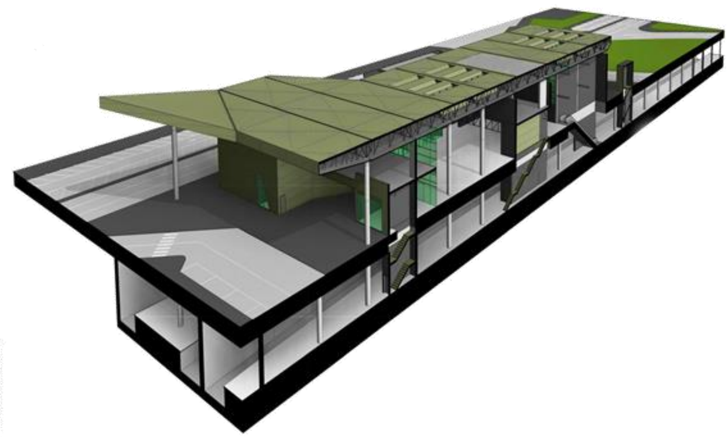
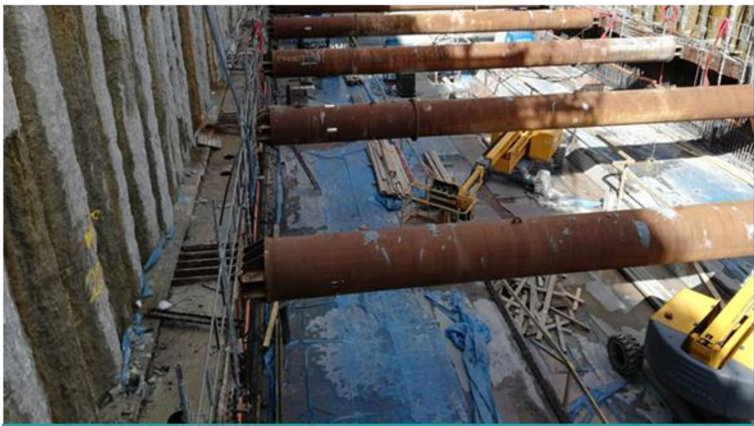
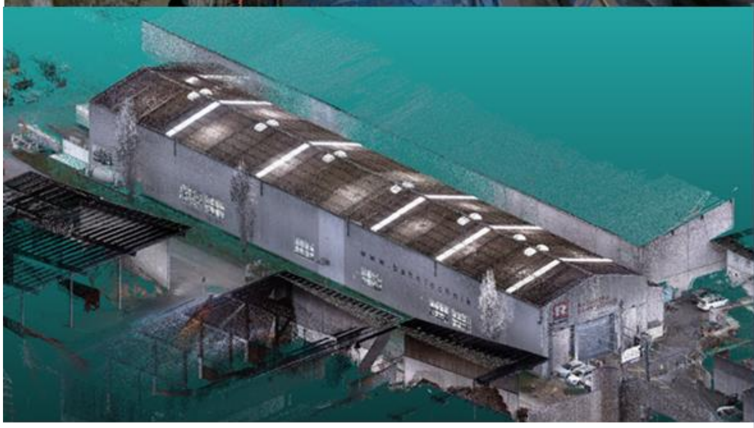






COGITO

CONSTRUCTION PHASE
DIGITAL TWIN MODEL

cogito-project.eu

D5.7 - User interface and AR enabled In-situ QC Visualisation v1

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

D5.7 – User interface and AR enabled In-situ QC Visualisation v1

Dissemination Level:	Public
Deliverable Type:	Demonstrator
Lead Partner:	CERTH
Contributing Partners:	Hypertech, UEDIN, FER, OLOD, RSRG
Due date:	31.05.2022
Actual submission date:	31.05.2022

Authors

Name	Beneficiary	Email
Thanos Tsakiris	CERTH	atsakir@iti.gr
Evangelia Pantraki	CERTH	epantrak@iti.gr
Apostolia Gounaridou	CERTH	agounaridou@iti.gr
Michalis Chatzakis	CERTH	mchatzak@iti.gr
Martin Bueno Esposito	UEDIN	martin.bueno@ed.ac.uk
Frederick Bosche	UEDIN	f.bosche@ed.ac.uk

Reviewers

Name	Beneficiary	Email
Agnieszka Mikołajczyk	ASM	a.mikolajczyk@asmresearch.pl
Bohus Belej	NT	belej@novitechgroup.sk

Version History

Version	Editors	Date	Comment
0.1	CERTH	15.03.2022	Table of Contents
0.2	CERTH	04.05.2022	Update ToC, Section 1, Sub-section 2.1.
0.3	CERTH	09.05.2022	Sub-section 2.3.1, 2.3.2, Sub-section 2.8
0.4	CERTH	13.05.2022	Sub-section 2.3.3, Sub-section 2.4, 2.5
0.5	CERTH	17.05.2022	Submission for peer review
0.6	ASM, NT	23.05.2022	Internal review complete
0.7	CERTH	27.05.2022	Review comments addressed
0.8	CERTH	27.05.2022	Final version
1.0	CERTH	31.05.2022	Submission to the EC portal

Disclaimer

Executive Summary

The COGITO Deliverable D5.7 “User interface and Augmented Reality (AR) enabled In-situ QC Visualisation v1” documents the first version of the COGITO Digital Twin visualisation with Augmented Reality (DigiTAR) module and focuses on the development activities concerning the T5.4 “User Interface for Construction Quality Control”. Overall, DigiTAR operates as a Graphical User Interface (GUI) for visualising construction Health/Safety and Quality Control (QC) results as well as for on-site visual data capturing and pre-processing.

The DigiTAR module has three main functionality modes: the QC mode, the Safety mode, and the Pre-processing mode. Within the fore mode, the user can visualise the QC results provided by the Geometric and Visual QC COGITO components in situ in order to confirm the automatically detected defects and propose the necessary remedial works. In addition, by selecting the Safety mode, the potentially hazardous areas generated by the Safety-related COGITO components can be displayed on-site, in order to confirm them and propose additional mitigations. Furthermore, DigiTAR could be considered as the GUI of the Visual Data Pre-processing module; hence, the user is able to create a new pre-processing job and add the related information (e.g., filter parameters, capture device properties). Finally, since DigiTAR runs on Hololens, it is considered as a data acquisition tool, which captures visual data on-site in order to forward it for pre-processing and further Quality Control. The DigiTAR module overall architecture is analysed in this deliverable. The tool is composed of three main layers: the main Application layer, the Viewers’ layer and the Communication layer. The first oversees implementing the main functionalities of the tool, while the second one is responsible for visualizing 2D and 3D data on site; the last one manages the communication with other COGITO components.

The present documentation of the COGITO User interface and AR enabled In-situ QC Visualisation, along with its sub-components, is oriented towards the functionalities they broadly deliver, the technology stacks they build upon, the inputs, outputs and APIs they expose, the installation instructions, the assumptions and restrictions, the usage walkthrough, the development and integration status, and the requirements coverage. In the second release, more functionalities will be implemented (i.e., for Safety-related results) or existing functionalities will be refined.

Table of contents

List of Figures

List of Tables

List of Acronyms

Term	Description
AR	Augmented Reality
COGITO	Construction Phase diGItal Twin mOdel
DigiTAR	Digital Twin visualisation with Augmented Reality
DT	Digital Twin
GUI	Graphical User Interface
H&S	Health and Safety
IFC	Industry Foundation Classes
QC	Quality Control

1 Introduction

1.1 Scope and Objectives of the Deliverable

This deliverable reports on the work conducted from M12 to M19 on the User Interface for Construction Quality Control that is developed as part of task T5.4. The scope of the Digital Twin visualisation with Augmented Reality component (DigiTAR) is to visualise the results of the Quality Control (QC) components (Geometric and Visual) as well as potential construction site hazards (Health and Safety issues, H&S) in-situ. In addition, DigiTAR is in charge of collecting as-built data and implementing part of the visual data pre-processing on-site. Subsequently, DigiTAR operates as a Graphical User Interface (GUI) for both construction Quality Control and on-site visual data pre-processing: (a) The stakeholders can view and confirm the results generated by the QC and Safety components in-situ as well as propose remedial works and mitigations for them respectively, (b) they can capture image data on-site, assign them to specific jobs and IFC components, prepare and display them after pre-processing; the data are eventually sent for Visual QC.

More specifically, this deliverable reports on the development of the first release of the DigiTAR component focusing on its main layers listed below:

- **Viewers Layer:** is in charge of visualising the necessary data (2D images and 3D IFC models).
- **Application Layer:** includes the basic functionalities of the application such as data capture, mode selection, job and report generation.
- **Communication Layer:** manages the communication and the data exchange between the DigiTAR component and other COGITO components (namely the Digital Twin (DT) Platform and the Visual Data Pre-processing module).

1.2 Relation to other Tasks and Deliverables

T5.4 “User Interface for Construction Quality Control” and consequently D5.7 “User interface and AR enabled In-situ QC Visualisation v1” are related to the following COGITO tasks and deliverables:

- The second version of the COGITO architecture in the corresponding deliverable “D2.5 COGITO System Architecture v2” provided an overview on the DigiTAR module, its requirements, and the communication with other components.
- The DigiTAR module, similarly to all components, relies on a shared ontology and a common data model developed within T3.2 “COGITO Data Model, Ontology Definition and Interoperability Design”; the second version of COGITO ontologies and data models have been documented in D3.3 “COGITO Data Model, Ontology Definition and Interoperability Design v2”.
- The DigiTAR module uses as input the results generated by the Geometric QC component to visualise and confirm on-site the detected geometric defects and their location (D5.5 “BIM-based Standard Test Methods for Geometric Quality Control v1”).
- The DigiTAR module uses as input the results generated by the Visual QC component to visualise and confirm on-site the detected visual defects and their location (D5.3 “Deep-Learning -based Visual QC component v1”).
- The DigiTAR module uses as input the results generated by the SafeConAI tool to visualise and confirm on site the identified hazardous regions (D4.1 “Preventive Health & Safety Application v1”).
- The DigiTAR module communicates with the Visual Data Pre-processing module to apply filters in 2D images captured by Microsoft Hololens on-site, preview and finalise the processed data that is finally sent for Visual Quality Control (D3.7 and D3.8 “Visual Data Pre-processing Module” v1 and v2 respectively).

1.3 Structure of the Deliverable

Remaining sections of this deliverable are organised as follows:

- Sub-section 2.1 presents the overall architecture of the DigiTAR module, introducing its sub-components and its workflow diagrams.

- Sub-section 2.2 describes the technologies, libraries and tools utilised for the implementation of this specific module.
- Sub-section 2.3 notes the inputs and outputs of the component as well as the APIs documentation.
- Sub-section 2.4 provides a brief manual (guidelines) on how to use this specific component.
- Sub-section 2.5 refers to information about the source code repository, the delivery form and the license of the component.
- Sub-section 2.6 describes how the DigiTAR module is accessible by the user.
- Sub-section 2.7 presents the status of the component and provides a plan regarding the integration procedure.
- Sub-section 2.8 notes the functional, non-functional and stakeholder requirements that are covered by the component.
- Sub-section 2.9 describes the assumptions already made for this component, as well as restrictions and measures that will be taken into consideration for the COGITO System Integration.
- The document concludes with Section 3, where the progress, the next steps and the contribution to the overall COGITO objectives are being reported.

2 Digital Twin visualisation with Augmented Reality (DigiTAR)

In this section, the DigiTAR module is presented in detail. In Section 2.1, the overall architecture of the component as well as the different sub-components it is comprised of is presented. Information about the implementation tools is presented in Section 2.2. The input and output data of the DigiTAR module are presented in Section 2.3, along with the API documentation. A detailed usage walkthrough is provided in Section 2.4. Information regarding the licensing of the DigiTAR module is presented in Section 2.5, while installation instructions are provided in Section 2.6. The development and integration status of the tool is discussed in Section 2.7. Finally, Sections 2.8 and 2.9 present the requirements coverage for this version of the DigiTAR module and some basic assumptions and restrictions, respectively.

2.1 Overall Architecture of the DigiTAR module

The DigiTAR module has three separate modes: i) the Quality Control mode, ii) the Safety mode and iii) the Visual Data Pre-processing mode. The first one is in charge of visualising the QC (Geometric and Visual) results on-site in order to be confirmed by the relevant stakeholders. Based on their decision, additional remedial works can be assigned to specific IFC components (Figure 1).

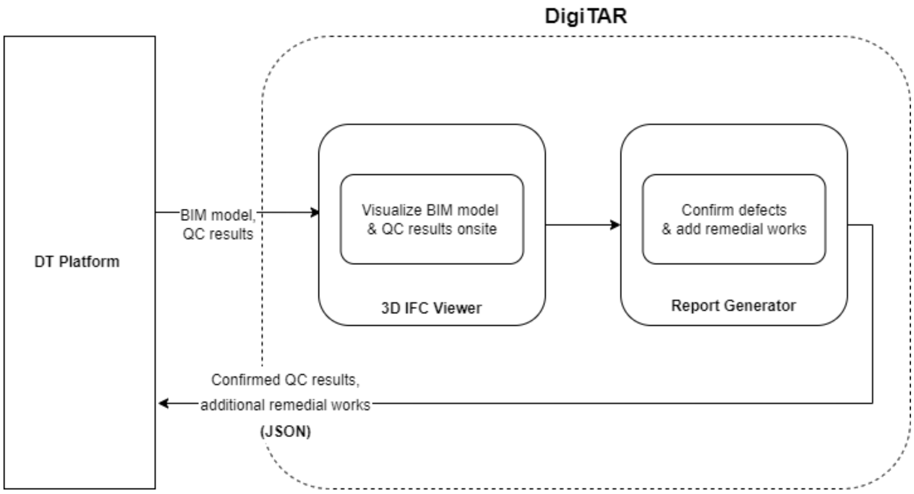


Figure 1 - DigiTAR Workflow: Quality Control mode

Furthermore, by selecting the second one, the generated H&S results can be also confirmed in-situ and thus the relevant stakeholder can view the potential hazardous regions of the construction site and propose additional mitigations (Figure 2).

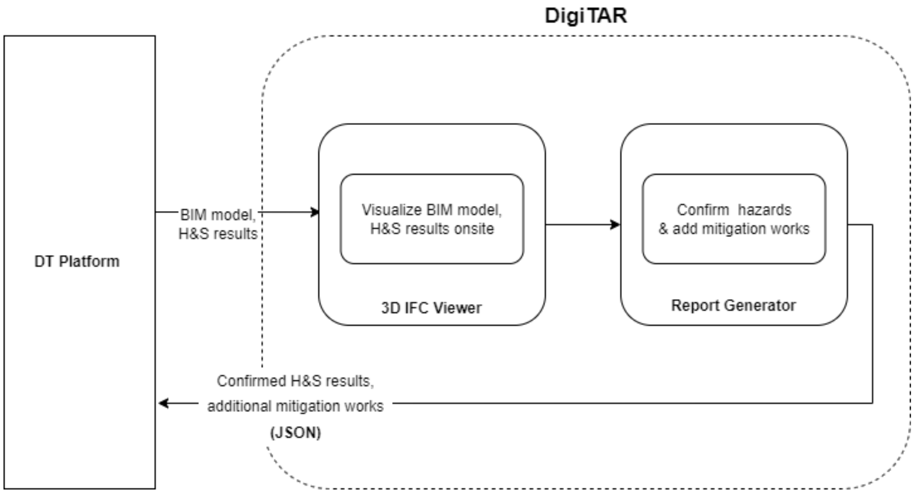


Figure 2 - DigiTAR Workflow: Safety mode

In addition, the Pre-processing mode provides a GUI facilitating data capture (i.e., images) as well as the performance of partial, on-site visual data pre-processing. Based on the work order definition provided by the WODM component, the user can create a new pre-processing job, associate it with specific IFC elements and link the latter with visual data (images) in order to be sent for Visual Quality Control. The DigiTAR module communicates internally with the Visual Data Pre-processing module in order to apply filters to the captured data. The processed data, along with the job metadata, are pushed from the Visual Data Pre-processing module to the DT Platform, where they can be assessed for further QC from the relevant COGITO components (Figure 3).

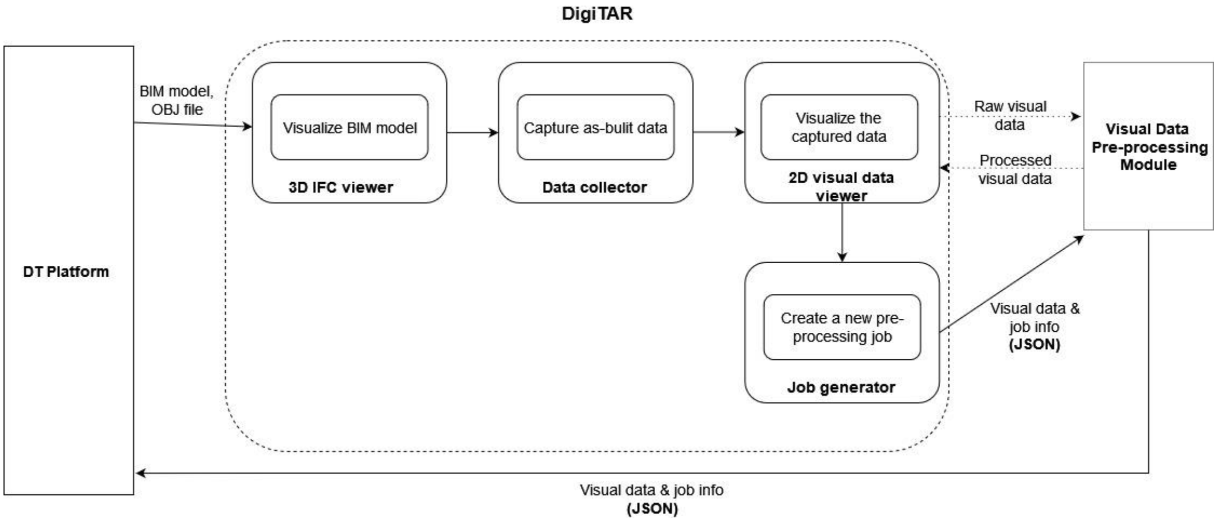


Figure 3 - DigiTAR Workflow: Pre-processing mode

The overall architecture of the DigiTAR tool is depicted in Figure 4. The application consists of three layers: the viewers' layer, the application layer, and the communication layer. The viewers' layer refers to the sub-modules developed for displaying existing data (2D images or 3D BIM models). The application layer includes all the sub-modules that handle the interaction of the user with the application. This layer provides access to the files received by the communication layer (e.g., QC results, H&S results) and displays this information to the users in the form of notifications or in the form of interactive AR menus. This layer is also responsible for handling the input of new data by the user, e.g., the confirmation of the QC results and the suggestion of a remedial work. Finally, the communication layer aims at connecting the DigiTAR module with the DT Platform and the Visual Data Pre-processing module to enable the necessary data exchange across them. All the aforementioned sub-modules and layers are further presented in the next sub-sections.

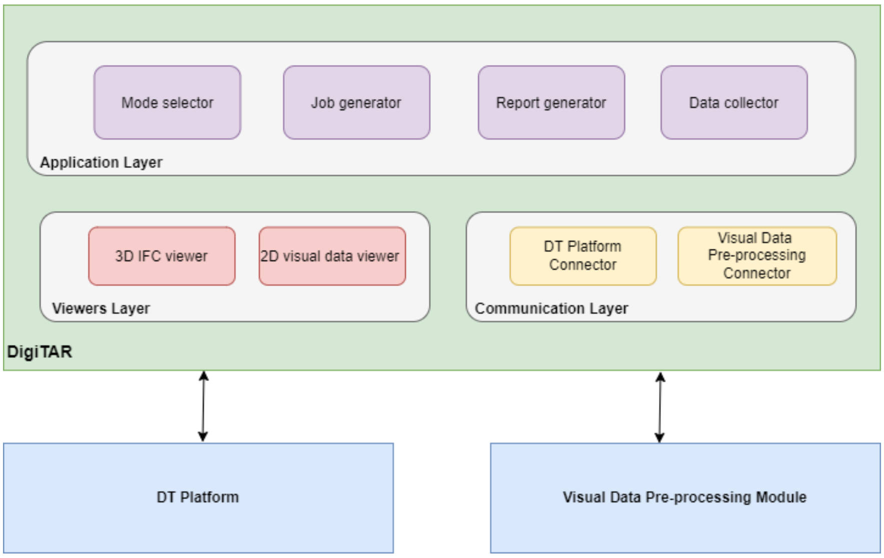


Figure 4 - Overall Architecture of the DigiTAR module

2.1.1 Viewers Layer

2.1.1.1 3D IFC viewer

The DigiTAR tool is responsible for displaying the 3D BIM model on-site. BIM model visualisation is a key functionality of the application as it is utilised by all three modes of DigiTAR; it is handled by the 3D IFC Viewer sub-module. The architecture of the 3D IFC Viewer component of the DigiTAR tool is graphically illustrated in Figure 5.

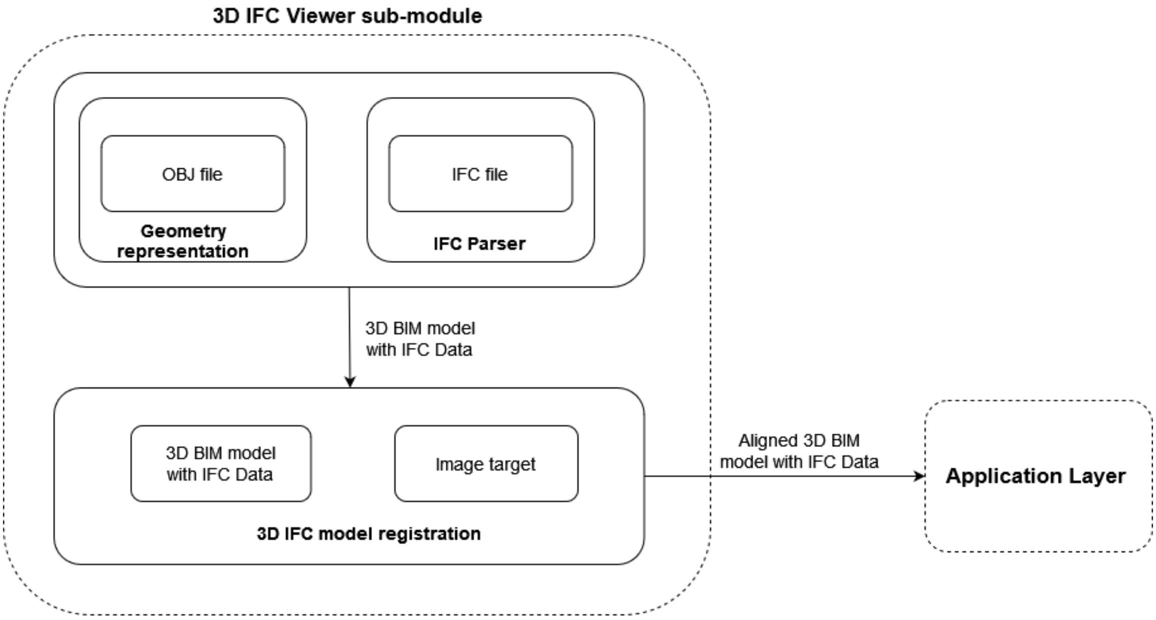


Figure 5 - Architecture of the 3D IFC Viewer component

The development of the DigiTAR tool is based on the Unity 3D Engine, as described in detail in Section 2.2. The Unity Game Engine does not support the generation of the 3D geometry directly from the IFC file. To overcome this obstacle, an OBJ file, that includes the geometric representation of the model, is generated from the IFC, and is imported, along with the IFC, as input to the DigiTAR tool. The 3D IFC Viewer of the DigiTAR tool requires both the geometry representation of the BIM model (OBJ file) and the parsed IFC data:

1. **Geometry Representation**: The Geometry representation of the BIM model is achieved through the transformation of the IFC file to a file format supported by the Unity Game Engine; such a format is the OBJ file format. The DigiTAR receives the OBJ file from the DT Platform, along with the IFC file. Figure 6 presents a visual representation of the OBJ file when imported in Unity.
2. **Parsed IFC Data**: Importing the IFC file into Unity, extracting the IFC data, and mapping those data to the 3D model are handled by custom C# classes based on the Xbim library [1].

Figure 6 - Visual representation of an OBJ file in Unity

The parsing process of the IFC file takes place in the IFC Parser sub-module. This process is recursive and can be divided in three tasks, as shown in Figure 7. The three subtasks are:

1. **GameObject Creation/Mapping**: In this task, the IFC Parser reads an IFC file, searches for the imported OBJ GameObjects based on the IfcGlobalId, and renames them with the name of the individual IfcElement. Since each IfcElement is uniquely referenced by an IfcGlobalId, each GameObject uniquely references a particular IfcElement. In the case of IFC elements as IfcProject or IfcSite that cannot generate geometry, the IFC Parser creates empty GameObjects with the name of the corresponding IfcElement, so that the hierarchy is complete. In the case of IFC elements as IfcDoor or IfcWall that generate geometry, the GameObjects of these IfcElements exist in the OBJ file (named with the IfcGlobalID of the corresponding IfcElement) and are renamed to hold the name of the corresponding IfcElement.
2. **Metadata Extraction**: In this task, the extraction of properties/materials quantities of each IfcElement takes place. All metadata corresponding to a particular IfcElement as retrieved by the queries are stored in a custom C# class and are added as scripting components to the corresponding GameObject.
3. **Hierarchy Creation**: In this stage, the IFC Parser creates the hierarchy of the BIM model by setting parent-child relationships between the GameObjects. The following relationships are used:

    a. IfcRelAggregates defines the spatial hierarchy between the IFC products,

    b. IfcRelAssignsToGroup associates the IfcSpaces to the IfcZones.

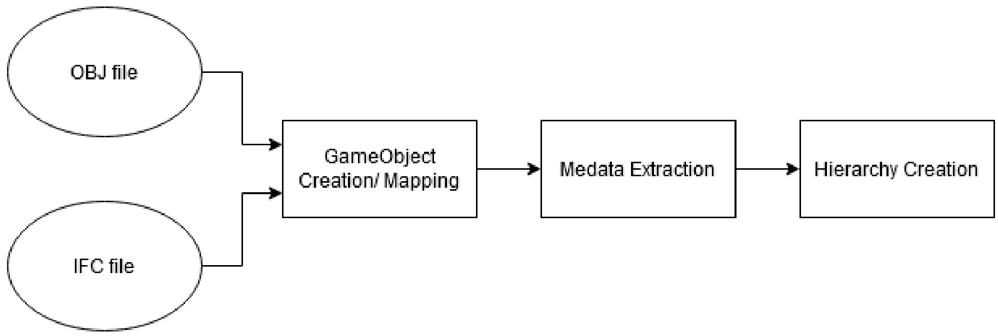


Figure 7 - Process of creating the IFC hierarchy in Unity

The connection between the geometry and the IFC information of the BIM model takes place in the IFC Parser sub-module. The parsing of the IFC and all the necessary data by the IFC Parser, is achieved using LINQ for optimal performance. LINQ [2] stands for a Language-Integrated Query and is a set of technologies based on the integration of query capabilities directly into the C# language. It implements deferred execution, which means it can chain the query statements. Figure 8 shows how the visualisation sub-module interacts with an IFC file through LINQ.

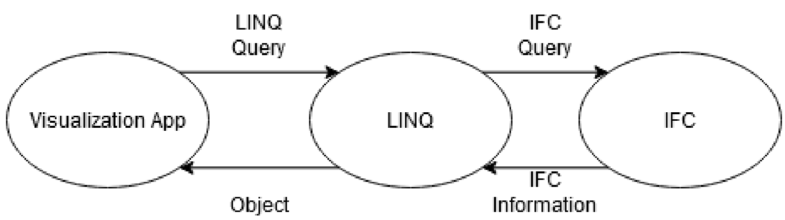


Figure 8 - LINQ to IFC

A query is generated by the IFC Parser and the retrieval of data from the IFC file is accomplished through LINQ. For this process to be successful, the correct query must be made, so knowledge of the IFC Schema is essential.

The purpose of the first query made by the IFC Parser is to find the IfcProject. Then, a recursive process begins:

- An empty GameObject is created and named after the IfcProject; its IFC information is added as an IFC Data Scripting component to the GameObject. In Figure 9, an example of the IFC information included in the form of an IFC Data component is shown.
- Then, through the relationship with other IFC elements the process starts over with the new next IfcElement – as defined in the IFC hierarchy, as a child GameObject to the parent GameObject previously created.
- If a GameObject with the same name as the IfcGlobalId of an IfcElement exists, then it is renamed with the name of the corresponding IfcElement and an IFC Data component with the IFC properties of the IfcElement, is added to the GameObject. The GameObject is set as a child to its parent IfcElement and the procedure is iterated until the whole IFC hierarchy is parsed.

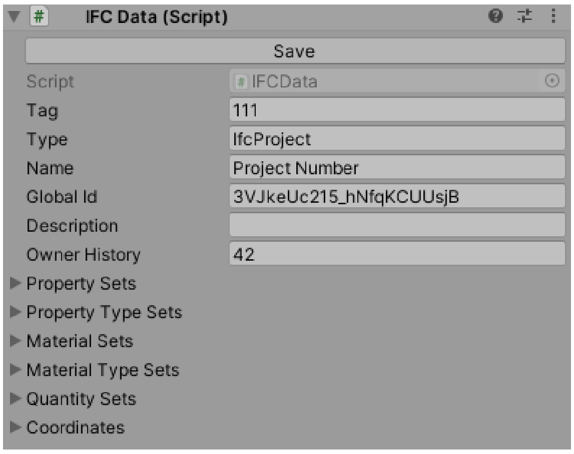


Figure 9 - IFC Data component

Upon completion of this process, the result is a hierarchical structure, where each GameObject has its own IFC Data component. Also, during this process, for each component a mesh collider is added. Colliders are necessary to interact with the GameObjects, e.g., to select them using Hololens gestures as discussed in detail in Section 2.4. Figure 10 shows the structure of the BIM model in the Unity *Hierarchy* tab as originally imported from the OBJ, while Figure 11 depicts the tree structure which is constructed as a result of the IFC Parser.

Figure 10 - Non-hierarchical structure of imported OBJ in Unity

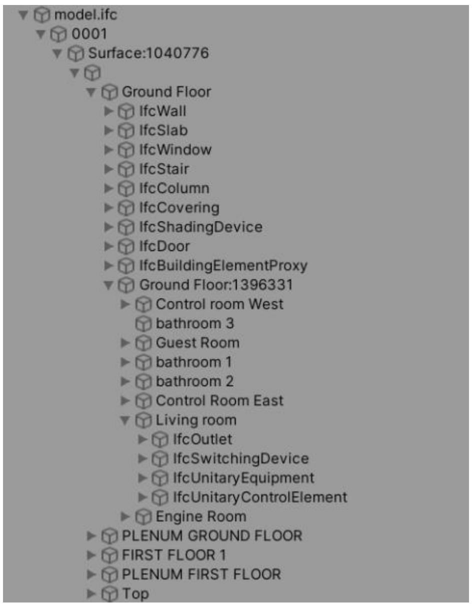


Figure 11 - Hierarchical structure created according to the IFC structure

The IFC fields that the IFC Data component stores are listed in Table 1.

Table 1 - IFC properties [3] that the IFC Data component stores.

Property	Description
Tag	The tag identifier at the particular instance of an IFC product.
IfcType	The type of an IfcObject (e.g., IfcSlab, IfcWall).
Name	The name of an IfcObject.
GlobalID	The global unique identifier attribute of an IfcObject.
Description	The description of an IfcObject.
Owner History	The IfcOwnerHistory defines all history- and identification-related information.
Property Sets	The Property Sets for Objects describes how an object occurrence can be related to a single or multiple property sets. A property set contains a single or multiple properties. The data types of an individual property are single value, enumerated value, bounded value, table value, reference value, list value, and combination of property values.
Property Type Sets	The concept template Property Sets for Objects describes how an object type can be related to a single or multiple property sets. A property type set contains a single or multiple properties. The data types of an individual property are the same as the data types of the Property Sets. The property values assigned to an object type apply equally to all occurrences of this object type.
Material Sets	The Material Association concept describes how a product relates to its associated materials that indicates the physical composition of an object. The material association can be of the Association type Material Single, Material Layer Set, Material Layer Set Usage, Material Profile Set, Material Profile Set Usage, and Material Constituent Set.
Material Type Sets	The Material Association concept that describes how a product type relates to its associated materials that indicates the physical composition of an object type. The material type association is the same as the material association in the Material Sets.
Quantity Sets	Contain all multiple quantity occurrences that relate to an object. The data type of quantity occurrence values is count, length, area, volume, weight, time, or a combination of quantities.
Coordinates	Refers to the world coordinates of an IfcObject calculated from the relative coordinates of the Project.

Figure 12 shows an example of a BIM model, as seen in the Unity Editor Scene. In this example, a particular component (an external wall) is selected. The created IFC hierarchy is depicted on the left (in the *Hierarchy* of the scene), while the IFC properties for the selected object are displayed on the right (in the *Inspector* section). As can be seen, the IFC Data component stores and displays the IFC properties for the selected object as listed in Table 1.

Figure 12 - Wall selection

After loading the 3D BIM model, the next step is to align the 3D model of the site to the actual site. This procedure is called **registration**. Within the DigiTAR application, registration relies on image targets. An image target is an image that the application running on the Hololens will detect and track. This image will be the link between the static 3D world (BIM model) and the real world.

The image target is printed and positioned at a location in the real world, ensuring that it is accessible to the person wearing the Hololens. At the same time, an identical image is placed in exactly the same spot in the 3D BIM model. To enable the detection of the image target, the user should use speech command “Scan”. This way, the data that are captured by the Hololens sensors and cameras are utilised for image target detection. More specifically, features are extracted from the Hololens camera stream and are compared to the reference features already extracted from the image target. In the context of pattern recognition, the features that are extracted in advance from the image target constitute the pattern that the algorithm searches across the continuous flow of data streams. When the person wearing the Hololens looks at the image target, the features extracted from the data stream of Hololens are matched to the pattern of features belonging to the image target. Therefore, the image target is detected. The image target used in the DigiTAR application, as well as an illustration of the extracted features, are depicted in Figure 13.

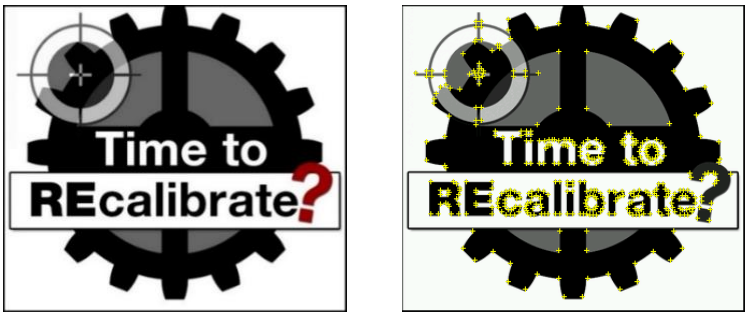


Figure 13 - The image target used in the DigiTAR application,
along with an illustration of the extracted features that are detected by the Hololens camera

After successful registration of the 3D BIM model and in order to maintain it, the registered 3D BIM model is continuously tracked. In the DigiTAR application, the registration of the 3D BIM model is tracked using **spatial anchors**; spatial anchors represent important points in the world that the Hololens coordinate system keeps track of over time. The registered 3D BIM model can be set as a spatial anchor using the speech command “Anchor”. This way, the next time a user opens the DigiTAR application, the 3D BIM model is loaded aligned to the real world without the need to repeat the registration process. The workflow of the registration process is graphically illustrated in Figure 14.

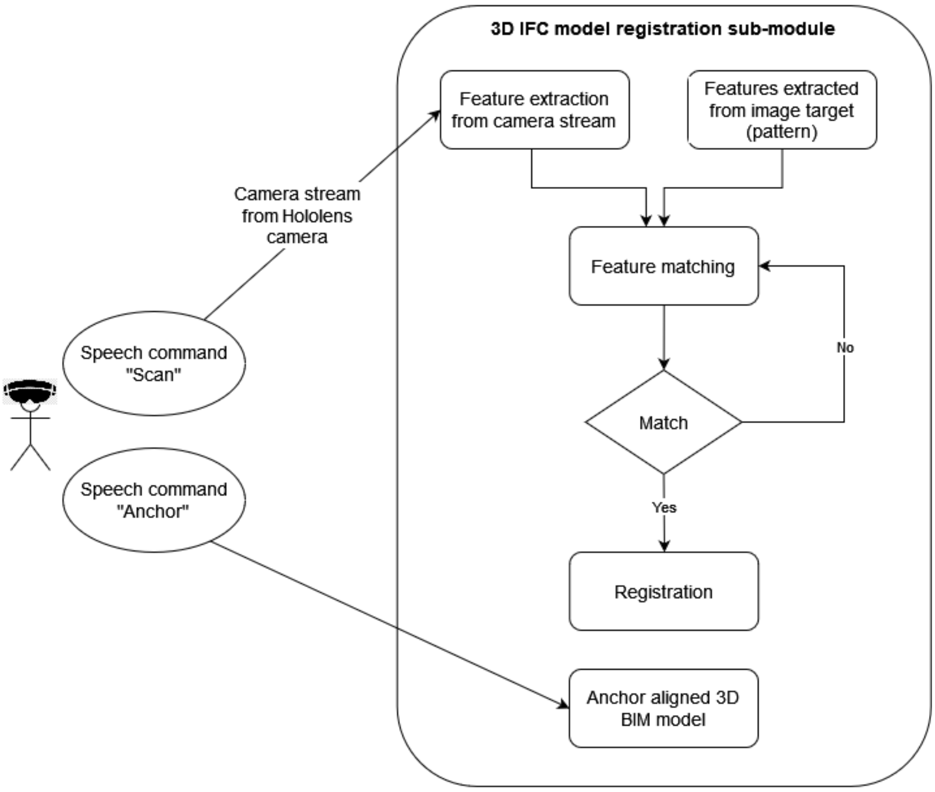


Figure 14 – 3D BIM registration process

Registration is also important for successful **localization** of the user in the 3D BIM model. Knowing the position and/or orientation of the user in the 3D BIM model is necessary for the pre-processing jobs created within the DigiTAR tool. As explained in detail in Section 2.1.2.2, the capture device specification is a necessary step to create a pre-processing job. Within the DigiTAR, the capture device is always the Hololens device, which the user wears on their head and on which the application runs. The location and orientation of the capture device, i.e., the Hololens, are essential properties included in the device specification for each pre-processing job. To locate the user in the 3D BIM model, the relative position of the user to the origin of the IFC model is calculated within the DigiTAR tool. This localization is combined with the georeferenced coordinates that will also be available through the DT Platform, the position and orientation of the user are computed and included for the capture device specification fine in JSON format that is sent to the Visual Data Pre-processing module, during the creation of a pre-processing job within DigiTAR.

2.1.1.2 2D Visual Data Viewer

Within the 2D Visual Data Viewer, the user can view visual data, both raw and processed. The user can either capture photos at runtime or select photos from the Hololens gallery and attach them to a pre-processing job. In both cases, the user can view the attached photos within the 2D Visual Data Viewer component of the application. The raw images (either captured at runtime or selected from the gallery) are displayed in the *Preview* menu of the application. The *Preview* menu also provides the option to select a filter and apply it to the raw image. Finally, the pre-processing job, the filter specification and the raw photo are sent to the Visual Data Pre-processing module, which returns the processed image back to the DigiTAR when the filtering is applied. The processed image can be viewed in the *Preview* menu. This way, the user can select different filters and preview the processed images in the *Preview* menu. When the user is satisfied with the processed image, they can select to finalise the pre-processing job, i.e., send it to the DT Platform. Figure 15 illustrates the basic workflow for capturing and/or selecting raw photos, selecting filters and finally, receiving the processed data (from the Visual Data Pre-processing tool) in the DigiTAR tool.

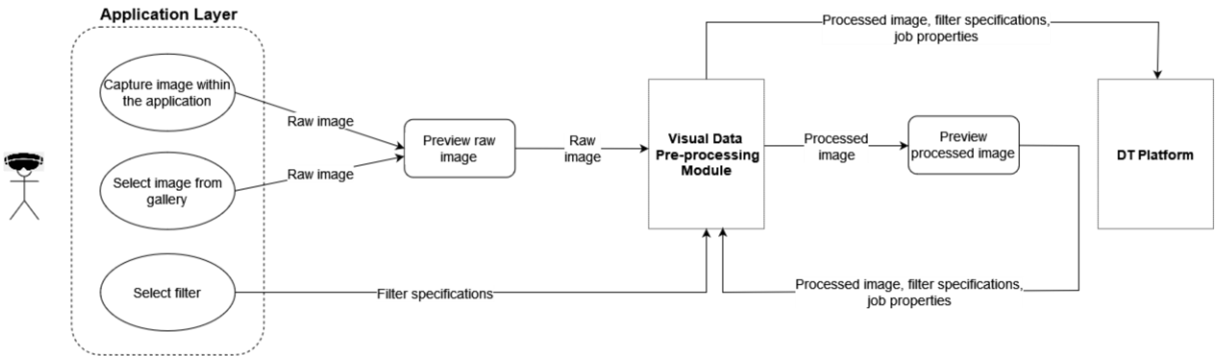


Figure 15 - Workflow for the 2D Visual Data Viewer of the DigiTAR tool

2.1.2 Application Layer

2.1.2.1 Mode selector

Within the mode selector, the user can choose between the three different modes of DigiTAR: i) the visualisation of the Quality Control results (Geometric and Visual), ii) the visualisation of the H&S results and finally iii) the Pre-processing mode. Figure 16 illustrates the mode selection process.

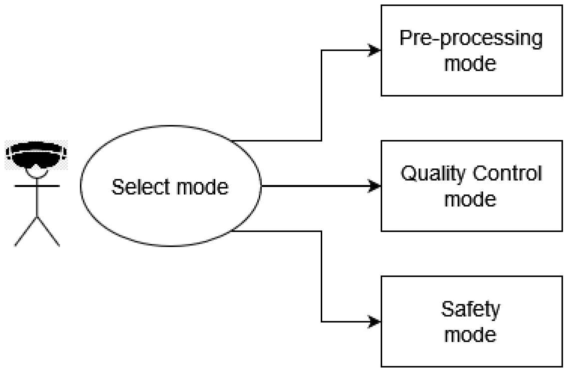


Figure 16 - Selecting mode in the DigiTAR tool

2.1.2.2 Job generator

Using this sub-module, the user can generate new jobs and link them with new data; the jobs are ultimately submitted for pre-processing to the Visual Data Pre-processing module. The Visual Data Pre-processing module is responsible for uploading the pre-processing jobs created within the DigiTAR to the DT platform so that they can be used for defect detection.

The DigiTAR user navigates on-site through the 3D BIM model (which is aligned to the actual site due to the registration process, described in Section 2.1.1.1). Figure 17 presents the generation of a pre-processing job within the DigiTAR tool. The first step when creating a pre-processing job, is to specify the capture device properties. The DigiTAR tool firstly ensures to register the capture device, i.e., a Hololens, as well as the current position and orientation of its camera in the 3D space. The current position and orientation of the user in the 3D space is provided by the 3D IFC Viewer sub-module that belongs to the Viewers Layer and is described in detail in Section 2.1.1.1. The location and orientation information, along with other device properties such as the Name and Type are included (in JSON format) in the metadata file that is pushed to the Visual Data Pre-processing module.

After the metadata for the involved capture device are sent from the DigiTAR module to the Visual Data Pre-processing module, the Visual Data Pre-processing module adds the capture device to its local database. Subsequently, the user of the DigiTAR tool can create a new pre-processing job and associate it with the newly defined device. Properties to be included in the generated job JSON file are the Name of the job, the data format (e.g., “Photo”), the Type (e.g., “.png”), the Frequency (e.g., “Once”) and the StartDate/EndDate. For the jobs created within the DigiTAR tool, the Frequency will always be “Once”, since they involve the current position/orientation of the user at the time that the photo is captured. The StartDate refers to the time that the user created the job. Since the job will have no duration, the EndDate coincides with the StartDate.

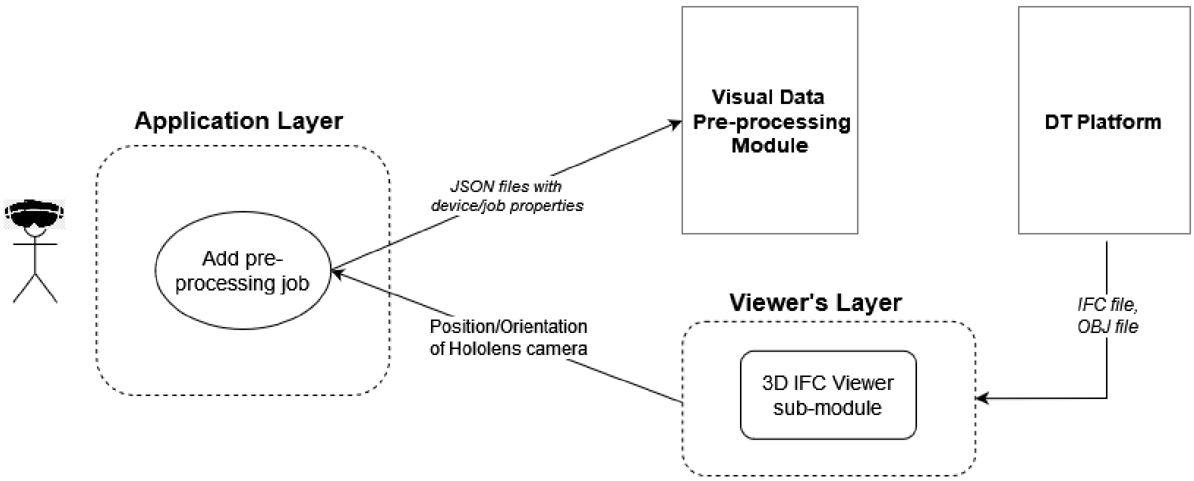


Figure 17 – Pre-processing job generation within the DigiTAR tool

2.1.2.3 *Report Generator*

This sub-module generates confirmation reports or additional remedial works and mitigations that are going to be delivered to the DT Platform. For the Quality Control mode, the user firstly views the Quality Control results and selects whether to confirm or reject them. If the user selects to confirm the QC result, the DigiTAR tool prompts the user to provide a remedial work. The same procedure applies for the Safety Control mode, where the user can confirm the H&S results and provide mitigation works.

2.1.2.4 *Data Collector*

The Data Collector is in charge of capturing on-site raw data that is going to be sent to the Visual Data Pre-processing module for pre-processing with filter application. There are two workflows for capturing raw data on-site within the Pre-processing mode of the DigiTAR tool; 1) the user can create a pre-processing job and select to capture an image to be linked to the job, 2) the user can capture images without relating them to a job. In the second workflow, the images are saved to the Hololens gallery with metadata that include the position and orientation of the user in the 3D BIM model at capture time. Thus, the user can select these images at subsequent sessions and link them to a job. Figure 18 presents the workflow for the data collection functionality of the DigiTAR tool.

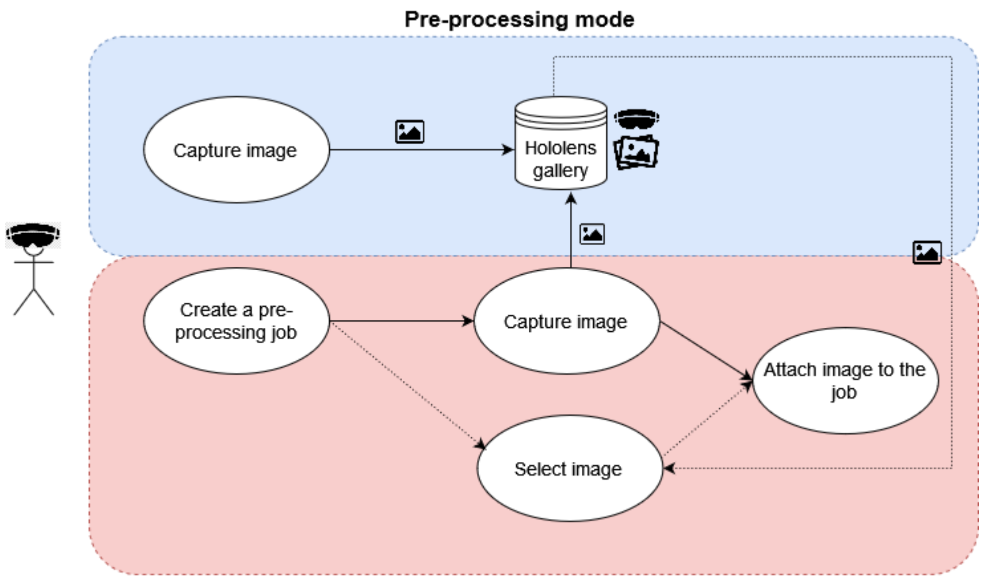


Figure 18 - Workflow for the Data Collector component of the DigiTAR tool

2.1.3 Communication Layer

2.1.3.1 DT Platform Connector

The DT Platform Connector enables the communication with the DT Platform. This component is responsible for uploading and downloading files and relative metadata through the DT Platform API.

2.1.3.2 Visual Data Pre-processing Connector

The Visual Data Pre-processing Connector handles the direct communication between the DigiTAR tool and the Visual Data Pre-processing tool. Communication is achieved using the Visual Data Pre-processing API, which is described in detail in Section 2.3.3.

2.2 Technology Stack and Implementation Tools

The DigiTAR application is developed from scratch. The development is performed using the Unity3D engine and the C# programming language. Several libraries and packages, such as the Mixed Reality Toolkit (MRTK), are used for the development of the application (see Table 2). MRTK enables access to the Hololens' capabilities and sensors. During the implementation phase, testing was merely performed on a Hololens2 device by building and deploying the application within Unity and Visual Studio. Alternatively, testing is always possible on the official Hololens emulator provided by Microsoft.

Unity3D: Unity [4] is a Game Engine created by Unity Technologies. The engine is widely used to build 3D and 2D applications, simulations and games as well as Virtual Reality (VR) and AR applications.

Programming Language C#: Programs based on/around Unity can be written in the object-oriented programming language C#. C# enables the development of a variety of secure and robust applications that run in the .NET ecosystem. The scripts to implement the DigiTAR application are developed in C#.

Microsoft Hololens2: The necessary hardware for the DigiTAR application is a pair of smart glasses with all the necessary sensors and AR capabilities. Microsoft Hololens2, depicted in Figure 19, is a pair of mixed reality smart glasses developed and manufactured by Microsoft [5]. Microsoft Hololens is the first fully self-contained holographic computer; the users can move freely, with no wires or external packs needed.

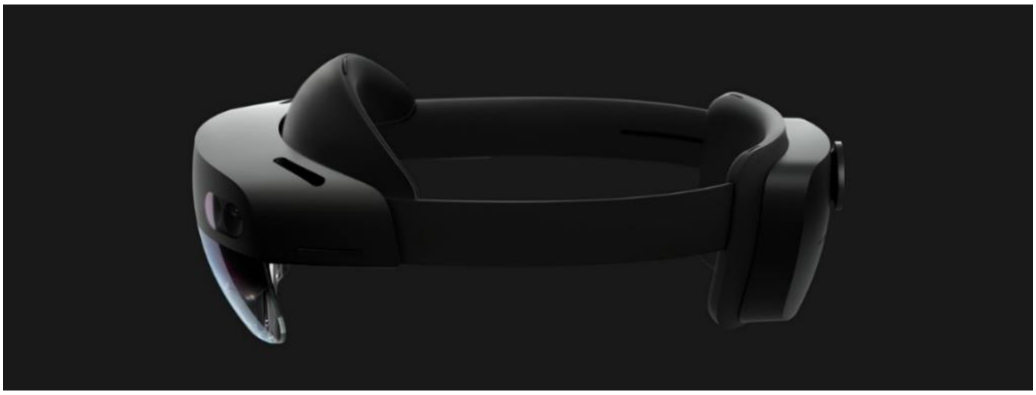


Figure 19 - Microsoft Hololens smart glasses [5]

Mixed Reality Toolkit: Microsoft provides substantial support for both users and developers regarding its Hololens product; the Mixed Reality Toolkit (MRTK) [6] for Unity provides essential sets of components and features for developing applications for Microsoft Hololens 1st gen and Hololens2. MRTK provides useful interfaces and implementations for creating Mixed Reality applications, accompanied by full documentation and support.

Hololens Emulator: To facilitate the development of Hololens applications, Windows Mixed Reality Software Development Kit (SDK) provides the Hololens Emulator [7]. The Hololens Emulator enables the testing of holographic applications on a computer without the need for a physical Hololens product. The human and environmental inputs that are usually read by Hololens sensors are simulated from keyboard, mouse, or

Windows Mixed Reality motion controllers [8]. Notably, applications running on the emulator do not need to be built from scratch to be deployed on the Hololens device. The main window of Hololens Emulator is depicted in Figure 20.

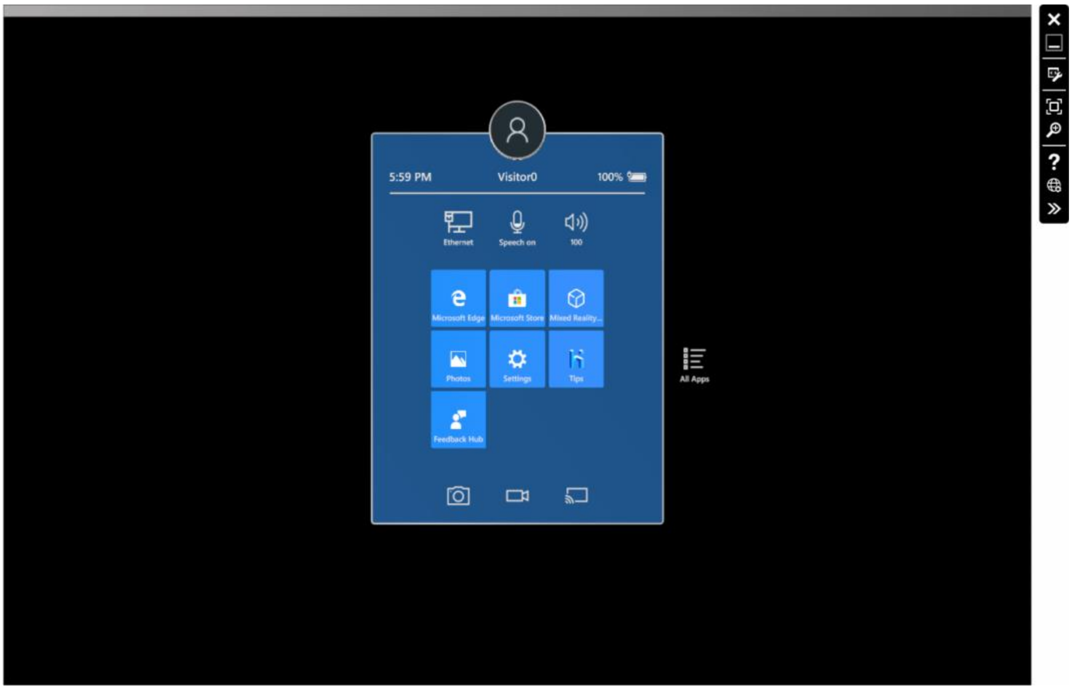


Figure 20 - Main window of Hololens2 Emulator [7]

Table 2 – Libraries and Technologies used in User interface and AR enabled in-situ QC Visualisation v1

Library/Technology Name	Version	License
Xbim toolkit	5.1.323	Common Development and Distribution License (CDDL)
OBJ Runtime Importer	2.02	Extension Asset
Microsoft Hololens	2nd gen	-
Windows Mixed Reality	SDK 10.0.19041	Software licensing
Unity 3D Editor	2020.5	Software licensing
Mixed Reality Toolkit (MRTK)	2.5.3	Software licensing
Vuforia Engine	9.6.4	Proprietary

2.3 Input, Output and API Documentation

The DigiTAR module interacts with the DT Platform and the Visual Data Pre-processing module for data exchange. The input data and output data for each mode of the DigiTAR tool are presented in Sections 2.3.1 and 2.3.2, respectively. Finally, the API is documented in Section 2.3.3.

2.3.1 Input Data

DigiTAR, when used in Quality Control mode, would require the following data as input:

- **Project ID:** the project ID that the user wants to check.
- **QC results:** a file in JSON format containing all the information generated by the relevant COGITO QC components
- **IFC file:** the BIM model exported in IFC and in OBJ file formats in order to connect the as-designed data and the QC results generated by the respective COGITO QC components.

In the case of Safety mode, DigiTAR would require the following data as input:

- **Project ID:** the project ID that the user wants to check.
- **IFC file enriched with Safety information**: BIM model exported in IFC and in OBJ file formats, enriched with the information generated by the COGITO Health and Safety-related components.

Finally, in the case of operation in Pre-processing mode, DigiTAR would require the following data as input:

- **Project ID:** the ID of the project the user wants to check.
- **IFC file:** BIM model exported in IFC and in OBJ file formats in order to connect the as-designed data and the as-build data (IFC components' attribution).
- **Work Order ID:** the working order ID to associate the image data and extract the list of components that are ready for quality control.

2.3.2 Output Data

The DigiTAR module in the case of Quality Control mode would provide the following data as output:

- **Metadata:** a file in JSON format containing information regarding the defects' confirmation and the additional remedial works.

In the case of Safety mode, DigiTAR would provide the following data as output:

- **Metadata:** a file in JSON format containing information regarding the hazards' confirmation and the additional mitigation works.

In the case of Pre-processing mode, DigiTAR would provide the following data as output:

- **Raw and processed as-built 2D image data:** processed images (i.e., .png, .jpeg) of the as-built data that will be utilised by the Visual Quality Control tool.
- **Metadata:** files in JSON format containing information accompanying the processed image, the capturing device properties and the pre-processing job properties.

2.3.3 API Documentation

Regarding the API, a first version is implemented in this first release of the User interface and AR enabled In-situ QC Visualisation. The API is mainly used for storing, retrieving, updating, and deleting data used within the DigiTAR application. The API is also used for communicating with the Visual Data Pre-processing module. To this end, the API is used to send raw images to the Visual Data Pre-processing module and receive the pre-processed images along with metadata. The requests were tested using the Postman API Platform[1]. A few examples of all the available requests are presented in the next sub-sections. However, the communication with the DT Platform will be established within the T8.1 "End-to-end ICT System Integration, Testing and Refinement" (M18-M34). Therefore, in the second release, the component will expose an execution interaction with the DT Platform in order to be consistent with the rest of the COGITO solution.

2.3.3.1 BIM Model Request

By making a request to the DT Platform for a specific project, the IFC file along with the OBJ file of the site are downloaded. For the Safety mode, the IFC file enriched with the Safety information is requested and loaded in the application.

2.3.3.2 QC Results Request

In the Quality Control mode, the DigiTAR receives the QC results for the specific project. An example of the JSON file is depicted in Figure 21. The properties received in the JSON body request are presented in Table 3.

[1] https://www.postman.com/

```
{
  "Project_rstadvancedsampleproject_QC372": {
    "QC_UID": "Project_rstadvancedsampleproject_QC372",
    "Dictionary_ID": "QC_1",
    "OriginDocument": "BS EN 13670-2009",
    "Label": "ColumnsAndWalls_a",
    "Description": "Inclination of element",
    "Result": "Pass",
    "Involved_Components": [
      "1M1TYAh$5Ba86TXvlzu46Z"
    ],
    "TimestampSchedule": "17/01/2022",
    "TimestampPerformed": "2/5/2022",
    "Unit": [
      "distance",
      "metres"
    ],
    "ScalarResult": [
      "0.000589440111"
    ],
    "ToleranceReference": [
      "0.0149999997"
    ],
    "AuxiliaryOutputFile": [
      "\"C:\\COGITO_Repos\\COGITO_GeomQC\\build\\Output\
       \RevitSchoolCylinders/18YHwga450Mw4Fy6M5t_8F_Project_rstadvancedsampleproject_QC372.ply\"",
      "\"C:\\COGITO_Repos\\COGITO_GeomQC\\build\\Output\
       \RevitSchoolCylinders/18YHwga450Mw4Fy6M5t_8F_Project_rstadvancedsampleproject_QC372.obj\""
    ]
  }
}
```


Figure 21 - The JSON file with the QC results received by DigiTAR in Quality Control mode

Table 3: QC results properties

Property	Description
QC_UID	The unique ID of the QC
Dictionary_ID	The dictionary ID
OriginDocument	The origin document
Label	The label of the QC result
Description	Description of the testing that is performed
Result	The QC result ("Pass"or "Fail")
Involved_Components	The unique IFC identifiers of the components involved in the QC result
TimestampSchedule	The scheduled timestamp for the QC test
TimestampPerformed	The timestamp when the QC was actually performed
Unit	The unit measurement
ScalarResult	The scalar result of the QC test
ToleranceReference	The tolerance of the QC test
AuxiliaryOutputFile	Auxiliary files with the output of the QC result

2.3.3.3 Pre-processing Mode Requests

Device request: In the Pre-processing mode, DigiTAR performs as a data acquisition tool. In order to create a pre-processing job, a request defining the capture device's specifications must firstly be pushed to the endpoint provided by the Visual Data Pre-processing module. An example JSON file holding the metadata of the Hololens camera specifications can be seen in Figure 22. As can be seen, the local position and orientation of the Hololens camera (i.e., the user's position and orientation) are included in the metadata. This request generates a unique device id for the capture device, i.e., the Hololens camera, at that particular location and orientation. The properties that should be passed in the JSON body request are presented in Table 4.

```json
{
  "name": "Hololens",
  "type": "Hololens Camera",
  "ifcFileId": 1,
  "ifcFileName": "IFC file 1.ifc",
  "status": "Online",
  "url": "https://cameraurl/2",
  "position": {
    "measurement": "metres",
    "x": 9.72,
    "y": 5.6,
    "z": 0.54
  },
  "orientation": {
    "x": 15.48,
    "y": 54.6,
    "z": 10
  }
}
```


Figure 22 - The JSON file defining the capture device properties for the Pre-processing mode in the DigiTAR

Table 4 – Capture device properties

Property	Description
name	The name of the device
type	The type of the device (RGB Camera, Laser Scanner...)
ifcFileId	The unique identifier for the IFC file to which the device belongs
ifcFileName	The name of the IFC file
status	The device status, i.e., Online or Offline
url	The url/IP address of the device when capable of streaming data
position	The position of the device
orientation	The orientation of the device

The position property has the sub-properties illustrated in Table 5.

Table 5 - Sub-properties of position

Position property	Description
measurement	The measurement that will be used for calculating the position of the device. *Metres* or *Centimetres*
x	The x coordinate of the device's position
y	The y coordinate of the device's position
z	The z coordinate of the device's position

The orientation property has the sub-properties illustrated in Table 6.

Table 6 - Sub-properties of orientation

Orientation property	Description
x	The x axis rotation of the device
y	The y axis rotation of the device
z	The z axis rotation of the device

Creation of the pre-processing job request: When the user of the DigiTAR application creates a pre-processing job, a request is made to the endpoint provided by the Visual Data Pre-processing module with the properties of the job, as depicted in the JSON in Figure 23. The "deviceID" field relates the job with the device definition of the previous step. The JSON is pushed to the Visual Data Pre-processing module which subsequently promotes the generated job to the DT Platform. The properties that should be passed in the JSON body request are presented in Table 7.


```json
{
    "name": "Job Test",
    "format": "Photo",
    "type": ".jpg",
    "frequency": "Once",
    "startDate": "2022-05-12",
    "endDate": "2022-05-12",
    "status": "On-going",
    "elementID": "1M1TYAh$5Ba86TXvlzu46Z",
    "deviceId": 1
}
```


Figure 23 - The JSON file for the pre-processing job that is created within the DigiTAR tool. This JSON is sent to the Visual Data Pre-processing module.

Table 7 – Pre-processing Job properties

Property	Description
name	The name of the job
format	The format of the job (Photo, Point Cloud Data)
type	The type in which the file will be stored (jpeg, png, etc)
frequency	The frequency in which the device will perform this job
startDate	The starting date of the job
endDate	The ending date of the job
status	Status of the job (*On-going* or *Finished)*
elementID	The unique IFC identifier of the component that are involved in the pre-processing job
deviceId	The unique identifier of the device to which the job belongs

Upload raw photos request: The raw photos that are attached to each job are either captured by the user during the runtime of the DigiTAR application or selected by the Hololens' gallery. The raw images are passed to the Visual Data Pre-processing module using the API endpoint depicted in Figure 24.

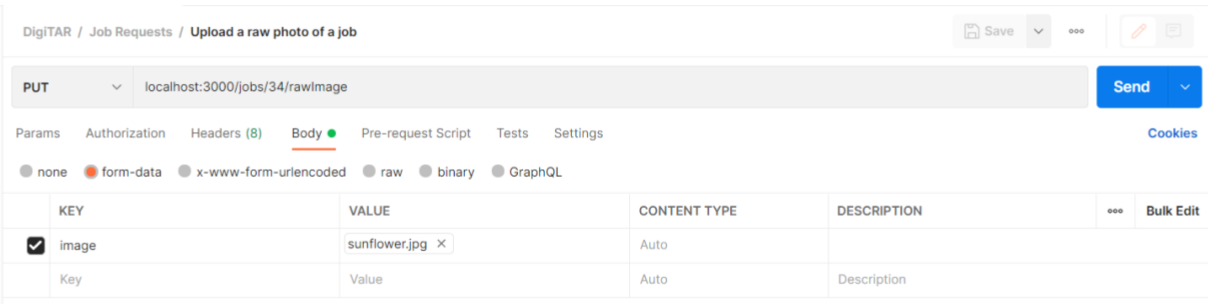


Figure 24 - API endpoint for uploading a raw photo linked to a particular job

Apply filter request: When the user selects a filter to be applied to the image from the GUI of DigiTAR, the corresponding request is performed to the API endpoint of the Visual Data Pre-processing module. In the following, examples of filtering requests are provided (i.e., blurring, resizing, cropping and changing brightness/contrast).

Example request for blurring an image: The API endpoint for blurring an image is depicted in Figure 25. A successful request returns HTTP 200 Status and the processed image. The properties that should be passed in the JSON body request are presented in Table 8.

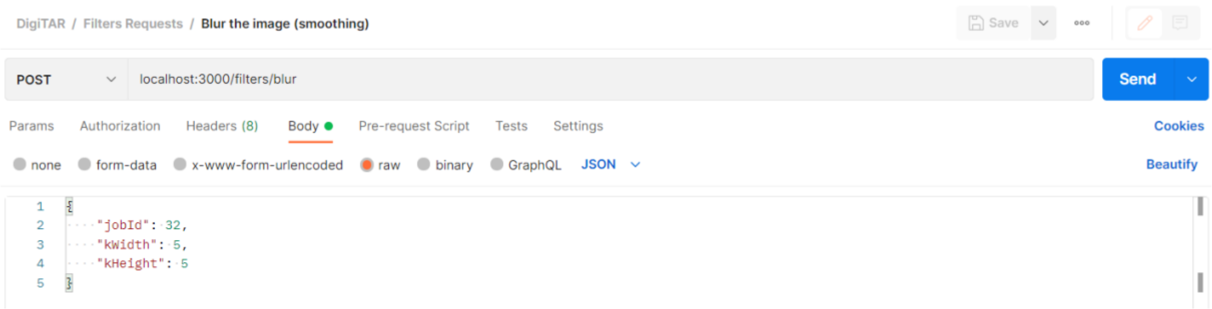


Figure 25 - API endpoint for requesting the blurring filter to be applied to the image

Table 8 - Blur properties

Property	Description
jobId	The unique identifier of the related job
kWidth	The width of the kernel used for Gaussian Blurring (should be odd)
kHeight	The height of the kernel used for Gaussian Blurring (should be odd)

Example request for resizing an image: The API endpoint for resizing the image related to a job is depicted in Figure 26. A successful request returns HTTP 200 Status and the processed image. The properties that should be passed in the JSON body request are presented in Table 9.

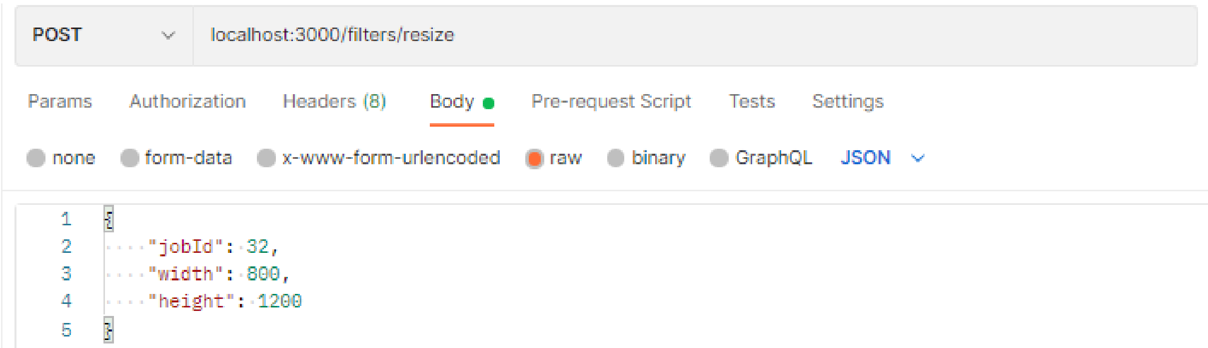


Figure 26 - API endpoint for resizing an image related to a job

Table 9 - Resize properties

Property	Description
jobId	The unique identifier of the related job
width	The desirable width of the processed image
height	The desirable height of the processed image

Example request for cropping an image: The API endpoint for cropping the image related to a job is depicted in Figure 27. A successful request returns HTTP 200 Status and the processed image. The properties that should be passed in the JSON body request are presented in Table 10.

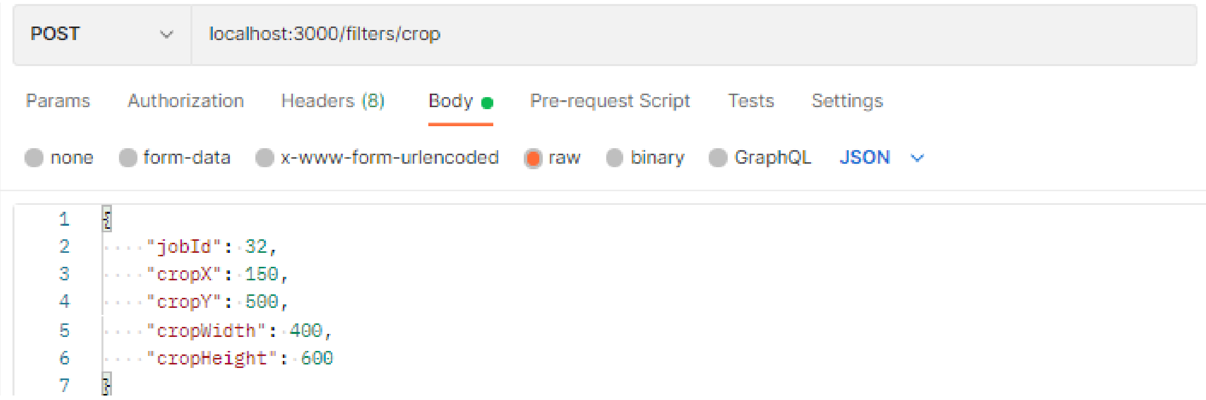


Figure 27 - API endpoint for cropping an image related to a job

Table 10 - Cropping properties

Property	Description
jobId	The unique identifier of the related job
cropX	X position of the starting cropping point (upper left point)
cropY	Y position of the starting cropping point (upper left point)
cropWidth	The desirable cropped width
cropHeight	The desirable cropped height

Example request for changing contrast and brightness of an image: The API endpoint for adjusting the contrast and brightness of the image [9] related to a job is depicted in Figure 28. A successful request returns HTTP 200 Status and the processed image. The properties that should be passed in the JSON body request are presented in Table 11.

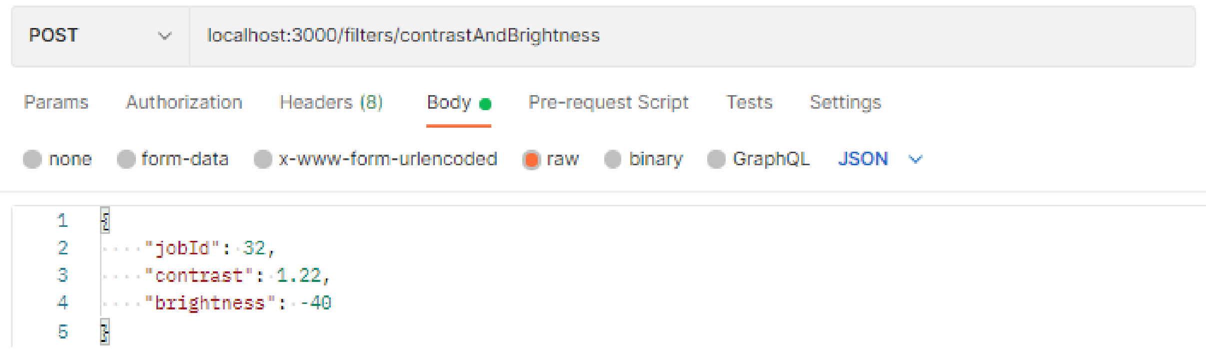


Figure 28 - API endpoint for changing contrast and brightness of an image related to a job

Table 11 – Contrast and brightness properties

Property	Description
jobId	The unique identifier of the related job
contrast	The desirable contrast of the processed image (values from 0.30 – 2.00)
brightness	The desirable brightness of the processed image (values from -127 – 127)

2.4 Usage Walkthrough

In this section, a brief introduction to some basic actions and gestures that Hololens recognizes is firstly enclosed followed by a detailed walkthrough guide describing the usage of the DigiTAR tool.

2.4.1 Introduction to basic Hololens actions and gestures

The basic interactions of the user with Hololens are:

- **Touch**: With Hololens the user can reach out and touch holograms. When Hololens sees the user's hand, a floating pointer (like a mouse pointer) appears near the tip of the user's index finger to help the user target elements (see Figure 29).
- **Hand ray**. To use a hand ray, the user holds their hand in front of them, with their palm facing away. A laser pointer (hand ray) appears (see Figure 30).

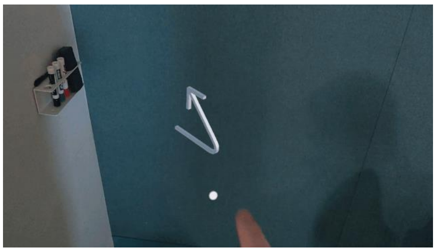


Figure 29 - Touch gesture on Hololens

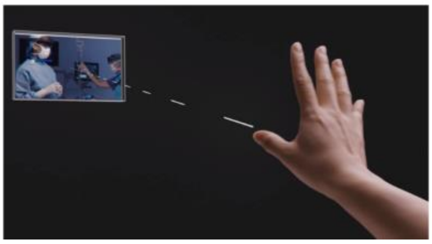


Figure 30 - Hand ray gesture

- **Gaze**: When the user gazes at an item, the pointer in Figure 31 corresponds to where the user's eyes are looking at. Often, the gaze is used together with the air tap. When the user gazes, they should turn their whole head, not just their eyes. The pointer will follow the movement of the user's gaze.
- **Air tap**: For the air tap gesture, the user holds their hand straight out in front of them in a loose fist, point their index finger straight up toward the ceiling, tap their finger down, and then quickly raise it back up again, as depicted in Figure 32.

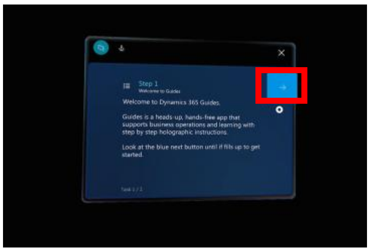


Figure 31 - Gaze on Hololens

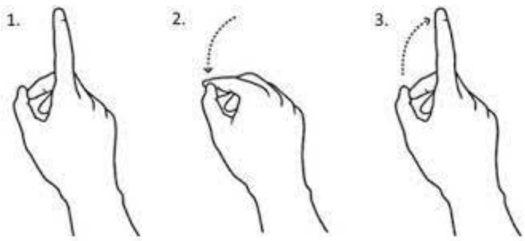


Figure 32 - Air tap gesture on Hololens

- **Hand-tracking frame**: Hololens has sensors that can see a few feet to either side of the user. When the user uses their hands, they need to keep them inside that frame, or Hololens won't see them. However, the frame moves with the user as the user moves around. The hand tracking frame on Hololens is depicted in Figure 33.

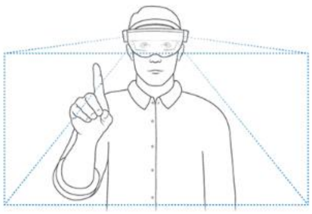


Figure 33 - Hand tracking frame on Hololens

- **Hololens2 gestures to open the Start menu:** The Start menu of Hololens is depicted in Figure 34. The Start menu on HoloLens is where the user can open apps, see important status info and access tools like the camera.

There are two ways to open the Start Menu (see Figure 35):

1. **Open the Start menu with two hands**
   - The user should hold out one of their hands with the palm facing up and look at their wrist. They should see a holographic Windows logo.
   - With the index finger of their other hand, they should touch the Windows logo.
2. **Open the Start menu with one hand**
   - The user should hold out one of their hands with the palm facing up and look at their wrist. They should see a holographic Microsoft Windows logo.
   - With the hand that they are holding out, they should touch their index finger to their thumb in a pinching motion.

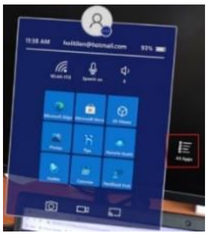


Figure 34 - Start menu on Hololens

Figure 35 - Gesture to open the Start menu in Hololens with (1) both hands, (2) one hand

2.4.2 Usage Walkthrough for the DigiTAR application

Open the DigiTAR application: To open the application, the user should:

1. Perform the Start gesture to open the Start menu of Hololens.
2. Use the Touch gesture to select **"All apps"** to the right (see Figure 36).
3. From the list of applications, select DigiTAR using the Touch gesture.

The first time the application is opened, the user's permission is asked to let DigiTAR access the Hololens camera and microphone. The user should accept these permissions since they are necessary for the application to capture images/videos and use speech commands.

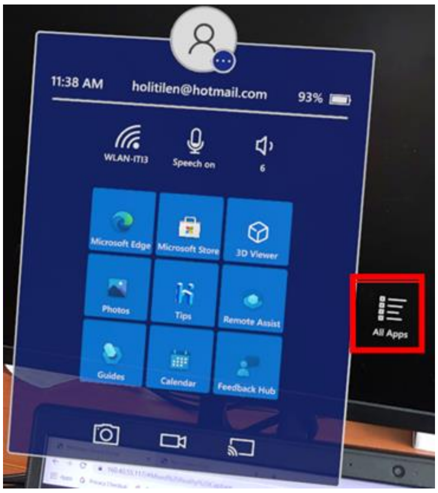


Figure 36 - Select "All apps" to see all applications installed on Hololens

Close the DigiTAR application: To close the application (at any time), the user can perform the Start gesture (depicted in Figure 35) to open the Start menu of Hololens and select the "Home" button (see Figure 37). Subsequently, the user should close the 2D box of the application (see Figure 38) to completely terminate the application. It is important to always make sure to close the 2D box before opening the application again, to avoid messing the calibration of the 3D BIM model and malfunctioning of the application.

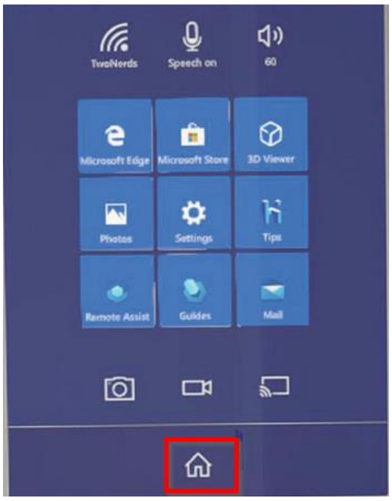


Figure 37 - To close the application, the user can select the "Home" button of the Start Menu

Figure 38 - The user should make sure to also close the 2D box of the application in order to completely terminate the application

Login menu: When the application is opened, the user is prompted to provide their credentials to login, as illustrated in Figure 39. To insert their username, the user can gaze and perform the air tap gesture on the username input field and the keyboard of Hololens opens-up, as depicted in Figure 40. The user can type on the Hololen's keyboard using the Touch gesture. To use dictation, the user can air tap on the dictation button (in red in Figure 40) to activate it. After inserting their credentials, the user can air tap on the "Login" button in the *Login* menu (see Figure 39) to proceed with the authentication. The user can also choose the "Exit" button to close the *Login* menu without proceeding with the application.

Figure 39 - DigiTAR tool user's login menu

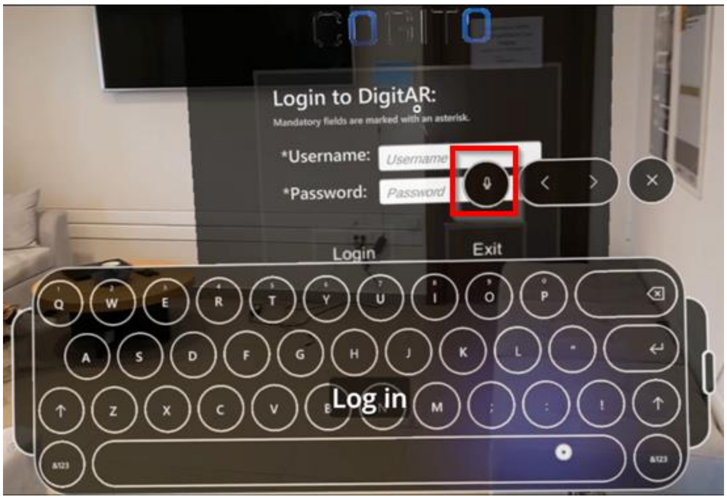


Figure 40 - When the user gazes and performs the air tap gesture on the input fields, the keyboard of Hololens opens-up

Speech commands: The user can select the buttons in DigiTAR by gazing at them and performing the air tap gesture. Most buttons in DigiTAR can be alternatively selected with speech commands, without having to use hand gestures. The user is notified for the speech command that activates a button, by gazing at it. This way, a tooltip appears with the corresponding speech command, as depicted in Figure 41.

Figure 41 - When the user gazes at a button that can be activated with speech command, a 2D tooltip box appears with the corresponding speech command

Pin/Unpin menus: By default, all displayed menus in DigiTAR follow the user's movement. By using speech command "Pin", the currently displayed menu is pinned in its current position and stops following the user's movement. To restore the movement of the currently displayed menu, the user scan use speech command "Unpin".

Select project: After successful authentication, the menu in Figure 42 opens-up so that the user can select a project. The user can perform the air tap gesture on the input field so that a drop-down list with the available projects appears, as depicted in Figure 43. The user can select a project from the drop-down list with the air tap gesture.

Figure 42 - The menu of DigiTAR tool for the user to select a project

Figure 43 - The user can air tap to select a project from the drop-down list

Select mode: After selecting a project, the user is prompted to the *Select mode* menu of the DigiTAR tool, which is depicted in Figure 44. As already mentioned, there are three modes for the DigiTAR: the Pre-processing, the Quality Control and the Safety Control modes. The user can air tap on the corresponding button to select a mode. Alternatively, the user can select a mode by gazing at the desired button and use the corresponding speech command. The speech commands assigned to the buttons are "Pre-processing", "Quality" and "Safety", respectively. To go back to the *Select project* menu, the user can air tap on the "Back" button at the bottom of the menu. To close the menu without proceeding with the mode selection, the user can air tap on the "Exit" button.

Figure 44 - The menu of the DigiTAR tool that allows the user to select a mode for the DigiTAR tool

Pre-processing mode: If the user selects the "Pre-Processing" button in the *Select mode* menu, the menu in Figure 45 opens-up. The user can air tap on the "Download BIM model" button to download the BIM model from the DT Platform or air tap on the "Visualize BIM model" button to visualise the 3D BIM model. The first option, i.e., to download the BIM model from the DT Platform, will be functional in the next version of the application and will be based on the endpoints that the DT Platform will provide to the COGITO tools for accessing the files and resources of each project.

Figure 45 - For the Pre-processing mode, the user should first load the BIM model of the project

3D BIM model registration: The position of the image target in the 3D space for each IFC file will be known in advance to the user. To perform registration, the user can use speech command **"Scan"** to initialize the image target detection process. Afterwards, the user is expected to approach and look at the printed image target in the real world. After the image target has been detected, the IFC model will be visualised to be aligned to the real world. To maintain the registration for subsequent sessions, the user can use speech command **"Anchor"** to set the aligned 3D model as a spatial anchor, so that it can be loaded the next time the application is initialized eliminating the need to repeat the registration process.

Pre-processing mode main menu: The main menu of the Pre-processing mode of the DigiTAR tool is displayed in Figure 46. The "Main Menu" button closes the menu in Figure 46 and enables the *Select mode* menu in Figure 44. The "Toggle BIM model" button toggles on and off the 3D BIM model visualisation. If the user selects this button while viewing the 3D BIM model, the BIM model visualisation is switched off. To switch on again the 3D BIM model visualisation, the user can select this button again. Finally, the "Add job" button enables the generation of a pre-processing job within the DigiTAR tool.

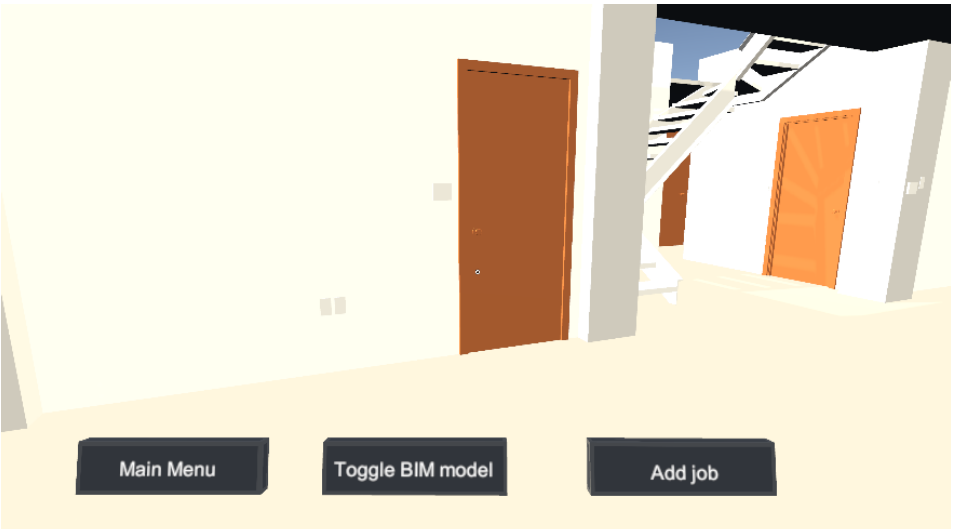


Figure 46 - The GUI displayed at the Pre-processing mode

Generate pre-processing job: The user can create a pre-processing job by selecting the corresponding "Add job" button from the displayed GUI of Pre-processing mode of the DigiTAR tool, which is displayed in Figure 46. After selecting the "Add job" button, the menu depicted in Figure 47, opens-up where the user is prompted to provide a name for the new pre-processing job. When the user performs the air tap gesture on the white input field, the keyboard of Hololens opens-up and the user can type the desired name.

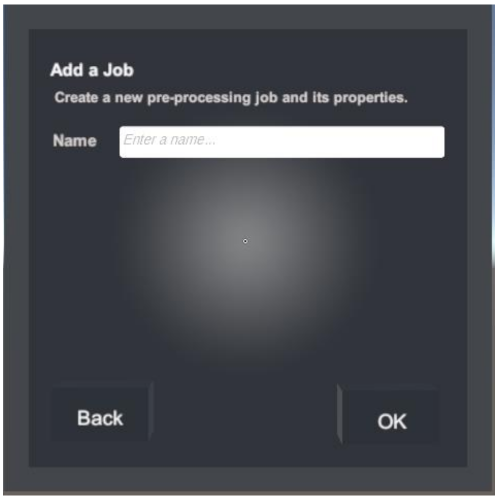


Figure 47 - Menu to add a pre-processing job

Attach 3D IFC elements to a pre-processing job: The user can gaze at a 3D IFC element and use speech command "Add" to attach this element to the newly generated job. The gazed 3D IFC element is highlighted in yellow to visually notify the user that the element has been selected. Subsequently, the menu depicted in Figure 48 opens-up that displays the name of the selected IFC element. The user can either change the IFC element that is associated to the job by selecting the "Change element" button or proceed by selecting the "OK" button.

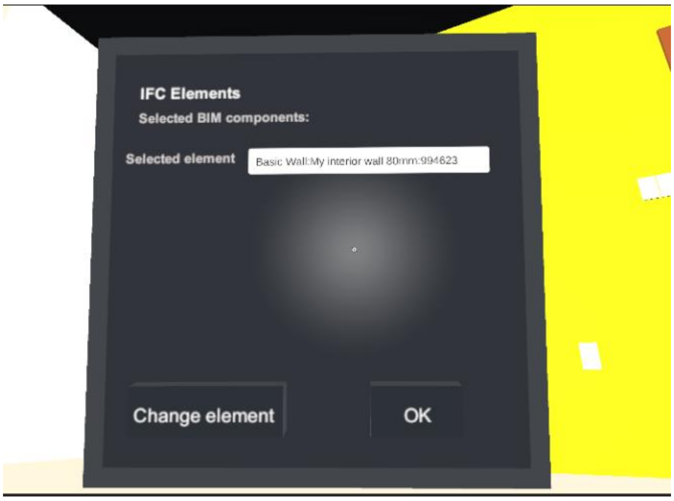


Figure 48 - The name of the selected IFC element (highlighted in yellow) that is attached to the pre-processing job is displayed

Link images to a pre-processing job: After an IFC element is attached to a pre-processing job, the menu in Figure 49 is displayed. Again, the user has the option to toggle between enabling and disabling the 3D BIM model visualisation using the "Toggle BIM model". The user can capture an image to be linked to the job by selecting the "Capture image" button. Afterwards, the user can gaze at the desired location and use speech command "Capture" to capture an image that is linked to the job. The user can also attach images from the Hololens gallery to the job by selecting the "Gallery" button. After the button has been selected, the picture folder of Hololens opens-up, as depicted in Figure 50. The user can navigate to the desired folder and select an image, as can be seen in Figure 51.

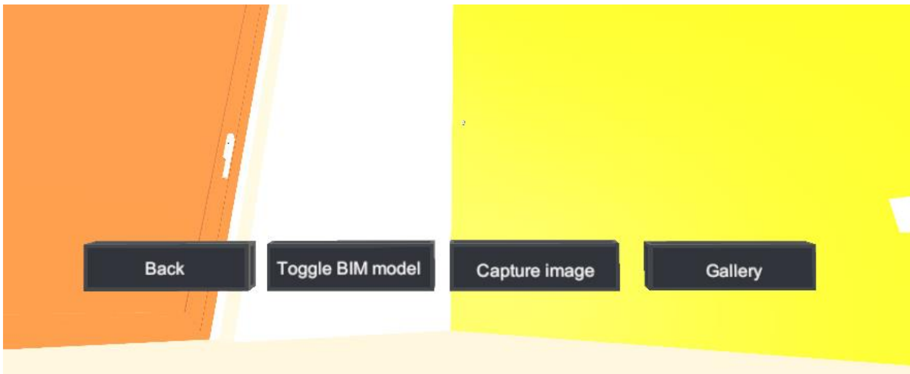


Figure 49 - The menu used for attaching images to a pre-processing job

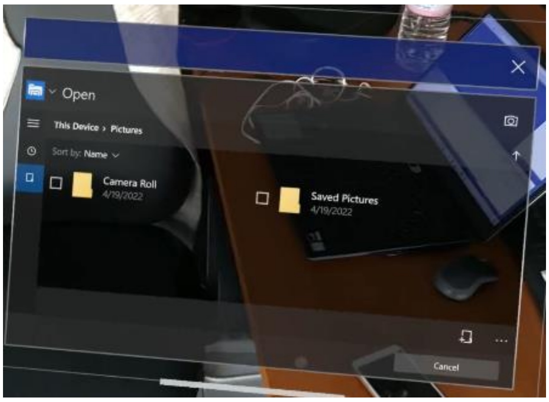


Figure 50 - The pictures folder on Hololens

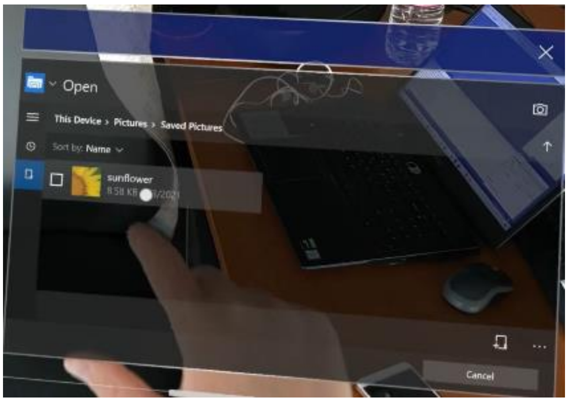


Figure 51 - The user can select an image to be attached to the pre-processing job

Preview menu: After linking an image to a pre-processing job (either by capturing an image at runtime or by selecting an image from the Hololens gallery), the menu in Figure 52 opens-up that displays the attached image.

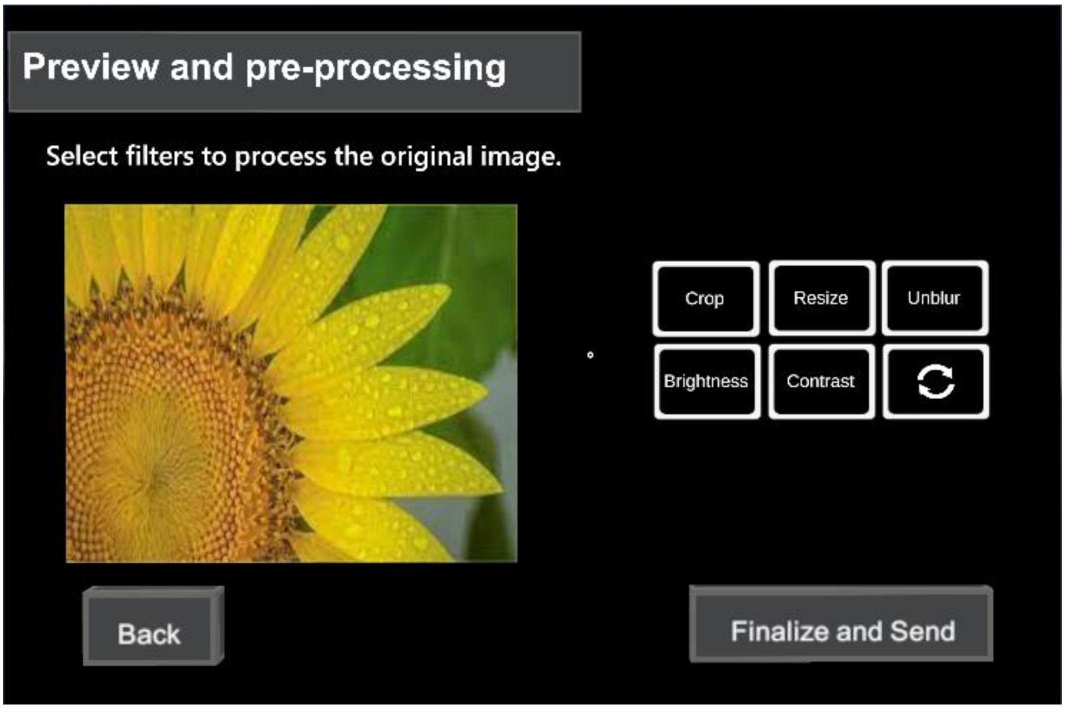


Figure 52 - Menu for displaying the images attached to a pre-processing job

Apply filters and preview processed image: The user can select a filter from the menu displayed in Figure 52. For example, if the user selects the "Crop" filter, as displayed in Figure 53, the corresponding filter request is sent to the Visual Data Pre-processing module, using the API described in Section 2.3.3. Subsequently, the processed image is received from the Visual Data Pre-processing module and is displayed, replacing the original image that was displayed before. If the user selects the "Finalize and Send" button, the processed image is promoted back to the Visual Data Pre-processing module which uploads it to the DT Platform so that it is accessible for the Visual Quality Control module. After uploading to the DT Platform is completed, a confirmation message is depicted, as can be seen in Figure 54.

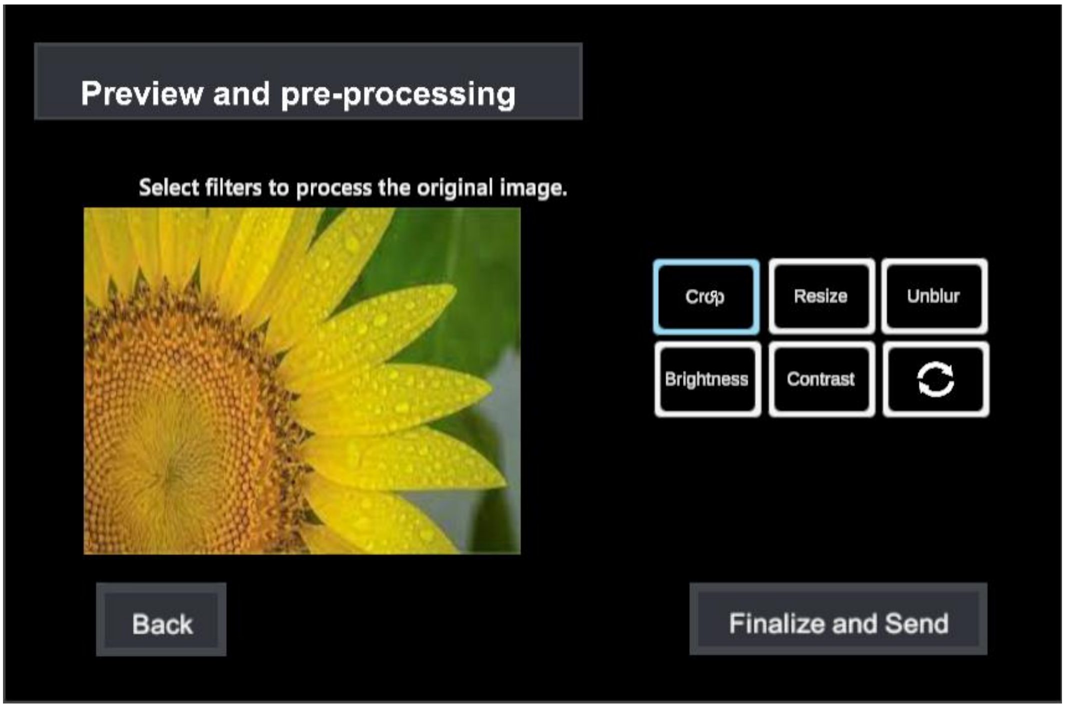


Figure 53 - The user can select a filter by selecting the corresponding button

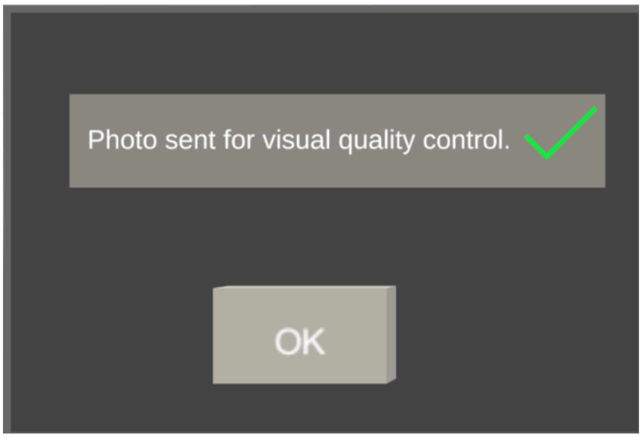


Figure 54 - Confirmation menu when processed image has been uploaded to the DT Platform and is accessible for the Visual Quality Control module

Quality Control mode: If the user selects the Quality Control mode in the *Select mode* menu in Figure 44, the DigiTAR tool downloads the Quality Control results for the specific project from the DT Platform. Since the DT Platform's API will be established in the following months, the application currently loads the QC results from its own internal storage.

QC results notification: To notify the user for the IFC elements that have a "Fail" QC result, the corresponding 3D IFC elements are displayed in red. An example can be seen in Figure 55. The user can air tap on the red 3D IFC elements to view details for the QC result. More specifically, the menu in Figure 56 is displayed for an example of a "Fail" QC result. The corresponding menu for an example of a "Pass" QC result is depicted in Figure 57. The user can select the "Confirm" button or the "Reject" button to confirm or reject the QC result, respectively.

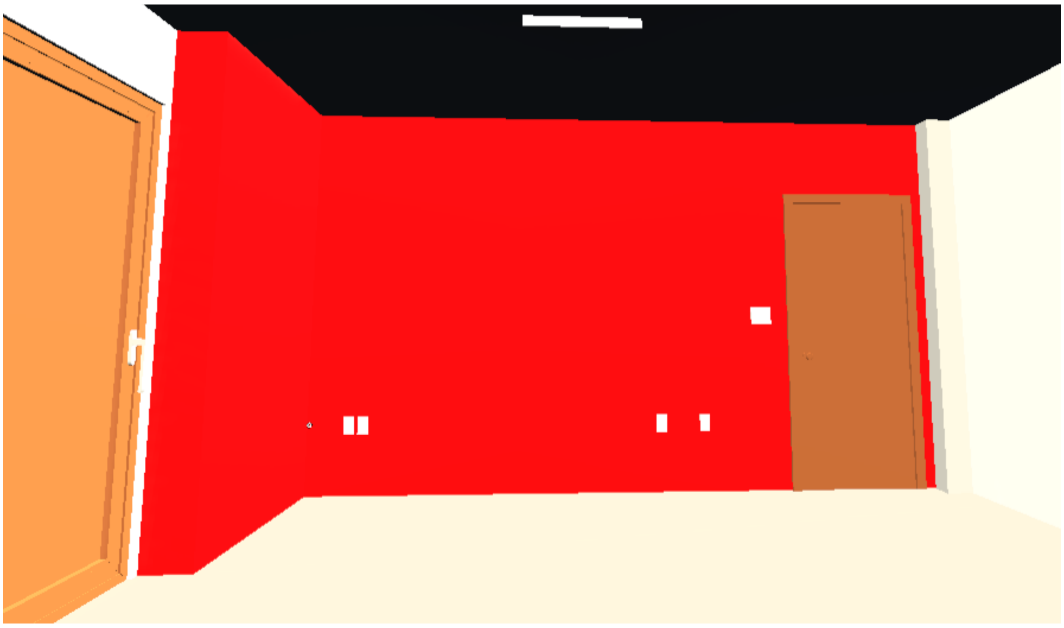


Figure 55 - The IFC elements with "Fail" QC result are highlighted in red

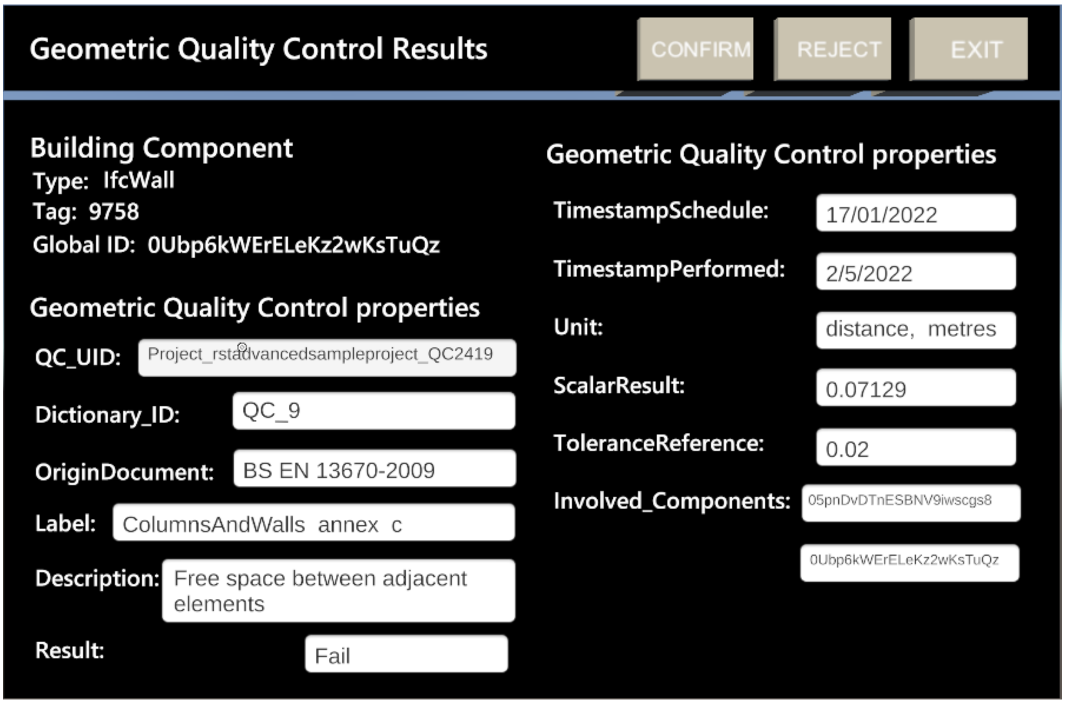


Figure 56 - Menu displaying a "Fail" QC result

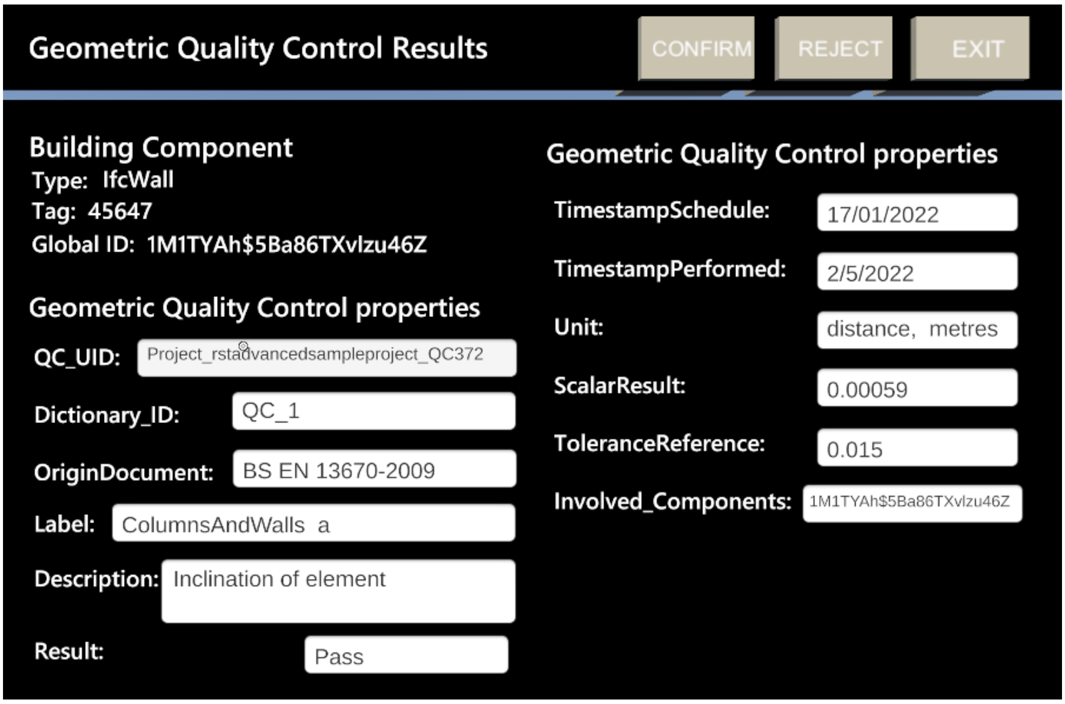


Figure 57 - Menu displaying a "Pass" QC result

Add remedial work for QC result: After confirming the QC result by selecting the "Confirm" button in the *QC results notification* menu (in Figure 56 and Figure 57), the menu in Figure 58 opens-up so that the user can add a remedial work. In this version of the DigiTAR tool, as can been in Figure 58, the remedial work report contains three fields: 1) a description of the remedial work, 2) the time schedule, 3) the priority of the remedial work. The generated JSON file with the confirmed QC result and the generated remedial work is stored locally in this version of the application. In the next version, where complete communication with the DT Platform will be established, the files will be promoted to the DT Platform.

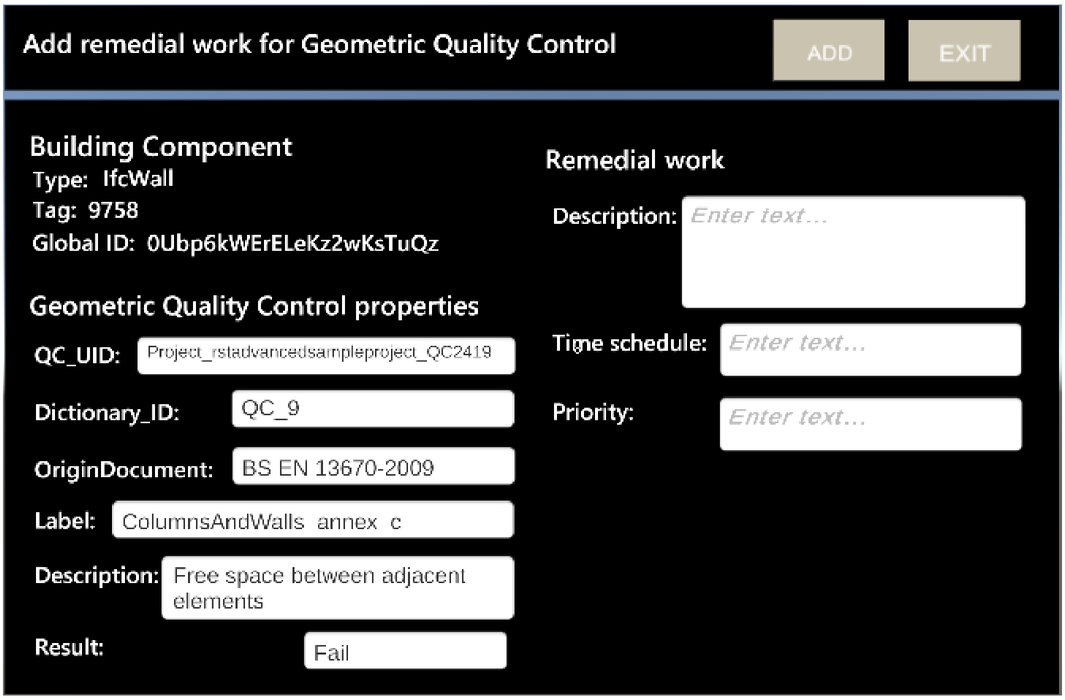


Figure 58 -The user can add a remedial work to a confirmed QC result

Safety Control mode: If the user selects the Safety Control mode in the *Select mode* menu in Figure 44, the DigiTAR tool downloads the H&S results for the specific project from the DT Platform. Downloading files from the SafeConAI via the DT Platform will be implemented in the next version of the DigiTAR tool. In this version, the Safety Control mode of the DigiTAR tool relies on an example IFC enriched with information regarding safety spaces provided by the SafeConAI. The BIM model as visualised in Unity is depicted in Figure 59. In Figure 60, Figure 61 and Figure 62 a walkable space, a fall hazard space and their overlap, as visualised in Unity, are depicted, respectively. The user is notified when they enter a hazardous space as depicted in Figure 63.

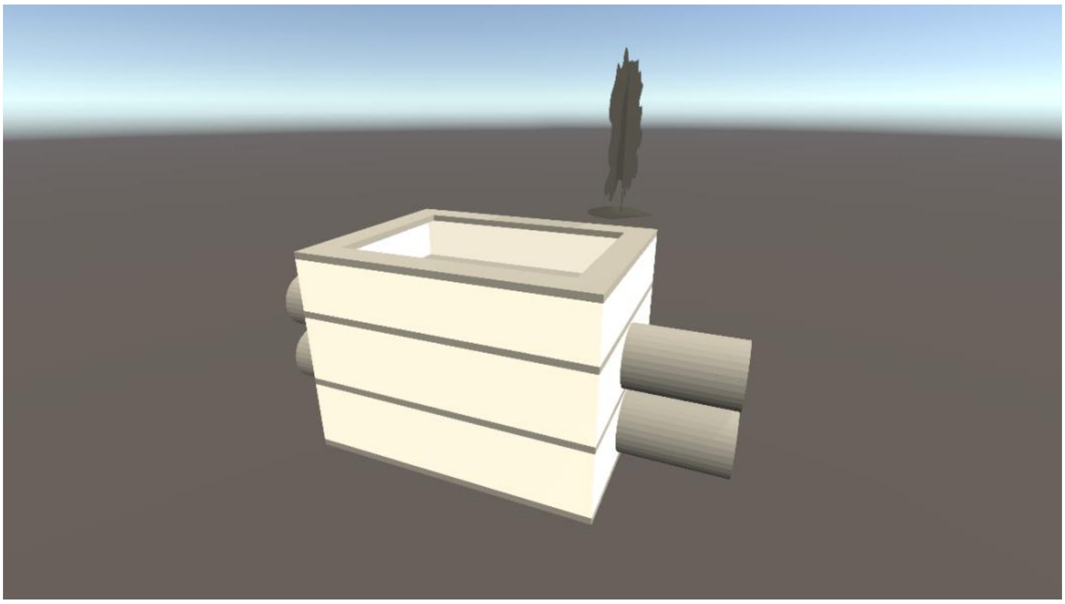


Figure 59 - The BIM model of the example IFC file enriched with safety information regarding safe spaces as visualised in Unity

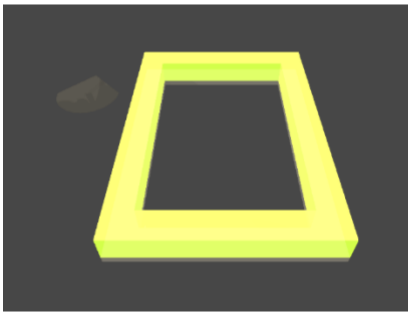


Figure 60 - Walkable space

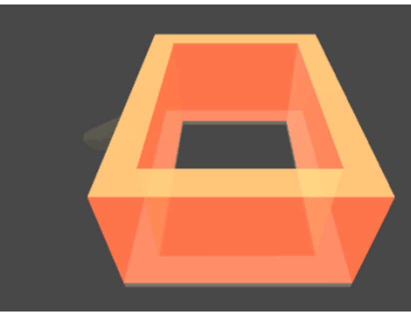


Figure 61 - Fall hazard space

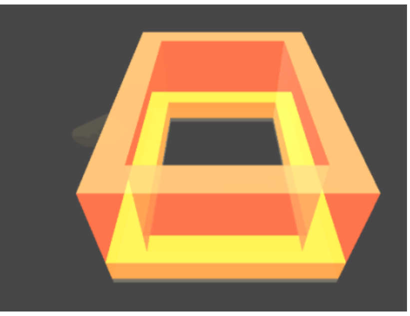


Figure 62 - Walkable and fall hazard space overlap

Figure 63 - The user is notified when entering a fall hazard space

2.5 Licensing

The DigiTAR module is a closed source component.

2.6 Installation Instructions

The DigiTAR tool can be installed via the Windows Device Portal of Hololens. The detailed instructions for the user to enable the Windows Device Portal feature on Hololens are[2]:

1. Power on the Hololens and put on the device.
2. Use the Start gesture to launch the main menu.
3. Gaze at the “Settings” tile and select it using the Touch gesture or using a Hand ray.
4. Select the “Update & Security” menu item.
5. Select the “For developers” menu item.
6. Enable the “Use developer features”.
7. Scroll down and enable “Device Portal”.

Subsequently, to install the DigiTAR application:

1. First, the certificate file (.cer file) that accompanies the DigiTAR application should be installed. To install it, the user should open the Windows Device Portal and navigate to "Views", "Apps" and select "Install certificate". Subsequently, the user should select "Choose File" and browse to their computer to find the certificate file. The user should select it and then press "Install".

[2] https://docs.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/using-the-windows-device-portal#setting-up-hololens-to-use-windows-device-portal

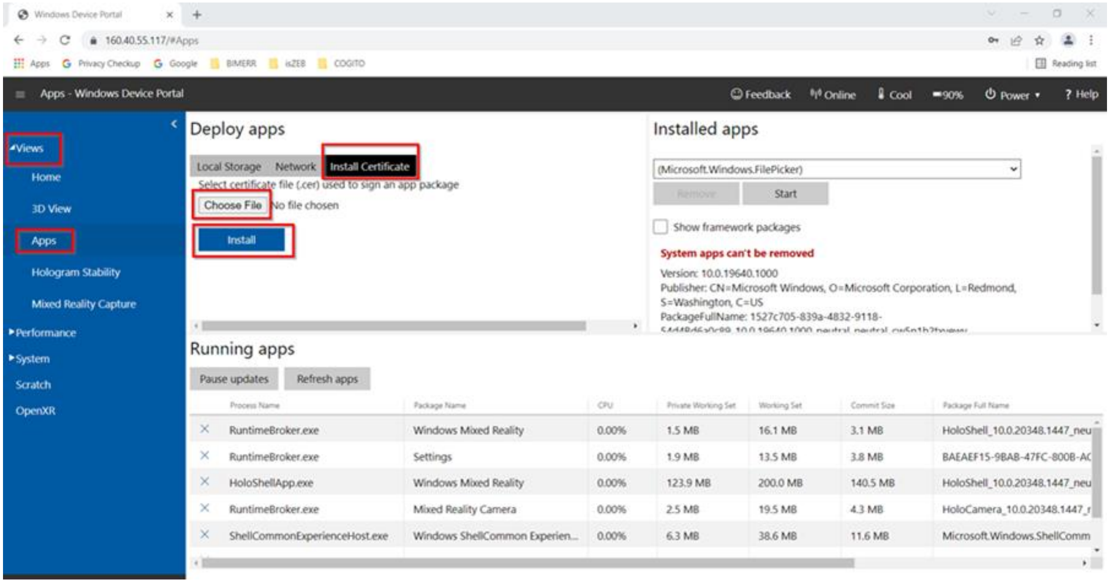


Figure 64 - Certificate installation via Windows Device Portal

2. To install the application (.appx file), the user should open the Windows Device Portal and navigate to "Views", "Apps". From "Local Storage", they should select "Choose file" and browse to their computer to find the application file. The user should select it and then press "Install".

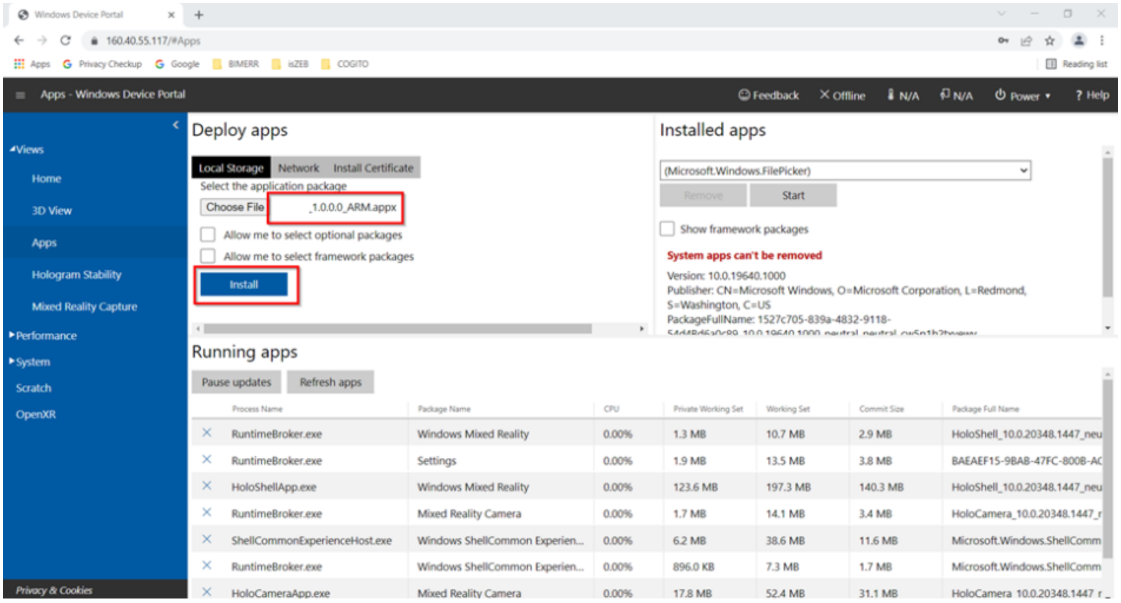


Figure 65 - Application installation via the Windows Device Portal on Hololens

3. The user is informed for the uploading and installation progress (see Figure 66), as well as when the installation is complete (see Figure 67).

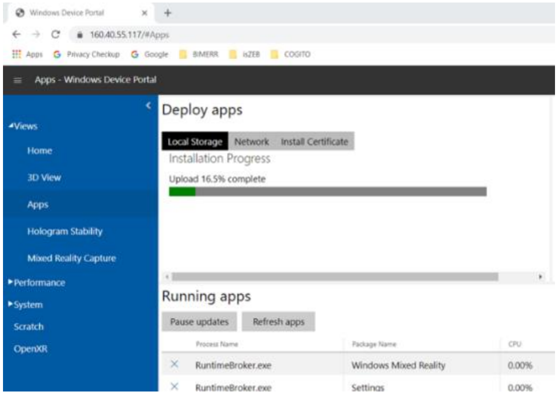


Figure 66 - Notification for installation progress

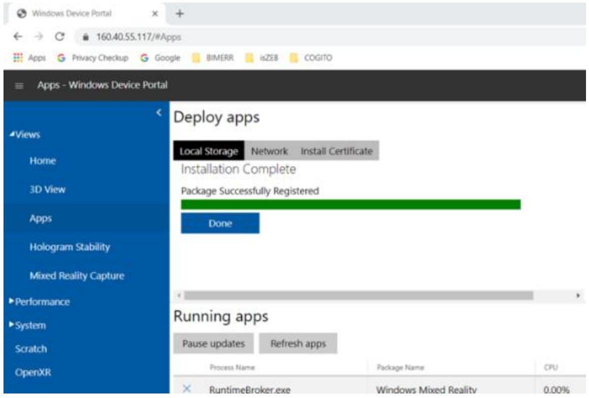


Figure 67 - Notification that installation is complete

2.7 Development and integration status

The DigiTAR module is currently under development. The alpha version of the tool has been developed and focuses on the Pre-processing and the Quality Control modes. The main functionalities, such as job addition, attachment of captured/selected images to a job, attachment of an IFC element to a job, filter selection, sending/receiving metadata and images with the Visual Data Pre-processing module have already been implemented and are ready to be used. Regarding the second release of the component in M24, the intention is to focus on the Safety Control mode. In this version of the DigiTAR, internal communication between the DigiTAR tool and the Visual Data Pre-processing module has already been established. In the final version of the tool, the communication with the DT Platform should be established and fully functional.

2.8 Requirements Coverage

Table 12 presents the stakeholder requirements as documented in D2.1 and are relevant to DigiTAR [10]. COGI-CS-2, COGI-CS-4 and COGI-CS-10 are merely covered by the programming language selected for the development of the DigiTAR tool. With regards to COGI-WF-2 and COGI-WF-4, the tool allows the Project Manager (PM) and Site Manager (SM)/Quality Manager (QM) to share information (design data, photos, point cloud etc.). In addition, concerning COGI-WF-5, the SM is able to share information with Subcontractors, Foreman and Workers (design data, photos etc.). Therefore, COGI-WF-2, COGI-WF-4 and COGI-WF-5 are going to be considered totally achieved at the end-to-end ICT COGITO System Integration.

Regarding the QC mode, COGI-QC-23, COGI-QC-24 and COGI-QC-25 are considered partially achieved since the basic functionalities have already been developed. However, refinements and updates must be implemented in order to visualise and validate the visual QC results, using graphic indicator, colourisation and accompanied by contextual information. Finally, as concerns the H&S issues, COGI-SF-3 and COGI-SF-4 will be covered in the second version of the DigiTAR tool in order to have the ability to visualise and confirm hazard identifications, as well as to provide feedback and mitigation works.

Table 12 - DigiTAR Stakeholders' requirements coverage from D2.1

ID	Description	Type	Priority	Status
COGI-CS-2	Runs on laptop and be usable on the construction site (remote access)	• Operational	Should	Achieved
COGI-CS-4	Runs on Windows	• Operational	Must	Achieved
COGI-CS-10	Allows visual comparison of current to planned status, either by AR glasses, mobile phones, or tablets	• Functional • Design constraint	Could	Achieved

COGI-WF-2	Allows the PM and Site Manager to share information (design data, photos etc.)	• Functional • Design constraint	Must	Partially achieved
COGI-WF-4	Allows the PM and Quality Manager to share information (design data, photos, design issues, schedules, work orders, materials schedule and usage, costs)	• Functional • Design constraint	Should	Partially achieved
COGI-WF-5	Allows the SM to share information with Subcontractors, Foreman, and Workers (design data, photos etc.)	• Functional • Design constraint	Should	Partially achieved
COGI-QC-23	Allows the QM to visualise and validate automated defect detections	• Functional • Design constraint • Performance	Must	Partially achieved
COGI-QC-24	Shows visualQC results using: graphic indicator; colourisation; and text	• Functional • Design constraint	Must	Partially achieved
COGI-QC-25	Shows visualQC defect with contextual information (link to components, defect history, etc)	• Functional • Design constraint	Should	Partially achieved
COGI-SF-3	Allows the HSE personnel to validate the automated hazard identification	• Functional • Design constraint	Could	Not yet supported
COGI-SF-4	Allows to employees to provide feedback and identify hazards	• Functional • Design constraint • Process	Should	Not yet supported

The functional and non-functional requirements were updated based on D2.5 “COGITO system architecture v2” [11] and are presented in Table 13. Concerning the functional requirements of the tool, Req.1-1 is covered while Req.1-2 is partially achieved. More specifically, the DigiTAR tool tracks every building component and defect that has been registered, but regarding the safety hazards, this version only tracks safe/unsafe spaces. As for the Req.1-3, it is partially achieved, since currently the local position of the user in the 3D BIM model is determined, but in the next versions the form of the coordinates may be different based on the final IFC files provided with georeferenced coordinates. Req-1.4 is achieved for the current form of received defects and safety hazards, but of course, any modification made to these files will lead to the corresponding modification on the DigiTAR. Req.1-5 and Req.1-6 are partially achieved and will be well covered in the final version of the DigiTAR tool. Req.1-7 is also partially achieved, since the communication with the DT Platform will be implemented in the T8.1 “End-to-end ICT System Integration, Testing and Refinement” (M18-M34). Finally, Req.1-8 and Req.1-9, that involve the communication between the DigiTAR and the Visual Data Pre-processing module, are achieved in this version of the DigiTAR tool.

Concerning the non-functional requirements, the status of Req-2.1 is considered to be partially achieved given that the application will be further updated, improved and extended in the next and final version. Req-2.2 is also partially achieved since it regards the WiFi connectivity while using the application on-site. Req-2.3, and Req-2.4 are covered since the DigiTAR is an AR application that already provides menus to let the user choose among the application’s modes and functionalities. Req-2.5 is considered to be partially achieved, since although the architecture and technologies used in DigiTAR tool allow for both vertical and horizontal scaling, processing and resource demands may be restricted by the headset’s computing capabilities. Req-2.6 can be considered as well covered by this version of the component, while Req-2.7 will be completely achieved in the next version of the tool when Keycloak integration will be established.

Table 13 – DigiTAR Functional and Non-Functional Requirements coverage from D2.5

ID	Description	Type	Status
Req-1.1	Maps the BIM 3D model enriched with defects and safety information, to the site.	Functional	Achieved
Req-1.2	Tracks every building component, defect and safety hazard that has been registered.	Functional	Partially achieved
Req-1.3	Determines user's position and orientation.	Functional	Partially achieved
Req-1.4	Confirms annotations about defects and safety hazards.	Functional	Partially achieved
Req-1.5	Creates annotations about safety hazards.	Functional	Partially achieved
Req-1.6	Creates task annotations for remedial work and safety hazard mitigation work.	Functional	Partially achieved
Req-1.7	Sends relevant information about annotations to DT platform.	Functional	Partially achieved
Req-1.8	Captures and sends visual data to Visual Data Pre-processing module.	Functional	Achieved
Req-1.9	Receives and displays processed visual data from Visual Data Pre-processing module.	Functional	Achieved
Req-2.1	User friendly interface.	Non-Functional	Partially achieved
Req-2.2	WiFi connection on site.	Non-Functional	Partially achieved
Req-2.3	Menu to discern the different functionalities.	Non-Functional	Achieved
Req-2.4	AR Application.	Non-Functional	Achieved
Req-2.5	Scalability.	Non-Functional	Partially achieved
Req-2.6	Reusability.	Non-Functional	Achieved
Req-2.7	Security.	Non-Functional	Partially achieved

2.9 Assumptions and Restrictions

The first version of the DigiTAR tool is accompanied by certain assumptions and restrictions, which are presented in the following:

- The current version of the tool is oriented to the Pre-processing and the Quality Control modes. For the Safety mode, it currently supports only the notification for safe and unsafe spaces in the IFC file enriched with safety information. When the ontology in D4.1 "Preventive Health & Safety Application v1" and the corresponding IFC files are created by the SafeConAI, the safety-related functionalities of the DigiTAR tool will be extended in the final version of tool.
- The current version of the tool communicates with the Visual Data Pre-processing module using their REST API endpoints that support queries regarding jobs, capture device and filter specifications. Additional refinements to simplify and enhance the efficiency of this bilateral communication could be made in the next version of the tool.
- The communication with the DT Platform and the rest COGITO components will be established and tested next months during the integration process (T8.1 End-to-end ICT System Integration, Testing and Refinement) and the DigiTAR tool should be adjusted to arising needs.

- In this first version of the tool, no credentials are used to login to the AR application. The authentication and authorization of the users will be implemented through Keycloak in the next months, during the System Integration process.

3 Conclusions

This deliverable introduced the main functional components of the DigiTAR module and their use. Several information regarding the development tools, the data exchange and an alpha version of the API that the module uses were also presented.

The DigiTAR module is an AR application, which offers many functionalities to the user. The DigiTAR module has three different modes: the Visual Data Pre-processing mode, the Quality Control mode and the Safety mode. Within the DigiTAR tool, the user is able to view the 3D BIM model on-site with the 3D IFC components overlaying the physical components. Due to the Data Collector functionality of the DigiTAR tool, the user can also capture image data on-site. The user can create pre-processing jobs, attach IFC elements to them, capture or select images to be linked to the pre-processing job and send job-related data and metadata to the Visual Data Pre-processing module. The user can view the processed/enhanced images within the DigiTAR tool and push them to the Visual Data Pre-processing module which further promotes them to the DT Platform, where they can be assessed by other COGITO components (Geometric and Visual Quality Control tools). The tool also notifies the user for Quality Control results and requests for on-site confirmation, along with the suggestion of a remedial work. Finally, the DigiTAR module notifies the user for potential construction site hazards. In the current version of the tool, notification concerns fall hazard spaces, but will be further extended in the next version of the tool based on the modifications made to the IFC enriched with safety results generated by the SafeConAI module.

The work presented here introduces a first version of the DigiTAR module. Thus far, the module has been tested only with basic and sample data, e.g., sample QC results, sample IFC enriched with safety results. In the future and as COGITO tools evolve, more functionalities will be added to the module, to support all the emerged demands as well as the connectivity and communication with the other COGITO tools. The final version of the DigiTAR module will be provided at M24 and will be tested on the pre-validation (T8.2) and validation (T8.4) sites with real case data during M21-30 and M28-34, respectively.

References

[1] S. Lockley, C. Benghi and M. Cerny, "Xbim.Essentials: a library for interoperable building information applications," *Journal of Open Source Software,* vol. 2, no. 20, p. 473, 2017.

[2] [Online]. Available: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/.

[3] [Online]. Available: https://standards.buildingsmart.org/.

[4] [Online]. Available: https://unity.com/.

[5] [Online]. Available: https://www.microsoft.com/en-us/hololens/hardware.

[6] [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/develop/unity/mrtk-getting-started.

[7] [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/develop/platform-capabilities-and-apis/using-the-hololens-emulator.

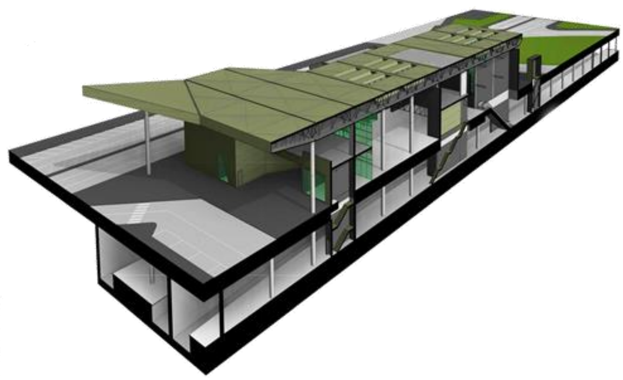
[8] [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/design/motion-controllers.

[9] [Online]. Available: https://docs.opencv.org/3.4/d3/dc1/tutorial_basic_linear_transform.html.

[10] D2.1-COGITO, "Deliverable 2.1: Stakeholder requirements for the COGITO system," 2021.

[11] D2.5-COGITO, "Deliverable 2.5: COGITO System Architecture v2," 2022.

[12] [Online]. Available: https://microsoft.github.io/MixedRealityToolkit-Unity/Documentation/Updating.html#upgrading-to-a-new-version-of-mrtk.



COGITO

CONSTRUCTION PHASE
DIGITAL TWIN MODEL

cogito-project.eu

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