COGITO

CONSTRUCTION PHASE DIGITAL TWIN MODEL

cogito-project.eu

D6.8 – Personalized On-site Works Support Apps v2

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958310

D6.8 – Personalized On-site Works Support Apps v2

Dissemination Level:	Public
Deliverable Type:	Demonstrator
Lead Partner:	NT
Contributing Partners:	Hypertech, UCL, UEDIN, BOC-AG
Due date:	31-10-2022
Actual submission date:	30-11-2022

Authors

Name	Beneficiary	Email
Ján Varga	NT	varga@novitechgroup.sk
Martin Straka	NT	straka@novitechgroup.sk
Bohuš Belej	NT	belej@novitechgroup.sk
Theofania Charbi	Hypertech	t.charbi@hypertech.gr

Reviewers

Name	Beneficiary	Email
Tobias Hanel	FER	thanel@ferrovial.com
Raúl García-Castro	UPM	rgarcia@fi.upm.es

Version History

Version	Editors	Date	Comment
0.1	NT	01.09.2022	ToC and initial version
0.2	NT	01.11.2022	Draft version ready for internal review
0.3	UPM, FER	21.11.2022	Internal review
0.4	NT	22.11.2022	Review comments addressed
0.5	NT	28.11.2022	Final version
1.0	NT, Hypertech	30.11.2022	Submission to the EC portal

Disclaimer

Executive Summary

The COGITO deliverable D6.8 “Personalized On-Site Works Support Apps v2” aims at documenting the second iteration of the WOEA application initially presented in the previous version of this deliverable, D6.7. WOEA is considered as a main on-site tool for workers, foremen, quality surveyors, surveyors and HSE supervisors to manage, monitor and report the status of the tasks assigned to them as part of the whole construction process.

The Work Order Execution Assistance tool, WOEA, is part of an application toolset delivered under the I3D platform[1] adjusted to fit the needs of COGITO. The application is a derivation of the currently developed mobile device application, mostly focusing to be used with Smart Glasses. Other mobile devices like Android smartphones or tablets are also supported. The development and delivery of WOEA’s functionalities were driven by the stakeholders’ requirements identified in the deliverable D2.1 “Stakeholder requirements for the COGITO system” and the functional and non-functional requirement documented in D2.5 “System Architecture v2”. The second version of WOEA aims to deliver a bundle of functionalities enabling the handling of end-users’ interactions while executing the tasks assigned to them. In this version, the following functionalities are provided:

- Integration with COGITO’s Digital Twin Platform - user authentication using COGITO’s Identity Provider;
- Overview of workorders/tasks assigned to a user;
- Accessing detailed information about the assigned workorders/tasks;
- Assistance during execution of the tasks;
- Integrated tool for progress reporting; and
- Support for offline execution.

The notification and in-task object recognition features have been removed as they fall outside the scope of the project. Also, possibilities to edit work orders as well as the user role assignment have been disabled – such functionality is offered by other COGITO tools. The application has been simplified removing unnecessary libraries and frameworks.

In this second, final version of WOEA’s report, the application’s integration with the overall COGITO ecosystem in terms of realising all the interactions with other COGITO components is provided.

[1] https://i3d.econtentstore.com/

Table of contents

List of Figures

List of Tables

List of Acronyms

Term	Description
COGITO	COnstruction Phase diGItal Twin mOdel
DTP	Digital Twin Platform
HSE	Health, Safety & Environment
ICT	Information and Communications Technology
QC	Quality Control
RAMS	Reliability, Availability, Maintainability and Safety
UI	User Interface
WODM	Workflow Management Automation Tools
WOEA	Work Order Execution Assistance tool
WP	Work Package

1 Introduction

1.1 Scope and Objectives of the Deliverable

This document aims to report on the work that was carried out in the context of "T6.4 – Personalised On-site Works Support and Relevant Apps Development" to develop and deliver the final release of a Personalized On-site Works support Application, hereafter named Work Order Execution Assistance (WOEA) application, to support activities during the construction work. Intended to be used by on-site construction stakeholders, the final version is being developed to be an application for workers that provides functionalities to assist them on reporting work progress and alert them for hazardous components and areas. On-site assistance is provided in various ways, e.g., (1) managing work orders and the corresponding tasks assigned to users; (2) getting acquainted with the necessary information e.g., RAMS or health & safety instructions; (3) reporting work progress and issues; and (4) providing important communication from the construction manager or foreman.

The WOEA application builds upon existing I3D industrial services, provided under an existing platform of integrated ICT tools and services that aim to support the whole life cycle of projects, from definition of process templates, over issuing task based on the defined templates to execution and reporting. The overall architecture of I3D is described in Figure 1. In COGITO the I3D system is adjusted according to the requirements defined in D2.1 to provide functionality of executing the workflow and on-site guidance for workers and on-field stakeholders. WOEA in I3D system operates as one of the two frontends to the main system which is in COGITO environment WODM. Basic overview about I3D architecture is presented on Figure 1. More detailed information of I3D environment is provided in "D6.6 – Adaptive Workflow Management and Automation Tool v2".

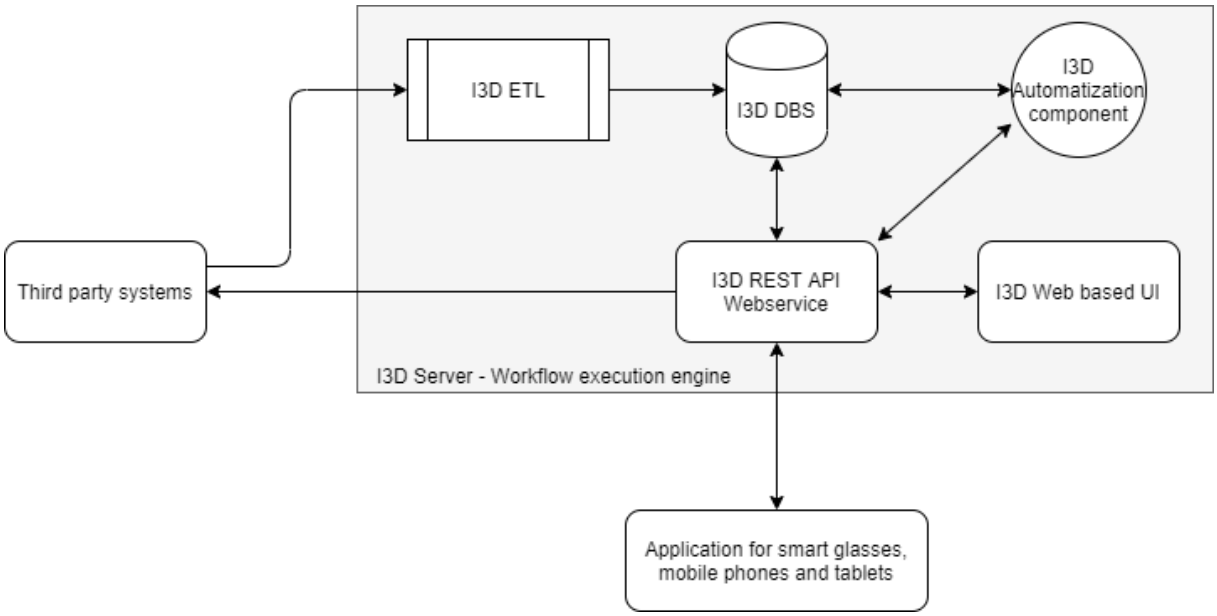


Figure 1 – Architecture of I3D

1.2 Relation to other Tasks and Deliverables

This deliverable is closely related to the task "T6.3-Adaptive Workflow Management and Automation" and its deliverable "D6.6-Adaptive Workflow Management and Automation tool v2". It is also related to other WP6 tasks, namely "T6.1-Blockchain & Smart Contracts on the Workflow Modelling and Management ", T6.2 "Adaptive Processes/Workflow Modelling and Simulation-based Optimization", and their deliverables, and to task "T7.1-Digital Twin Platform Design & Interface Specification" and its deliverables. The end-user requirements for WOEA were gathered and described in "D2.1-Stakeholder requirements for the COGITO system". Furthermore, the specifications, the functional and non-functional requirements, as well as the interactions of WOEA with other components of the COGITO ecosystem are presented in "D2.5-System Architecture v2".

1.3 Structure of the Deliverable

This deliverable contains the following sections:

- Prototype overview, where the WOEA application is introduced,
- technology stack and implementation tools used,
- input, output and API documentation,
- usage walkthrough,
- licensing information,
- installation instructions,
- development and integration status overview,
- requirements coverage, and
- assumptions and restrictions.

1.4 Updates to the first version of WOEA

Building on top of the first version of WOEA (as presented in the first version of this deliverable, namely D6.7), this second version offers new, additional features and functionality. The screenshots and the usage walkthrough presented are based on the “School Project” datasets obtained from the DTP as explained in the corresponding sections below.

First of all, all frameworks and libraries have been updated to the current versions, the application has been reassembled taking into account the removed functionalities and ensuring the cleanliness of the code. Also, the application has been prepared to be published in the Google Play portal.

Another new feature is the integration with the DTP’s Identity Provider service. Despite the original idea of Identity Provider integration proposed in D6.7, a different approach has been chosen, i.e., to provide more transparent and efficient way of DTP - WODM - WOEA integration. To provide the potential of simple login method - especially on smart devices without keyboard - QR code login has been implemented. Also, we removed activity counter to re- login after 10 minutes of no-activity as it could be disturbing for users during implementation and testing.

Finally, the application graphical design has been updated in accordance with the COGITO design templates.

2 Work Order Execution Assistance tool

2.1 Prototype Overview

For the WOEA application's development, the Unity3D Engine has been used. It has been selected due its multiplatform nature, providing out-of-the-box support and deployment to various operating systems and devices. This allows to maintain a single version of the code for several types of hardware. Currently, the application is being designed to run on Smart Glasses, but some Android smartphones and tablets have been successfully tested. Since Unity3D is widely used, future support for devices to be launched in the market is highly expected. The code is written in C# programming language. Key features of WOEA that have been considered and are already implemented include:

- Secured access to and provision of information that is being exchanged between various construction stakeholders;
- Personalised display of data according to the credentials entered to login, (e.g, projects and list of tasks assigned to the logged-in user);
- Display of guidelines and description of actions that should be considered for a successful completion and delivery of a selected task;
- Provision of various means for reporting progress on pre-defined tasks and identified issues that affect the completion of a workorder;
- Provision of functionalities for operation with voice commands or gestures (not for android smartphones and tablets).

For the early testing and refinement of all WOEA's functionalities, the application is being initially deployed on a smart wearable tablet, Realwear HMT-1, a fully rugged Android computer that is worn on the helmet and replaces the touch screen with an articulated micro display. Additional features such as hard-hat compatibility, rugged drop and waterproof design and outdoor display makes it a very suitable candidate to be used on construction sites. The selected wearable device allows the on-site crew to maintain full situational awareness and maximum productivity. This device is a current standard in heavy duty industrial environment.

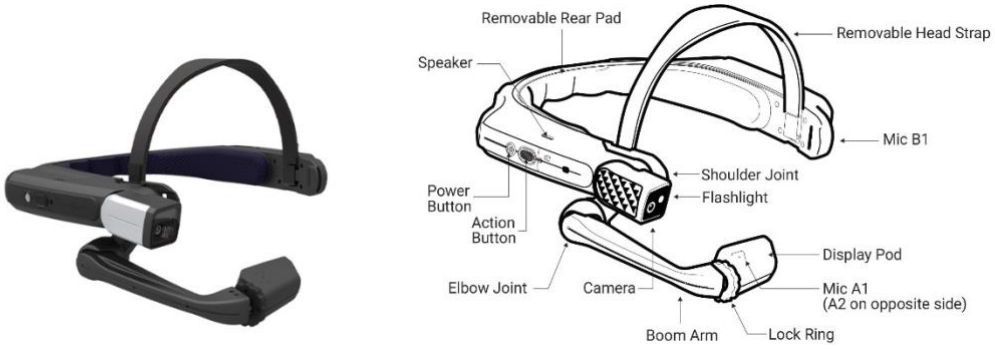


Figure 2 – RealWear HMT-1 headset[2]

WOEA is configured to utilise the built-in voice recognition feature, which gives the user the ability to operate applications and the device mostly solely by voice commands.. Besides this preferred device, the application is ported also to Microsoft Hololens2, even this device is not suitable for on-field tasks.

2.2 Technology Stack and Implementation Tools

WOEA is developed as a part of the I3D system, originally serving as a UI for on-field stakeholder support. The development is performed using the Unity3D engine and the C# programming language. To ensure wide compatibility with different devices and replicability, we tried not to use too many proprietary libraries and we

[2] https://www.realwear.com/hmt-1/

mostly used the standard features that came preinstalled in Unity. However, we still needed to import some specific libraries for some of the features:

- **NatMic 1.3.0** – Native Microphone API;
- **html-agility-pack** 1.11.46 – Displaying the html components in the app UI;
- **OpenCV for Unity 2.3.7** – Object and character recognition used for QR code login;
- **NatCoder 1.5.0** – Straightforward way to record videos and capture other multimedia;
- **Runtime File Browser** 1.5.7– Simple file browser saved on the device;
- Mixed Reality OpenXR Plugin 1.6.0 – Library for implementation of AR;
- JetBrains Rider Editor 2.0.7 - C# editor for Unity

Unity3D: Unity is a Game Engine created by Unity Technologies. The engine is widely used to build Virtual reality and Augmented Reality application in entertainment and industrial sector.

Programming language C#: Programs based on Unity3D are written in the object-oriented programming language C#, language supports development of secure and robust applications that run in the .NET ecosystem.

Unity for Windows: To facilitate the development process, Unity for Windows is used, providing the possibility to test and readjust its functionalities locally without requiring any installation on wearable devices.

2.3 Input, Output and API Documentation

The first interaction occurs between the WODM and WOEA, and it revolves around the workorders that were created and assigned to specific workers. After being created in WODM UI by managers, the tasks contained in the work order are stored in the WODM's database. Once the worker logs into the WOEA, the communication between the WOEA and WODM happens in two layers.

On the first layer, the WOEA requests all workorders assigned to this specific worker from the WODM. The data that it receives is surface-level and not too detailed since the worker does not need all details at this step. It then populates the workorder data into the list of available work for the worker to choose from. Once the worker chooses and starts the specific workorder, the second layer of communication occurs. Here, the WOEA requests the detailed information about the specific workorder (including the detailed tasks to be performed), so that it is available for the worker in its entirety. This two-layered communication is designed to avoid the overload of the webservice and network. Both requests use the REST API and receive a JSON containing all the data (see Figure 3).

The next interaction occurs between the WODM/WOEA and the DT platform. Its main point is the authorisation and authentication of the user that is trying to use the system. The WODM currently supports this type of authorisation from outside services using the Identity Provider, and after a successful login, the user is automatically inserted into the database if he or she is not there already. The current process of logging into the WOEA is a little bit different. It doesn't contain any sort of support for the authorisation from an outside service, like the WODM does, so it strictly relies on the WODM database. This is a very straightforward and simple implementation, as it simplifies the WOEA logging system. To avoid necessity to be already registered in WODM, on WODM side there is a new webservice created which connecting WOEA with Identity Provider service in DTP. WOEA currently supports two ways of authentication. The first one is called "developer access" and it is used for login outside from Identity Provider directly to WODM, especially for development tasks or testing purposes. The second one is "COGITO login" which connecting WOEA and Identity Provider in DTP via WODM webservice. There is not necessity to be already logged in WODM to be logged to WOEA as in previous version of application. There is added new functionality, QR code login, especially useful for Smart Glasses which using internal camera and object recognition service to quick login using QR code. This QR code currently contain authentication data of user in Identity Provider. In future development, IoT personal tag will be used to authenticate user to WOEA when it will be linked with Identity Provider credentials.

```
{
   "process_id":null,
   "i3d_id":"i3d-wo-cogito_prototype-10",
   "execution_status":"Created",
   "name":"Test:WOL LEVEL 4",
   "storey":"",
   "apartment":null,
   "plannedStartDateTime":"2022-07-04 15:04:00",
   "plannedFinishDateTime":"2022-07-04 16:04:00",
   "actualStartDateTime":null,
   "actualFinishDateTime":null,
   "related_project":null,
   "creationDate":"2022-07-04 15:06:35",
   "version":1.1,
   "space_id":null,
   "space_name":" ",
   "provider_id":"i3d-prov-cogito_prototype-8",
   "manager_login":"i3d-prov-cogito_prototype-8",
   "tasks":[
      {
         "i3d_result_id":"i3d-task-cogito_prototype-65",
         "id":null,
         "name":"Piles and Caps",
         "result":"Empty",
         "provider_id":"i3d-prov-cogito_prototype-8",
         "start":null,
         "duration":0,
         "planned_start":null,
         "planned_end":null,
         "real_start":null,
         "real_end":null
      },
      {
         "i3d_result_id":"i3d-task-cogito_prototype-76",
         "id":null,
         "name":"Piles and Caps",
         "result":"Empty",
         "provider_id":"i3d-prov-cogito_prototype-8",
         "start":null,
         "duration":0,
         "planned_start":null,
         "planned_end":null,
         "real_start":null,
         "real_end":null
      },
      {
         "i3d_result_id":"i3d-task-cogito_prototype-78",
         "id":null,
         "name":"Concrete Rd Columns",
         "result":"Empty",
         "provider_id":"i3d-prov-cogito_prototype-8",
         "start":null,
         "duration":0,
         "planned_start":null,
         "planned_end":null,
         "real_start":null,
         "real_end":null
      }
   ]
}
```


Figure 3 – WODM to WOEA Workorder JSON

After the worker using the WOEA finishes a task of a workorder, all information gets serialized into a JSON file, which is then sent via the REST API directly to the WODM. There are two JSONs to be sent to WODM. One is the task progress, and it is sent after finishing the task (Figure 4).


```
{
  "workorder_result_id": 1,
  "answer_type_id": 1,
  "result_value": 1,
  "text": "it went well"
}
```


Figure 4 - Task status

The second one is information about Workorder status. This information is sent when workorder is finished (Figure 5).


```
{
  "workorderId": 1,
  "execution_status_id": 1,
  "startTime": "1/1/20020"
}
```


Figure 5 - Workorder status

It then deserializes the data and stores it into the database. The WODM generally needs to know the result of the task, the result text (if the worker wrote any), the result value (if requested by the manager), any captured multimedia, and the execution start and end time. Execution end time is calculated in WODM when it receives Workorder status. Those data synthesize an instance of a Workorder Result. Aside from the result, there is a brief information exchange any time the task changes its state on any of both sides.

2.4 Usage Walkthrough

Prerequisite to access the WOEA App is a user account that must be created in WODM or DTP. WODM is providing an internal identity and access management system and Identity Provider (DTP) access management

as well. Unique credentials are generated and should be communicated to the end-user to login the WOEA App (see Figure 6). User login name is usually their e-mail address. WOEA does not support any mechanism to manage user passwords. Any change to the user credentials should be done through WODM or DTP, depends on type of login.

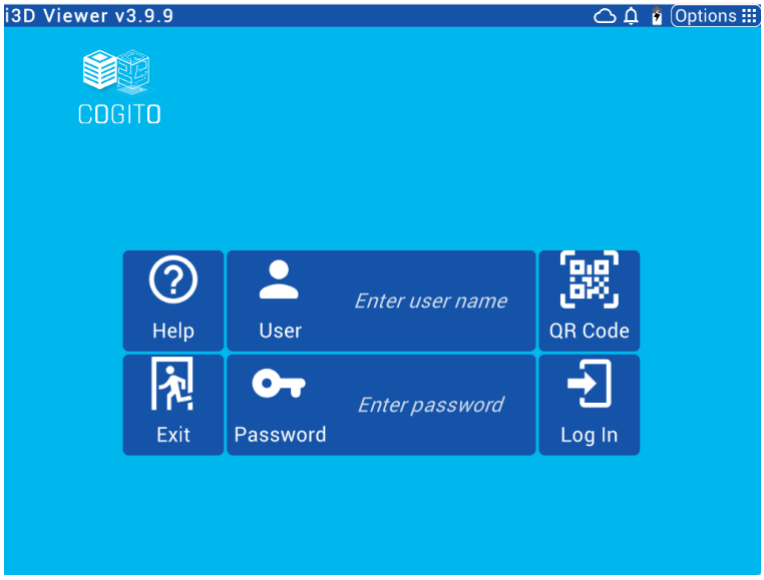


Figure 6 - Login screen

For testing purposes there is possibility to login directly to WODM outside of DTP. For that, there is an option to switch to “Developer mode”. (Figure 7)

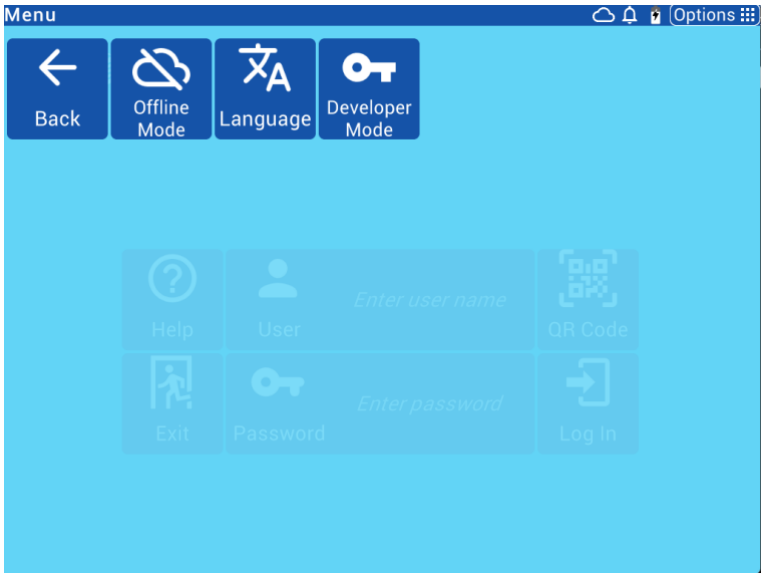


Figure 7 - Developer mode

After a successful login, the user selects a project from a list of running/current user projects (Figure 8) accessible to him. The list of projects depends on type of login, for the projects which are stored in the WODM backend there will be list of all projects valid for COGITO development, including testing and temporary projects. When login is done via Identity Provider, list of the projects valid for the user in DTP will be shown.

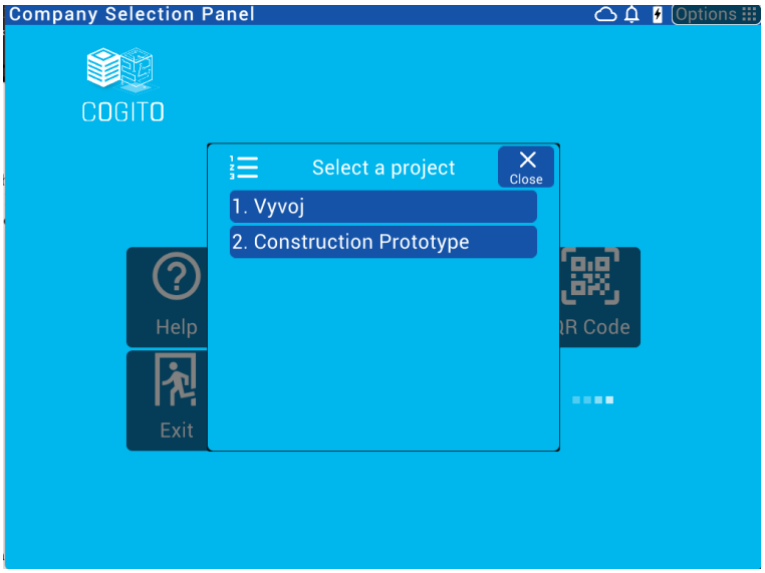


Figure 8 – Projects' list and selection

In the previous version of the application, the application had an idle counter of 10 minutes, after which the application locked the screen, and the user had requested to re-login to further use of the application. According to user experiences in other projects, we decide to remove this counter. The user can lock the screen or log out from the application intentionally (Figure 9).

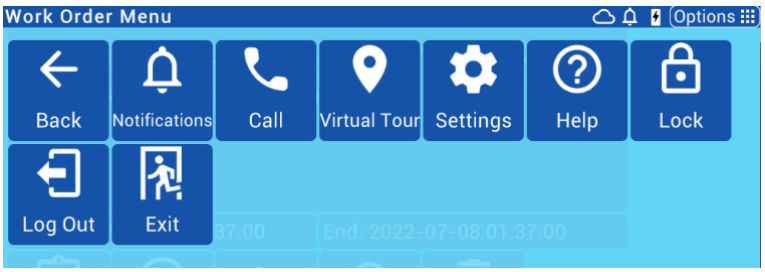


Figure 9 – Lock / Exit /Logout screen

WOEA is designed to be adjusted to the needs of the user (Figure 10). The user can switch the language of the application, change video recording quality, and set the application to offline mode. In this mode, only the work orders previously downloaded are available. The user can execute the pre-downloaded work orders in offline mode. In that case, the report and evidence of work will be synchronised with WODM after the device is re-connected to the internet. It does matter for devices without mobile connection like HMT-1.

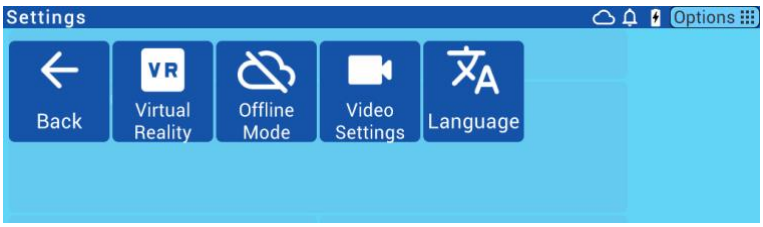


Figure 10 – Settings

After project selection, the list of workorders assigned to the user is displayed. For every workorder, the basic information about expected date and time of execution is displayed (Figure 11).

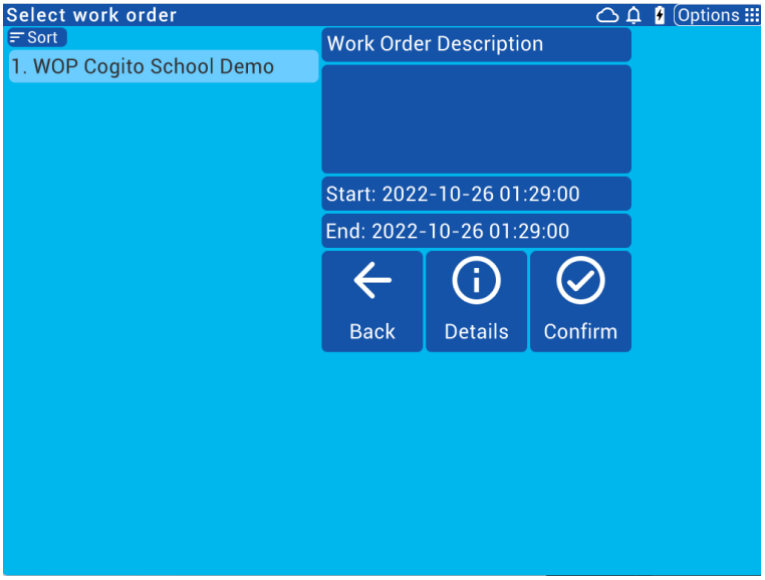


Figure 11 – List of assigned workorders

The user can check the details of any of the assigned work orders (Figure 12), download them to the device for offline use (button Download) or start the execution of the chosen workorder (button Start) (Figure 13).

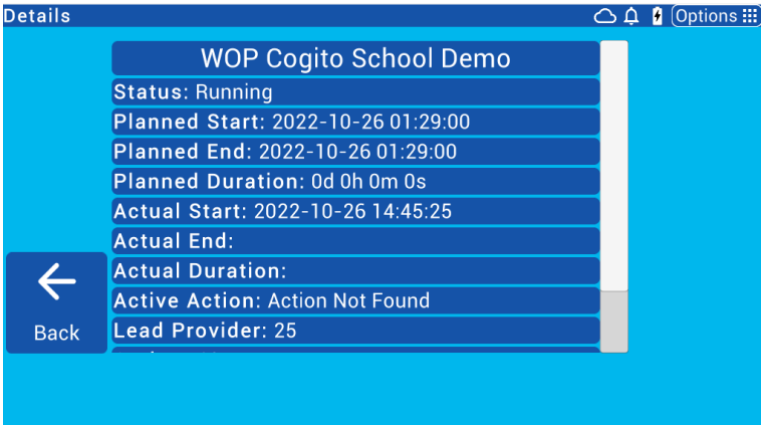


Figure 12 – Details of an assigned workorder

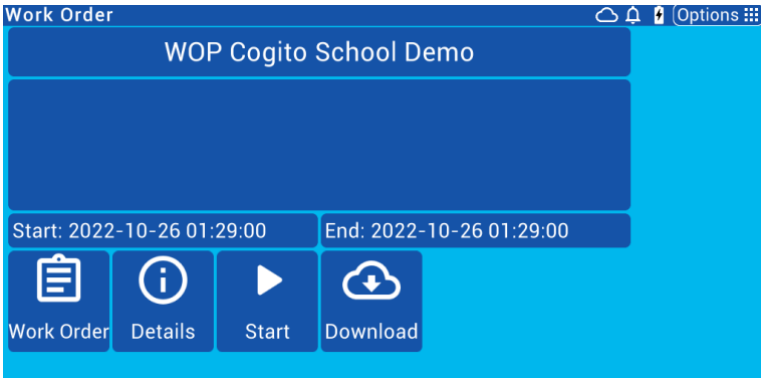


Figure 13 – Workorder screen and menu

After the worker starts the execution of a workorder by clicking on “Start” (Figure 13), the system guides them step by step through the tasks of the workorder. The worker is asked to execute the popped-up action (Figure 14), while detailed description of the work to be done is displayed.

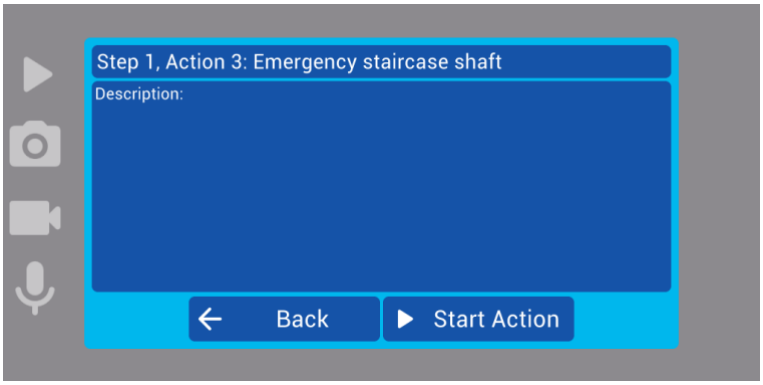


Figure 14 – The start of the task execution

Once the worker starts the execution of the task, the application allows the attachment of any kind of multimedia information to the report as evidence of work completion (Figure 15). These multimedia files are linked to the tasks of the workorder. When the worker finishes the execution of the task, he has to click on stop button on the left side of screen or use voice command “stop”. Then application requires from the worker to set the status of the task (Figure 16). For every task a text note can be attached, in case that a numerical value is expected (i.e., quality assurance tasks), the value can be entered to dedicated field (this screenshot is not shown as school project doesn’t contain such type of task currently).

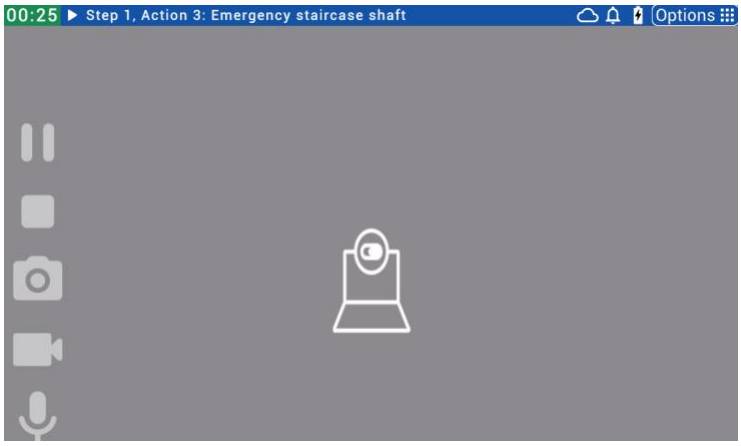


Figure 15 – Task in progress

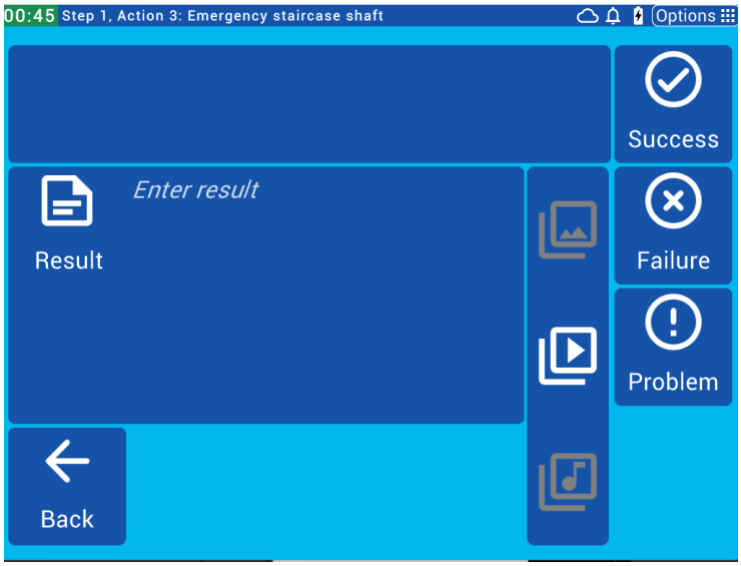


Figure 16 – Task result confirmation

The user can check the list of tasks of the actual workorder, in which the status of tasks execution is indicated (Figure 17).

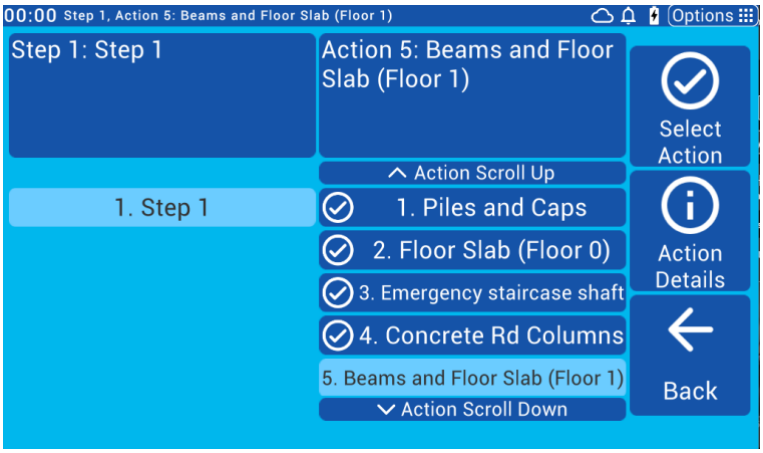


Figure 17 – List of tasks of the workorder

When all tasks from the workorder are finished, or, for some reason the worker needs to quit execution, the application requires to set the status of the workorder (Figure 18). Available options are: Completed, Aborted or Paused.

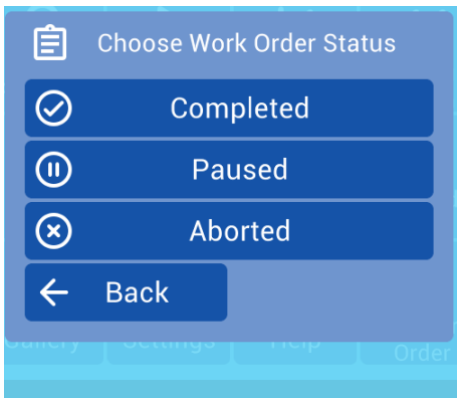


Figure 18 – Possible workorder statuses

The execution of tasks is supported by additional tools (Figure 19), like Remote assistance where user can call to another user (this communication works only within WOEA users).Other buttons are for easier navigation in the list of assigned tasks. Button for Notifiacation is inactive as it is prepared for future notification system to be defined in next phases of the COGITO project.

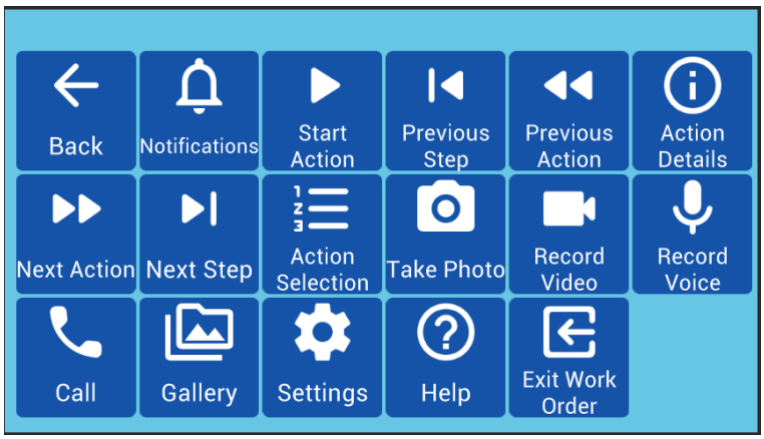


Figure 19 – Tool set for task execution

The user also can customise the quality of the video recordings based on the actual preferences (Figure 20). In case a short video is required to document the status, high quality of the video recording is recommended. In contrast, if a long process is recorded for evidence, it is recommended to record in medium or low quality to keep a reasonable size of the recording. The resolution of the three predefined qualities is 854 x 480px for High Quality, 512 x 288px for Medium Quality and 256 x 144px for Low Quality.

Automated video recording of all actions is also supported by tapping on “Record each action. The user can deactivate this option at any time.

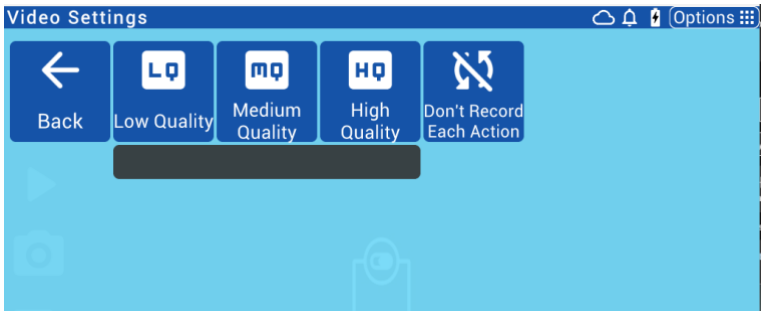


Figure 20 – Video quality settings

2.5 Licensing

WOEA is provided as a closed source software component.

2.6 Installation Instructions

WOEA application can be downloaded through the following links:

- WOEA for Realwear HMT-1 and other Android devices (operating system Android 10.0): WOEA Android
- WOEA for Windows desktop: WOEA for Windows

To install WOEA on Android devices, download the .apk file. Use a file browser application to reach the folder where the downloaded file is located. Tap on the .apk file. The installation process will start. For Windows desktop installation, unzip the downloaded compressed file to any folder and run the “I3D Viewer.exe” executable. The version for Windows is designed for development purposes only, when you first start the

application, Windows warning about non- approved application will be shown. Proceed further and there will be no messages again.

The following credentials are being provided for early testing purposes in developer mode (Figure 7):

- Login: `CogitoUser`
- Password: `rdh486o38qw9sf4jz`

Also, user with Identity Provider credentials (DTP) and assigned project can login or use temporary demo user (Identity Provider mode):

- Login: `info@novitechgroup.sk`
- Password: `qB6USqBGJ8Wed4A`

2.7 Development and Integration Status

The application is ready for use. All features related to work order management and task reporting are functional and have been tested using the example "School project"; the latter is commonly used by the various COGITO tools during/after their development for testing/verification purposes. During the integration phase in WP8 adjustments will be conducted to address potentially new requirements; the latter also applies during the deployment, namely addressing any pilot partner-/end-user-specific needs. After these modifications and the selection of the target platform for WOEA, the application will be published in an Android repository for safe and quick installation.

2.8 Requirements Coverage

In Table 1, the computing requirements extracted from the deliverable "D2.1-Stakeholder requirements for the COGITO system" are listed. Requirement COGI-CS-9 has been achieved in this version of application. COGI_CS-2 is achieved via Windows Unity application, even if this is not preferred way of use. COGI-CS-3 is achieved for Android devices (both mobile phones and tables). Apple devices are not supported due to licencing reasons, but technically it is possible to release an iOS version as well.

Table 1 – Computing system requirements

ID	Solution	Priority	Status
COGI-CS-2	runs on laptop and is usable on the construction site (remote access)	Should	Achieved
COGI-CS-3	runs on mobile phones or tablets, without brand restriction	Must	Achieved
COGI-CS-4	runs on Windows	Must	Achieved
COGI-CS-6	runs on Android	Should	Achieved
COGI-CS-7	allows access to the whole data in one location	Must	Achieved
COGI-CS-8	maintains communication and data security	Must	Achieved
COGI-CS-9	differentiates data and system access levels and modification rights	Must	Achieved

In Table 2, the requirements for Workflow Execution and Monitoring are extracted from the deliverable D2.1. COGI-WF-29 will be implemented after successful implementation and testing of all required functionalities.

Table 2 – Workflow execution and monitoring requirements

ID	Solution	Priority	Status
COGI-WF-1	allows the PM and Client to share information (design data, photos, videos, schedules, design issues, cost)	Could	Supported
COGI-WF-11	updates the activity status during work execution and monitoring	Should	Supported
COGI-WF-14	allows work progress reports	Must	Supported
COGI-WF-15	updates work progress weekly	Must	Supported
COGI-WF-18	displays only current information or document versions, related to the project, to all stakeholders	Must	Supported

COGI-WF-26	allows efficient reporting of work completion (using sensor data or simple app interface)	Could	Supported
COGI-WF-27	offers simple, easy to use, and intuitive interface to avoid workforce over-burdening	Must	Supported
COGI-WF-29	facilitates swift tool adoption by easily available video tutorials and other learning materials online	Would	In progress

In Table 3 the Functional and Non-Functional Requirements are extracted from the deliverable D2.5 ”COGITO System Architecture v2”. Req-1.1 has been achieved after the successful integration of the authentication mechanism with the DT platform’s identity provider/management service.

Table 3 – WOEA Functional and Non-Functional Requirements

Type	ID	Description	Status
Functional	Req-1.1	Connect and Authenticate to the DT platform (User login)	Achieved
	Req-1.2	Display assigned tasks	Achieved
	Req-1.2	Report progress of assigned tasks	Achieved
Non-Functional	Req-2.1	User-friendly	Achieved
	Req-2.2	Scalability	Achieved
	Req-2.3	Stability	Achieved
	Req-2.4	Multiplatform	Achieved
	Req-2.5	Security	Achieved

2.9 Assumptions and Restrictions

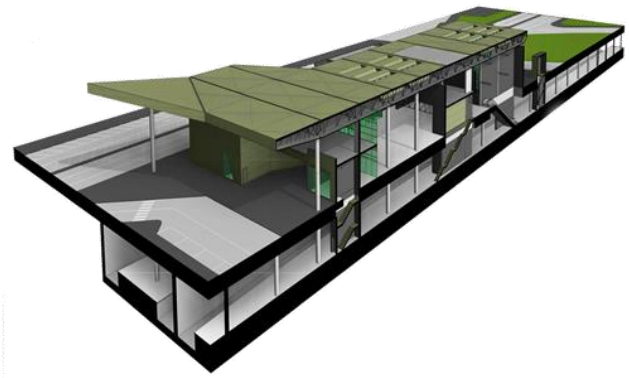
The second version of the WOEA tool is accompanied by the assumptions and restrictions briefly presented below:

- The application is ready to support construction tasks monitoring and progress reporting. Tasks related to Quality Control (QC) and Health Safety Environment (HSE) issues need to be tested with actual data (e.g., during the pre-validation phase).
- User authentication via QR code is ready for use, facilitating the authentication via e.g., QR codes printed on the IoT Tags assigned to workers. This functionality will be tested during the integration phase in WP8.
- Based on prior experience from other previous projects, it is highly required to offer mobile applications via trusted platforms (e.g., Google Play or Apple App Store). This will be done before the full-scale pilot roll-out phase.
- The application UI is currently offered only in English; nevertheless it is possible to create a multilanguage version depending on the target users.

3 Conclusions

This deliverable presents the second version of the WOEA application, a COGITO component offering a UI for on-site stakeholders (e.g., workers, foremen, quality surveyors, etc) to handle the various work orders; the latter are issued by COGITO's WODM component. More particularly, as described and presented in the provided example walkthrough, the application enables the documented execution of work orders along with monitoring of their execution process.

The application is currently ready for integration and deployment with the rest of COGITO's tools. Thanks to the multiplatform nature of the Unity engine used at the application's core, it is possible to deploy it to different target devices according to the project needs. Finally, there are provisions in the application's code to accommodate potential changes required by the end users during the demonstration and validation phase of the project, especially changes involving the UI and the IoT tags management implementation.



COGITO

CONSTRUCTION PHASE DIGITAL TWIN MODEL

cogito-project.eu

HYPERTECH energy labs

UCL

AARHUS UNIVERSITET

THE UNIVERSITY of EDINBURGH

BoC www.boc-group.com

iti

POLITÉCNICA

QUE TECHNOLOGIES

Novitech New Information Technologies

ASM

ferrovial construction

ΟΛΥΜΠΙΑ ΟΔΟΣ

RHOMBERG SERSA RAIL GROUP

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958310