Ref. Ares(2024)6560709 - 17/09/2024

Ref. Ares(2024)656532 - 29/01/2024

Ref.Ares (2022) 8272732 - 30/11/2022

European Commission

ReInHerit

Redefining the Future of Cultural Heritage, through a disruptive model of sustainability

ReInHerit
Redefining the Future of Cultural Heritage

www.reinherit.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004545

Project

Project Number	101004545
Project Acronym	ReInHerit
Project Title	Redefining the future of cultural heritage, through a disruptive model of sustainability
Starting Date	01/03/2021
Duration in Months	36
Funding Scheme	Coordination and Support Action
Call (part) Identifier	H2020-SC6-TRANSFORMATIONS-2020
Topic	TRANSFORMATIONS-19-2020 *Culture beyond borders – Facilitating innovation and research cooperation between European museums and heritage sites*
Website	www.reinherit.eu

Deliverable

Work Package	3
Task	3.4
Deliverable	3.7
Dissemination Level	Public
Type of Deliverable	Demonstrator
Leader	CYENS – Centre of Excellence
Due Date	28/02/2023
Submission Date	
Keywords	

Version History

Version	Date	Author	Notes
V1	24/02/2023	Marco Bertini Chiara Zuanni Kleanthis Neokleous Fotos Frangoudes Christodoula Gavriel Alexis Polydorou Andreas Papapetrou	
V2	28/02/2023	Marco Bertini Chiara Zuanni Kleanthis Neokleous Fotos Frangoudes Christodoula Gavriel Alexis Polydorou Andreas Papapetrou	
V3	20/03/2023	Kleanthis Neokleous Fotos Frangoudes Christodoula Gavriel Alexis Polydorou Andreas Papapetrou	
V4	1/2024	Marco Bertini, Paolo Mazzanti	Revision - based on the “General Project Review Consolidated Report” No. 2
V5	8/2024	Marco Bertini, Paolo Mazzanti, Kleanthis Neokleous Fotos Frangoudes Christodoula Gavriel Alexis Polydorou Andreas Papapetrou	Revision - based on the “General Project Review Consolidated Report” No. 3

Acronyms and abbreviations

Field of View	**FOV**

Disclaimer

This document reflects only the author's view and the Research Executive Agency is not responsible for any use that may be made of the information it contains.

Table of Contents

Executive Summary

This final version of the Mobile devices that includes links to the ReInHerit Digital Hub, together with the source code, guidelines, manuals and relevant material. In particular, new sections have been added regarding the usage of the apps (see section 2.5) and updated information has been reported based on the revised versions of the mobile apps the corresponding sections of the deliverable.

Mobile devices have become ubiquitous in our everyday lives, offering an accessible and easy touse medium that can showcase content at cultural heritage sites and museums. Toward this goal, a number of mobile applications were developed as part of the project that can be used in different scenarios. The report presents four such applications, with details on how they were developed, used, and how they can be integrated into future projects.

The applications are divided into two subsections. The first corresponds to the Immersive Performance on the House of Hadjigeorgakis Kornesios on the 29th of October 2022 and the second section to the mobile applications for museums, Smart Lens, Strike-a-pose and Face fit.

D3.7: Demonstrator Mobile Applications: Final Version Report

1 Introduction

A set of mobile applications were developed that can be used by audiences to interact and view content at cultural heritage sites and museums. In particular, is somehow a continuation of deliverable regarding the toolkit strategy (D3.3) and an introduction to the forthcoming D4.4 and D4.5 be reports on the immersive performances' apps and data.

In total, four different applications were developed that can be used by audiences to:

a) interact with the performance environment (including both hardware equipment and performers)
b) act as smart guide tools that adapt to the actions and interests of the visitor of a museum on site
c) act as tools based on gamification and learning-by-doing using techniques such as style transfer and "deep fakes" applied to user-generated content (e.g. transforming participants' photos into paintings or painting/sculpting the visitors in museum exhibits)

These applications utilize novel technologies like machine learning, computer vision, etc., and collect anonymous user data for analysis and studying user behaviors.

The remaining of the report, presents each of the four applications, describing the purpose of their development, information about their development, how they were used in the project, and guidelines on how to reuse them.

2 Application 1: Immersive & Interactive Music Performance

2.1 Purpose

The purpose of the application is to create an immersive audio experience that can be setup at museums. Immersive audio stations are placed in various areas of a museum, offering visitors an interactive experience. Users that enter in the corresponding areas within the museum, based on their behavior (e.g., based on how many people are in the area, or how active they are), trigger specific sounds that are layered together creating a music composition.

2.2 Application Description

The project consists of two applications which both run on smart mobile phones and were developed using Android Studio:

a) The first one is called "Station", and its main purpose is to detect users in an area while streaming the appropriate sounds to a speaker.

b) The second one is called "Remote", and its purpose is to connect with the various stations that are spread across the museum and control them remotely.

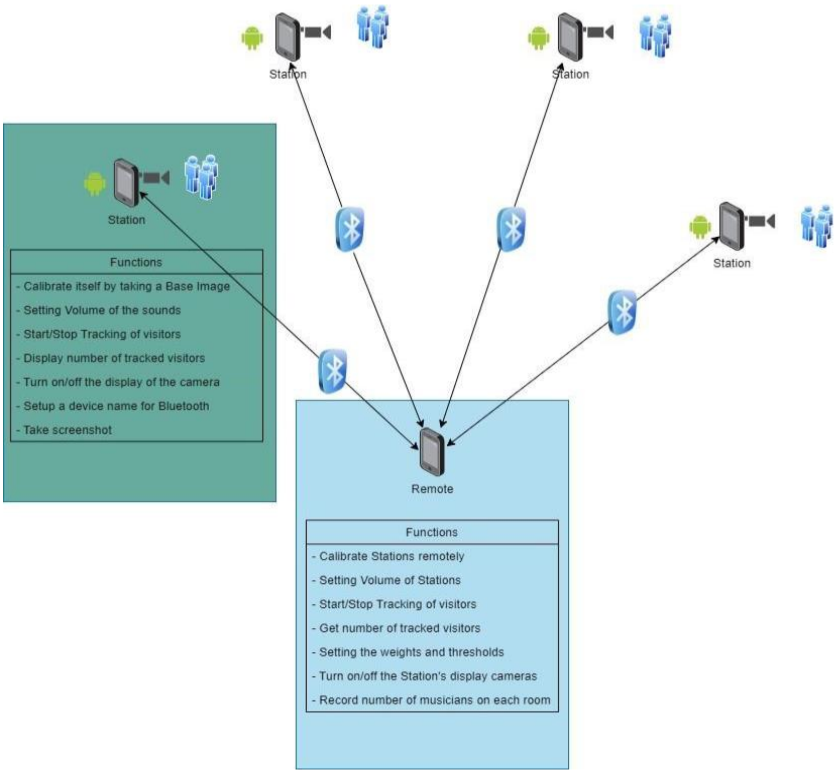


Fig. 1 - Setup of both applications.

2.2.1 Station application

The station application is responsible for tracking users' behavior within their Field of View (FOV). The application, using the integrated back-facing camera of the phone, scans the room for changes in their FOV, and triggers different sounds based on these changes. When greater changes are observed, additional sounds begin playing in additional layers.

By default, the application is programmed to play 5 different sounds, however this number can be adjusted based on the desired setup. At the same time, a constant sound also plays in the background regardless of the user's behavior.

2.2.1.1 People detection and tracking

An initial approach for analyzing user behavior was people detection and tracking in 3D space using OpenCV. More specifically, the Kernelized Correlation Filter (KCF) Tracker[1] was used, first for detecting people and then tracking them as they moved through space. The specific algorithm was used because of its speed and relatively good accuracy which was essential for the requirements of our use case. A chequerboard was also used to calibrate the camera, which could then provide the 3D coordinates of people in space.

[1] J. F. Henriques, R. Caseiro, P. Martins and J. Batista, "High-Speed Tracking with Kernelized Correlation Filters," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 3, pp. 583-596, 1 March 2015, doi: 10.1109/TPAMI.2014.2345390.

Even though this solution was feasible and operational under normal conditions, when it was tested at an actual museum it failed to work as intended. The museum that the application was tested was the House of Hadjigeorgakis Kornesios since this was where the main event would eventually take place. The tests were held prior to the event throughout the month of October, and it was made clear from the first trials that this solution would not be feasible (At least for the open spaces). The main issues with it were the often low lighting conditions in some of the areas, as well as restrictions on where the phones could be placed in space, which prevented having a clear view of the people.

2.2.1.2 Frame Changes

Due to the above limitations, an alternative processing framework was developed. More specifically, instead of tracking exact users, changes within the captured frames were analyzed directly. The first step in the process is the capturing of a “base” image of the area that is going to be observed. Afterwards, the application continuously checks every frame and compares it to that first “base” image and identifying differences between the pixel values. Since this procedure produces high CPU load, each frame is converted to a grayscale image and is also rescaled to a lower resolution. That is how the framerate of the application is always stable without any interference.

Moreover, the process can change the implemented algorithm to detect frame-to-frame changes by adjusting certain weights and thresholds by calculating the different values between consecutive frame pixels. This is done by comparing each frame to the last received frame. That way, the application can not only detect differences between the initial state of the area, but also changes that occur due to the movement of the crowd that is exploring the museum. Currently, the weights control how much the movement of the crowd is affecting the algorithm results while the thresholds affect the general sensitivity of the device.

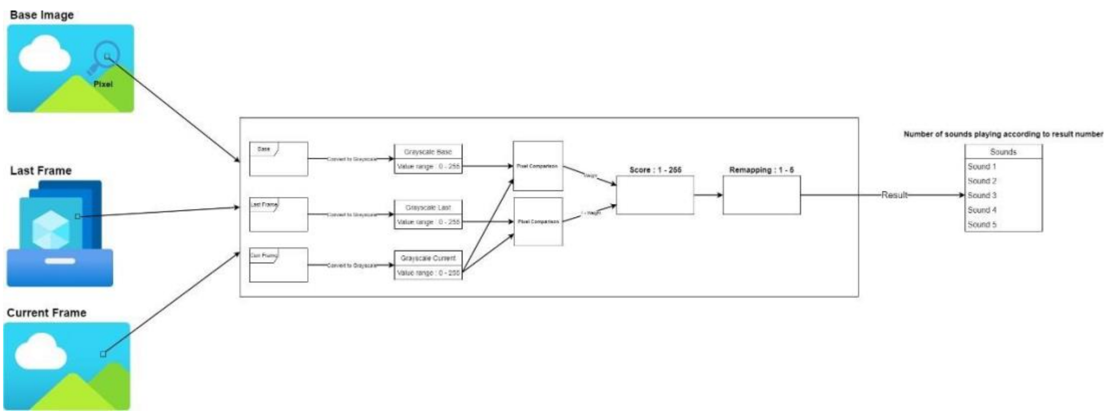


Fig- 2 - Description of the algorithm for image comparison.

2.2.1.3 Data Logging

Finally, the station application has integrated logging capabilities, which record the analyzed values in every frame to a CSV file that is easily accessible on the phone. The primary purpose of the data logging is for post-processing analysis of the activity at each station. The logging provides detailed measurements of the activity during each frame, both compared to the base image, but also to the prior frame. Therefore, one can simulate the activity that occurred at a station during the event, even after it finished. However, these logged data can also be used for debugging purposes during the setup of the stations. For example, values can be recorded from a specific location and then analyzed in order to validate whether the chosen parameters (i.e., thresholds and weight) provide accurate results or if they need adjustments.

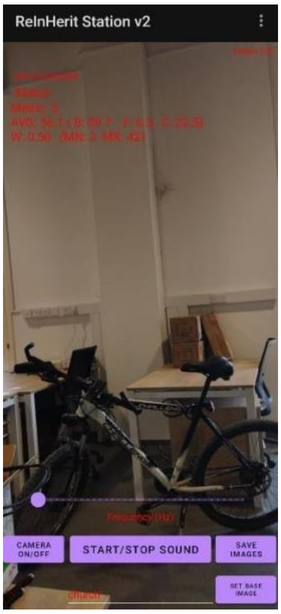


Fig. 3. - Station application main UI.

2.2.2 Remote application

The second application can connect through Bluetooth to the Stations that are around the museum and control them. Its purpose is to be used by the staff members of the museum while the exhibition is taking place. The staff can walk around the rooms and control the Stations remotely. If there are any changes in the lighting or around the area where a Station is monitoring, the Remote application can easily reset the Station's settings (most notably the weights and the thresholds for calculating the values between the pixels).

2.2.1.4 Main functionalities

The remote application haste following functions:

a) Raise or lower the volume of the Stations.

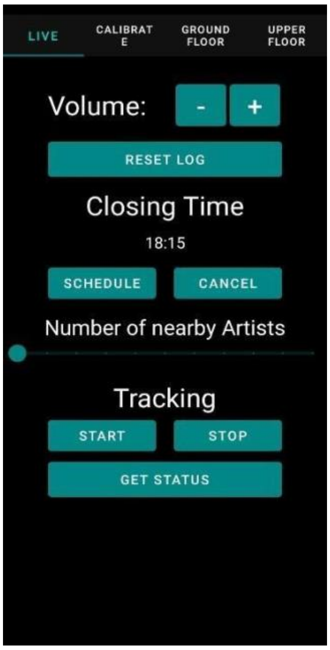
b) Start/Stop the Station's tracking ability.

c) Get the status of the Station. That means that the user can get the number of people that are currently being detected along with the weights that are being used by the station to determine that.

d) Set the weights and thresholds that a Station is using.

e) Turn on/off the display of the Station to save battery since the duration of a museum exhibition can last for 5 to 6 hours.

f) Record the number of artists and/or visitors that are joining a room in a file. This data can be later used to extract valuable information about the movement of the crowd during the exhibition.

(a) 

(b) 

Fig. 4 - Remote application (a) main UI, and (b) Calibration menu UI.

2.2.1.5 Additional information logging

The remote application also had the ability to track and log information about events happening near each of the tracked areas. For each area, where a station is located, staff can track by using a slider (ranging from 0 to 10) the external stimuli within the area.

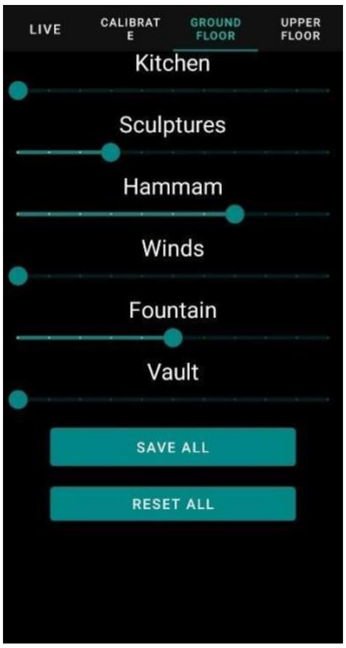


Fig. 5- Menu to record manually how many artists are currently located in each room. Every time a value changes, it is saved internally in a csv file.

2.3 Usage example

2.3.1 Venue and Setup

The application as described above was used during an immersive performance that took place at The House of Hadjigeorgakis Kornesios (Ethnological Museum) on the 29th of October 2022 and spanned both outdoor and indoor areas. In total, interactive stations were placed in 6 locations on the ground floor and 8 locations on the first floor as shown in Figure 6.

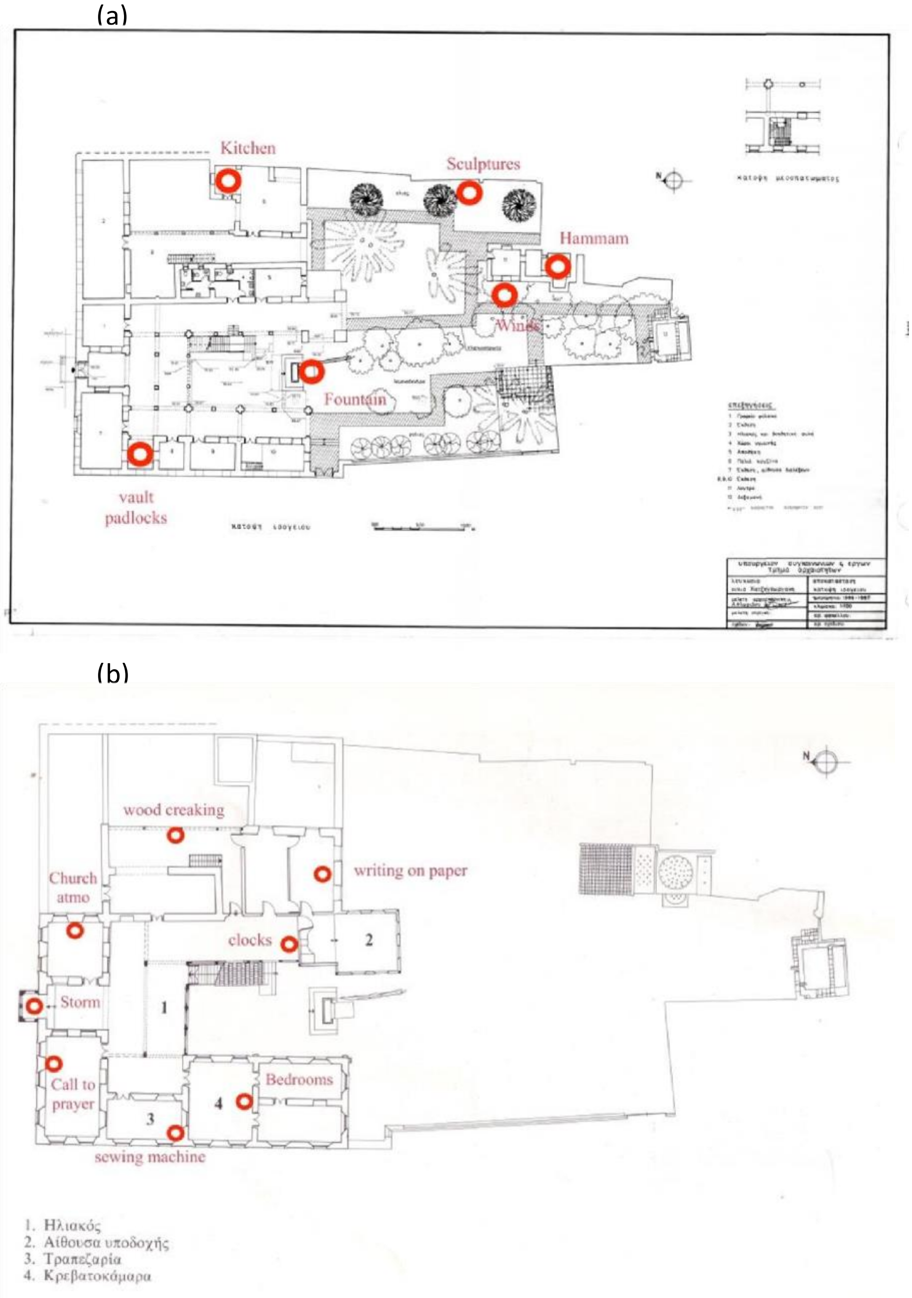


Fig- 6 - The layout of The House of Hadjigeorgakis Kornesios and all the locations where stations were placed on (a) the ground floor and (b) the upper floor.

Fig. 7 - The exhibition at The House of Hadjigeorgakis Kornesios.

2.3.2 Equipment

Around the mansion, 14 mobile phones with cameras were scattered in various areas. Each station was equipped with a Xiaomi Redmi 10C mobile phone, and each was connected to a Bluetooth speaker that would then output the audio. During the performance, 3 mobile phones were also equipped with the Remote application in order to control the stations.

Fig. 8. - Phones used as station and remote for the exhibition.

2.4 Guidelines for reuse

These are the proposed steps to be followed when setting up a system for such a performance:

a) The mobile phones that contain the Station application should be placed at elevated positions where they can maintain a good FOV. Also, visitors should not be able to reach them and move them out of position since this will affect the functionality of the application. Also, if it is possible, the devices should include locking mechanisms to avoid theft.

b) The Bluetooth speakers should also be hidden from the eyes of the visitors and placed near points of interest. This will make the exhibition more interactive and engaging.

c) When every mobile phone and Bluetooth speaker is set up, the Remote application must be used to configure each Station. Primarily, set the "base" image, configure the volume, and define the weights to be used for the detection.

d) If during the performance something inside the room is changed (lights, objects moved, etc.), a reset must be performed to the Station application.

It is important to mention that data collected during the event do not contain any sensitive private information about the visitors or the performers. The application only tracks the number of people passing in front of the cameras and the number of the performers that are located inside a room at any given time. Also, the stations do not record any video nor have the ability to do so which protects the anonymity of all guests during the event.

More details regarding the specific app are provided in dedicated sections of the ReInHerit digital hub with manuals, instructions, guidelines and links to the source code for the recreation of the experience, in a more "personalized" context for other potential cultural heritage sites across Europe.

2.5 Technical Guidelines

The code for the application can be accessed from the following GitHub repository link: https://github.com/CYENS/Reinherit-Hadjigeorgakis-Kornesios-Mansion.

In this section we are going to describe some important features of both the applications and review parts of their code and the necessary files that are needed in order to re-create a similar experience.

2.5.1 Setting up the Base Image

Firstly, the Base Image need to be setup. It is the base image that the Base Station will compare every frame when the exhibition starts. By detecting differences in pixel values, it would be able to estimate how many people are in front of the camera and will play the appropriate sounds. The Base Image is taken before the exhibition so that no visitors are present in the picture. Another important step that needs to be taken in order to correctly setup the Base Station is to calibrate three more parameters:

1. The Weight.

2. The Minimum Threshold.

3. The Maximum Value.

The weight describes how sensitive the Base Station would be in detecting rapid changes in the pixel values. Its value ranges from 0 to 1 and by default is set to 0.5. By changing this value, the base station gets more sensitive in detecting people moving in front of the camera or it gives more weight in comparing the Base image with the current frame. The Minimum Threshold determines the least amount of pixel values that need to be changed to register a visitor while the Maximum Value is the upper limit of pixels.

2.5.2 The "Pixel comparison algorithm"

The Base Station application uses a pixel comparison algorithm that determines the number of visitors located in a room. The implementation of the algorithm can be found in the "CameraInputManager" class. The algorithm has the following steps:

1. Setup the Base Image. It is required in order for the algorithm to be able to do comparisons.
2. Get a new frame from the camera.
3. Convert the frame to grayscale and rescale it.
4. Check the value of every pixel in the new frame with the value of every pixel in the Base Image and the previous frame that was received from the camera.
5. Determine how many visitors are in the room.
6. Repeat for the next frame.

The grayscale conversion and rescale of the processed frame is done to make the pixel comparison between the two images faster since we are checking every frame the Base station's camera provides. Based on the previous parameters (Weight, Minimum Threshold and Maximum Value), the algorithm then determines the number of visitors and plays certain number of sounds. The Base Stations need to be loaded with the sound files beforehand, and they all follow a specific name format, "sound" plus a number between one to five.

Fig. 9 - Sound files in the Base Station using the specific name format with the prefix "sound".

The sound file with the number zero is the default background sound that is always playing and acts as ambiance. The rest of the sounds are played simultaneously when there are visitors present. The number in the sound file name represents the number of visitors that should be in the room for the sound to be played.

2.5.3 The “SoundManager” class

To play all these sounds at the same time we use the “SoundManager” class which creates a new thread each time a sound needs to be played.

```java
//MusicThread class that extends java Thread. Run multiple instances of this class to play simultaneously multiple sources of audio
public static class MusicThread extends Thread{
    public MediaPlayer mSoundPlayer;
    private boolean mIsRunning;
    private final boolean mLoopSound;

    public MusicThread(int id, String filePrefix, String fileExtension, Activity parentActivity, boolean loop){
        this.mSoundPlayer = createMediaPlayer(id, filePrefix, fileExtension, parentActivity);
        mLoopSound = loop;
    }

    public void stopPlayback() { this.mIsRunning = false; }

    //Main function that runs everytime a new thread is created
    public void run(){
        if (mSoundPlayer == null) return;

        this.mIsRunning = true;

        mSoundPlayer.start();
        mSoundPlayer.setLooping(mLoopSound);

        try {
            while(mSoundPlayer.isPlaying() && mIsRunning){
                Thread.sleep( millis: 100); // wait
            }
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        mSoundPlayer.release();
    }
}
```

Fig. 10 - The “MusicThread” class creates a new thread each time a new sound needs to be played. The sound will stop playing when the “stopPlayback” function is called.

2.5.4 The “Remote station”

Both applications, Base Station and Remote, communicate by using the Bluetooth protocol which enables them to exchange messages between them. Primarily, the Remote sends most of the requests to the base station to calibrate it since most of the time the stations are in difficult to reach positions. In order to create new requests to be sent from the Remote, one can edit the “PerformanceController” class and add new message requests by adding listeners to new or existing buttons. Similarly, by editing the “MainActivity” class, we can write new functions or edit the existing ones that will execute every time we receive a specific message in the Base Station.

```java
Button btn1 = (Button) view.findViewById(R.id.startButton);
btn1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        try {
            Toast.makeText(getActivity(), "Starting...", Toast.LENGTH_LONG).show();
            mViewModel.sendMessage(new JSONObject().put("id", Constants.COMMANDS.START).put("val", "0").toString());
        } catch (JSONException e) {
            e.printStackTrace();
        }

    }
});
```

Fig. 11 -. Adding a button that will send a message to the Base Station to start monitoring a room.

```java
switch (id) {
    case GET_STATUS:
        //mService.write(mStatus.getText().toString().getBytes()); <--new code follows. Uncomment this if needed
        mTrackerText = findViewById(R.id.textViewTracker);
        String textToSend = "s"+mTrackerText.getText();
        mBluetoothManager.write(textToSend.getBytes());

        //save a screenshot of the camera
        ImageUtilities.saveBitmap(mPreviewView.getBitmap(), "gs_");
        break;
    case RESET_CAMERA_POSE:
        break;
    case SET_FREQ:
        //set the frequency of the test sound and adjust the position of the slider UI
        Slider slider = findViewById(R.id.sliderFreq);
        slider.setValue(valueInt);
        break;
    case START:
        //start all sounds
        soundStopped = false;
        break;
```

Fig. 12 - Handling the messages that can be received from the Remote.

Besides the calibration, the Remote can control the volume of the Base Stations. It can also start or stop the camera detection and it can request from the Base Station to send the number of visitors that are currently in front of the camera along with all the current parameters.

Another aspect that the Remote is controlling is the number of musicians that are currently present in a room. Each Base Station is typically associated with one room and with the Remote someone can register manually the number of musicians that are in it and save that information in .csv logs which are created at the Base Station. Along with the number of musicians, the Base Station automatically saves the Day of the recordings, the station's parameters and the number of visitors that pass through.

2.5.5 Setup of the specific performance

For the House of Hadjigeorgakis Kornesios event, there were 14 rooms in total which were in two floors. The Remote application can be edited to accommodate more than two floors and more rooms for each occasion.

3 Application 2: Smart Lens

3.1 Purpose

The purpose of this application is to recognize details of artworks and provide associated information. The goal is to induce the user to be more proactive when looking at an artwork, searching for its details using his mobile phone (or tablet) as if it was a smart magnifying lens. The app uses a variety of computer vision techniques (CBIR, classification or object detection) to recognize the desired details. The details that are recognized are shown in the interface and clicking on them the app provides the related multimedia information. The app can be used as a smart guide to a collection or for a set of artworks that are of particular relevance and that can necessitate of a thorough analysis. The type of interaction elicited from a user is more active than that of a typical guide, e.g. based on QR codes, since it calls for an active analysis of the exhibition.

3.2 Application description

The application allows the recognition of artwork details using three alternative methods:

- Content-based Image retrieval (CBIR): this modality allows to avoid re-training the neural network used to recognize the details, since details are represented using visual "embeddings" representing the content of parts of the framed image computed with an ANN for image classification.
- Classification: this modality requires to re-train a neural network for image classification, although this is relatively low cost in terms of required computational performance; typically, it provides a better recognition than CBIR.
- Object detection: this modality allows to use the app also when users are relatively far from it. The retraining of the neural network is computationally more expensive than that of classification, but the results are typically better in terms of precision.

3.2.1 Content-based Image retrieval (CBIR)

The CBIR mechanism allows the recognition of the works and their details through the comparison between the visual features extracted from a dataset of images (the collection of details stored in a database, i.e. in the guide) and those obtained from the frame produced by the user's camera.

To allow the recognition of the details of the works, the features have been extracted not only from the entire work but also from some of its parts obtained through a rigid subdivision as shown in the following figure. The system has been designed to use 4 o 5 partitions, i.e. in the 4-parts version the center patch shown in the figure is not computed, providing a slightly faster computation.

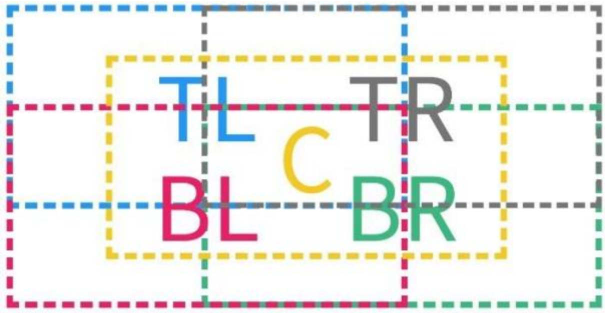


Fig. 13 - Image subdivision/augmentation used for CBIR

3.2.2 Classification

This second approach allows the recognition of the framed detail / work through a classification network on which a fine-tuning operation has been performed on the paintings and their details present in the dataset. This requires, differently from the previous approach, a retraining of the neural network; it must be noted that typically this retraining can be performed even using CPUs.

In this version, therefore, the classification of the entire frame is performed, which is identified with the most probable class only if the degree of confidence is greater than a specific threshold for each image.

In order to improve the training results a backend system has been implemented to create "augmentations" of the images of the details, i.e. creating from a reduced set of images a large number of variations subjected to a set of distortions and transformations that account for different points of view (e.g. geometric and perspective distortions), imaging sensors (e.g. RGB color shifts), motion (e.g. blur and motion blur), noise (e.g. simulated dirty lenses). The images created with this backend system allow to create more robust image classifiers.

3.2.3 Object detection

In this version, the recognition is entrusted to a network for Object Detection on which fine tuning has been performed on the details of the work. This retraining is more expensive than the previous one and requires a GPU.

In this case, only the details within the work are identified by taking the most probable bounding-box (i.e. a rectangle that most probably contains the object of interest) for each class provided that its degree of confidence exceeds a specific threshold for each detail.

A detector based on SSD and MobileNet backbone has been used. Similarly, to the backend system for image augmentation used for classification, another backend system has been developed. The transformations are the same as that designed for classification, since the same distortions and modifications apply in this case, the additional functionality is that this augmentation system must account for the need to keep bounding boxes of the desired details visible in the augmented images.

3.2.4 GUI

The interface is entirely made in HTML5 and CSS and has a fluid behavior adapting to the size of any screen (tablet or smartphone).

It is dynamically modified through JavaScript scripts that allow you to show the results of the matches with the images framed by the camera in real time. In the camera view, the work or detail is framed, and the recognition operation is carried out in real time. This view remains hidden until the networks to be used have been loaded correctly as well as the data from the server (see following figure).

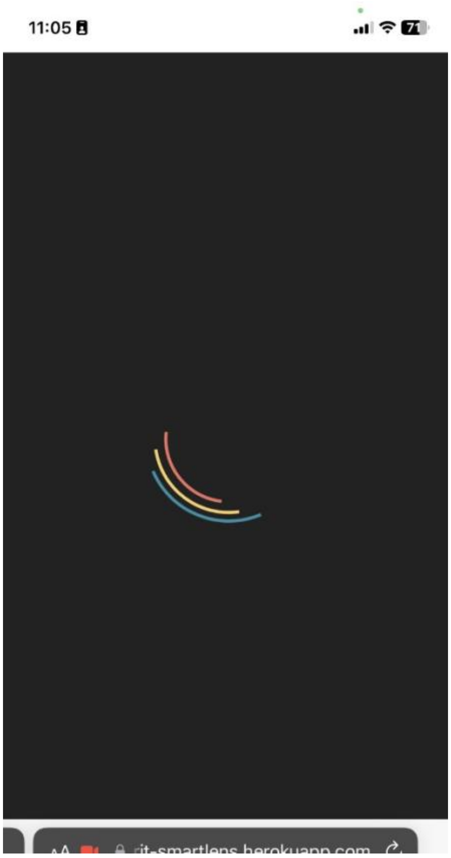


Fig. 14 - View hidden until loaded correctly.

The recognized details are shown as small thumbnails in the GUI. Selecting one of them leads to the associated information, as shown in the following figure. It is interesting to note how using image augmentations it is possible to obtain a system that works even with rotated images. Note also how the system works on 3D artifacts, paintings, prints, etc. For each work or detail is provided a textual description, the relative image and any audio and video content.

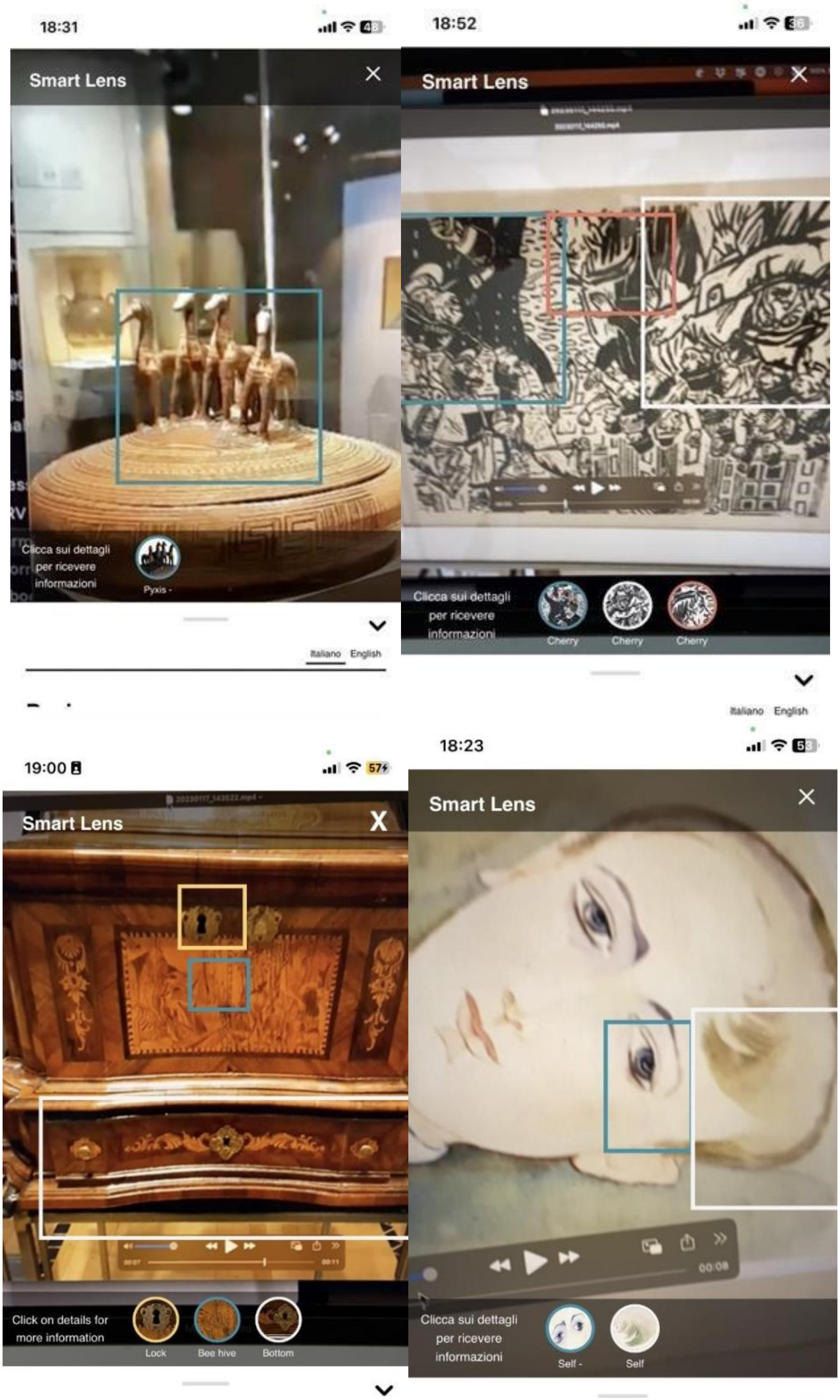


Fig. 15 - Details and thumbnails.

3.3 Usage example

There is need to use a server to host the backend and the web app itself; it is possible to use a server on-premises or hosting services such as Heroku. A QR code can be used to avoid typing the URL of the web apps.

3.4 Guidelines for reuse

The application is composed of a backend that maintains information on the details of the artworks. The first step for reuse is to create the guide itself in all the languages that the museum wants to support or to extend the current set of details to add specific or desired ones.

The types of interaction vary according to the techniques used to recognize the details: using object detection it is possible to analyze artworks also from afar, getting hints at what parts of the artwork are the "hotspots" for which the information is provided. Using the classification network the user must inspect the artwork more thoroughly, mimicking more the use of a magnifying lens. Using CBIR it is possible to create a user experience between those of object detection and classification, reducing the time to create a dataset (no neural network must be retrained) but at the possible cost of having a reduced recognition performance in case of similar artworks or too similar details. Adding logging capabilities is possible to obtain an anonymized information on which details attracted more the attention of the visitors to better understand their behavior and plan improvements in the visit experience.

The app can be used also to perform artwork recognition, without considering the details.

3.5 Ethical aspects

Ethical aspects of this application have been analyzed in the related Ethic Card reported in D3.2. In this section the main aspects are briefly reported.

The neural network used in the app are based on the SSD MobileNet network trained by Google. The network has been finetuned to recognize the artworks selected in the WP6 exhibition by WP6 partners, that provided all the materials, i.e. images, the selection of relevant details and guide content, so to assure the correctness and reliability of the information. The application has been tested by WP6 partners. The application has been designed to avoid the collection of any personal data, and no data is stored or shared with third parties. No tracking or profiling of the user experience has been envisaged during the design of the app nor implemented.

4 Application 3 and 4: gamification and user generated content

4.1 Purpose

The purpose of these two applications is to implement examples of gamified interactions with users. In particular, users are requested to use their body and face to replicate artworks. Successful completion of the challenges results in creation of personalized new media that can be downloaded and shared with friends and on social media. The new media is accompanied with information related to the original artworks that were replicated. The two applications are called “Strike-a-pose” (the app replicates poses using the body of the user) and “Face-fit” (the app the asks to replicate facial pose and expression). These applications use deep neural networks to implement computer vision functionalities required to match user’s body movement and face with the artwork. Computer vision and image processing techniques are also used to create the new media provided as feedback to the users.

4.2 Application description

Details on the apps have been provided in D3.3 “Toolkit Phase I” and in “D3.5: Demonstrator Mobile Applications: Intermediate Version Report”. In this paragraph we report only the updates and improvements with respect to those versions.

4.2.1 Strike-a-pose

The code of the application has been thoroughly cleaned and revisited to ease maintenance and extensibility. A Docker (https://docs.docker.com/get-started/overview/) file has been implemented to ease the installation and experimentation of the app. The changes to the code have allowed to add a new functionality to let two users compete in the challenge, allowing thus an additional type of interaction.

4.2.2 Face-fit

Similarly, to Strike-a-pose the app has been dockerized to ease its deployment. The code has also been cleaned to improve its maintenance and reuse.

Safari compatibility

The Strike-a-pose and Face-fit apps were tested on different operating systems, both desktop and mobile (iOS and Android) using different browsers during their development, testing and deployment. At the end of the project it was observed that under some combinations of O.S. (macOS / iOS) and Safari, probably due to some interactions with content blockers, the apps may not work correctly. To solve the issue a Browser Warning has been added to these applications deployed on Heroku.

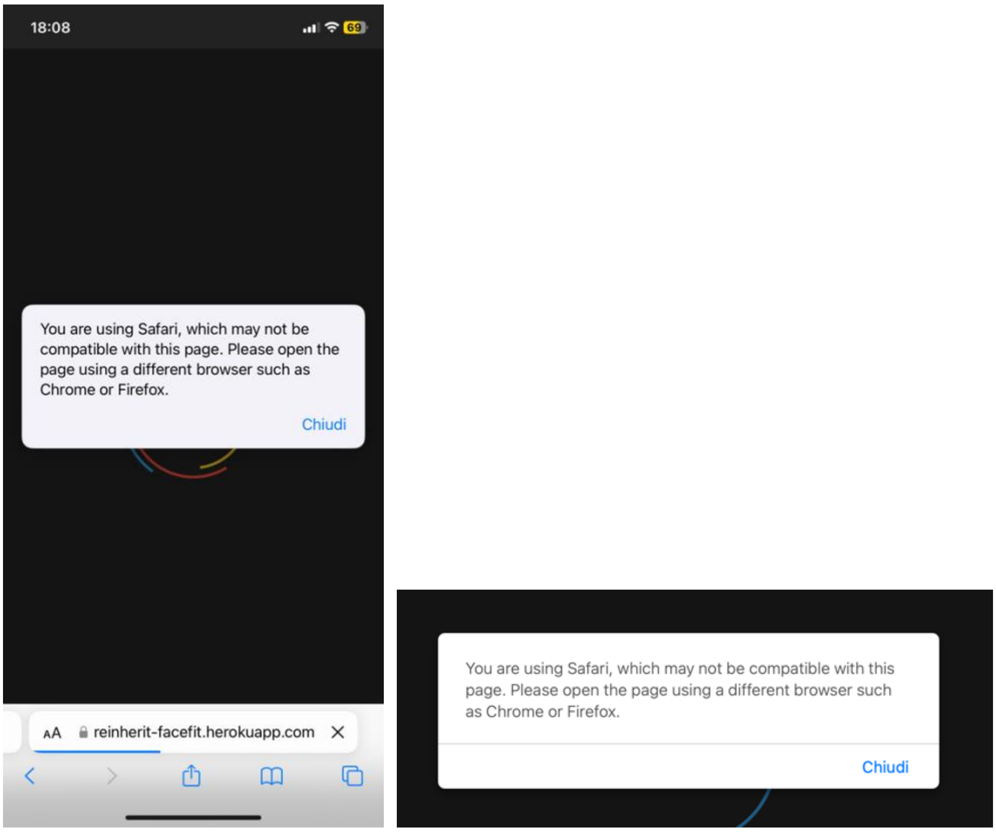


Fig. 16 – browser warnings for Safari

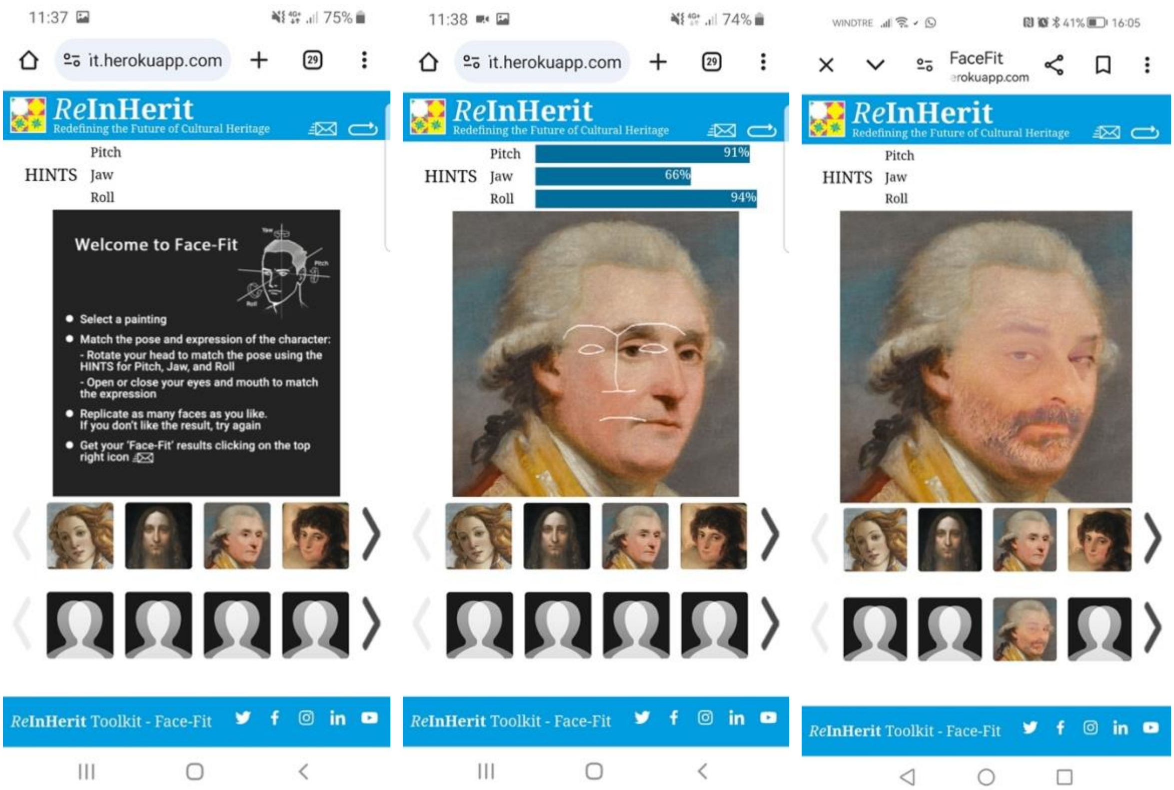


Figure 17 – examples of use of Face-fit on Chrome for Android

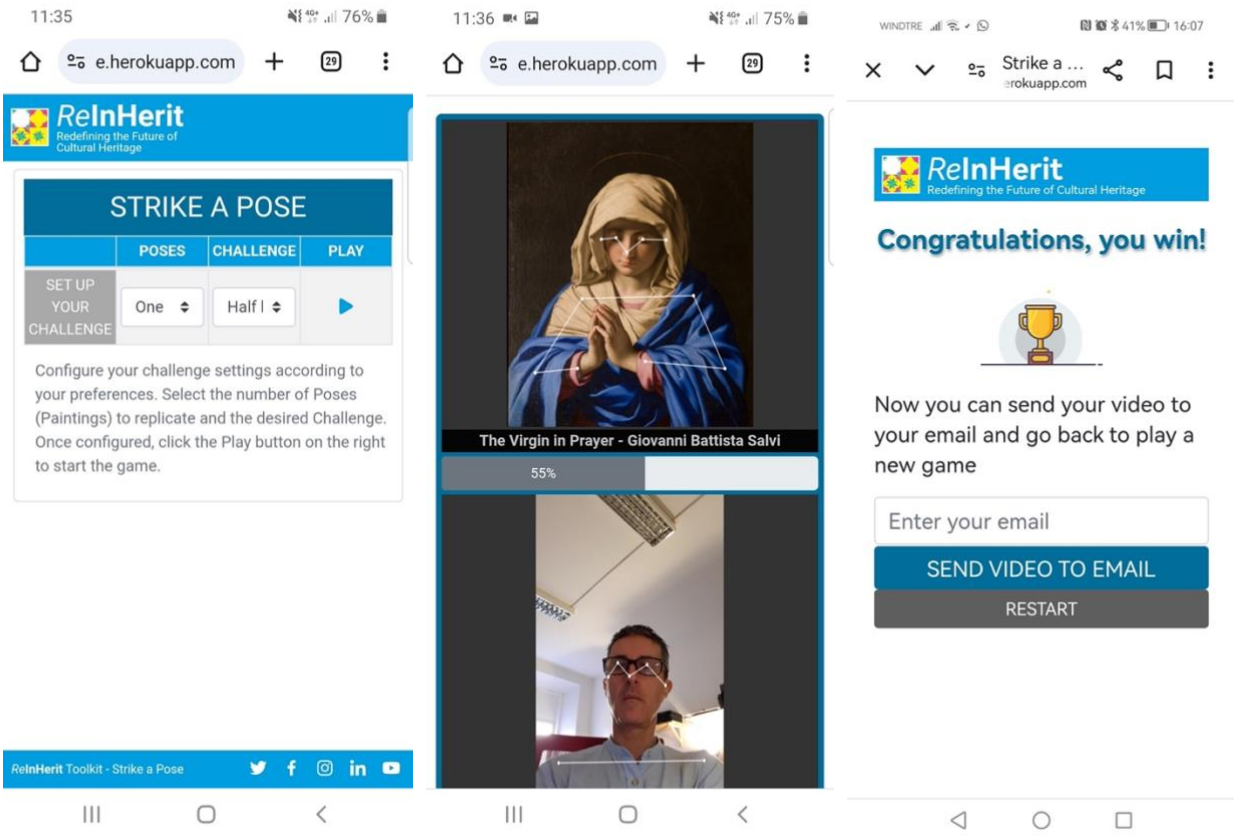
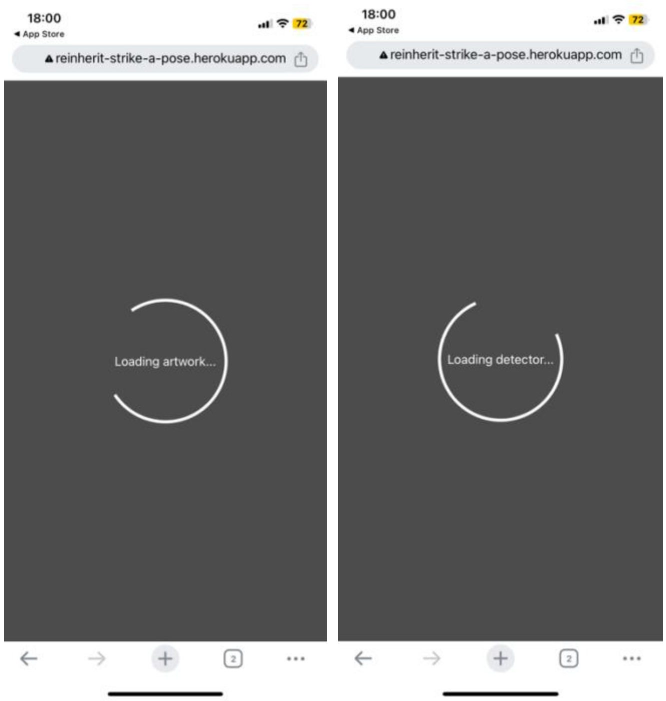


Fig. 18 – examples of use of Strike-a-pose on Chrome for Android



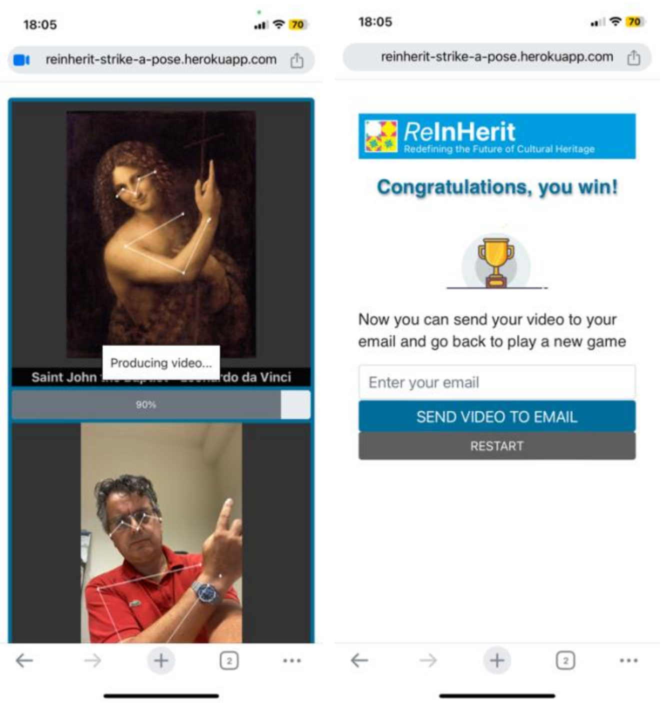


Fig.19 – examples of use of Strike-a-pose on Chrome for Apple iOS

4.3 Guidelines for reuse

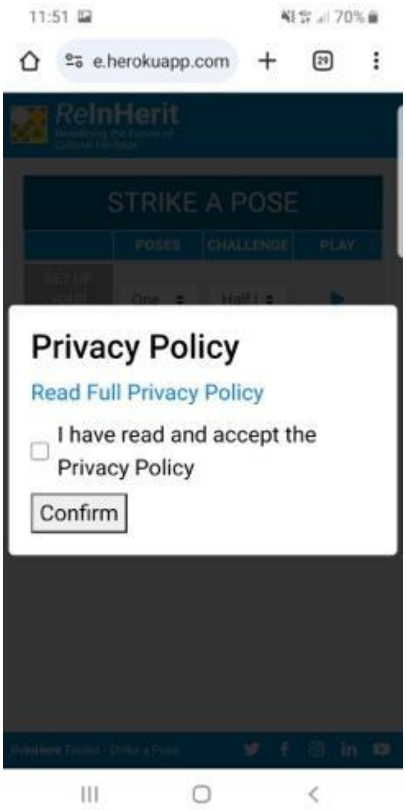
The simplest type of reuse is substituting the selected sample artworks with those of the collection of the museum/organization that desires to customize the apps, along with the associated information. Setup of the apps is based on Docker, to simplify the installation of the backend. The new competitive setup added to Strike-a-pose shows how to extend the functionality of this type of apps.

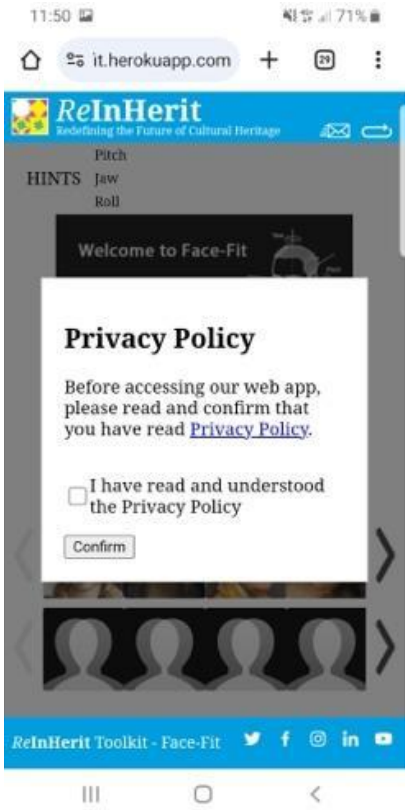
Changing the GUIs is relatively easy, since it is needed to update the HTML5 of Strike-a-pose and for the mobile version of Face-fit, or the Kivy code of desktop version of Face-fit.

4.4 Ethical aspects

Ethical aspects of this application have been analyzed in the related Ethic Card reported in D3.2; further analysis has been described in Sect. 3.4.3 in D3.3. In this section the main aspects are briefly reported.

The user experience of the users is not analyzed by these applications, nor it is tracked or used for any type of profiling. The goal of the applications is to engage the users in a playful approach, and an emotional response is something that may be part of the user experience of the application, but that is not analyzed, measured or tracked in any form. Users have to visualize and accept the privacy policies to use both Strike-a-pose and Face-fit, as shown in the following two screenshots.





The neural networks used in these applications have been trained by the authors of these models. As reported in the model cards (see D3.2) these networks have been shown to behave fairly with respect to different demographical characteristics of the users, such as gender, age, geographical origins, skin colors, etc. Possible future updates of these applications should consider these fairness aspects, in case updates or changes to the models are considered.

Great care has been taken to avoid leakage of the personal data when using the applications: no personal data is ever stored by the apps, temporary files used to generate the materials given bac to the users are stored in randomized temporary directories and automatically eliminated as soon as they are sent to the users. The dockerization of the apps guarantees further safety, since the apps and the libraries used are further isolated from the operating system of the server. The models used in the apps cannot be used, by design, to identify or track the users. No third-party service is used to collect or store data.

5 Application 5: multimedia chatbot

5.1 Purpose

A new mobile application to provide a new type of interaction in the form of a chatbot has been developed. The web app allows users to ask questions about content and context of artworks using natural language. It is possible to type the questions or ask them through a voice-to-text system. This type of interaction has become extremely popular in the latest months thanks to the inception of chat systems like ChatGPT.

5.2 Application description

The system allows to use different techniques to answer the questions provided by the users. A Visual Question Answering system composes three neural networks, one to classify the type of question, i.e. if it is related to the visual content of the artwork or + its context, and then uses two specialized neural networks to answer the question based on the classification of the question. A second modality uses GPT-based neural networks trained to answer questions.

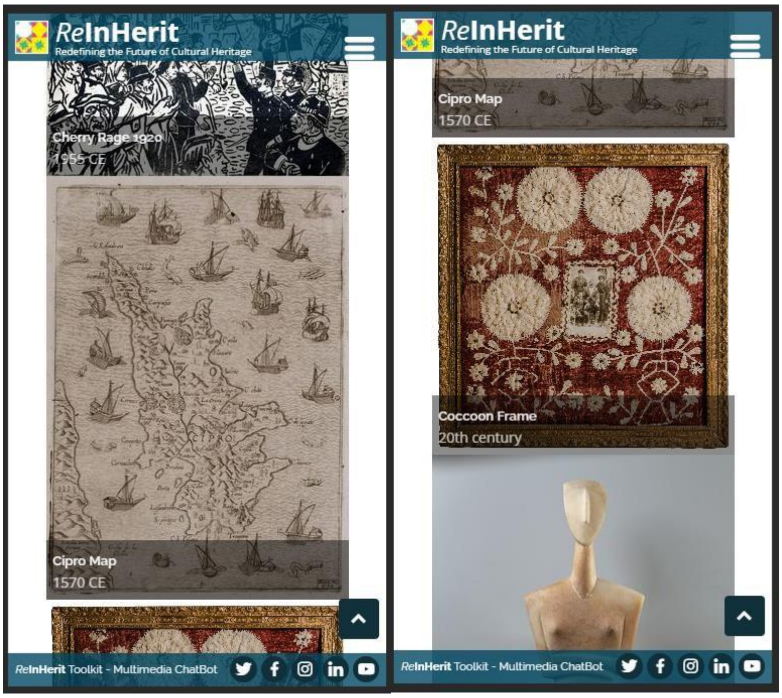


Fig. 20. Initial pages of the chatbot app.

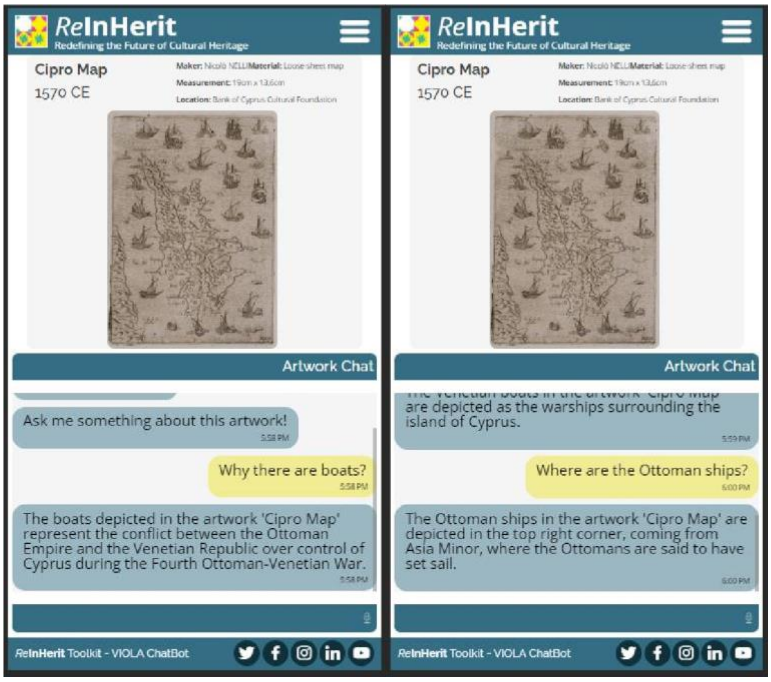


Fig. 21. Examples of chat-based interaction on a mobile device: the system can answer questions related to the visual content (e.g. general content or about details) and to the context of the artwork (e.g. where a painting was made or about artists inspired by it)

5.3 Usage example

An example of the app is shown in the previous and following figures. The app is web based and is composed of a backend system where the visual-question answering system is executed to provide the chat-based functionalities, and an HTML-based interface to implement the end user interface on the frontend.

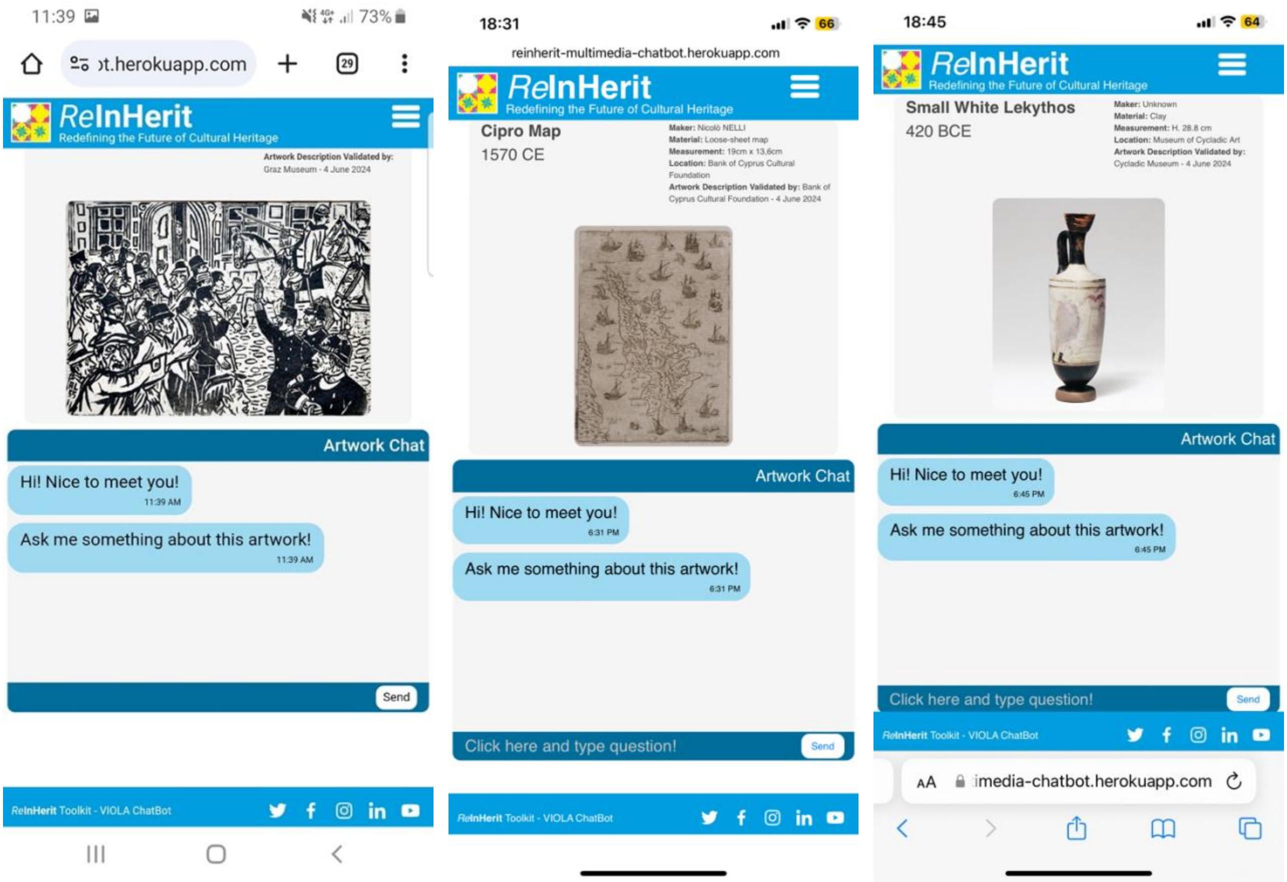


Fig. 22. Examples of chat-based interaction on: (left) Chrome on Android; (center) Chrome on Appeli iOS; (right) Safari on Apple iOS.

5.4 Guidelines for reuse

The simplest form of reuse is to change the database information composed of images of the gallery and the textual information related to the visual and contextual aspects of the artworks that are used by the neural network that provides the chat functionalities to compose its answers. A backend system allows to easily update this information, using a schema based on ICOM standard.

The system can be used also as a base of more complex systems, e.g. to implement a chat-based retrieval system, or to design serious games or implement other types of interactions based on natural language processing.

5.5 Ethical aspects

Details on the ethical aspects of this application are provided in the related Ethics Card reported in D3.2. An additional discussion is reported in Sect. 3.4.1 in D3.8 regarding data accuracy and ethical aspects. In this section these considerations are briefly reported.

The new version of the chatbot uses a selection of materials provided by WP6 partners, based on elements of the Digital Exhibition. Images and texts used by the chatbot to answer questions have been provided directly by the curators of the partner museums, instead of being based on the publicly available Computer Vision dataset (ArtPedia – used by the Computer Vision scientific community to test Visual Question Answering systems) used before. The backend of the chatbot allows to update collections used by the chatbot using an interface that allows to collect information compliant with ICOM schema.

The chatbot itself has been designed, using prompt engineering to answer using only the contextual information provided by curators, to avoid the risk of “hallucinations” typical of chatbot systems that rely only on the internal knowledge of the domain. This approach has another benefit: it does not require to fine tune the model for different artwork collections, reducing thus the deployment costs.

The latest version of the chatbot continues not to register or store any personal information, but to allow the iterative participatory design process of the knowledge base used to answer, as detailed in Sect. 3.4.2 in deliverable D3.8, it stores anonymously the questions that could not be answered. For this reason, a privacy policy is shown to the users, for their acceptance, as shown in the following screenshot.

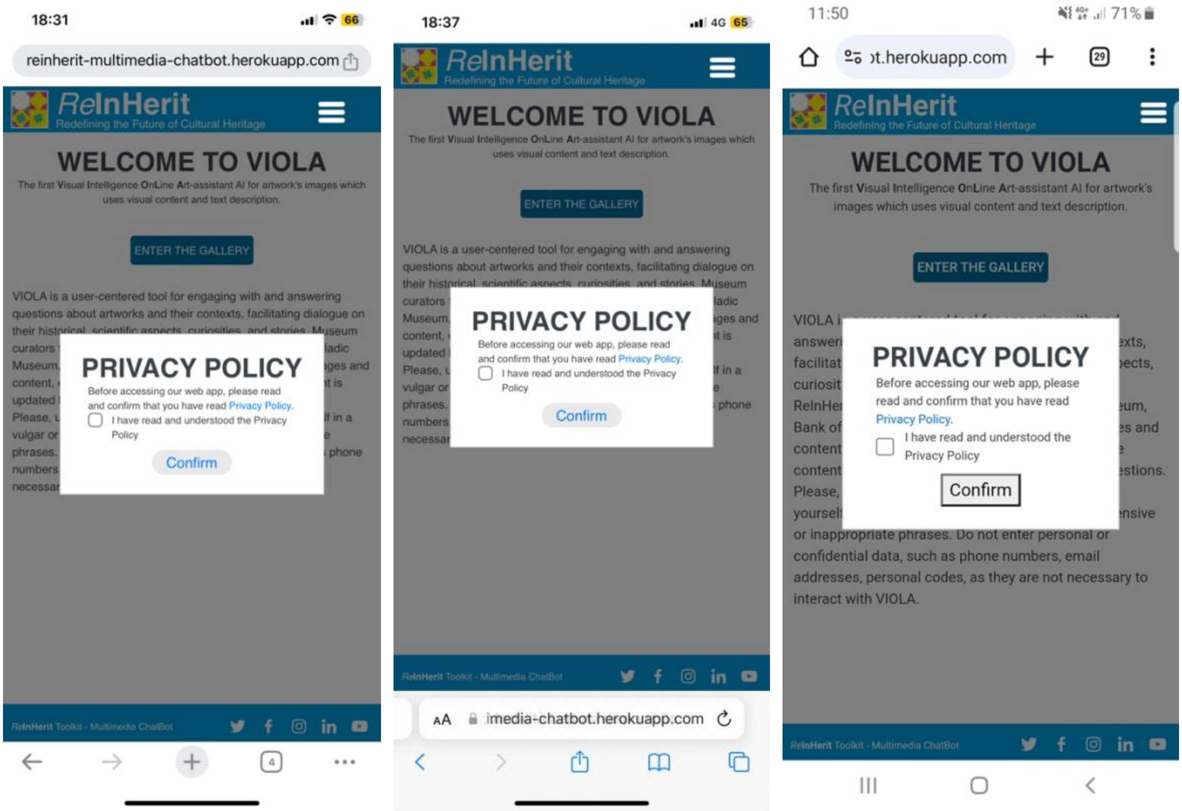


Fig. 23 – Privacy policy boxes in: (left) Apple iPhone iOS using Chrome, and Safari (center) browsers; Chrom eon Android phone (right)

6 Concluding remarks

As described above, Deliverable 3.7 provides an overview of the mobile applications developed under two main paths of the project: the immersive performance on the House of Hadjigeorgakis Kornesios on the 26th of October 2022 and the mobile applications for facilitating visitors' experiences in museums.

All applications have been tested and displayed in real environments with interesting and generally positive feedback comments from end users. Documentation, lessons learned and potential suggestions for adaptations will all be included in dedicated sections of the ReInHerit hub, with explicit focus towards professionals working in the cultural and creative industries.

The latest versions of VIOLA, Strike-a-pose and Face-fit provided a privacy policy; and have been revamped to show a similar look-and-feel. When needed, a disclaimer that asks to use a browser different from Safari is used, to deal with recent incompatibilities that were introduced in the latest versions of this browser that may cause malfunctions; this issue seems to be related to optional ad-blockers that may hinder the functionality of the apps, but as shown in the screenshots above using other browsers make it possible to run also on iPhones. We'll monitor the evolution of Safari to enable again the possibility to use that browser and enable it again as soon as it's possible.

Fig. 24. Snapshots from the immersive performance rehearsals in The House of the Hadjigeorgakis Kornesios in Nicosia, Cyprus.