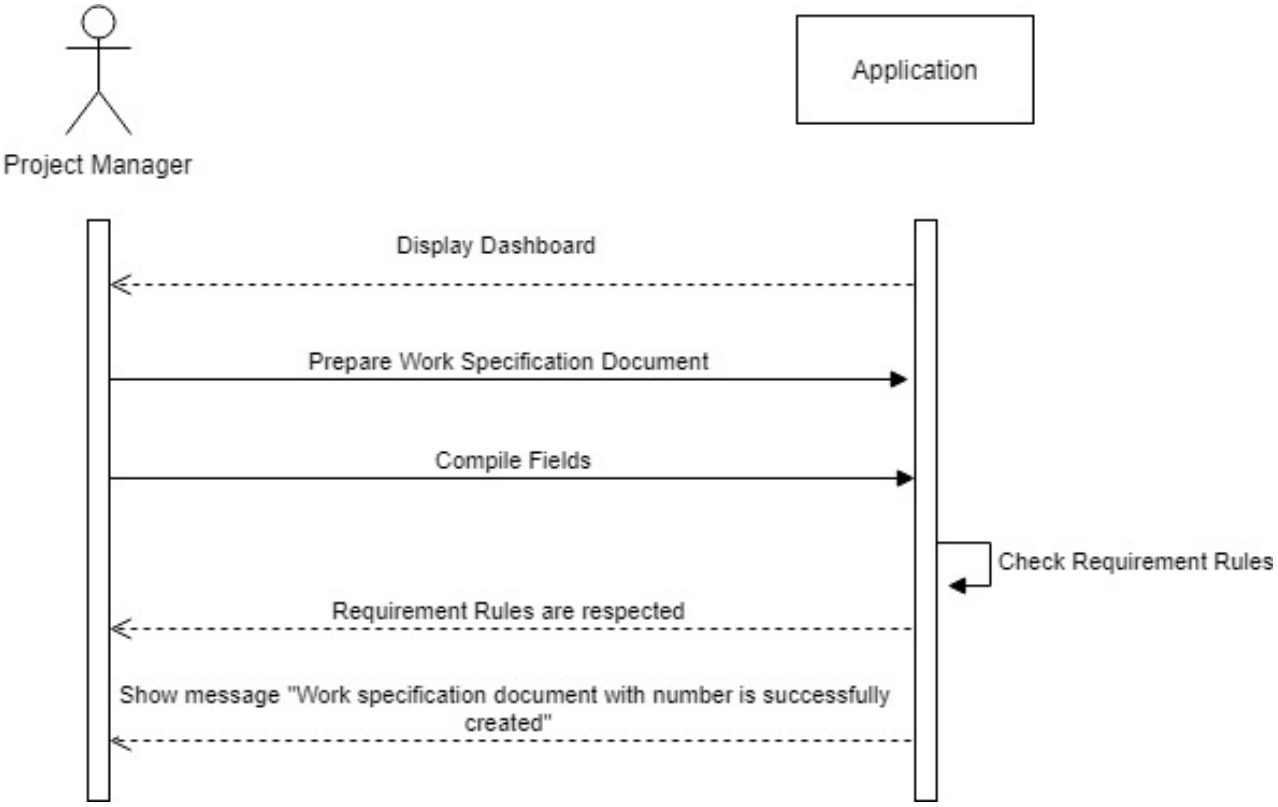


Figure 8: Project Manager

Project Manager can prepare the ”Work specification Document” by compiling the required fields.If all of the fields are respecting the rules the document will be successfully created.

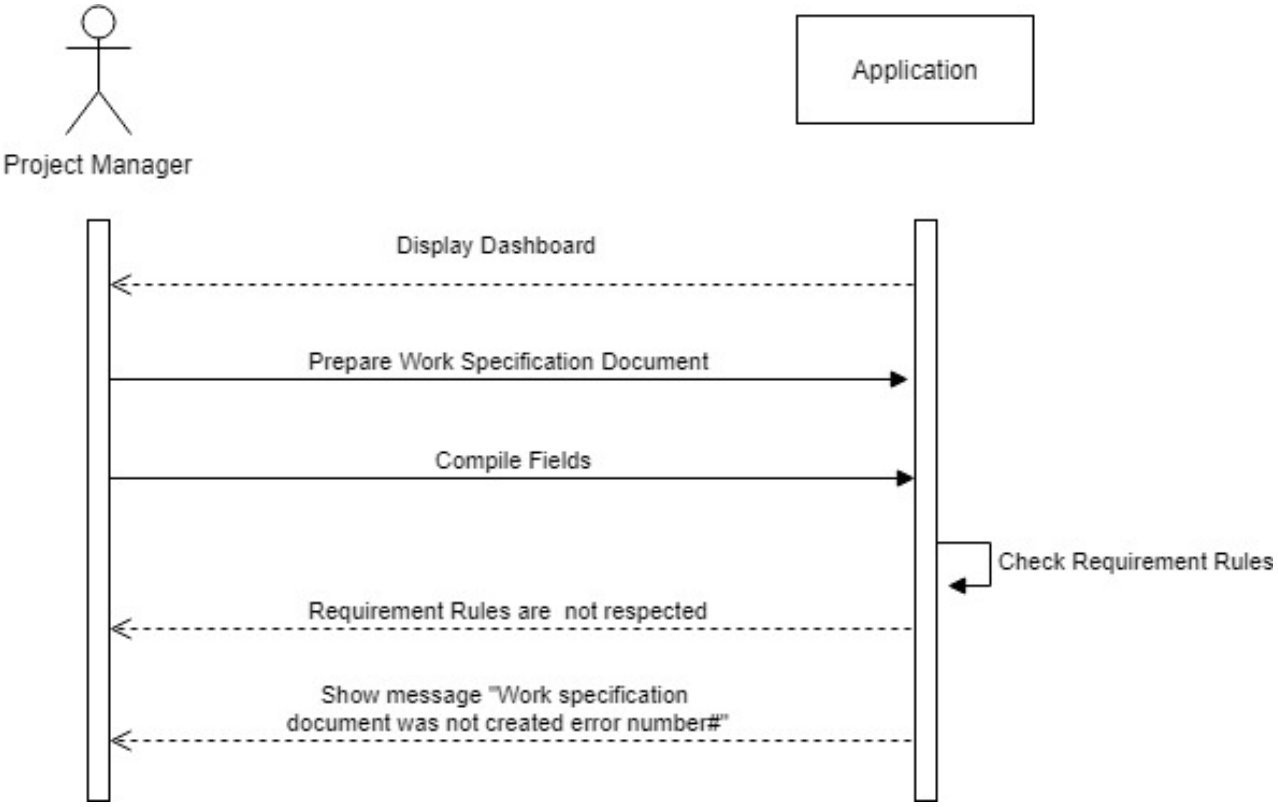


Figure 9: Project Manager Error

If the Project Manager is not respecting the enforced rules the document will not be created and a proper error message will be shown.

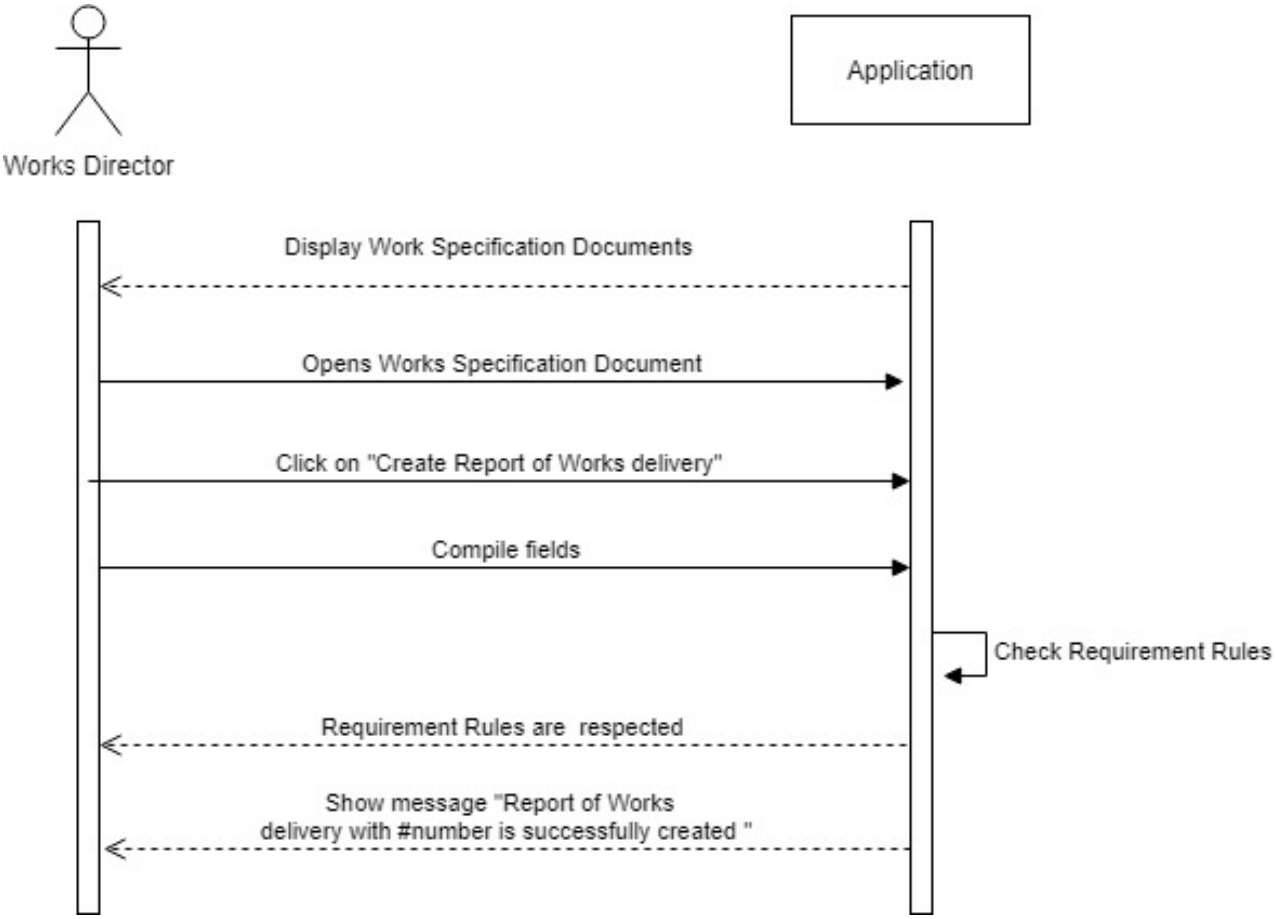


Figure 10: Works Director

Works Director here can view all of the ”Work Specification Documents”. Based on this documents he can create new document called ”Report of Works delivery”. After compiling the system will check if the enforced rules are respected and if they are the document successfully will be created.

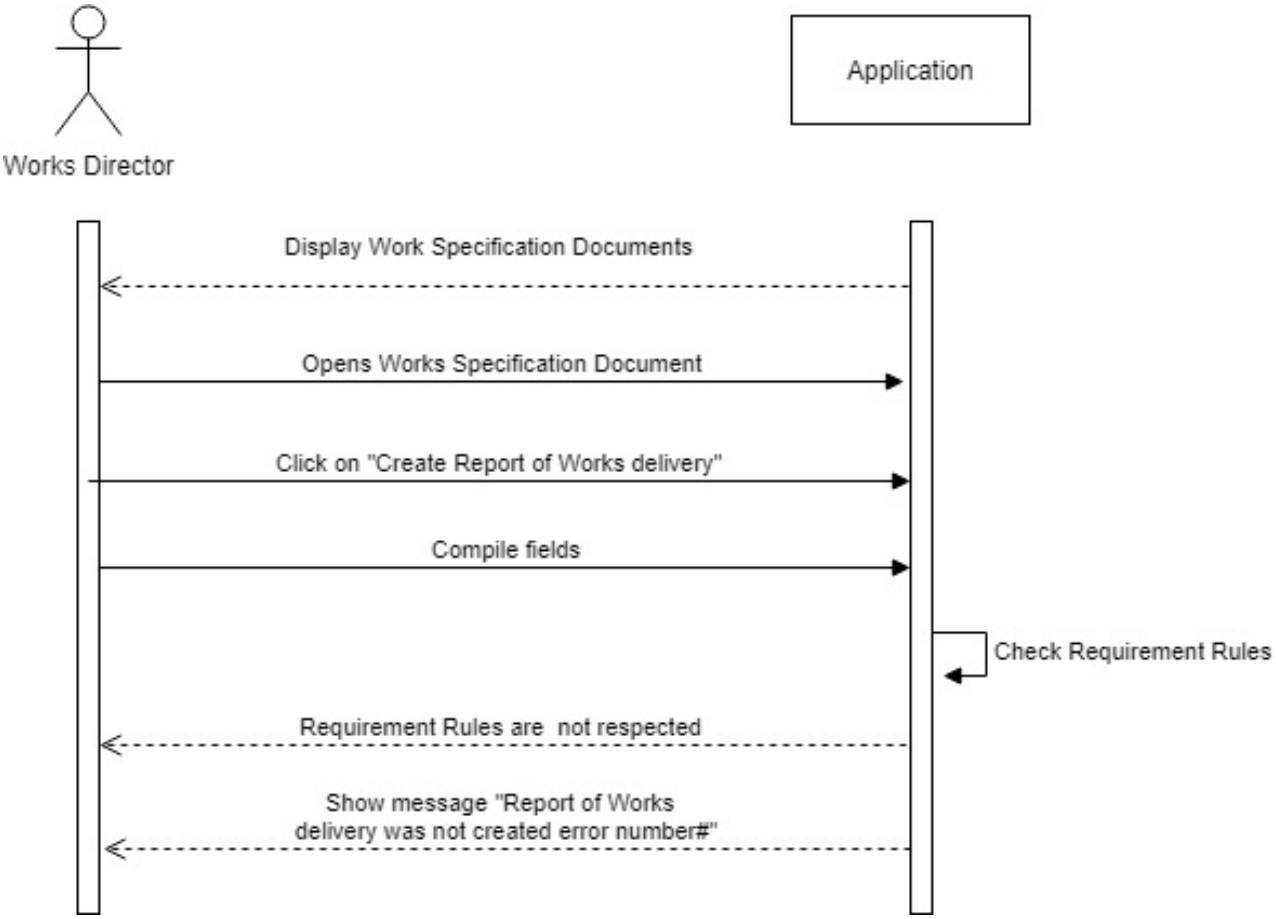


Figure 11: Works Director Error

If the Works Director is not following the enforced rules the system will reject the creation of the document and a proper error message will be shown.

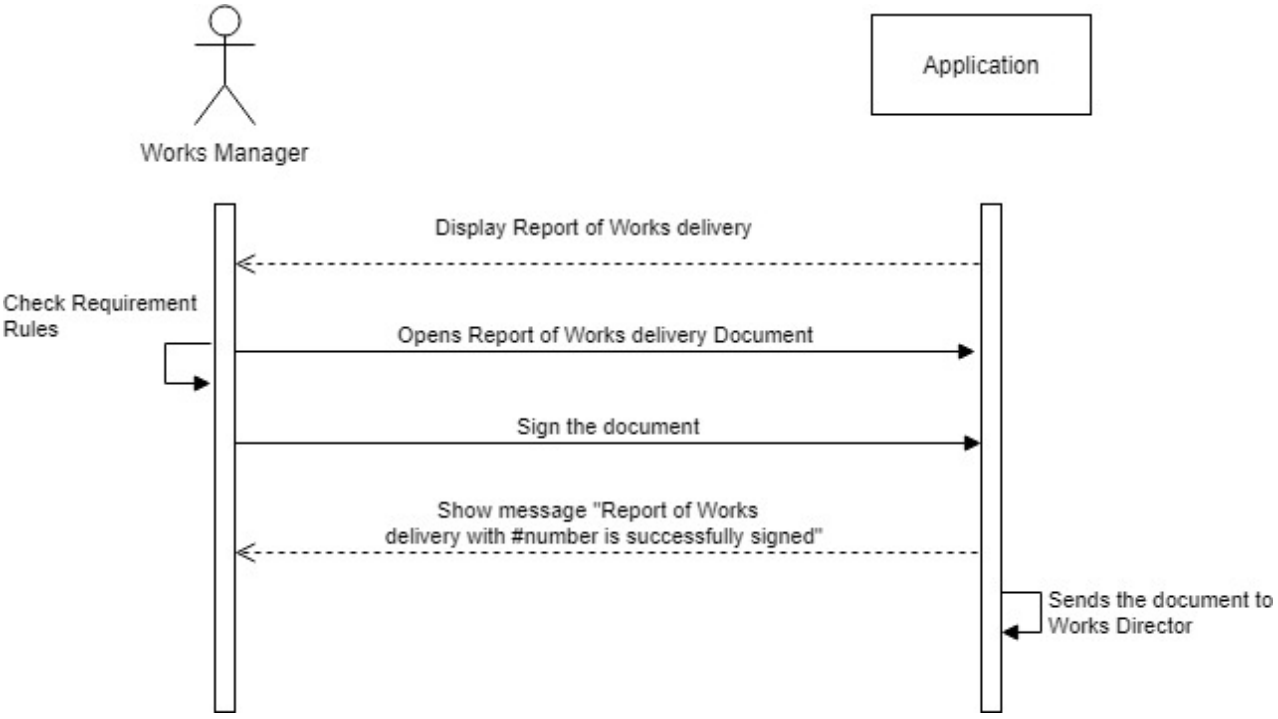


Figure 12: Works Manager

Works Manager will view all of the ”Report of Works delivery” documents that have been created. After close revision of the document the Works Manager will sign the document.

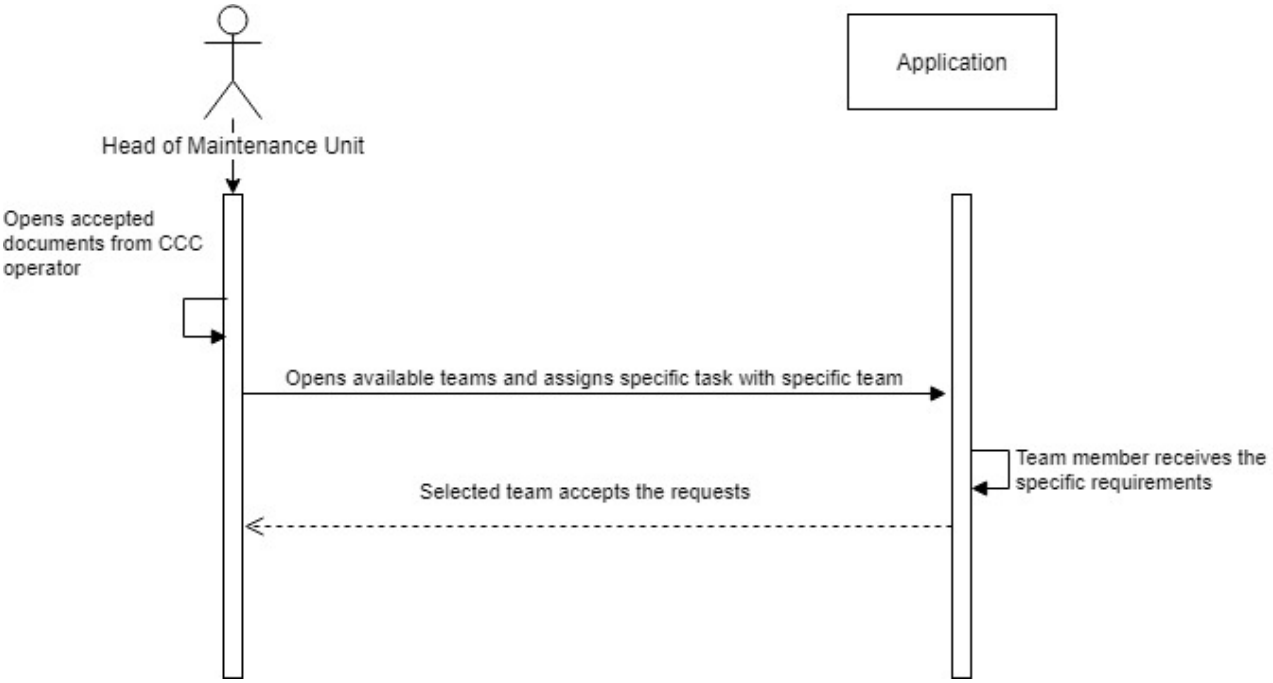


Figure 13: Worker Accepts

The Head of Maintenance Unit opens all of the documents that are accepted by the CCC operator. After revision he assigns proper maintenance team. The team that is selected will view the requirements and accept the work provided.

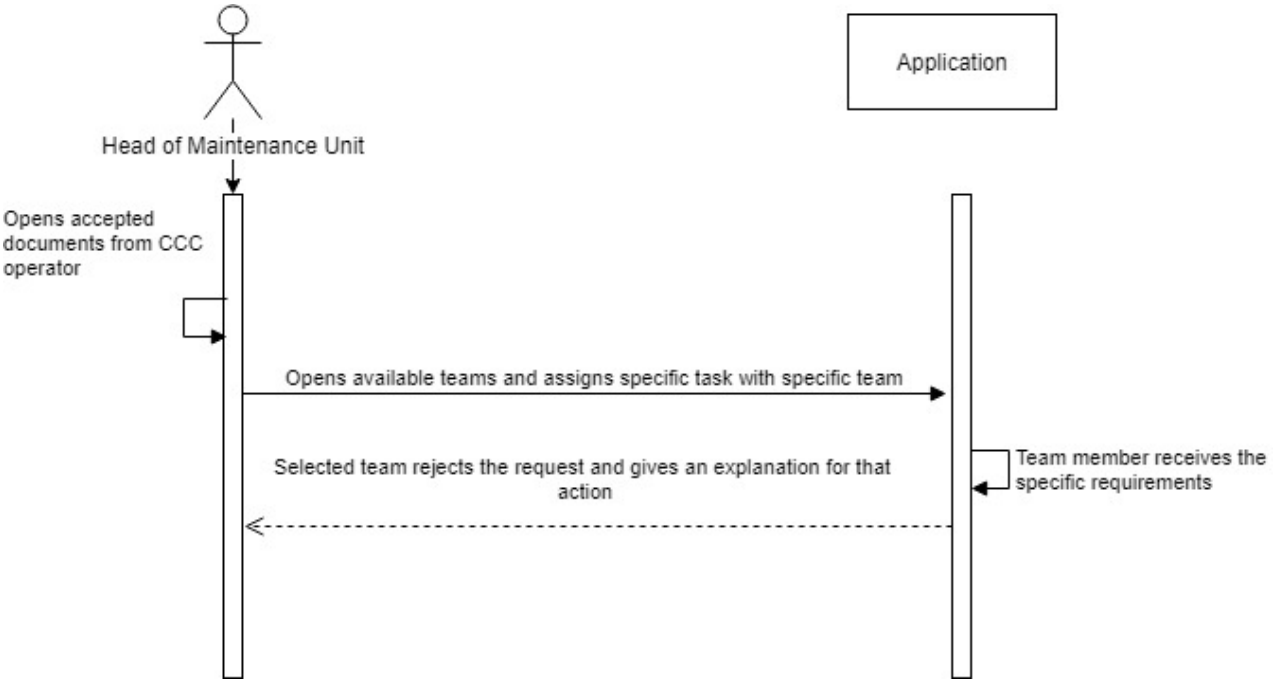


Figure 14: Worker Rejects

The selected workers team can reject the offered job because of many problems. But however they need to write description why the job was rejected.

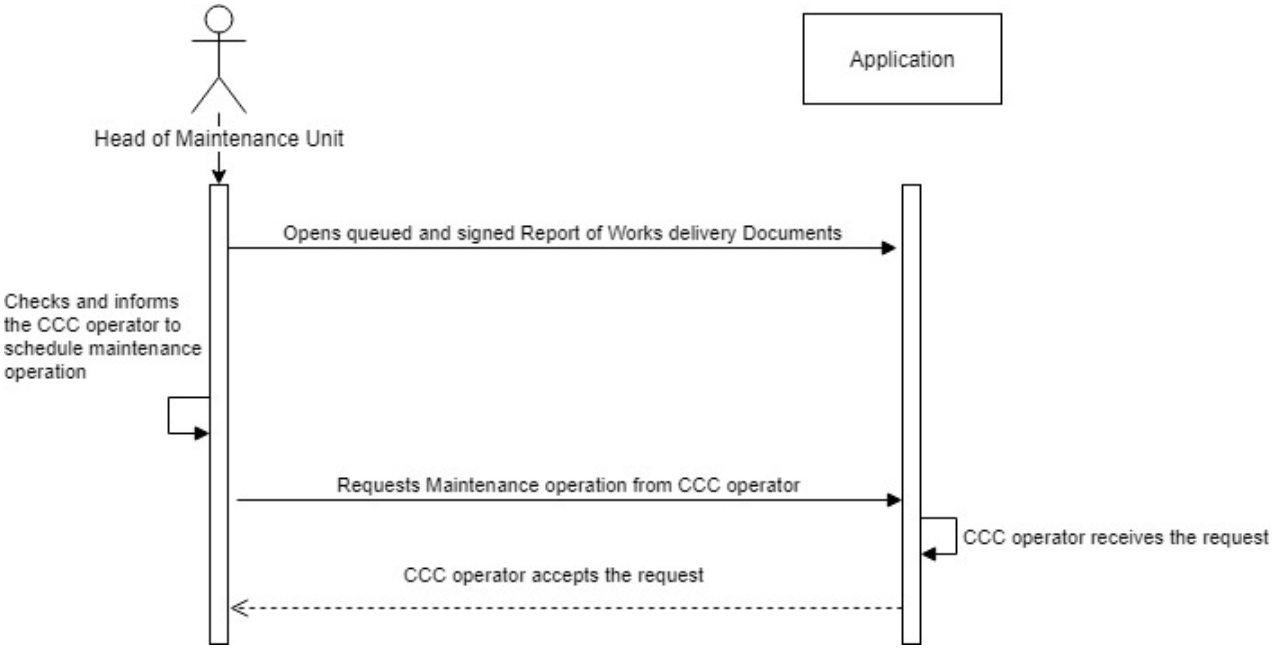


Figure 15: CCC operator accepts

If the CCC operator checks that on that day and on that track is possible to be done maintenance, he accepts the request and informs the Head of Maintenance Unit and the Workers.

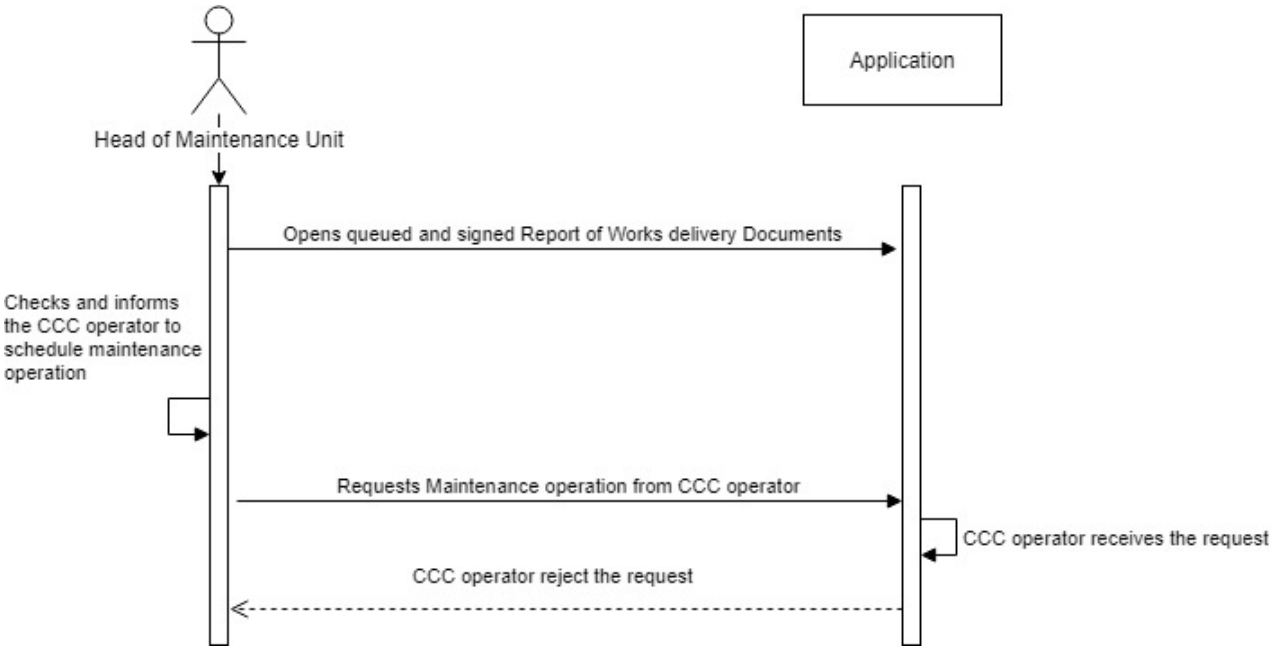


Figure 16: CCC operator rejects

If that certain request is not available on that day or because of any other reason he/she rejects the request and informs the Head of Maintenance Unit and the Workers.

4 User interface

In this section we will report the most relevant print screens of the PoC demonstrator for the solution, which is actually a backend of a smart contract solution based on blockchain, plus the technical interface for primary testing on the field with selected operators. Obviously, in real environment this interface should evolve in a more user friendly and mobile.

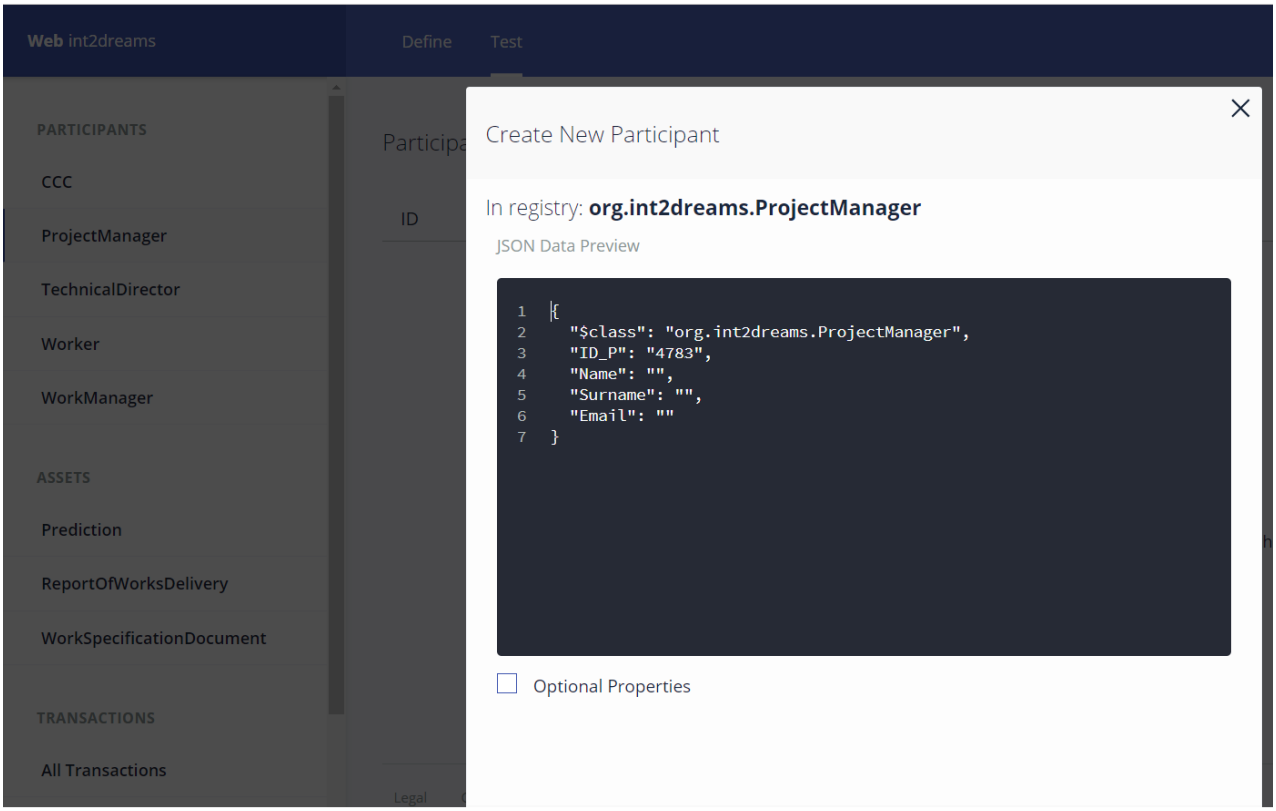


Figure 17: Create Project Manager

This picture represents the adding new Project Manager process in the system. When the network is deployed the admin of the network can add or delete Participants in the network.

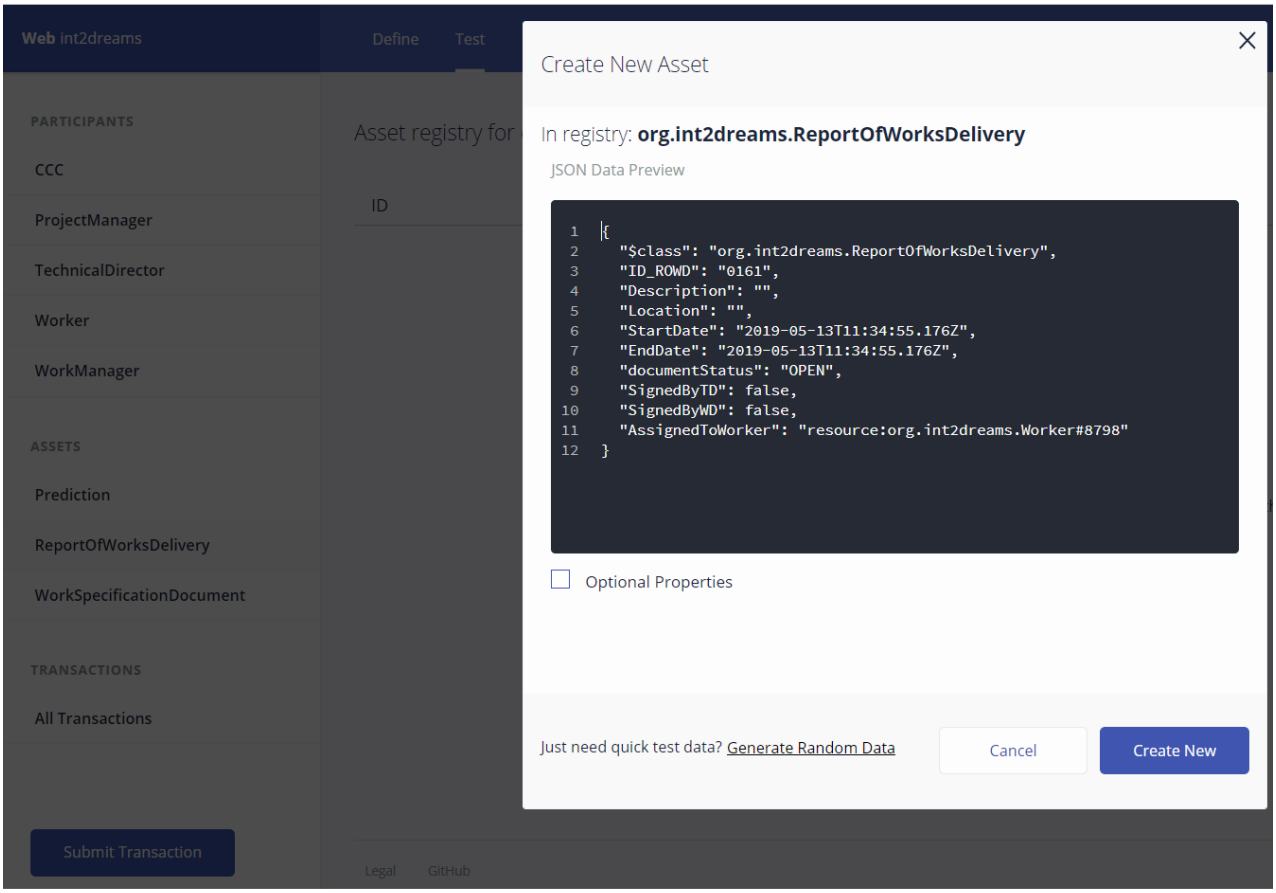


Figure 18: Create Works of Delivery document

This is the process of creating the ”Report of works delivery” document. You can see all of the required fields and example input of the data. If you click on the ”Optional Properties” you can see if there are some optional fields that can be inserted. In the end of the process the participant clicks on ”Create New” to create the document.

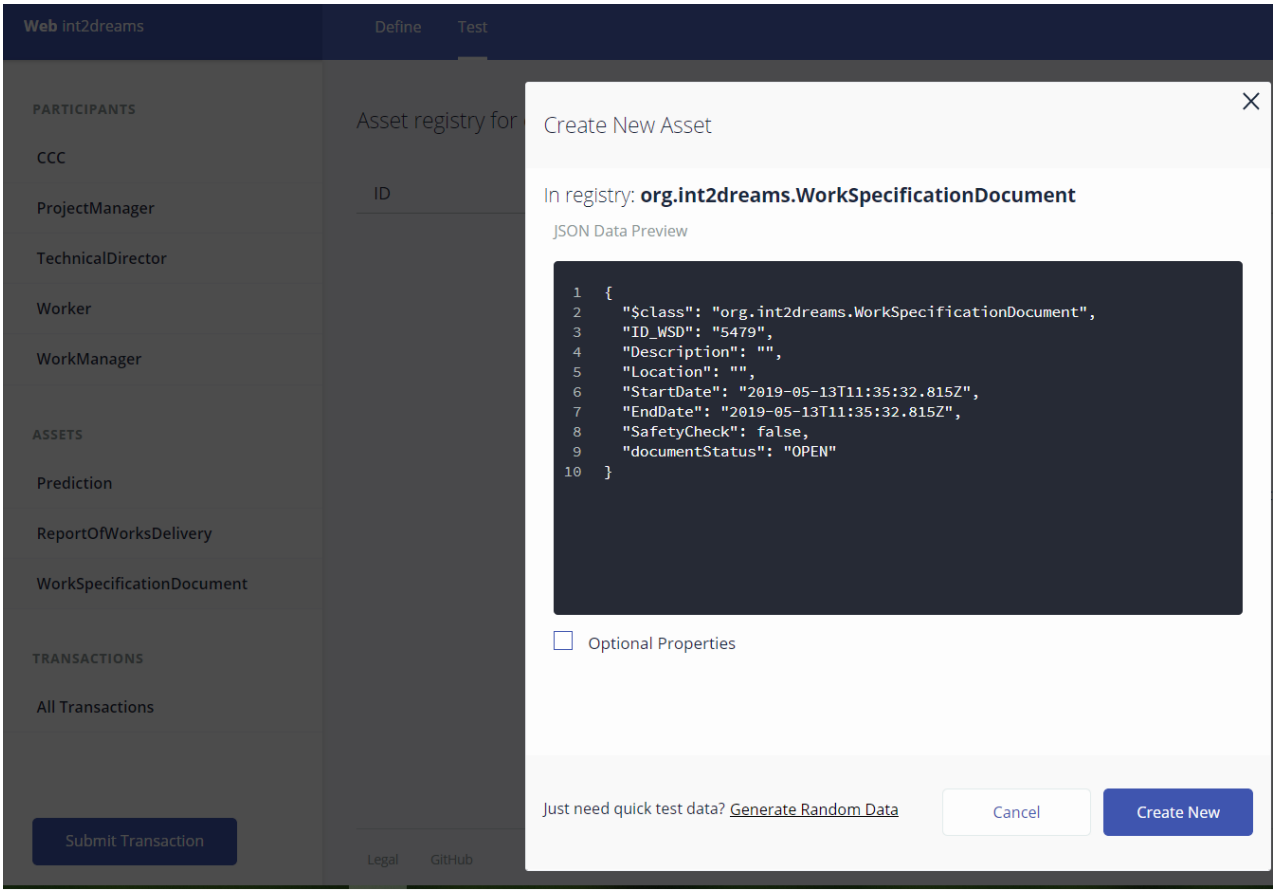


Figure 19: Create Works Specification Document

This is the process of creating the ”Work specification document” . You can see all of the required fields and example input of the data. If you click on the ”Optional Properties” you can see if there are some optional fields that can be inserted. In the end of the process the participant clicks on ”Create New” to create the document.

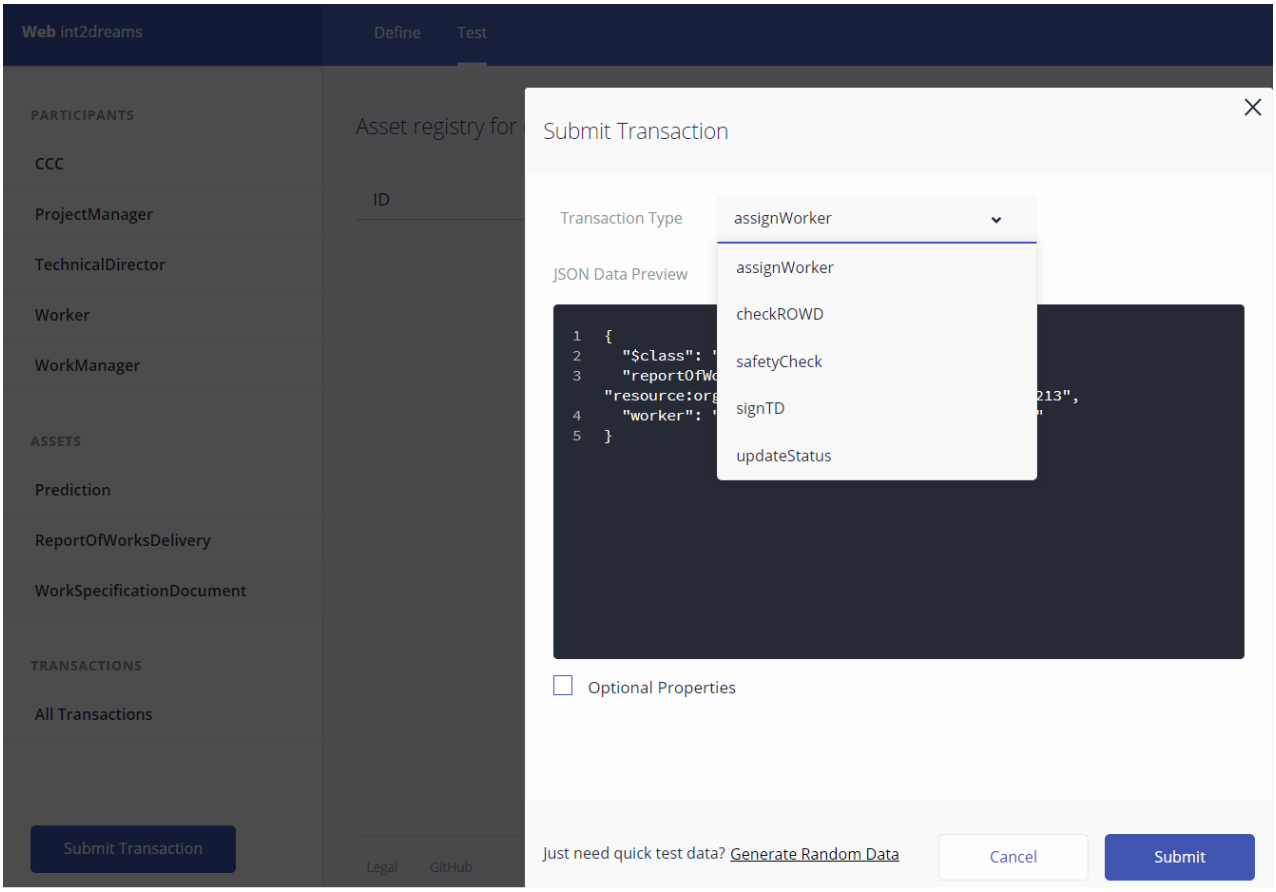


Figure 20: Submit Transactions

In this picture you can see the possible transaction that the participants can perform in the network. After choosing the right transaction the participant needs to insert the proper data and click on submit so that the transaction can be processed.

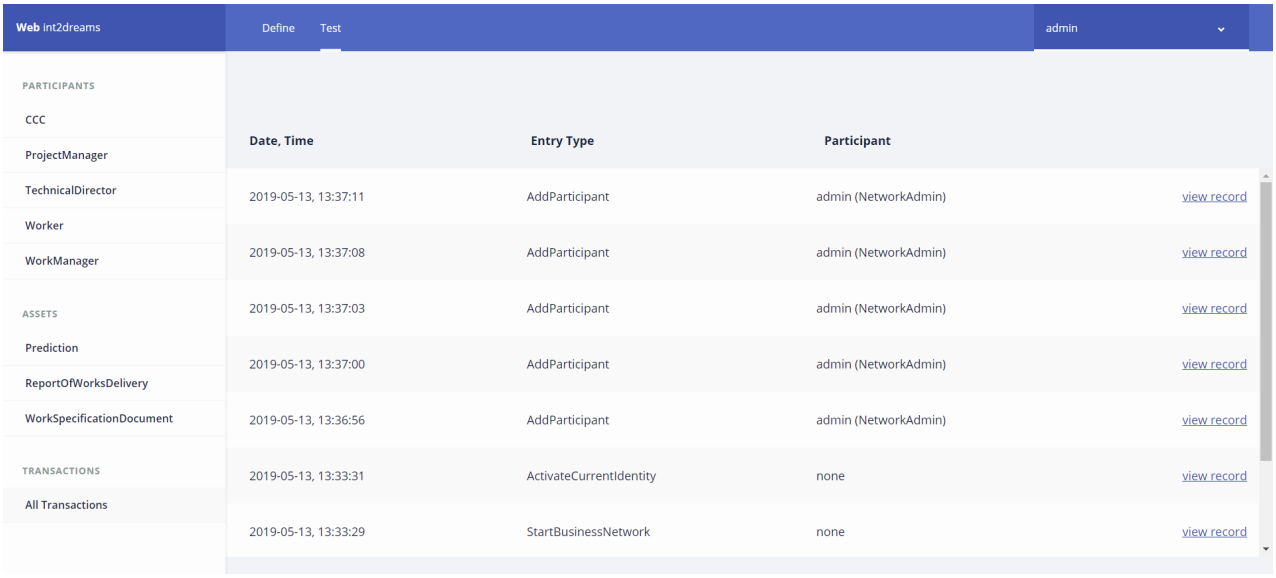


Figure 21: View the ledger

This pictures shows the ledger and all of the interactions happened in the network. The admin can analyze, check the actions performed inside the network.

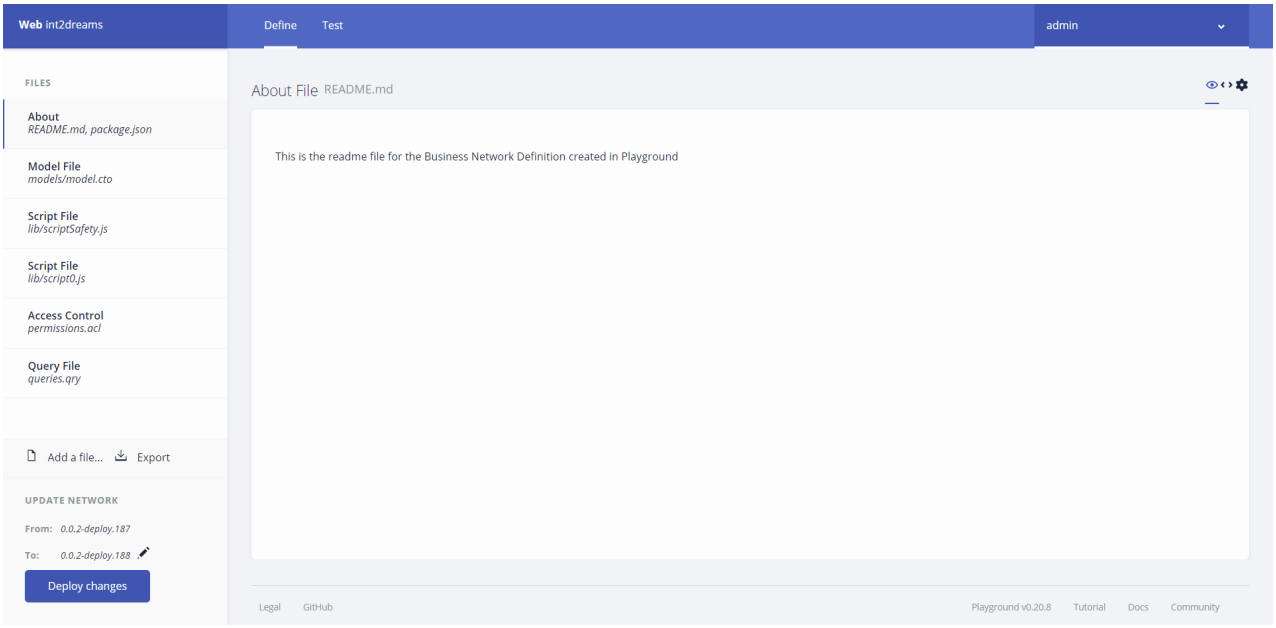


Figure 22: Backend

This picture shows the backend and the structure of the whole network

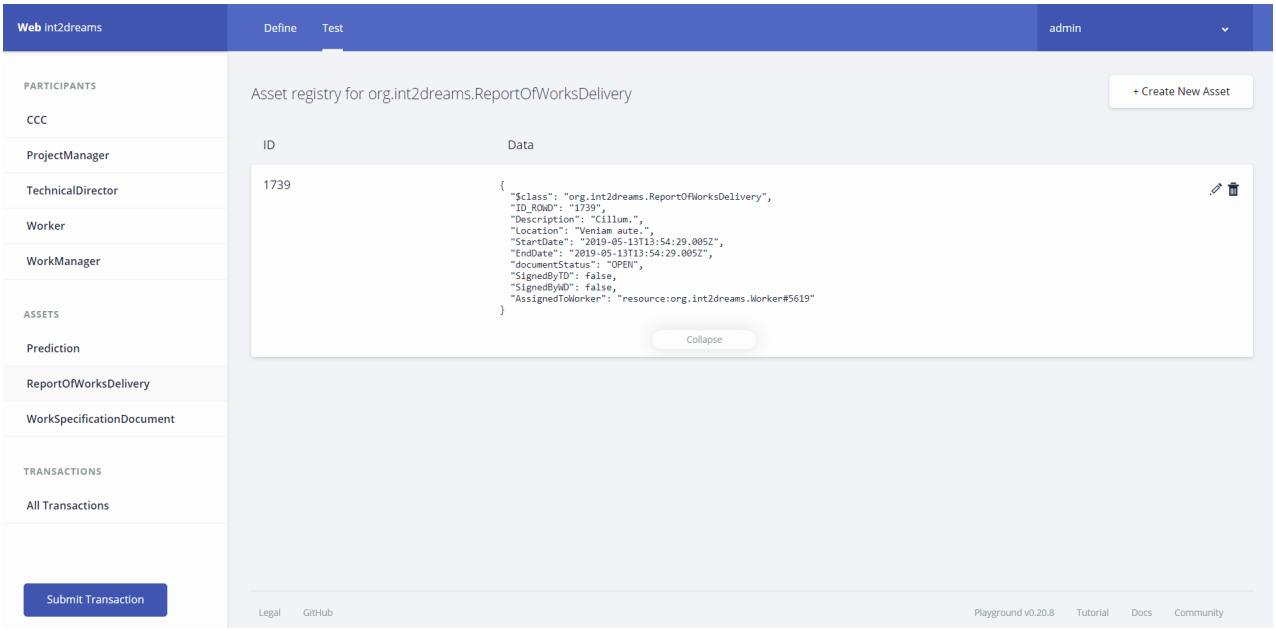


Figure 23: View documents

Here in this window we can see successfully creation of ”Report of Works delivery” document. After a successfully creation of the document, unique ID is assigned to that document.

5 Requirements

This chapter contains the functional requirements and the related Usage Scenarios of the TD for the Asset Maintenance[3]. It contains also the list of the the non functional requirements of the TD , related to its evolution in time, presentation, access, reliability and user interface. All the requirements listed in this chapter (functional and non functional) and all the usage scenarios have already been implemented in the actual TD delivered as output of task 4.3 of WP4. For the non functional requirements, the TD fulfill the requirements with a minimum grade and optimization are expected in future implementation and evolution of the TD.
All the requirements have been derived from the analysis that was conducted in the previous tasks of WP4, and can be retrieved in Deliverable 4.1. (for a description of the workflow, see the figure 1).

5.1 Functional requirements

Here are explained the operations and the activity’s that this system should provide. Users of this software should be able to understand how to use and operate this software with minimal computer knowledge and no technical knowledge. Bellow all of the functional requirements for each actor in the network are specified:
Project Manager

- Sign in
- Sign out
- Password reset

[3] The high level business context selected for the TD, Asset Maintenance, has been translated into the use case of Programmed Asset Maintenance and this technical use case has been splitted into technical scenarios for the development of the TD

- Manage personal information
- Able to create and edit “Work specification document”
- View all “Work specification documents” created in the system
- Change status of “Work specification document”

Works Director

- Sign in
- Sign out
- Password reset
- Manage personal information
- Able to view all “Work specification documents” created in the system
- Able to create “Report of works delivery”
- Able to sign “Report of works delivery”

Technical Director

- Sign in
- Sign out
- Password reset
- Manage personal information
- Able to sign “Report of works delivery”

Head of maintenance unit

- Sign in
- Sign out
- Password reset
- Manage personal information
- Able to view all “Report of works delivery”
- Able to assign maintenance workers
- Able to send and inform CCC operators

CCC operator

- Sign in
- Sign out

- Password reset
- Manage personal information
- Able to view all “Report of works delivery”
- Able to reschedule “Report of works delivery”

Worker

- Sign in
- Sign out
- Password reset
- Manage personal information
- Able to view only assigned “Report of works delivery”
- Able to create “Prediction” document

5.2 Usage Scenarios of the TD

In this section the usage scenarios of the TD are described. The term scenario here is used accordingly to Standard Glossary of Software Engineering Terminology.

5.2.1 Scenario 1

The Project Manager needs to prepare document ”Works Specification document” for certain railway maintenance. The standard procedure requires that the document is hand written and then send to the Works Director. This procedure is time consuming and lacks security, also if some kind of mistake happen in the phase of compilation of the document and if its not seen immediately can cause huge problems. With this system the Project Manager can easily compile the document and the system will check for possible mistakes and inform if they are present. Also he/she can easily see all of the previous documents that were created and sort them in various ways. Also the whole communication is secure and the process is tamper proof. The system makes objective view of the documents on the need to know basis that means that only the actor that needs and have permission to see the document will see it and no one else.

5.2.2 Scenario 2

Works Director receives the ”Works Specification document” compiled by the Project Manager. This document needs to be reviewed and based on it to create a new document called ”Report of works delivery”. In case if the previous document ”Works Specification document” is somehow mistaken the Works Director needs to return the document and to wait for the correct one and this is also time consuming process. In the process of preparing the ”Report of works delivery” mistakes can happen that can continue and cause critical hazards. With this system the Works Director is always sure that he/she receives the correct documents and they can compile the ”Report of works delivery” with high confidence. After finishing with the compile of the document the Works Director signs the document and sends it to Technical Director Also the whole communication is secure and the process is tamper proof. The system makes objective view of the documents on the need to know basis that means that only the actor that needs and have permission to see the document will see it and no one else.

5.2.3 Scenario 3

The Technical Director receives the document ”Report of works delivery” and he/she is doing evaluation of the document. In this point the Technical Director can not be sure that this is the original document and that all of the data inside are actually relevant. He/she is also risking to sign document that potentially may be malicious. This system ensures that all of the data inserted and all of the signs are authentic and relevant and also he/she is able to see all of the journey of the document and to be confident to sign it and to return it to the Works Director. Also the whole communication is secure and the process is tamper proof. The system makes objective view of the documents on the need to know basis that means that only the actor that needs and have permission to see the document will see it and no one else.

5.2.4 Scenario 4

Head of the Maintenance Unit receives the ”Report of works delivery” document and he can not be sure that this document is not intercept by someone or if the data is tampered by some malicious one. With all of that uncertainty he/she needs to inform the CCC operator for scheduling the maintenance and also to assign team of Workers for doing the job. With the help of the system and the blockchain technology the Head of the Maintenance Unit can trust the document and the data inside and he/she can make decisions without additional double checking about the relevance of the document. Also he/she can easily and in secure manner communicate with the Workers and CCC operator.

5.2.5 Scenario 5

The CCC operator will receive request for maintenance and his/her job is to make sure that location where the job needs to be done is secure and available for maintenance and the environment is safe so that the Workers can work risk free. But in any case he can not guarantee with high certainty that the document is relevant and not tampered in its journey. In case some mistake happen a big capital loss and potential risk can arise. With this system the CCC operator can easily manage all of the requests and can always be sure that the document that he/she is reading is secure and relevant.

5.2.6 Scenario 6

Workers are the last in this chain of operation that will receive information and requirements about the maintenance job. If they receive false information they risk their life or they can cause critical hazards if some damage is not fixed on time. With help of this system they can be sure that all of the in information’s are relevant and that they can work risk-free. They view all of the requirements on their mobile app, see the deadline and report problem. When they are going to finish with the job they can report and closed the document. Also they can filter documents based on location, date or problem for further analysis.

5.3 Nonfunctional requirements

This kind of requirements will describe aspects of the system that will relate to its evolution over time. These requirements will ensure the usability and effectiveness of the entire system. In our project not all of the nonfunctional requirements are needed to be satisfied. Usability
s

- Presenting data and information’s in clear, logical and understandable way
- Fast access to content, information

- User friendly interfaces (intuitive and easy to use)

Reliability

- The system should provide almost real-time information
- The system should provide accurate date

Availability

- The system should be available 24 hours, 7 days in the week

Security

- The system should be tamper proof
- The system should respect the data privacy rules

Maintainability

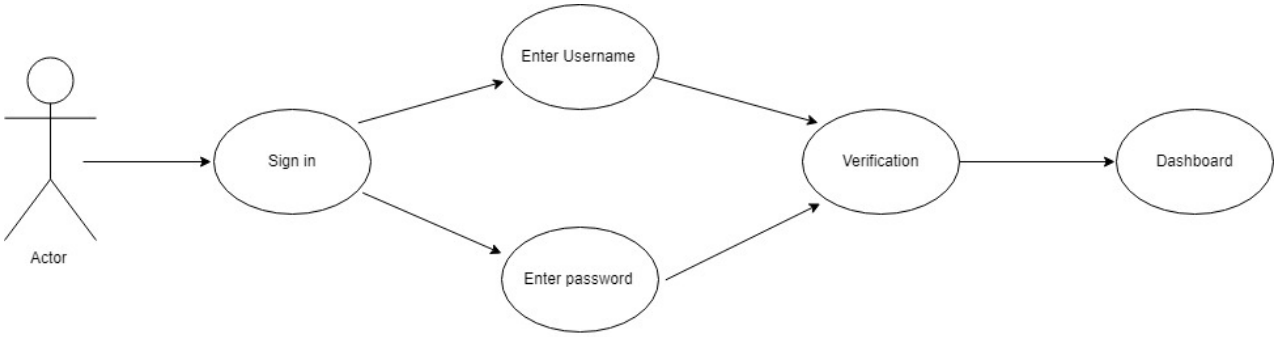
- The system should be designed in a way that new additions of features can be done without changing the already developed structure

Portability

- The system should be able to be access from different platforms and Operating systems

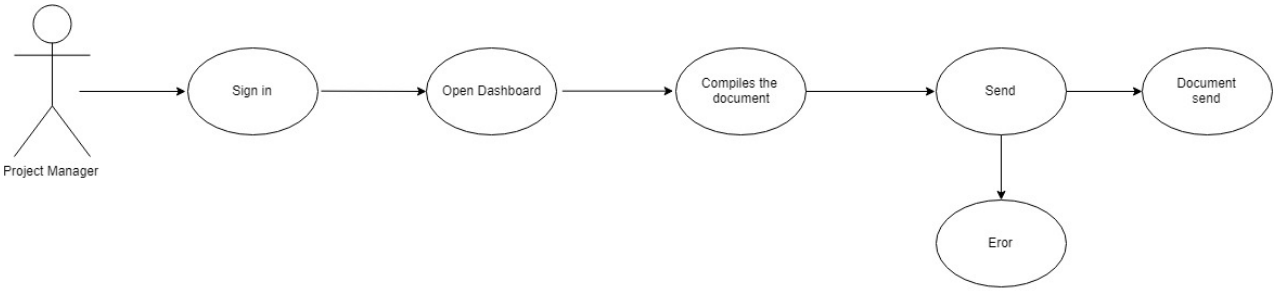
6 UML models

In this chapter, according to standard method ([4]), the journey of each of the participants in the networks is described visually and the related table with conditions is given (entry, exit, expeptions).



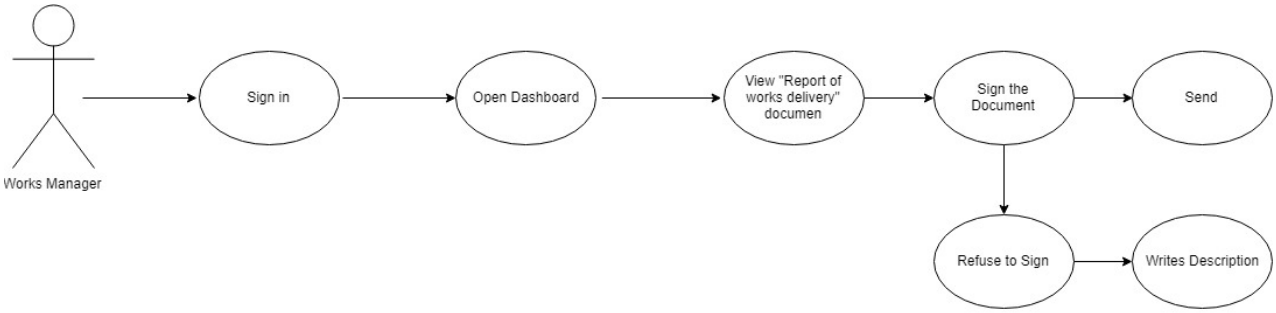
Name	User Log in
Actor	All of the actors in the system
Entry conditions	The user wants to log in into the system
Events flow	The user navigates to the system domain. The system shows login page and the user inserts username and password and clicks ont he sign in button The system validate the correctness of the inserted information, confirms and redirects to the proper dashboard
Exit conditions	The user has access to the system functionality
Exceptions	If the user inserts non valid information's the system shows an error message and the user can enter his/her credentials again

Figure 24: Login



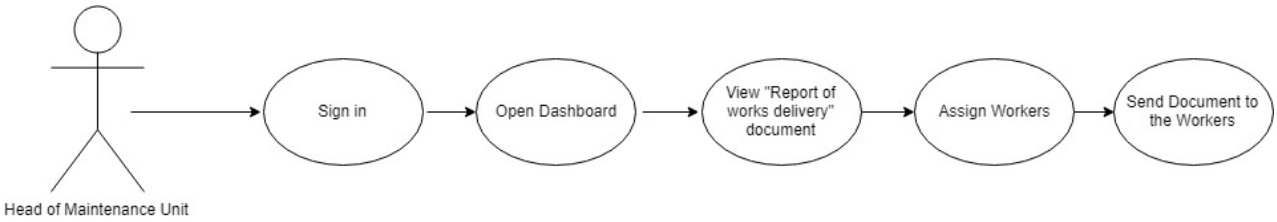
Name	Project Manager compiles "Works specification document"
Actor	Project Manager
Entry conditions	The Project Manager wants to compile "Works specification document"
Events flow	The Project Manager logs in and navigates to the dashboard. The Project Manager compiles the document by inserting all of the required fields. The system validate the correctness of the inserted information, confirms and redirects to the proper dashboard
Exit conditions	"Works specification document" with #number is successfully created
Exceptions	If the Project Manager inserts incorrect information's the system shows an error message and the user cant send the document and is asked to try again, If the start date is in the past error will occur.

Figure 25: Project Manager



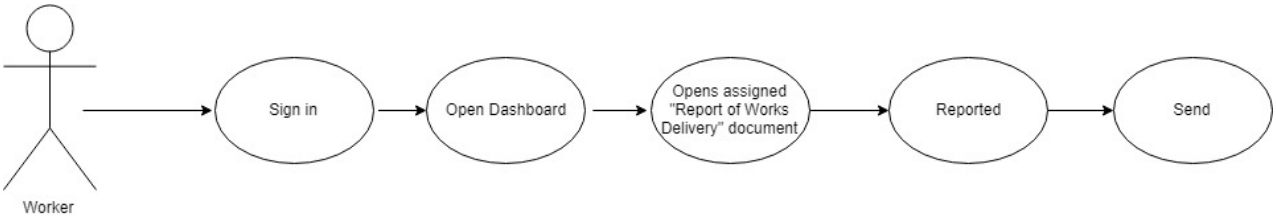
Name	Works Manager signs "Report of works delivery"
Actor	Works Manager
Entry conditions	The Works Manager reviews and signs the "Report of works delivery" document
Events flow	The Works Manager logs in and navigates to the dashboard. He/She opens the document check constraints and if everything is okay signs the document The system confirms and checks if the document is signed by the two parties Work Manager and Works Director and sends the document to the Head of the maintenance unit The system redirects to the proper dashboard
Exit conditions	"Works specification document" with certain #number and status is reviewed and signed by the Works Manager
Exceptions	"Report of Works Delivery" with status "Closed", "Assigned" and "Reported" can not be reviewed.

Figure 26: Works Manager



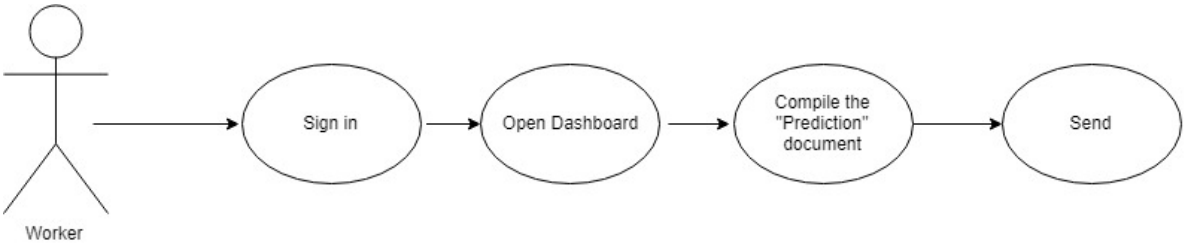
Name	Head of Maintenance Unit check the "Report of works delivery" document and assigns Workers and send the document to them
Actor	Head of Maintenance Unit
Entry conditions	The Head of Maintenance Unit reviews, assigns workers and sends the document to Workers
Events flow	The Head of Maintenance Unit logs in and navigates to the dashboard. He/She opens the "Report of works delivery" document checks and if everything is okay assigns proper Workers and sends the document to the Workers The system confirms and checks if the document is signed by the two parties Work Manager and Works Director and sends the document to the Workers The system redirects to the proper dashboard
Exit conditions	"Works specification document" with certain #number and signed is reviewed, workers are assigned and send to the Workers
xceptions	"Report of Works Delivery" with status "Closed", "Assigned" and "Reported" can not be reviewed.

Figure 27: Workers



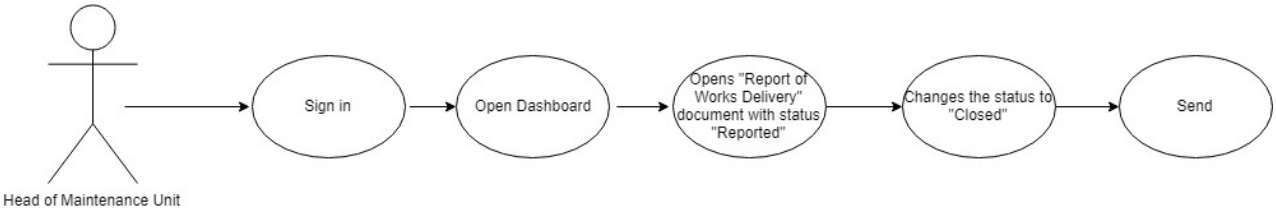
Name	Worker finishes his assigned job
Actor	Worker
Entry conditions	Worker change the status of his ongoing job
Events flow	The Worker logs in and navigates to the dashboard. He/She opens the "Report of Works Delivery" document and change the status. The system confirms and changes the status of the document. The system redirects to the proper dashboard
Exit conditions	"Report of Works Delivery" document status is changed
Exceptions	Only "Reported" status can be insert, If the worker tries to change to other status proper error will be displayed

Figure 28: Worker finishes



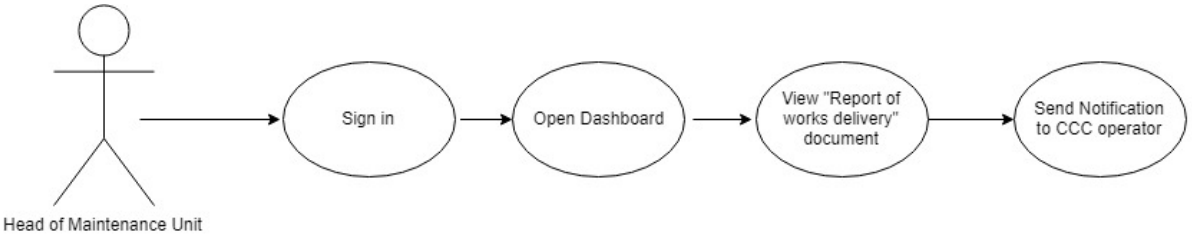
Name	Worker compiles the "Prediction" document
Actor	Worker
Entry conditions	Worker compiles the "Prediction" document and sends to the system.
Events flow	The Worker logs in and navigates to the dashboard. He/She opens the "Prediction" document and compiles the fields required. The system confirms and checks if the document is properly field assigns unique ID number and saves it. The system redirects to the proper dashboard
Exit conditions	"Prediction" document with certain #number is successfully created
Exceptions	If the worker inserted non valid data the system will show an error message. If the worker did not inserted required filed the system will show an error message

Figure 29: Worker Compiles Form



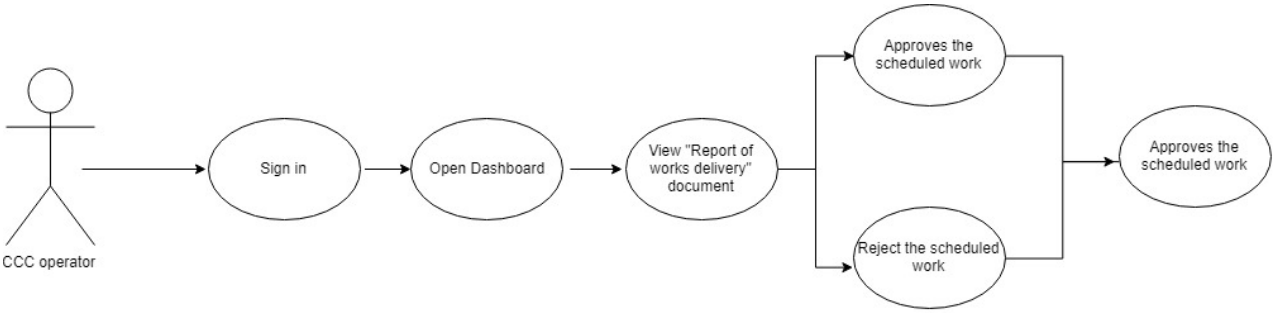
Name	Head of Maintenance Unit check if everything was properly done and closes the document
Actor	Head of Maintenance Unit
Entry conditions	Head of Maintenance Unit changes the status of the document
Events flow	The Head of Maintenance Unit logs in and navigates to the dashboard. He/She opens the "Report of Works Delivery" document and change the status. The system confirms and changes the status of the document. The system redirects to the proper dashboard
Exit conditions	"Report of Works Delivery" document status is changed
Exceptions	Only "Reported" documents status can be change to "Closed".

Figure 30: Head Of Maintenance Unit



Name	Head of Maintenance Unit check the "Report of works delivery" document and informs CCC operator
Actor	Head of Maintenance Unit
Entry conditions	The Head of Maintenance Unit reviews and sends the document to CCC operator
Events flow	The Head of Maintenance Unit logs in and navigates to the dashboard. He/She opens the "Report of works delivery" document checks and if everything is okay sends the document to the CCC operator The system confirms and checks if the document is signed by the two parties Work Manager and Works Director and sends the document to the CCC operator The system redirects to the proper dashboard
Exit conditions	"Works specification document" with certain #number and signed is reviewed and send to the CCC operator
Exceptions	"Report of Works Delivery" with status "Closed", "Assigned" and "Reported" can not be reviewed.

Figure 31: CCC operator



Name	CCC operator informs the Workers
Actor	CCC operator
Entry conditions	CCC operator reviews the document and send proper notification to the Workers
Events flow	The CCC operator logs in and navigates to the dashboard. He/She opens the "Report of works delivery" document checks and if everything is okay sends and sends the document to the Workers. The status is changed to "Assigned". The system confirms and checks if the document is signed by the two parties Work Manager and Works Director and sends the document to the Workers operator The system redirects to the proper dashboard
Exit conditions	"Works specification document" with certain #number and signed is reviewed and send to the Workers
Exceptions	"Report of Works Delivery" with status "Closed", "Assigned" and "Reported" can not be reviewed.

Figure 32: CCC operator informs Workers

7 Implementation remarks

The TD for the PoC is based on Hyperledger Fabric v1.4[4], the open source project of Linux Foundation supported by IBM, that is resulted as the best solution from the assessment done during the previous tasks of WP4 and that can be retrieved at D4.2.
According to this approach, the network is permissioned and restricted to identified users, moreover the network is not based on reward, Proof of Work mining and cryptocurrency (like Bitcoin or Ethereum). Fabric allows different consensus mechanisms based on nodes differentiation of roles in orderers and validators (SOLO, Kafka, Pratical BFT). Dedicated nodes, called orderers, are in charge this validation and leadership process. The complete guide of Fabric and the SW used are explained at 3.
For the PoC, it is important to remember that: i) the identification of users can be managed with the pluggable Membership Service Provider; it is possible to create multiple channels to select the corresponding portion of the ledger which is visible. The Smart Contracts in Fabric are the ”Chaincodes”. The storage of the ledger states may be different.
The TD uses the Hyperledger Composer, which is the high-level application development framework that runs on top of the Hyperledger Fabric blockchain infrastructure.
As described in the [3], and explained in some articles in [1] and [2], Composer framework is based on isolated runtime a connected to network through the connection profiles that allows deployment of Business Network Definition (BND). THE BND is generated by the Composer and it is made up of:

[4]The current version of the blockchain used is not relevant, since the TD may be deployed also in further version with the same code source

- The CTO modeling language;
- The user interface Playground;
- The Command-Line Interface (CLI) tools.

The CTO is for modelling the business network. The Playground provides provides a user interface for the configuration, deployment and testing of a business network. The CLI is used for the integration of business networks and running instance.

8 Installation Guidelines

In this part it is explained how to deploy the network and install all of the dependencies needed for proper run of the system.It is suggested to follow the following steps in the exact order as they are given for a proper installation. The following hardware [5] requirements are for the minimal performance needed for testing:

- Minimum of 4GB of RAM,
- Minimum of 1 core processor (2.60GHz),
- Minimum of 60GB of hard disk free space.

This requirements might change if the network needs to grow and scale from demonstrative purpose to real system (this is out of the scope of task 4.3). Hyperledger composer, by the date of writing this document, is only supported on Ubuntu and Mac OS.For proper running Hyperledger Composer, pre-requisites are needed and they are the following one:

- Operating systems: Ubuntu Linux 14.04 / 16.04 LTS (both 64-bit), or Mac OS 10.12,
- Docker Engine: Version 17.03 or higher,
- Docker-Compose: Version 1.8 or higher,
- Node: 8.9 or higher (note version 9 is not supported),
- npm: v5.x,
- git: 2.9.x or higher,
- Python: 2.7.x.

In order to install Hyperledger Composer on Ubuntu operating system correctly, it is important to be aware of some advice: avoid the adoption of root account for installing just login as normal user and also the use ”su” to root. When installing the prerequisites is advised to use ”curl” following with unzip using sudo. After this step it is recommended to run ”prereqs-ubuntu.sh” as normal user. Also is strongly advised to avoid the use ”npm” with ”sudo” or ”su”. And lastly it is recommended to avoid installing node globally as root.
In below, in order to deploy Hyperledger Composer on Ubuntu, installation instructions are provided step by step :

[5] Eventually cloud installation are possible, but suggested only for demonstrative purpose.

On the terminal window type the following commands for downloading the prerequisites and to change permissions:

```
curl -O https://hyperledger.github.io/composer/latest/prereqs-ubuntu.sh
```

Change permissions:

```
chmod u+x prereqs-ubuntu.sh
```

Next run the downloaded script (you can use sudo here) :

```
./prereqs-ubuntu.sh
```

After finishing this step it is required to log out and then log in. After performing this step need to install the final NPM packages :

```
npm install -g composer-cli
npm install -g composer-rest-server
npm install -g generator-hyperledger-composer
npm install -g yo
npm install -g composer-playground
```

Successfully were installed the prerequisites and next they need to be installed and run Hyperledger Fabric. First create and enter directory :

```
mkdir ~/fabric-tools && cd ~/fabric-tools
```

Then download Hyperledger Fabric and unzip it in the directory that was created:

```
curl -O https://raw.githubusercontent.com/hyperledger/composer-tools
/master/packages/fabric-dev-servers/fabric-dev-servers.tar.gz
tar -xvf fabric-dev-servers.tar.gz
```

Now download Fabric with the following command and when the downloading will finish run Hyperledger Composer

```
./downloadFabric.sh
composer-playground
```

The installation is done with the prerequisites and Hyperledger Fabric. Now is time to implement our Business Network Archive(BNA). BNA is is file which contains the model, script network access and query file that defines our solution.First navagite to the fabric-tools directory and then we are creating Network Peer Admin card.

```
cd fabric-tools
./createPeerAdminCard.sh
```

Next start Fabric and install ".bna" file in the fabric-tools directory:

```
./startFabric.sh
composer network install -a int2dreams.bna -c PeerAdmin@hlfv1
```

Now start the network with the following command:

```
composer network start -c PeerAdmin@hlfv1 -n int2dreams -V 0.0.2-deploy.181
-A admin -S adminpw
```

Next import the created admin card by the following command:

```
composer card import -f admin@int2dreams.card
```

This was the last step and everything is ready to be tested.

9 Conclusions

The goal of the presented TD is to prove the actual feasibility of implementation of smart contracts for the programmed Asset Maintenance. The TD has been developed by using the proposed software architecture selected as output of task 4.2 of WP4, while the scenario was the output of WP2. The proposed TD has focused on programmed maintenance.

The TD for the PoC is testable in a safe and limited environment, as was explained in Chapter 5, in accordance to the needs of IAMS. This report has explained the technical aspects and given a guidelines to installation and use. As next step, the TD will be integrated, at demonstrative level, with the other two TDs resulting as outputs of WP2 (related to predictive maintenance and visualization). The integration of the three TDs has already been explained in D5.1 and it was possible since, within the WS2 of the Project In2Dreams, a single scenario was selected across different work packages (WP4 and WP5), giving the opportunity of testing the integration of different emerging technologies (see also as result [9]).

The TD has demonstrated the feasibility of a decentralised smart contract based asset maintenance. It might be extended in future to integrate also the physical part like switches, signals, occupancy sensors as well as on in-cab signaling, integrating the so called Internet of Things in a blockchain.

Next steps of the Task 4.3 will be to test the TD and to collect feedbacks on the field (there will be planned demos with IMs operators on RFI premises in Italy site on next June 2019) . For this objective, interviews will be conducted involving different operators at different level in the IMs.

To conclude the activities of WP4, after testing and interviews, the TD solution will be evaluated in term of overall strengths, weakens, impacts and possible evolution will be suggested in term of improving the maturity level of the present system towards possible real application.

References

[1] IBM Developers. *Patterns*. IBM, 2019.

[2] IBM Developers. *Serverless emerging patterns*. IBM, 2019.

[3] Hyperledger Foundation. *Fabric Reference Guide 1.4*. Red Hat, 2018.

[4] IEEE. *IEEE 1220-2005 - IEEE Standard for Application and Management of the Systems Engineering Process*. IEEE Software Systems Engineering Standards Committee, 2005, reaffirmed 2011.

[5] Systems and software engineering – Lifecycle profiles for Very Small Entities (VSEs). Standard, International Organization for Standardization, Geneva, CH, June 2016.

[6] Rete Ferroviaria Italiana. *Convenzione per l'esecuzione di lavori di manutenzione all'Armamento.* 2011.

[7] Rete Ferroviaria Italiana. 2017.

[8] Rete Ferroviaria Italiana. *Condizioni Generali di Contratto per gli appalti di lavori delle società del Gruppo Ferrovie dello Stato Italiane.* RFI, 2017.

[9] R. Spigolon, L. Oneto, D. Anastasovski, N. Fabrizio, M. Swiatek, R. Canepa, and D. Anguita. Improving railway maintenance actions with big data and distributed ledger technologies. In *INNS Big Data and Deep Learning (INNSBDDL)*, 2019.