

ARCHES

H2020 – 693229

Deliverable 4.2 “Report on system integration – 1st version”

Lead Author: Ana Belén Rodríguez Arias

With contributions from: Christoph Bitzner, Nicola Milinkovic

Reviewers: Vladimir Kopric

Deliverable nature:	R: Report
Dissemination level: (Confidentiality)	PU: Public
Contractual delivery date:	09-2017
Actual delivery date:	09-2017
Version:	1.0
Total number of pages:	47
Keywords:	Mobile applications, iOS, Android, avatars, games.

Abstract

The present deliverable contains all the advances related to the design and development of the mobile applications in ARCHES. The design and functionalities arise from the needs and suggestions of the exploration groups and the requirements detected in deliverable D4.1 “Report on system requirements for handles devices” [3].

The adopted user-centred and iterative development strategy allows the continuous modification, enhancement and adaptation of the software based on the feedback provided by people with differences and difficulties associated with perception, memory, cognition and communication, being crucial to obtain more appropriate and effective results.

Document information

IST Project Number	H2020 - 693229	Acronym	ARCHES
Full Title	Accessible Resources for Cultural Heritage EcoSystems		
Project URL	http://arches-project.eu		
Document URL	http://arches-project.eu		
EU Project Officer	Luis García Domínguez		

Deliverable	Number	D4.2	Title	Report on system integration – 1 st version	
Work Package	Number	WP4	Title	Development of applications for handheld devices	
Date of Delivery	Contractual	M12		Actual	M12
Status	Version 1.0			final ☒	
Nature	Prototype ☐ Report ☐ Demonstrator ☐ Other ☐ Ethics ☐ ORDP ☐				
Dissemination level	Public ☒ Confidential ☐				

Authors (Partner)	Treelogic (TREE)			
Responsible Author	Name	Ana Belén Rodríguez Arias	E-mail	ana.rodriguez@treeologic.com
	Partner	TREE	Phone	+34 985 966 136

Abstract (for dissemination)	<p>The present deliverable contains all the advances related to the design and development of the mobile applications in ARCHES. The design and functionalities arise from the needs and suggestions of the exploration groups and the requirements detected in deliverable D4.1 "Report on system requirements for handles devices" [2].</p> <p>The adopted user-centred and iterative development strategy allows the continuous modification, enhancement and adaptation of the software based on the feedback provided by people with differences and difficulties associated with perception, memory, cognition and communication, being crucial to obtain more appropriate and effective results.</p>
Keywords	Mobile applications, iOS, Android, avatars, games.

Version Log			
Issue Date	Rev. No.	Author	Change
29-08-2017	0.1	Ana Belén Rodríguez Arias	Table of contents and sections
08-09-2017	0.2	Ana Belén Rodríguez Arias	Initial content
11-09-2017	0.2	Ana Belén Rodríguez Arias	Submitted to technical partners
25-09-2017	0.3	Christoph Bitzner	Integration of avatar-related descriptions
26-09-2017	0.4	Vladimir Kopric	New contributions and revision
26-09-2017	0.5	Ana Belén Rodríguez Arias	Integration of contributions

28-09-2017	0.6	Nikola Milinkovic	Minor changes
30-09-2017	1.0	Javier Gutiérrez Meana	Corrections and final version

Table of contents

Document information	3
Table of contents	5
List of figures	7
List of tables	9
Abbreviations	10
1 Introduction	11
2 OurStory	12
2.1 App schema	13
2.2 Platforms	14
2.3 User visual interface	14
2.3.1 Home page	15
2.3.2 New story	16
2.3.3 Edit images in the story	17
2.3.4 Story list	18
2.4 Application interface – current development state	19
2.4.1 Home	19
2.4.2 New story	20
2.4.3 Edit images in a story	22
2.4.4 Show story	23
2.4.5 Send story	24
2.4.6 Story list	25
3 Unity3D ARCHES test application	27
3.1 App schema	28
3.2 Platforms	28
3.3 Application interface – current development state	28
3.3.1 Main screen	28
3.3.2 Text recognition	29
3.3.3 Camera	30
3.3.4 Images	33
3.3.5 Multimedia resources	33
3.3.6 Recognition of paintings	35

3.3.7	QR codes	35
3.3.8	Text to speech.....	36
4	Museum application design	38
4.1	Home screen	38
4.2	Tours	39
4.3	Steps to find the artwork	40
4.4	Camera screen	41
4.5	Avatar component	44
5	Conclusions	46
	References	47

List of figures

Figure 1: <i>OurStory</i> logo.....	12
Figure 2: Ionic v2 schema.	13
Figure 3: Initial designs for the home screen.	15
Figure 4: Final design for the home screen.	16
Figure 5: Screen ‘New story’ (mock-up).	17
Figure 6: Screen ‘Edit story’ (mock-up).	18
Figure 7: Screen ‘List stories’ (mock-up).	19
Figure 8: Screen ‘Home’.	20
Figure 9: Screen ‘New story’: (a) Headings; (b) Camera roll; and (c) Online Drive.	21
Figure 10: Screen ‘Edit’.	22
Figure 11: Edit options: (a) Sound recorder; (b) New text; and (c) Drawing.	23
Figure 12: Screen ‘Show story’.	24
Figure 13: Screen ‘Send’.	25
Figure 14: Screen ‘Story list’.	26
Figure 15: Screen ‘Main page’.	29
Figure 16: Screen ‘Text recognition’.....	30
Figure 17: Screen ‘Camera’.....	31
Figure 18: Example of filters I: (a) Colour blindness material and (b) Grey material.	32
Figure 19: Example of filters II: (a) Invert material and (b) tint filter.	32
Figure 20: Static image example.....	34
Figure 21: Screen ‘Multimedia resources’.....	34
Figure 22: Screen ‘Painting recognition’.....	35
Figure 23 : Screen ‘QR’: (a) Preparation and (b) Recognition.	36
Figure 24: Screen ‘Text to speech’.....	37
Figure 25: Screen ‘Intro’.	39
Figure 26: Screen ‘Tour’.....	40
Figure 27: Screen ‘Steps’.	41
Figure 28: Screen ‘Camera’.....	42
Figure 29: Screen ‘Recognition’.....	43
Figure 30: Screen ‘Description’.....	44

List of tables

Table 1: Worldwide smartphone sales to end users by operating system in 4Q-2016 and 4Q-2015 (thousands of units) [8].	14
Table 2: Functionalities identified by the exploration groups in the UK.	27

Abbreviations

API: Application Programming Interface

AR: Augmented Reality

ARCHES: Accessible Resources for Cultural Heritage EcoSystems

Dx.x: Deliverable x.x

GUI: Graphical User Interface

IDE: Integrated Development Environment

OOP: Oriented Object Programming

OS: Operating System

OU: The Open University

TREE: Treelogic Telemática y Lógica Racional para la Empresa Europea S.L.

UI: User Interface

V&A: Victoria & Albert Museum

WC: The Wallace Collection

WP: Work Package

1 Introduction

This deliverable presents the advances in the development and design of the applications for handheld devices within the ARCHES project. The document follows the facts and recommendations already presented in deliverable D4.1 “Report on system requirements for handheld devices” [3].

The methodology selected for the design and implementation is based on the development of mock-ups and prototypes that can be tested by the target audiences: ‘showing is always better than telling’. This way, if people are allowed to directly interact with the ideas and functionalities from the very beginning, then it is easier for them to understand the scope and capabilities and, therefore, to propose modifications and improvements. Likewise, this strategy facilitates the communication and collaboration between a multidisciplinary team and with members of the exploration groups in United Kingdom, Spain and Austria. Developers and users are invited to play with a prototype and explore its limitations and possibilities all along the pilot exercises, so that any problem that may arise can be fixed earlier in the process.

The deliverable is divided into several sections. Each section is dedicated to each of the applications envisioned within ARCHES:

- **OurStory:** This application can be described as an accessible virtual research diary that can be used by anyone in any environment to record and present their experiences (especially when visiting a museum). In addition, it can serve as an inexpensive tool for creating simple, accessible tours within museum spaces and share them with other people.
- **Unity ARCHES test application:** This is a prototype designed and created to validate the functionalities previously identified by the exploration groups at both the Victoria and Albert Museum (V&A) and The Wallace Collection (WC). Some of them are: zoom, text to speech, QR codes, image processing (e.g. filters), etc.
- **Museum application Design:** The ARCHES app has been conceived to explore the museums and their collections in an easy manner using a smartphone or a tablet. It proposes different tours where the functionalities in the Unity ARCHES test application are fully exploited. The use of Augmented Reality (AR) allows the user to interact with works of art and discover contents specifically adapted to her profile.

2 OurStory

The *OurStory* app was brought into the project as a form of capturing, collecting and sharing the participant’s experiences. The application was originally created to enable parents to generate stories based around the interests of their young children so as to encourage their involvement in literacy activities. Subsequently, an early version of the application has been used with a range of different users, in a range of contexts and across the ages, in order to capture tales people may wish to tell. The original application has been effective in one operating system (iOS). However, it has limited functionalities and has a design that is not suited to the range of users associated with the ARCHES project.

The intention in redeveloping the *OurStory* application is twofold. Firstly, it is to serve as an accessible virtual research diary that can be used by anyone in any environment to record and present their museum experiences. Secondly, it can serve as an inexpensive tool for creating simple, accessible tours within museum spaces. In addition to these two functions, the redevelopment of the *OurStory* application will make it available to a far wider audience beyond museums who will be able to adapt it to their own particular needs.

Within the ARCHES exploration group, *OurStory* (see the logo in Figure 1) is used by participants to store their experience but also to plan out a project like the making of a video. Particularly for people with memory difficulties, storytelling is an important social and literacy skill which is being encouraged here with the storyboard on the app. The process of creating a story itself can teach participants valuable lessons such as helping each other during the making process, exploring an museum and/or object in depth, structuring their thoughts, facing and overcoming challenges within the museums environment and the value of sharing. The process encourages the participants to be creative and imaginative.

The stories itself are a rich source of qualitative data that will inform the project on the development of ideas, relationships with the museums and the group as well as the process of being a researcher and many more.



Figure 1: *OurStory* logo.

2.1 App schema

From a technical perspective, one of the key discussion points when deciding how to address the challenge of redesigning and redeveloping *OurStory* was to select the appropriate technologies. Based on previous experiences and the expertise in this area, the team decided to opt for Ionic v2 [6], a framework based on Angular v2 [2] for the front-end and on Cordova [4] for the back-end (see Figure 2).

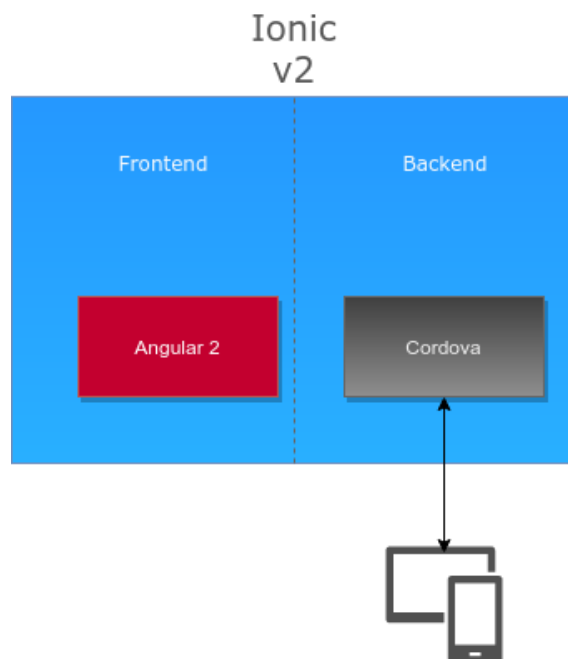


Figure 2: Ionic v2 schema.

The main technical reasons to use Ionic v2 in ARCHES include:

- **Cross-platform:** Ionic allows developers to deploy the same code in different platforms. Consequently, it is not necessary to be an expert in the different platform specific languages. In addition, development for a new platform it is always easier once the application is running in other platforms.
- **Common programming language:** The front-end of the Ionic App is based on Angular v2. Angular is a well-known JavaScript framework that is used to make easier web developments.

Particularly, the base programming language has changed from JavaScript to Typescript in the second version of Angular. It changes the structure to develop applications, but the programming paradigm is similar to oriented object programming (OOP). This modification was deemed to be positive in the context of ARCHES.

- **Easy access to device features:** As a back-end framework, Ionic uses Apache Cordova. Using Apache Cordova is straightforward to access to all the features that the device includes. In this way, it makes possible to access to the camera, file managing or database functionalities among others.

Apache Cordova depends on the device to run the different functionalities, so there may be some difficulties to run a specific functionality if the device is not ready for it. For example, it cannot run the camera functionality if the device has not a camera.

2.2 Platforms

Being a cross-platform application, *OurStory* is expected to easily reach a wider audience, especially in countries where iOS is not as popular as in the United Kingdom. At the same time, the possibility of releasing upgrades with new functionalities or designs will be open in the future.

Table 1: Worldwide smartphone sales to end users by operating system in 4Q-2016 and 4Q-2015 (thousands of units) [8].

Operating System	4Q16 Units	4Q16 Market share (%)	4Q15 Units	4Q15 Market share (%)
Android	352,669.9	81.7	325,394.4	80.7
iOS	77,038.9	17.9	71,525.9	17.7
Windows	1,092.2	0.3	4,395.0	1.1
BlackBerry	207.9	0.0	906.9	0.2
Other	530.4	0.1	887.3	0.2
Total	431,539.3	100.0	403,109.4	100.0

Two of the most used operating systems (OSs) for mobile phones are Android and iOS. Indeed their use is still increasing as shown in Table 1. This is the main motivation why these two have been selected for the development of *OurStory*:

- **Android** [1]: The app was tested successfully on mobile phones and tablets with Android version 5.0.0 or above (API level 21). Mobile phones and tablets with earlier versions of Android were not tested and the correct running with these platforms is not guaranteed. Android versions 5.0.0 and higher covers the 75% of the current Android market.
- **iOS** [7]: *OurStory* was tested successfully in iOS versions 8.0.0 and 10.3.0. Therefore, it covers at least the 97% of current iOS market. Earlier versions to 8.0.0 could not be supported.

Although the initial results are promising, the team will continue working on fine-tuning the application and fix any bugs and errors detected by the users.

2.3 User visual interface

This section shows the first design of the graphical interface of *OurStory*. Even though the ARCHES version has been designed based on the original application – developed by child psychologist and other specialists at The Open University (OU) – partners considered that significant differences should be introduced since the target audience (adults with differences and difficulties associated with perception, memory, cognition and communication) is no longer children.

The main objective of these designs was to validate the user interfaces with end-users in order to get the optimal user interface (UI) according to their needs.

2.3.1 Home page

This is the main screen of the application. From this screen the users can access to all the options in the app.

The design of both the home page and logo is more important than we often think since it:

- Establishes an identity.
- Makes a good first impression (or not).
- Provides visual and verbal clues to reveal the site’s underlying content and encourages deeper browsing.
- Provides a clear navigation with several entry points to diverse functionalities.

TREE worked together with The Open University, the exploration group in the UK and Ms Natalia Kucirkova (member of the External Expert Advisory Board and one of the creators of the original *OurStory* app) to create the main screen. As shown in Figure 3, multiple options were explored: different colours, different scenarios, different people, different resources, etc.



Figure 3: Initial designs for the home screen.

After several meetings and several initial designs, the final design was agreed. It tries to reflect diversity but not within a crowded space. It also highlights the importance of a collaborative environment and the crucial role of technology to support users. In this screen (Figure 4) there are three options:

- **‘Get started’**: This button gives access to the screen ‘Help’, which will contain the basic information to guide the user.
- **‘Create a new story’**: It allows the user to access to the screen ‘New Story’. It will contain the functionalities and widgets to create a new story and save it in different formats.
- **‘Use existing story’**: Access to the screen ‘Story List’, where the user can edit, share, delete, etc. existing stories.

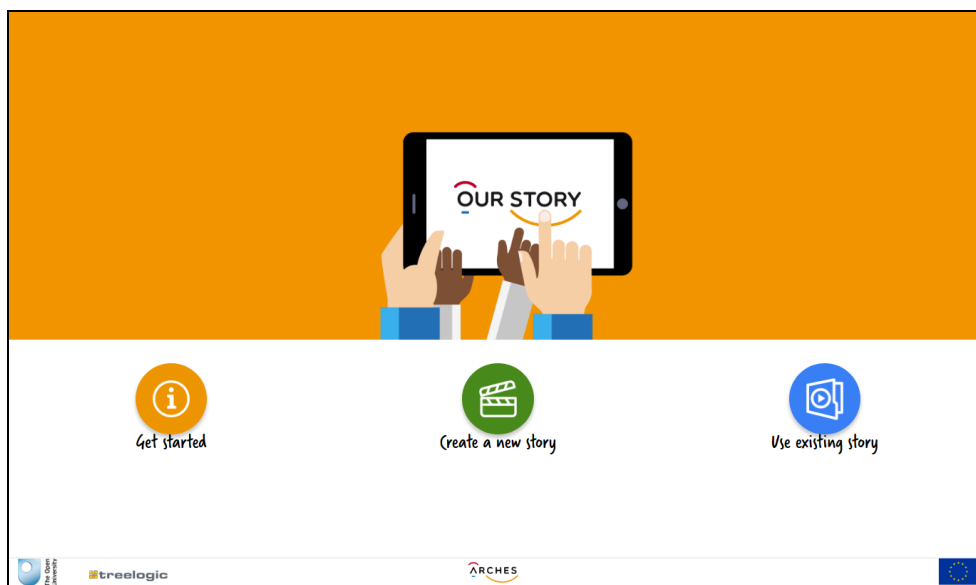


Figure 4: Final design for the home screen.

2.3.2 New story

When clicking on ‘Create a new story’, the user is redirected to this screen where new stories are created (see Figure 5). On top, there is bar consisting of the following elements:

- An input field to insert the title of the story
- Two buttons on the left: back button and help button. The latter allows the user to go to the application help system.
- Three buttons on the right:
 - **‘Export’**: Options such as send by email or Whatsapp, share in social networks and others are available.
 - **‘Print’**: The story can be printed in different formats. For this a PDF will be generated in advance.
 - **‘Save’**: The story is saved in the device. If the user has not introduced a title, this will be requested before proceeding.

Below the top bar, there are a couple of buttons on the right. The first one, ‘New video’ is intended to enable recording videos with the mobile device, while the second one, ‘New photo’ is for taking photos. In both cases, the generated resource is displayed on the gallery.

The central area of the screen presents three tabs:

- **‘Headings’**: The user can select the colour of a heading and insert this into the story. Multiple headings are possible (for example when a story is divided into different sections).
- **‘Camera roll’**: The photographs stored in a predefined folder are displayed so the user can select them for the story.
- **‘Dropbox’**: Photos and videos can be imported from the Dropbox account. To this end, the user has to log in.

At the bottom of the screen, the photos and videos would be displayed. The bottom on the left will initiate the presentation mode.

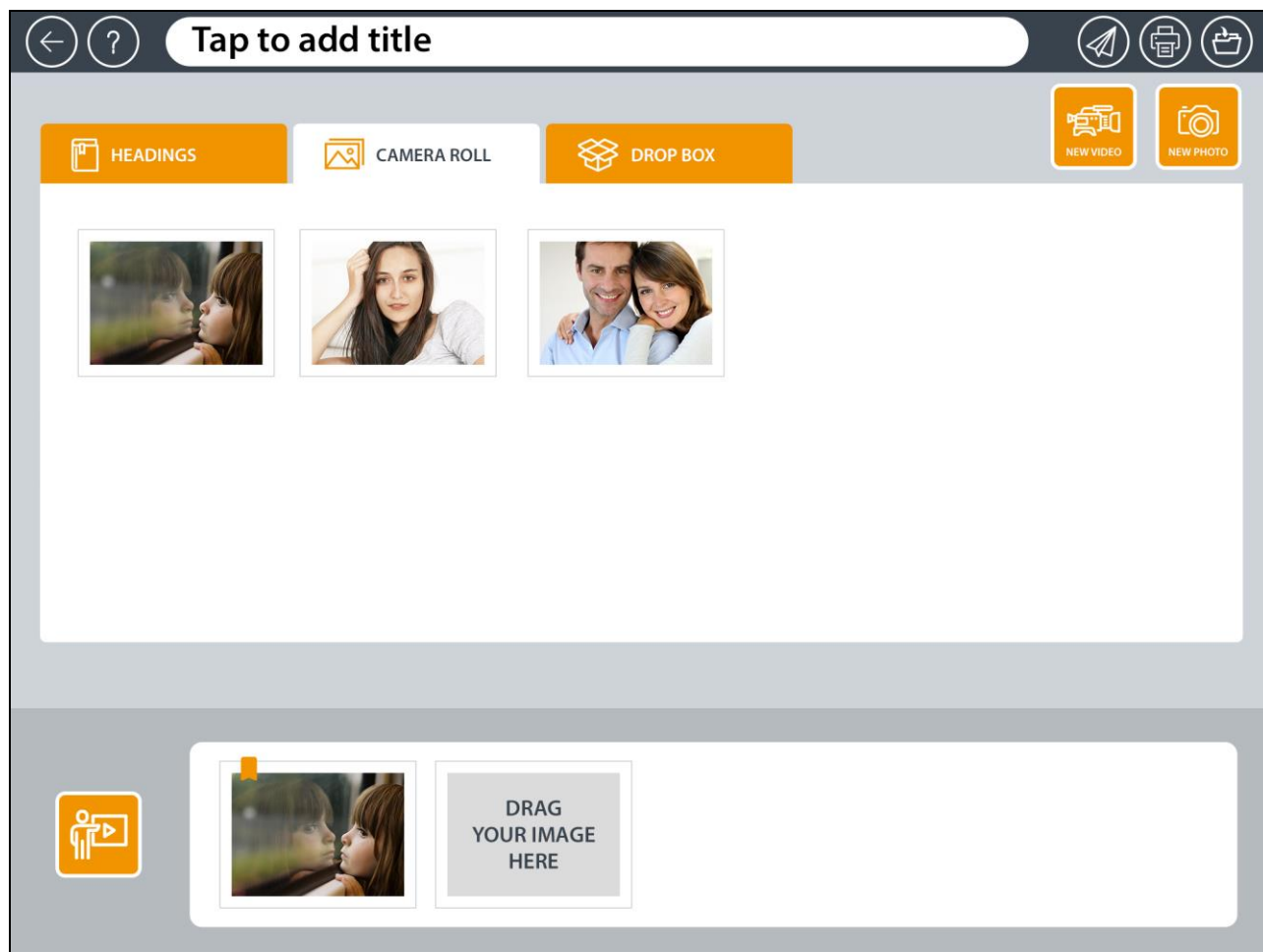


Figure 5: Screen ‘New story’ (mock-up).

2.3.3 Edit images in the story

Whenever a photo is selected, this can be edited. Likewise, each of the images in the story can be modified to personalise the content with text and audio. These functionalities were directly taken from the original *OurStory* app.

On top of the screen (see Figure 6), the header displays a couple of buttons to (from left to right) go back to the previous screen and get information about the functionalities available on this screen.

The image is presented in the central area and three possibilities are available:

- **‘Make this heading’**: The image is processed as a heading in the slideshow.
- **‘Microphone’**: It allows the user to add an audio recording to the current image.
- **‘Keyboard’**: It allows the user to introduce a text that will be placed at the bottom of the image.

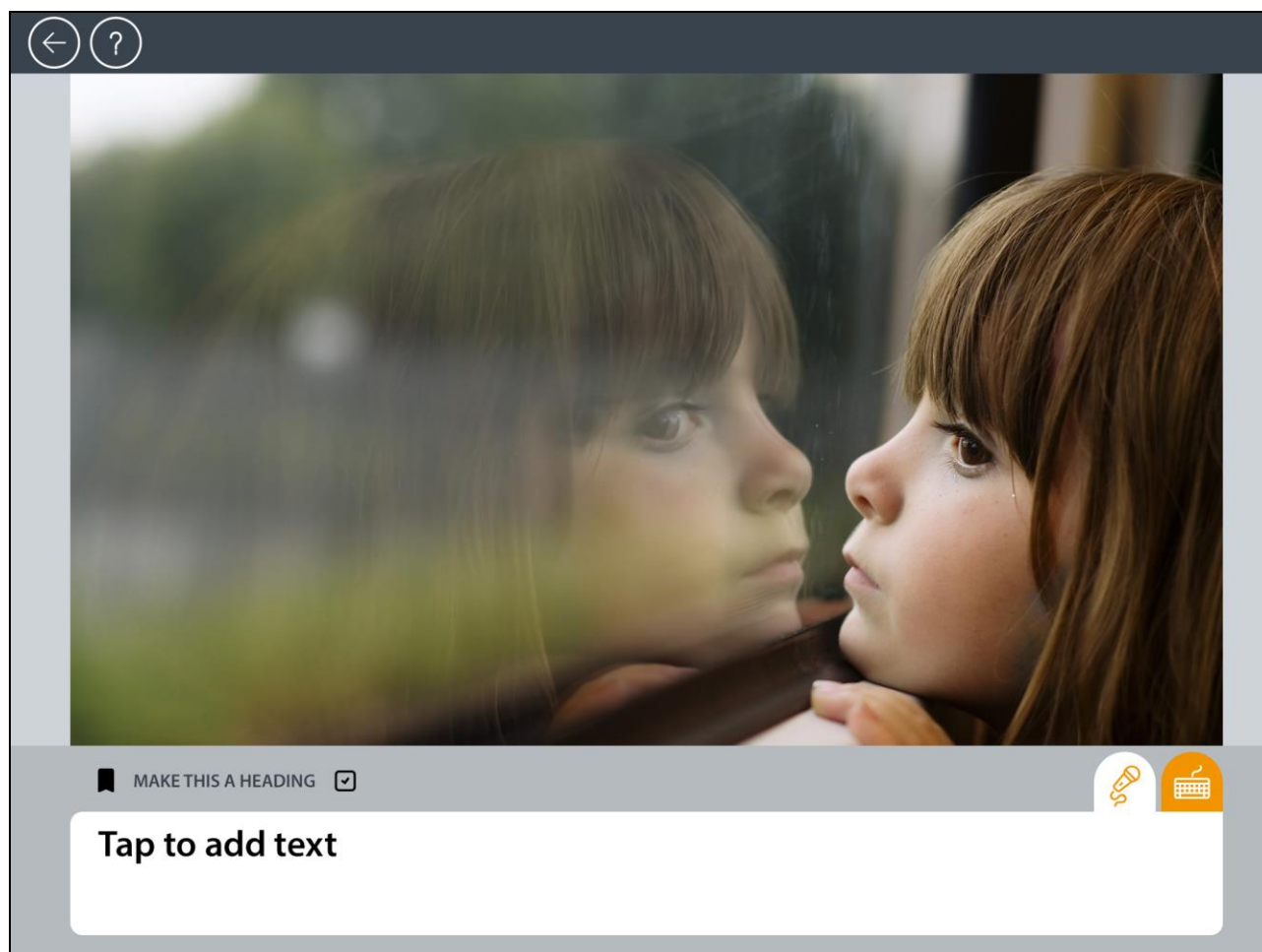


Figure 6: Screen ‘Edit story’ (mock-up).

2.3.4 Story list

On this screen the user can access at list of stories that were previously saved. For each story the user can play, edit, print, share or delete the story as presented in Figure 7. When clicking on these icons, the user is redirected to the corresponding screen.

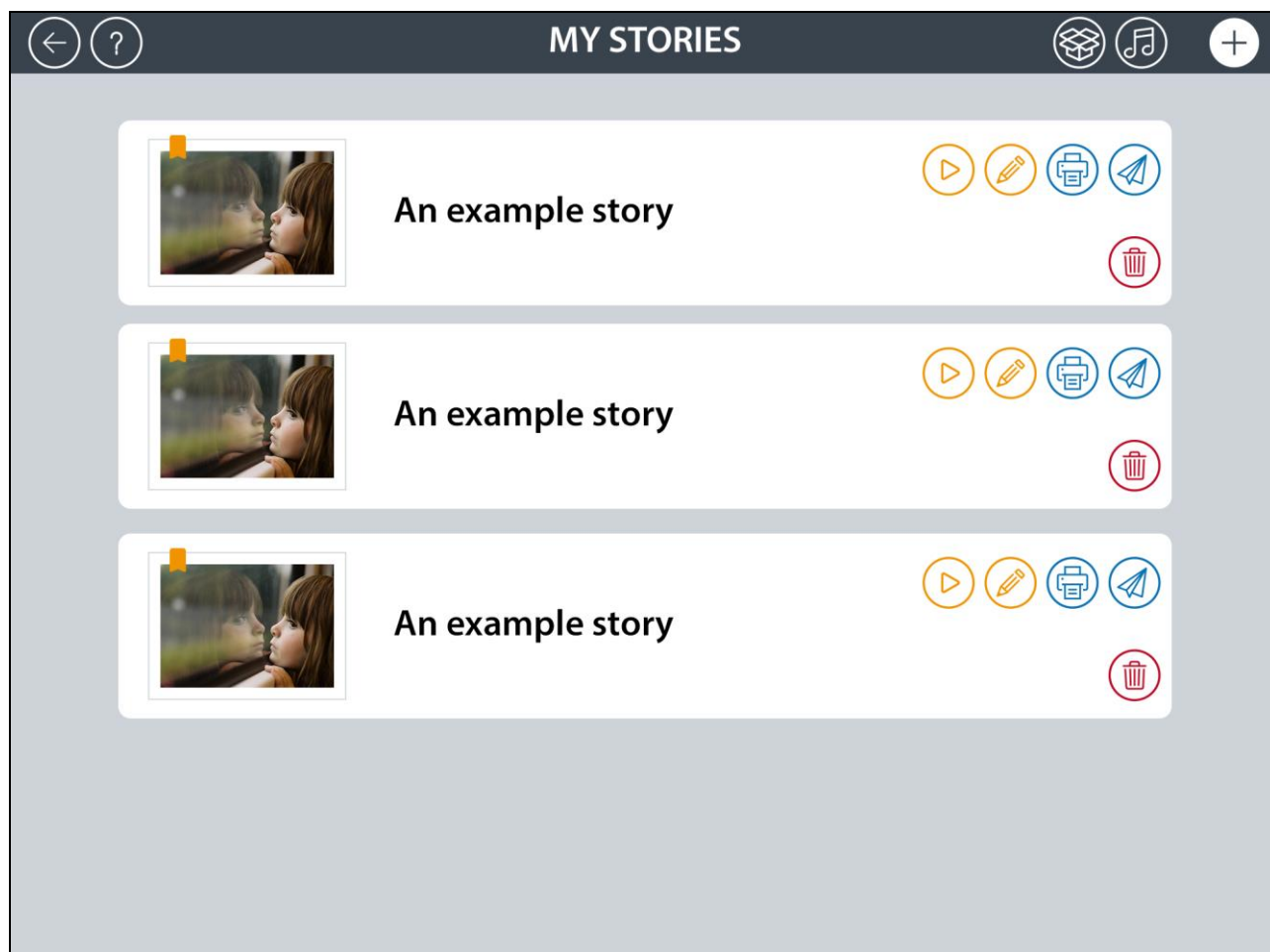


Figure 7: Screen ‘List stories’ (mock-up).

2.4 Application interface – current development state

After describing the mock-ups prepared to share the proposed design with the partners within the consortium (especially the museums) and with the exploration group in the UK, this section focuses on the current development state of the application. Taking advantage of the feedback provided by the aforementioned relevant actors, new functionalities have been detected and added to the initial design. Likewise, other options have been adapted accordingly.

Android and iOS versions will not differ significantly, at least in regards to the graphical interface and options. Nevertheless, some of the functionalities available in Android may not be available in iOS and the other way around. For example, access to the file system in Android to save/load content has nothing to do with the iOS approach.

2.4.1 Home

As pointed out before, this is the main screen of the *OurStory* app. The icons and texts are bigger than in the associated mock-up to facilitate the interaction of the target audiences and non-tech-savvy users (see Figure 8).

From here the user has three options that are linked to the three buttons in yellow, green and blue (the colours have not been changed to go with the ARCHES logo – yellow, blue and red – because they were validated by OU during the design process):

- **‘Get started’**: A manual where all the options and functionalities of the app are explained. This will be available once the exploration groups have validated the application and developers can be sure that the design will not be modified. Exploration groups will contribute to the design of the manual and how this is presented to the users.
- **‘Create new story’**: This is the core functionality of the application and it allows the user to create her own stories using the camera, multimedia content, etc.
- **‘Use existing story’**: Users can save and load/recover their stories.

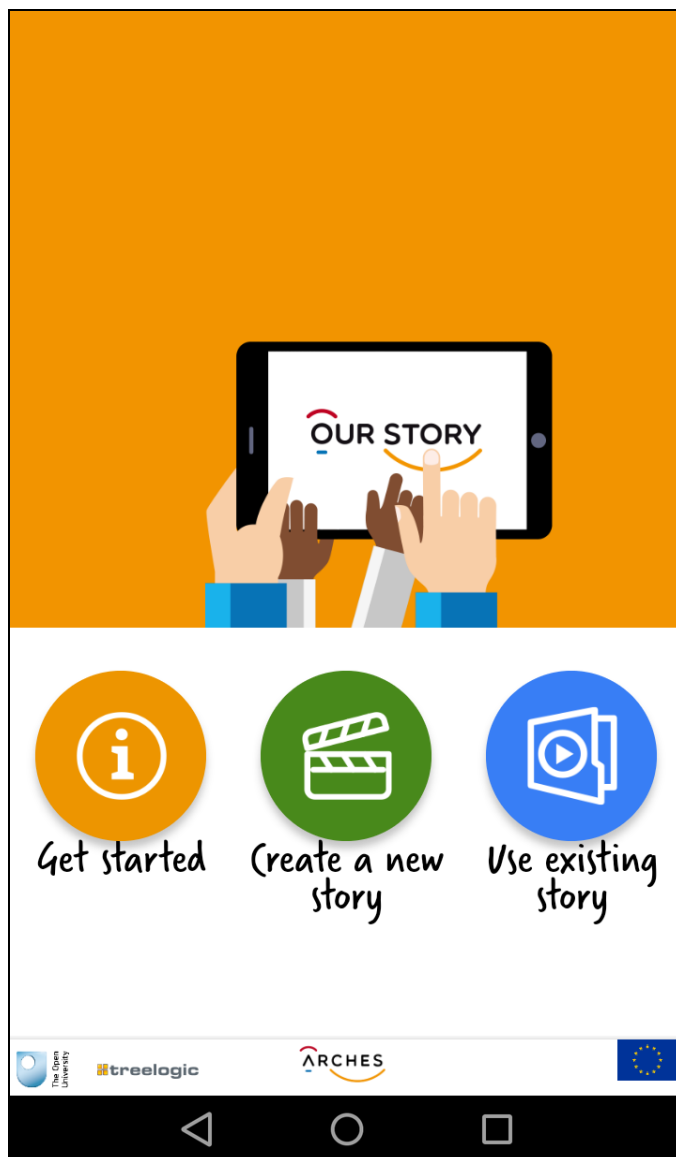


Figure 8: Screen ‘Home’.

2.4.2 New story

Once again, the icons and texts are bigger than in the mock-ups.

Under the header, we have decided to keep the three tabs. These can be used to add different items to the story (see Figure 8):

- **‘Headings’**: This tab (see Figure 9a) allows the user to insert a coloured slide (the colours have been agreed with the experts) in the middle of two other images or at the beginning/end of the story.
- **‘Camera roll’**: This tab (see Figure 9b) is used to import images and videos. There are three options:
 - **‘Camera’**: This icon enables the user to switch on the camera to take a photo.
 - **‘Video’**: This icon allows the user to record a video (the duration may be limited to some seconds or a minute to avoid the size of the story to grow exponentially).
 - **‘More’**: Access to the device gallery to get a photo or video already stored in the memory.
- **‘Online drive’**: This tab (see Figure 9c) is intended to retrieve multimedia content from online storage platforms:
 - **Dropbox**: Visualisation of the Dropbox folder to select images.
 - **Google Drive**: This option was suggested by the UK exploration group since they are using this platform for other similar purposes during their weekly sessions.

Downloading the images from any online drive platform will require logging in first.

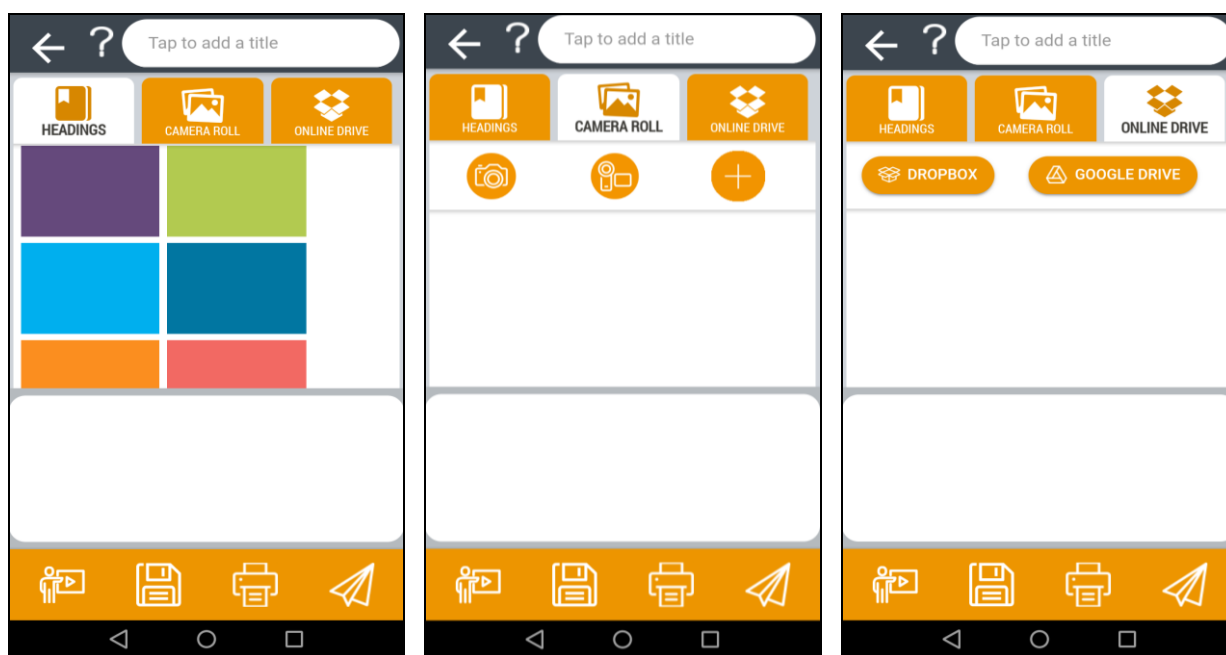


Figure 9: Screen ‘New story’: (a) Headings; (b) Camera roll; and (c) Online Drive.

All the contents which are selected in three tabs will be loaded in the area just below the three tab buttons. To confirm that an item should be added to the story, the user has click on the desired item, drag it and drop it in the white area at the bottom.

When an image is dropped in this white area, the user can click on it to open the screen ‘Edit’ – see next section.

If the user clicks and holds on one of the items in the story, a ‘cross’ icon will appear on the image, making it possible to remove the selected item from the story line.

The bar at the bottom displays four different buttons related to different functionalities concerning the current story:

- **'Slideshow'**: It activates the presentation mode. See section 2.4.4.
- **'Save'**: The current story is saved and will be displayed in the screen 'Story list'.
- **'Print'**: A PDF is generated and sent to the printer. Different configuration options are available.
- **'Share'**: It redirects the user to the screen 'Send'. See section 2.4.5.

2.4.3 Edit images in a story

This screen (see Figure 10) is intended to edit images that are part of a story. As usual, there are two buttons on the upper left corner ('Back' and 'Help'), but also a third button ('Accept') on the right :

- **'Back'**: It goes back to the screen 'New story' discarding all changes.
- **'Help'**: It leads to the help section.
- **'Accept'**: When clicking on it, the changes are saved and updated in the story line.

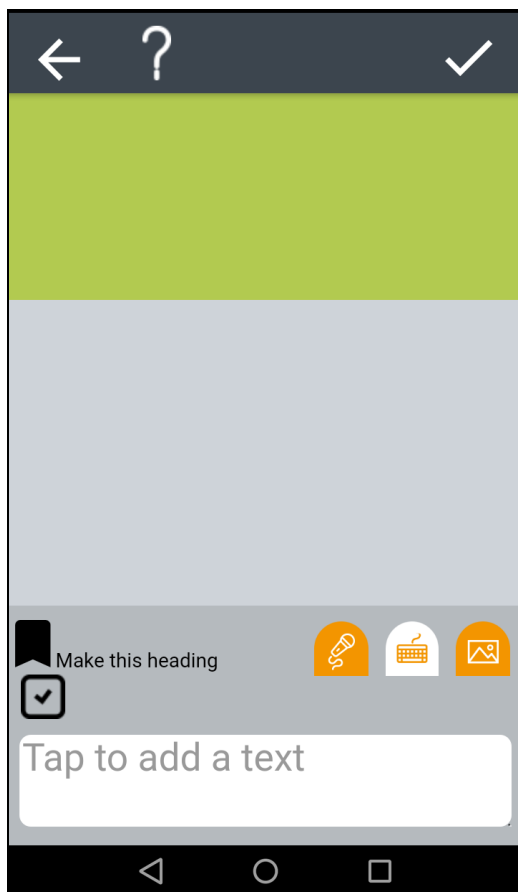


Figure 10: Screen 'Edit'.

The area at the bottom of the screen is reserved for the different functionalities:

- **'Make this heading'**: This checkbox enables the user to change a heading so their properties are modified, and the other way around (an image can be treated as a heading). When an image is presented as a heading, the position and font of the text inserted in the field 'Tap to add a text' will be different.

- **'Add audio'**: When clicking on the microphone icon, the user can record an audio file that will be played when the image is presented in the slideshow. For this, the *OurStory* app has to get access to an audio recorder (see Figure 11a).
- **'Add text'**: When clicking on the keyboard icon next to the microphone, the user can introduce a text (typically one or two lines) that will be added to the image (at the bottom) during the slideshow. This is shown in Figure 11b.
- **'Add drawing'**: The last button on the right will hide the text box and show a colour palette (yellow, green, blue, red, magenta, etc.). Users can select the colour and paint on the image with a pencil by swiping the finger over it (see Figure 11c). The traces can be deleted too.

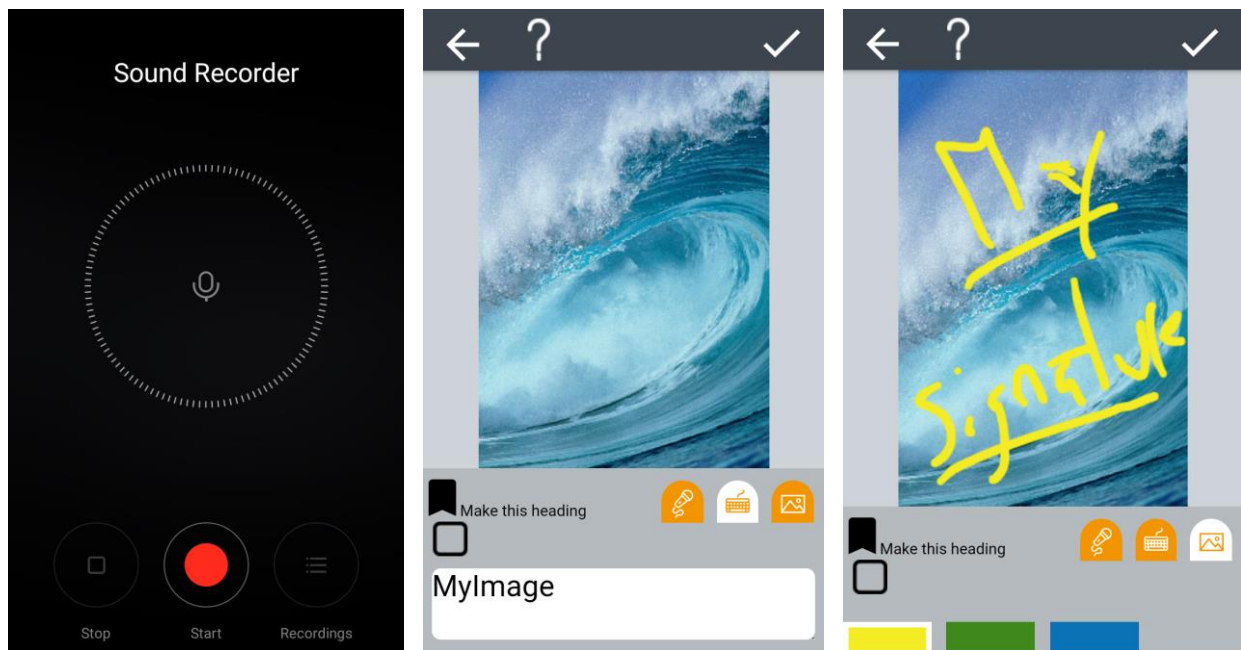


Figure 11: Edit options: (a) Sound recorder; (b) New text; and (c) Drawing.

2.4.4 Show story

The user can visualise the story at any time from the screen 'New story' (see Figure 9) through the button 'Slideshow' or from the screen 'Story list' (see Figure 14) by clicking on 'Play'. This way the user can check the content of the story in an easy manner.

Once the presentation mode is active, a new screen will emerge (see Figure 12). Clicking on the buttons on the header, the user can go back to the previous screen ('New Story' or 'Story list') or open the help screen.

By swiping the finger on the current slide from left to right or from right to left, the user can see the previous or the next slide respectively.

Each of the slides presented on this screen will contain all the changes made in screen 'Edit' (title text, signature or audio recording).

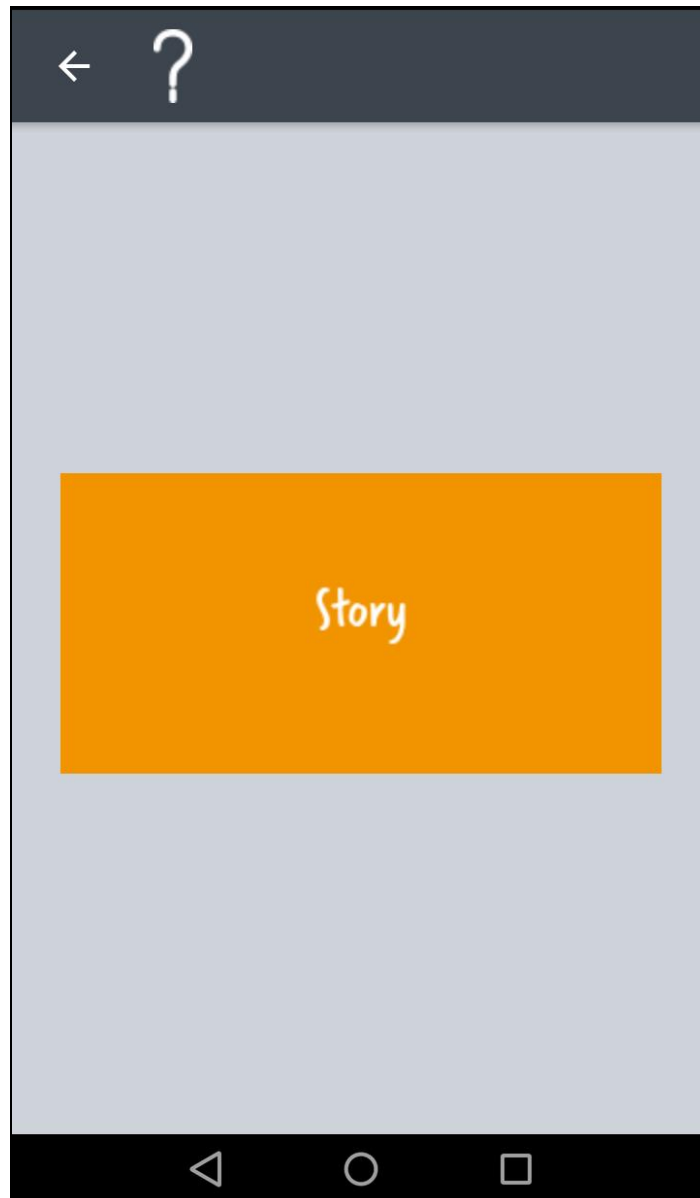


Figure 12: Screen ‘Show story’.

2.4.5 Send story

The screen ‘Send’ allows exporting the story in different forms. First, the user has the possibility of choosing a format to share the story ‘Choose the format’. The available formats include PDF (with one or four slides per page), ZIP and STORY.

The STORY format is the only export format which allows importing the story in *OurStory* after this was created. Defined by TREE, STORY is a lightweight format that permits to share the story quickly. One of the advantages is that key information can be extracted from the text file, facilitating the understanding by third parties to avoid incompatibilities in the future.

The ZIP format will save all the contents of the story in the same file (as STORY file), including multimedia resources, but it cannot be used to import a story. It is very useful since the user can access the contents directly without the *OurStory* App.

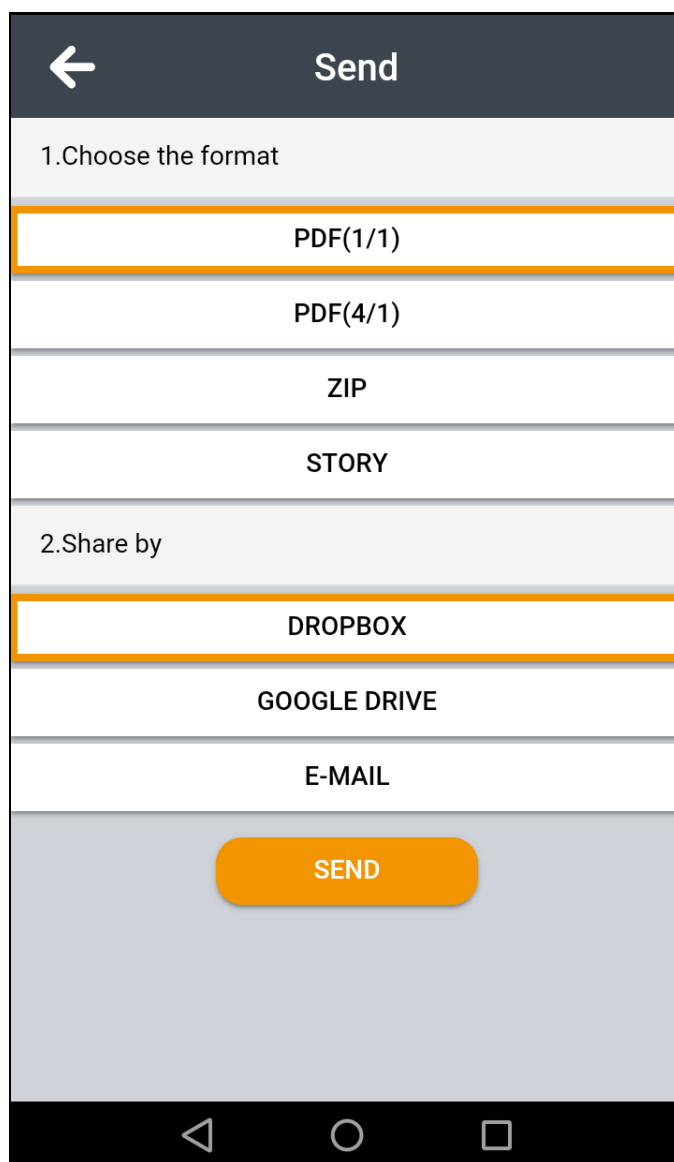


Figure 13: Screen ‘Send’.

The PDF format is probably the most versatile format since any file can be opened with a compatible reader installed on the device. It can be used to watch the contents of the story off-line or when *OurStory* has not been downloaded from the market. Analogously to the ZIP, it is not an importable format by the app.

Once the user has selected a format, the second step deals with choosing the option to share the selected file from a list containing three possibilities: Dropbox, Google Drive or email.

To complete the process, the user should simply click on the button ‘Send’ at the bottom of the page. Alternatively, the user can also click on the arrow on the top bar to cancel the operation.

2.4.6 Story list

This screen is used for getting access to the stories saved in the application and can be reached from the screen ‘Home’ (see Figure 8) by clicking on ‘Use existing story’.

On the header bar, together with the two well-known icons (i.e., ‘Go back’ and ‘Help’), a new button has been inserted. By clicking on it, the user will be allowed to import a story from Dropbox or Google Drive.

Nevertheless, only STORY formats will be valid. This means that importing a story requires that story to have previously been exported in STORY format as it was explained in the previous section.

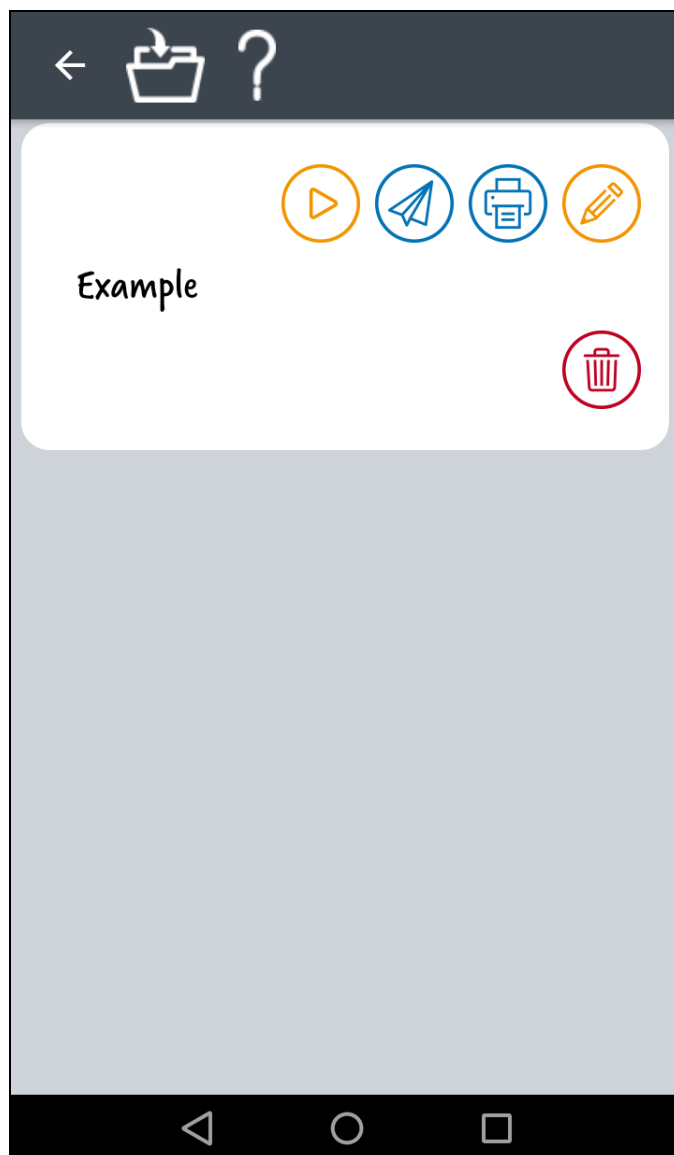


Figure 14: Screen ‘Story list’.

Under the header, the complete list of stories (imported or saved in advance) will be displayed. In addition, once the final version of the *OurStory* app is ready to be released, we will include an example to help users to understand the purpose of the application and its capabilities.

For each story, and next to its name, five buttons (described from left to right and from top to bottom) provide quick access to some of the functionalities already cited in this document:

- **‘Play’**: Access to the slideshow in the screen ‘Show story’ (see Figure 12).
- **‘Export’**: Access to the screen ‘Send’ (see Figure 13).
- **‘Print’**: Access to the print menu.
- **‘Modify’**: Access to the screen ‘New story’ (see Figure 9).
- **‘Delete’**: The story is remove from the internal memory.

3 Unity3D ARCHES test application

Unity3D ARCHES test application is a prototype designed and created to validate the performance of the functionalities developed in response to the needs identified by the exploration groups at the V&A and the WC. These are summarised in Table 2:

Table 2: Functionalities identified by the exploration groups in the UK.

Functionality	Description
Text enlarging	Get information from other fonts and enlarges the artwork’s description
Zoom	Enlarges an object or aspects of an object.
Text to speech	Reads text
Torch	The application acts as a torch to make dark object more visible
Multimedia resources	A functionality that ‘plays’ a short dramatization of the information about an object.
Object information	This functionality that would be able to tell users the height and width of an object and then encourage them to inspect in more detail the bits that are easily missed, e.g. the ornate carvings on the tops and bottom of objects.
Image processing	Shows images in photonegative.
QR	The application can read a Quick Response (QR) code and can verbally image.
Identified and show touch objects	The application can show touch objects/brightly coloured (friendly).
Guidance	Guide the visitor inside the museum, also the application will alert users about changes on the routes due to building or road works – important for wheel chair users too.
Caption and share information	Caption descriptions of tactile images available in the museums and make comments or share with other users.

3.1 App schema

Unity3D ARCHES test application has been developed under the multiplatform game engine Unity3D [10], so that it can be easily compiled and exported to be executed not only on the actually leading operating systems, i.e., Android and iOS, but also on Windows Mobile and Blackberry. This implies an even higher depth of penetration, making it accessible for almost everybody. Unity 3D is a development engine with built-in rendering system used for real- time rendering of 3D and 2D objects.

Our application is developed using Unity and several libraries. The main goal is to create an app with augmented reality and accessibility features, and visually attractive. For accomplishing these objectives application libraries include:

- **Vuforia [11]** : Used to recognise images of artwork and show the points of interest of them. Also used for the text recognition functionality. *“Vuforia is a software platform for creating Augmented Reality applications. Developers can easily add advanced computer vision functionality to any application, allowing it to recognize images and objects, or reconstruct environments in the real world”*.
- **EasyTTS [1]**: Used for text to speech functionality: “Easy TTS v2.2 (Text-to-Speech) is a plug-in that allows you to easily use the native text-to-speech feature of iOS and Android.”
- **ZXing.Net [12]**: Used for scanning and processing QR codes: *“A library which supports decoding and generating of barcodes (like QR Code, PDF 417, EAN, UPC, Aztec, Data Matrix, Codabar) within images.”*
- **I18n-unity-csharp [9]**: Used for the future internationalisation of the application (not in use in the test application). *“Lightweight internationalization for use with C#, uses common __('...') syntax.”*

3.2 Platforms

At the moment, the app is only deployed in Android platforms. An iOS version may be developed in the short term if the exploration groups want to test the functionalities with iPhones and iPads. Otherwise, since this application is intended to analyse whether the demands and needs presented in Table 2 are successfully fulfilled with the proposed functionalities, no more advances will be made in this context.

At present, Unity app is available for Android above 4.1 (API level 16).

3.3 Application interface – current development state

3.3.1 Main screen

Contrary to the strategy adopted for *OurStory*, the graphical interface was not designed by an expert because the application will not be released to the general public. The idea was to create an accessible and simple interface (see Figure 15) through which users can access to all the functionalities of the app:

- Text recognition
- Camera
- Images
- Multimedia
- Painting recognition

- QR Codes
- Text to speech

Each one of the functionalities will be explained in the following sub-sections.

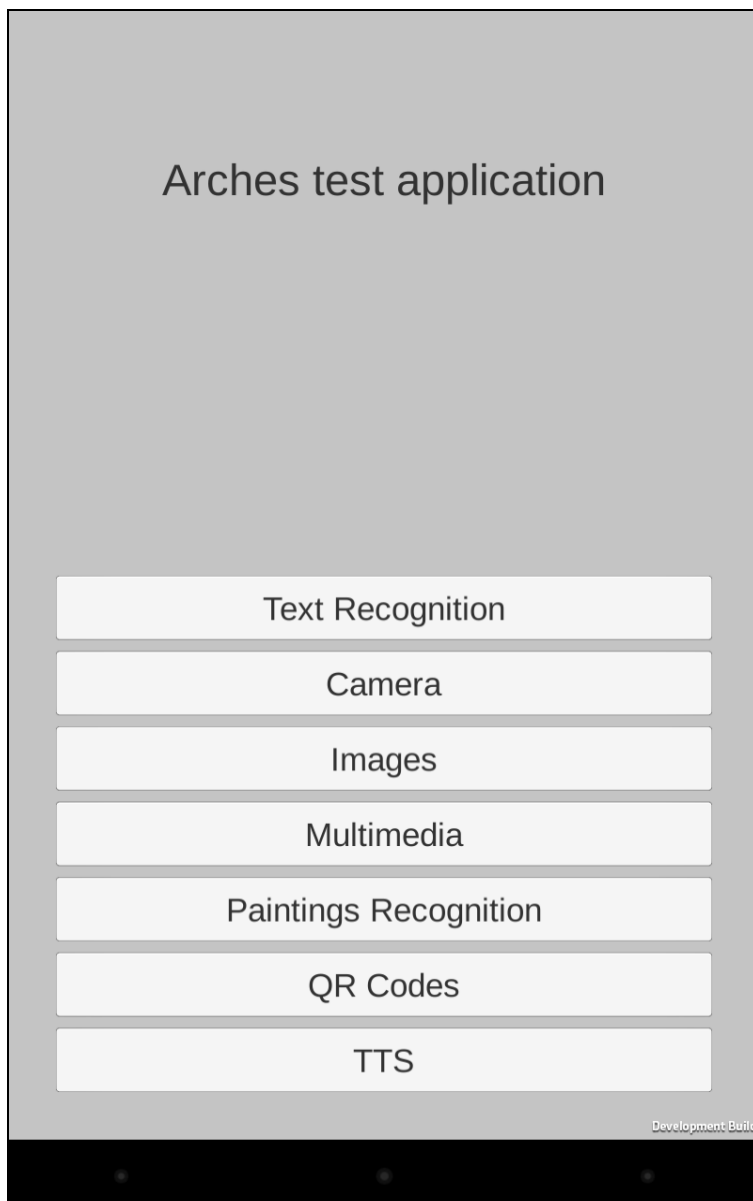


Figure 15: Screen ‘Main page’.

3.3.2 Text recognition

This functionality allows the user to recognise and process texts written in English in real-time by taking advantage of the camera embedded in the smartphone or tablet. Once the text has been recognised it is shown to the user in bigger size to read it more easily.

When this functionality is switched on (see Figure 16), the screen is divided into two areas. In the upper area, the camera is activated and the user has to fit the text to be recognised by the application inside the light grey rectangle. Once the text is recognised, each word is highlighted in orange and automatically displayed in the lower half of the screen with big black letters on white background – high contrast to help people with any kind of visual impairment.

Additionally, the flash of the device can be turned on by clicking on the button 'Flash' located at the upper right corner of the screen. This will allow the user to better illuminate the environment when trying to focus on a specific part of the text.

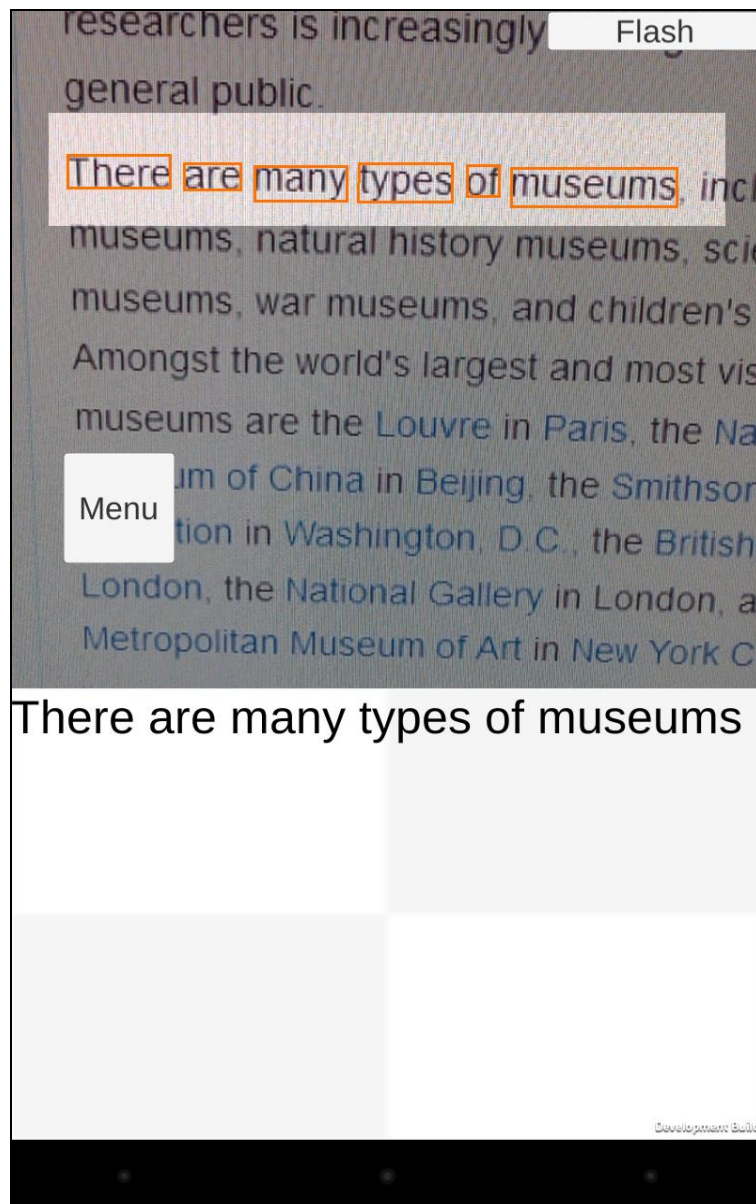


Figure 16: Screen 'Text recognition'.

3.3.3 Camera

The camera in the application has multiple functionalities:

- Different **filters** can be applied to the image on the screen in real time. There are several options, but the most important filters are: colour blindness simulation, black and white, colour inversion, sepia and colour tint.
- **Zoom** can be made to enlarge what the user is watching through the screen.
- **Photos** can be taken and the camera can be paused in case it will not be used for a large period of time. This will allow the user to save battery power.
- The active camera can be switched between the back and the front camera if both are available.

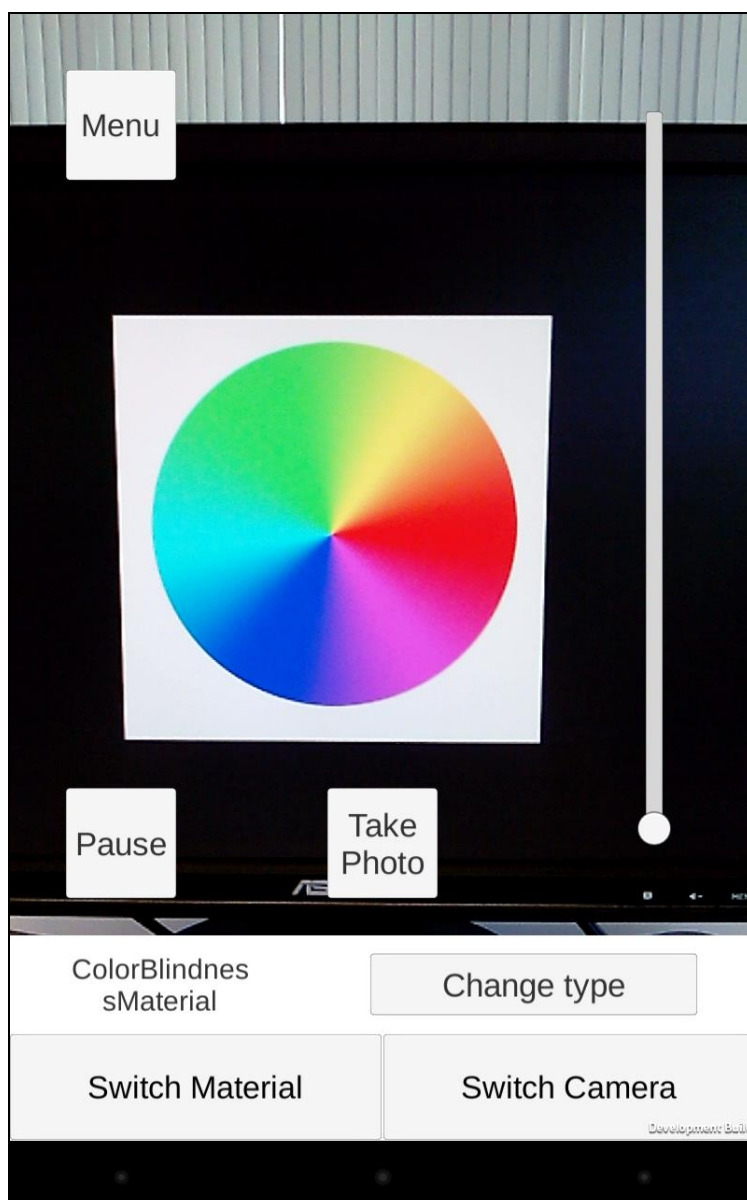


Figure 17: Screen ‘Camera’.

3.3.3.1 Filters

Filters are applied by means of the controls in the panel located at the bottom of the screen. A label on the top left of the panel shows the name of the filter applied on that moment. Below this label the button ‘Switch Material’ allows the user to change the filter. Depending on the specific type of filter, different options are enabled for their use:

- **Colour blindness:** Selecting the colour blindness simulation filter (see Figure 18a) the button ‘Change type’ appears next to the label with the name of the filter. This button is used to change the type of colour blindness.
- **Grey material, colour inversion and sepia:** With the filters grey material (see Figure 18b), colour inversion (see Figure 19a) and sepia, a slider shows up to change the intensity of the filter, being the lower intensity when the handle is on the left and maximum intensity when it is on the right.

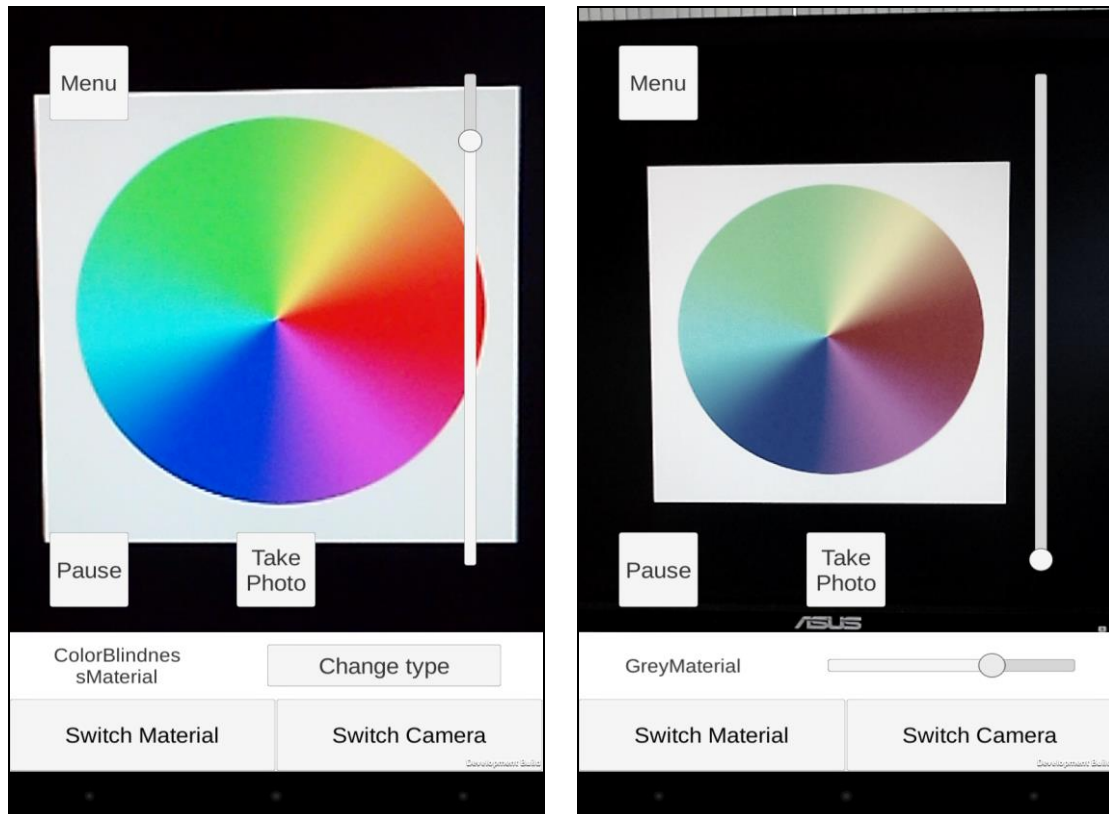


Figure 18: Example of filters I: (a) Colour blindness material and (b) Grey material.

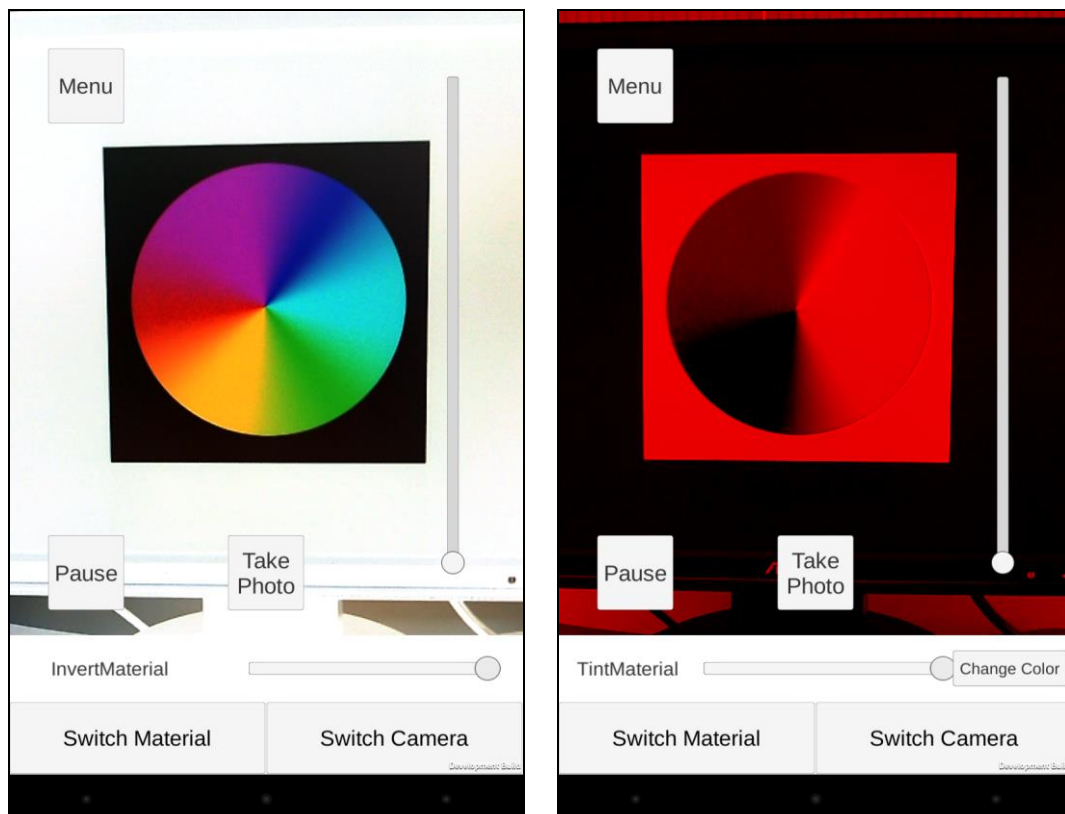


Figure 19: Example of filters II: (a) Invert material and (b) tint filter.

- **Tint filter:** With the colour tint filter (see Figure 19b) the user can select the colour that will be applied while the slider will help to vary the intensity of the result.

All these filters (and others that may be suggested by the exploration groups in the future) have been developed to be used with images but also with real-time videos when the camera is on.

3.3.3.2 Zoom

To use the zoom functionality there are two alternative actions the user can carry out: use the pinch-to-zoom gesture or the slider located on the right side of the screen (see Figure 17). The slider works as follows: If the handle of the slider is at the bottom, the image is presented in its original size while if the handle is moved to the top of the slider, it makes the image twice the original size.

3.3.3.3 Camera management

Three buttons have been inserted to use the camera:

- **'Take photo':** In the centre of the screen (see Figure 17), it allows the user to take a photo at any time. The image shown on the screen (with the zoom and filter applied at that very moment) will be saved to a directory in the device.
- **'Pause':** On the left of the screen (see Figure 17), the camera is paused the first time the user clicks on it. Then, the label will change to 'Play' to resume the functioning when clicking on it again. This functionality allows to save battery of the device by turning off the camera when it is not in use.
- **'Switch camera':** At the bottom right corner of the screen (see Figure 17), it switches the active camera from back to front and vice versa.

3.3.4 Images

This screen has almost the same functionality of the screen 'Camera' but applied to a static image. On the bottom panel two buttons are added (see Figure 20): 'Change image' and 'Share'. Indeed, except for the new buttons on this screen, the functioning is the same previously described for the screen 'Camera':

- **'Change image':** This button changes the image shown when the user clicks on it, alternating between a set of images previously loaded in the application.
- **'Share':** The user can share the image displayed on the screen with other people.

3.3.5 Multimedia resources

The screen 'Multimedia resources' (see Figure 21) shows a demo of a static image, a video and an audio in the application.

The screen has three buttons at the bottom:

- **'Set image':** It shows an image on the screen.
- **'Set audio':** It plays a video on the screen.
- **'Set video':** It plays an audio track on the screen.

Although quite simple, this part of the application is useful to test different formats and configurations before the multimedia content is integrated into the final version of the ARCHES app.

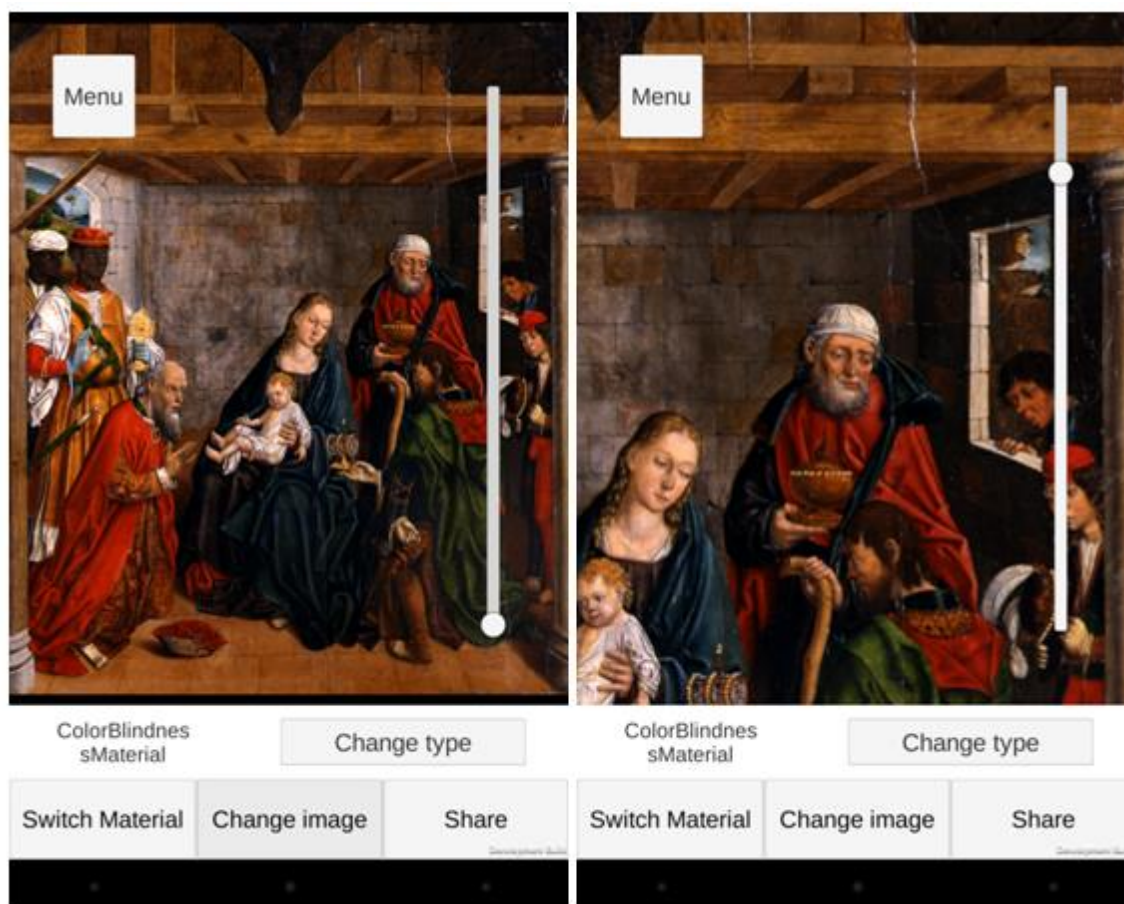


Figure 20: Static image example.



Figure 21: Screen ‘Multimedia resources’.

3.3.6 Recognition of paintings

When this functionality is chosen on the main screen, it automatically switches the camera on to recognise paintings from a collection of predefined (and trained) images.

To recognise a painting, the user has to aim the camera at it and if this is within the set of selected works of art, the system will automatically display red points (points of interest) on the painting (see Figure 22). These points contain additional information of the painting and the user can click on them to get further information (e.g. a brief description, a detail, a gossip, a link to a website, etc.).

The points of interest can be hidden by clicking on the button ‘Toggle points’. This way the user can take a photo with no interferences.



Figure 22: Screen ‘Painting recognition’.

3.3.7 QR codes

Other functionality demanded by the UK exploration group was the recognition of QR codes using the camera of the smartphone or tablet and subsequent presentation of the information to the user.

To recognise a QR code, first it must be fitted into the rectangle in the centre of the screen (see Figure 23a). Next, the user has to click on the button ‘Scan QR’ and maintain the QR code inside the rectangle while it is scanned. When the application has successfully scanned the QR code, the data contained in it will be shown in the grey box at the bottom of the screen (see Figure 23b). If it is a URL, the browser will automatically open it.

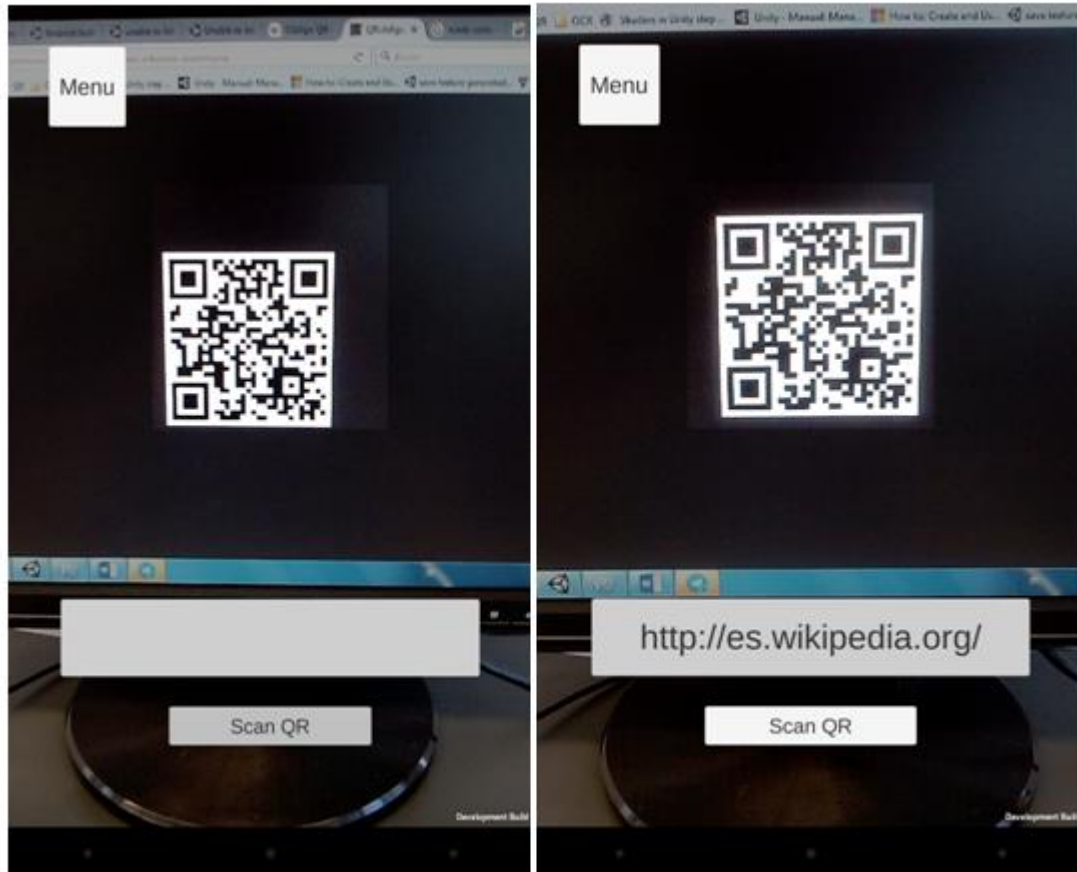


Figure 23 : Screen ‘QR’: (a) Preparation and (b) Recognition.

3.3.8 Text to speech

Text to speech converts written text into voice in various languages including: Spanish, English and German. As a complement to this, available libraries allows the developer to adjust some parameters of the voice like volume, rate and pitch.

In the context of this prototype, the first thing to do when using the text to speech functionality is to write the text that will be converted into voice. On the screen in Figure 24, the text should be introduced in the input text at the top. The next thing to do is to choose the language in the panel at the bottom of the screen – activate the checkbox with the desired language. With the sliders located in the centre of the screen, the volume, rate and pitch can dynamically be modified until the target values are reached. Once the configuration is ready, the user can click on the button ‘Speak’ and the device will start to reproduce the written text with voice. To stop the audio the button ‘Stop’ will be enabled.

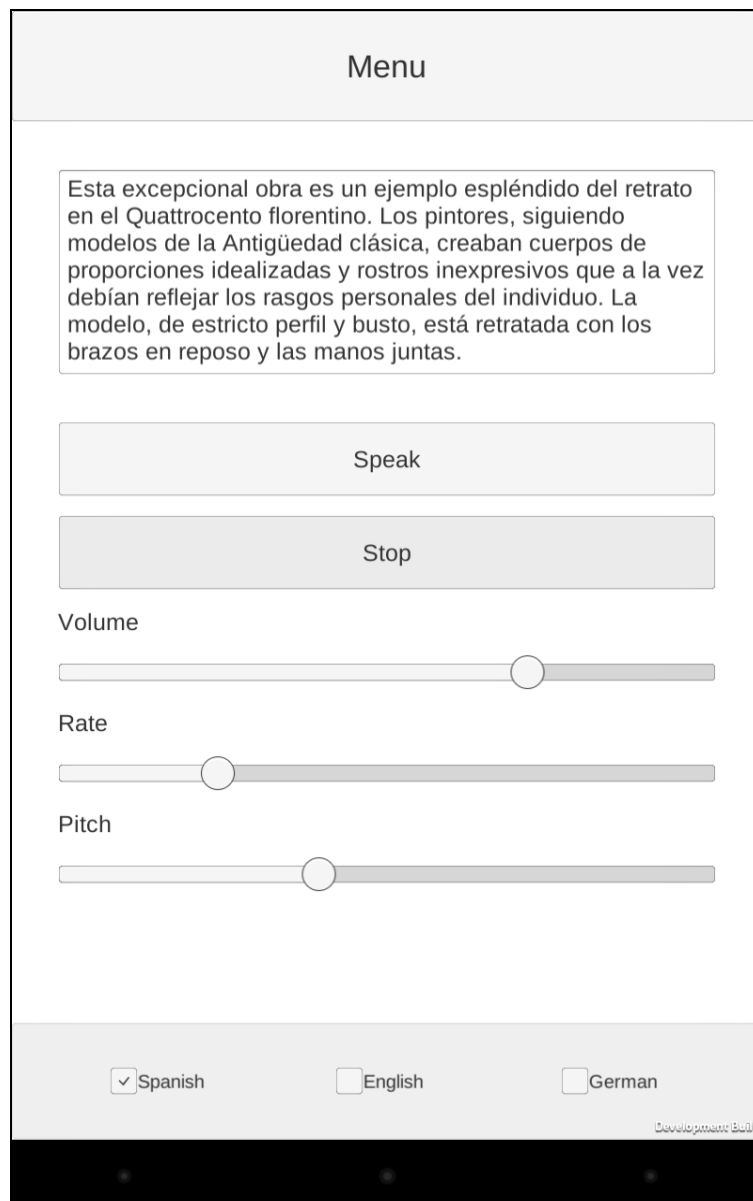


Figure 24: Screen 'Text to speech'.

4 Museum application design

The development of an accessible application for smartphones and tablets is one of the main challenges of ARCHES. In fact, the objective of the application is two-fold: On the one hand, the application is intended to provide specific and adapted content to the target audiences by taking advantage of the close collaboration with the participating museums. Therefore, they can actively cooperate with the developers to decide which is the most suitable design or the most appropriate multimedia resources; how the collection and artworks are presented to the users depending on what they can do or they like and not on their limitations; whether alternative texts should be used, etc. These and other actions will contribute to the generation of more personalised and attractive experiences.

On the other hand,

In the following sub-sections, we will describe a set of mock-ups that have been designed following the principles of accessibility – as summarised in the previous deliverables in WP3 and WP4 – as well as the guidelines defined by experts in social research within the consortium. However, this is an initial version that may be modified, improved and updated all along the continuous iterative development process on which the user-centred strategy of the project is based.

4.1 Home screen

The first screen of the app (see Figure 25) presents a brief introduction of the available functionalities, so that the user can get familiar with them at first sight. These functionalities are the same outlined in the previous section, but they are integrated into overall design to boost the experience of differences and difficulties associated with perception, memory, cognition and communication when visiting a museum.

The user can select between the options:

- **‘Guided tour’**: Start a predefined tour in the museum.
- **‘On your own’**: Freely explore the collection taking advantage of the functionalities of the application, such as filters, text to speech, etc.
- **‘More information’**: Additional information about the application and the ARCHES project.

The two last options will jointly be addressed with the exploration groups to decide the best way to present the content.

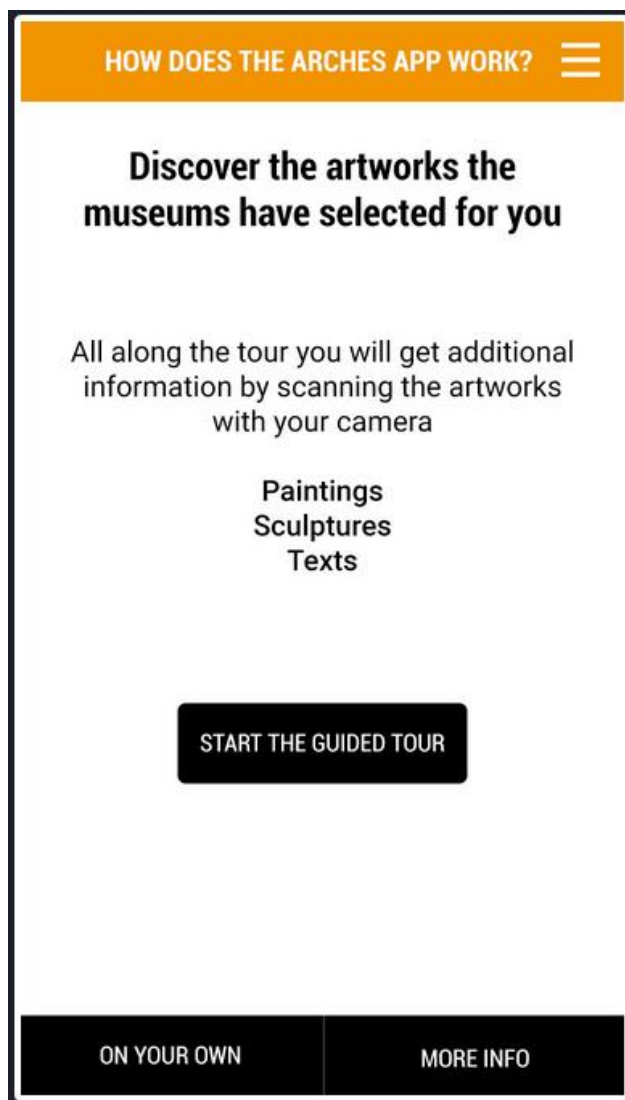


Figure 25: Screen 'Intro'.

4.2 Tours

In this screen (see Figure 27), a complete list of the artworks that compose the current tour are listed. There are different possibilities to establish the order of the elements and this will be decided by the museums. In principle, the next artwork in the tour will be the closest to the previous one to make the tour attractive and comfortable, especially for those with reduced mobility.

This screen may be preceded by other where the user should choose the museum being visited or the tour that will be followed (e.g. different styles, periods, theme, etc.). However, this option is still open to be agreed with the partners of the consortium.

By clicking on one of the artworks, the user will be redirected to the screen 'Steps'.



Figure 26: Screen 'Tour'.

4.3 Steps to find the artwork

This screen (see Figure 27) presents the necessary information about the steps that a user should follow to find the next artwork in the tour. Unfortunately the application does not implement an automatic guiding because of the limitations of global positioning systems, such as GPS and Galileo, in indoor environments. Other alternatives (e.g. the use of beacons) have been studied but they require an investment that goes beyond the scope of this project. However, the use of Augmented Reality will be analysed to be integrated into this approach.

The steps to follow are defined by the museums with the involvement of educators and experts in accessibility. Once the user has found the artwork in the museum, a click on the button 'Switch on the camera' at the bottom of the screen is enough to continue the tour. At this stage, the camera is switched on.

To facilitate the navigation, the user can click on the arrow on the top bar and go back to the previous screen.

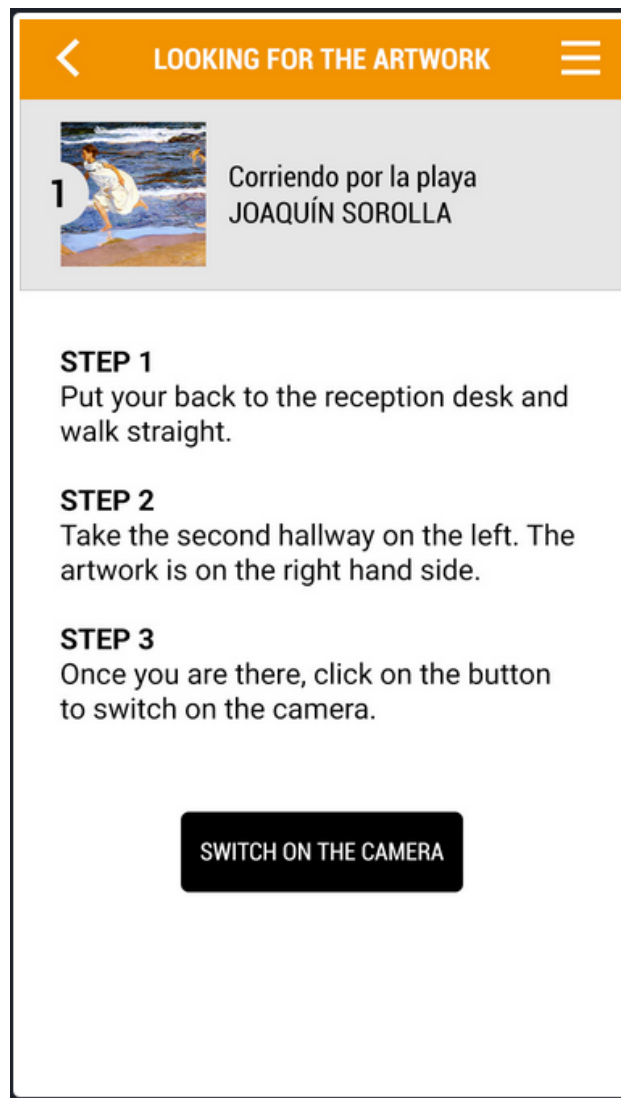


Figure 27: Screen 'Steps'.

4.4 Camera screen

The screen 'Camera' (see Figure 28) has different options to make easier the recognition of paintings as described in section 3.3.6. A button with a left arrow shape is displayed in the header to go back to screen 'Steps'. On the right, the pull-down menu gives access to information about the application and its functionalities.

Under the header, a full list of the artworks of the tour is shown. The current artwork is highlighted. Nonetheless, the user can swipe the finger from left to right and from right to left to see the remaining elements. Besides, it is possible to select one of them by clicking on it. This will redirect the user to the screen where the instructions to find the room and the artwork are summarised (see Figure 27).

The zoom functionality commented in section 3.3.3.2 has been implemented on the right hand side of the screen with a semi-transparent slider. This way the user can amplify the image captured with the camera without walking around.

In the middle of the screen the camera button takes a photo, which can be discarded or stored in the smartphone or tablet for future use.

The two buttons at the bottom of the screen are:

- **‘Filters’**: Apply any of the filters described in section 3.3.3.1.
- **‘Torch’**: Switch on the flash. This option may be deactivated in case the museums do not permit the use of flash inside the gallery.



Figure 28: Screen ‘Camera’.

When the user points at one of the paintings that can be recognised by the application and this is captured by the camera, the user will note this as reflected in Figure 29:

- The artwork is highlighted by means of a dotted red line. We are still discussing with other partners how to proceed afterwards, in the sense that perhaps the application should remember the last artwork the user has identified so she does not need to continuously aim the camera at the wall.
- A new bar emerges below the camera button. This option is intended to show or hide the points of interest or information points. When clicking on one of them, further details are presented: a brief text, a link to a website, a picture, a video, etc.

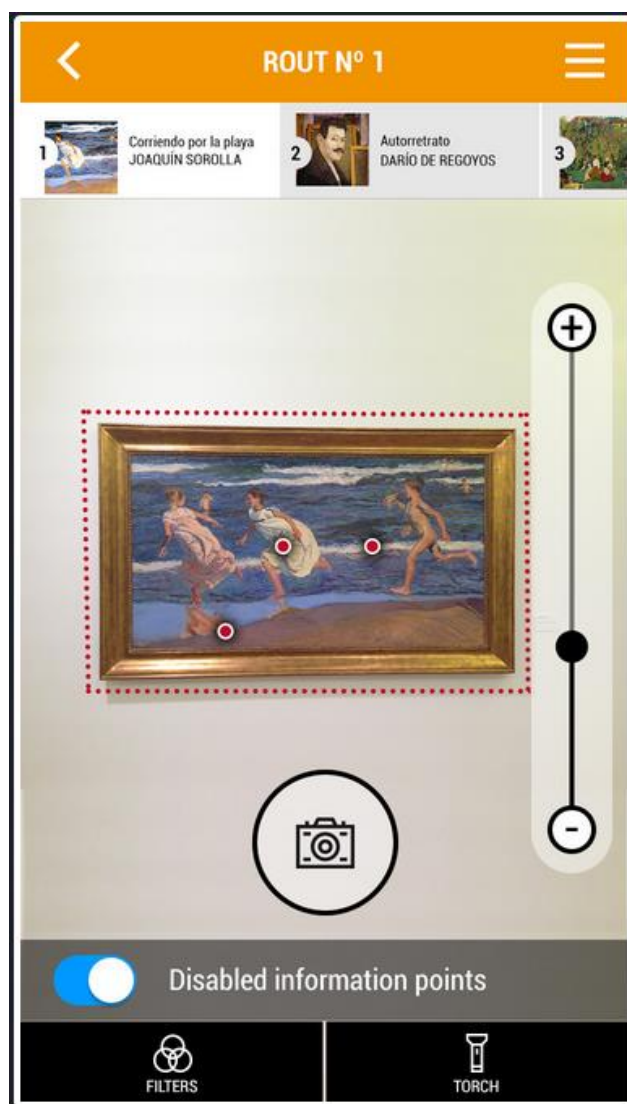


Figure 29: Screen ‘Recognition’.

By clicking on the central area of the recognised painting (not on one of the information points), the whole description of the artwork is presented in a new screen (see Figure 30). The structure of this page will be defined by each of the museums depending on their own methodology. In principle, it will show a high resolution image of the artwork, followed by its title and author. The amount of data included in the technical details will be adapted to the user needs, so alternative tags may be possible.

The main text will be available in different languages. Similarly to the technical details, the user may be allowed to choose the version that better suits her capabilities and interests. This is a common approach that is being applied to all technical work packages in ARCHES (see, for example, deliverable D5.2 “Test setup – 1st version”).

The text can be complemented with multimedia resources that the educators will select. Apart from images and videos, avatars can be used for sign language translation (see section 4.5). We will consider any possible restrictions, especially concerning the file size.

The options at the bottom of the screen are:

- **‘Filters’**: Same filters previously described but applied to static images.
- **‘Send to’**: Share the content by email and PDF generation.

- **'Share'**: The information can be shared and commented via social networks.
- **'Delete'**: Cancel previous actions. This button may be redefined in the next iterations of the development process.



Figure 30: Screen 'Description'.

4.5 Avatar component

An avatar component will be integrated into the screen where text descriptions are available such as those in Figure 30. The avatar will sign the texts about the artwork in various languages including Spanish, English and German.

In a first step the avatar component will have the ability to play pre-rendered videos so we can guarantee a satisfying look and a simple GUI to operate it. The user will be able to switch the language according to his preferred sign language.

In a second step we will attempt to replace the component partially with a real time rendering component for the mobile apps. For the user the avatar experience will stay like a pre-recorded video which includes one textured character with one static, basic light set up. Thus, only limited geometry, surroundings or effects are needed. The full capacity of the rendering component will be used to guarantee a nice looking

picture with smooth movements. By using mobile friendly frame rates, simplified shaders and proper light mapping, the needed computing power will also be reduced to a minimum.

The avatar component will have its own sign database where the translations and sentences are pulled from. The avatar component will build upon the rendering engine in order to combine the signs for single words to whole sentences. Text, signs and body movement of the character can be fixed before and saved as one data set into the application. Then it is passed to the rendering component.

5 Conclusions

This document defines the advances in the development and design of the web applications in the ARCHES platform. In this document, the sections have been divided per application:

- Our Story
- Unity3D ARCHES test application
- Museum app design

Some application developments have already started, others are still in the design phase. The designs have been evaluated with the exploration groups and this process will continue in the future all along the pilot exercises. The different prototypes will be evaluated by the museums and the volunteers collaborating with the project in United Kingdom, Spain and Austria.

The next advances and results will be included and detailed in the following deliverables:

- D4.7 “Report on system integration – 2nd version”: This second version corresponds to the updates after the feedbacks obtained as a result of the pilot exercises II but before the pilot III starts.
- D4.8 “Report on system integration – 3rd version”: This is the final version of the report on the integration once the pilot exercises have been completed. In addition, a brief user guide will be included in this report so as to give assistance on how to use the application.

References

- [1] Android. Available online at (26th September 2017): <https://developer.android.com/about/dashboards/index.html?hl=es-419>
- [2] Angular. Available online at (26th September 2017): <https://angular.io/>
- [3] D4.1 "Report on system requirements for handles devices". ARCHES project. Available online at (26th September 2017): <http://arches-project.eu/download/993/>
- [4] Cordova. Available online at (26th September 2017): <https://cordova.apache.org/>
- [5] EasyTTS. Available online at (26th September 2017): <https://www.assetstore.unity3d.com/en/#!/content/22315>
- [6] Ionic2. Available online at (26th September 2017): <http://ionic.io/2>
- [7] iOS. Available online at (26th September 2017): <https://developer.apple.com/support/app-store/>
- [8] "Gartner Says Worldwide Sales of Smartphones Grew 7 Percent in the Fourth Quarter of 2016". Available online at (26th September 2017): <http://www.gartner.com/newsroom/id/3609817>
- [9] I18n-unity-csharp. Available online at (26th September 2017): <https://github.com/MoonGateLabs/i18n-unity-csharp>
- [10] Unity3D. Available online at (26th September 2017): <https://unity3d.com/es>
- [11] Vuforia. Available online at (26th September 2017): <https://library.vuforia.com/getting-started>
- [12] ZXing.Net. Available online at (26th September 2017): <https://github.com/micjahn/ZXing.Net>