

# Enhancing museum visitors' experience through user-adapted interactive multimedia descriptions of exhibits

*Denis Sancov*

Thesis submitted in partial fulfillment of the requirements for the

*Masters' of Science degree in Computer Science and Engineering*

University of Crete

School of Sciences and Engineering

Computer Science Department

Voutes University Campus, 700 13 Heraklion, Crete, Greece

Thesis Advisor: Prof. *Constantine Stephanidis*

---

This work has been performed at the University of Crete, School of Sciences and Engineering, Computer Science Department.

The work has been supported by the Foundation for Research and Technology - Hellas (FORTH), Institute of Computer Science (ICS).




University of Crete  
Computer Science Department

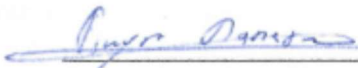
**Enhancing museum visitor's experience by delivering interactive multimedia  
description about exhibits based on user profile**

Thesis submitted by  
**Denis Sancov**  
In partial fulfillment of the requirements for the  
Masters' of Science degree in Computer Science


THESIS APPROVAL

Author:   
\_\_\_\_\_  
Denis Sancov

Committee approvals:   
\_\_\_\_\_  
Constantine Stephanidis  
Professor, Thesis Supervisor

  
\_\_\_\_\_  
George Papagiannakis  
Assistant Professor, Committee Member

  
\_\_\_\_\_  
Margherita Antona  
Principal Researcher, Committee Member

Departmental approval:   
\_\_\_\_\_  
Angelos Bilas  
Professor, Director of Graduate Studies

Heraklion, June 2019



# Enhancing museum visitors' experience through user-adapted interactive multimedia descriptions of exhibits

## Abstract

In the past decade, the availability and role of mobile phones have significantly increased. In the past, they were used solely for communication, while nowadays, modern smartphones have become hand-held portable high-performance computers. Their technological features allow people to use them for various tasks in various domains, including work, education, entertainment, etc., while their potential benefits and innovative use is still under investigation in other areas, such as the museum environment.

This thesis proposes a system supporting the management of museum exhibits as well as the authoring and presentation of its multimedia description to museum visitors. Multimedia descriptions are dynamically generated according to predefined conditions and user profiles. This research work is rooted in the use of smartphones in Ambient Intelligence Environments, and in the exploitation of Semantic Web technologies for interactive authoring of multimedia descriptions about exhibits. Various technologies stemming from different research domains have been employed by the designed system. Unity3D based mobile application and ReactJS administration panel are working on top of .NET Core web server framework with an RDF compliant storage system.

Regarding the system implementation, a Model-View-Controller oriented architecture was adopted, along with the cooperation of various subsystems. System components have been designed and built based on a generic architecture, in order to support knowledge management, annotation and extraction, multimodal interaction, content personalization and UI adaptation.

Furthermore, the proposed system facilitates the extraction of semantic knowledge according to user preferences and usage history. As a result, such information can help the system to generate more relevant and personalized interactive multimedia descriptions.

A heuristic evaluation was conducted with the participation of 5 experts to study the developed system in terms of usability and user experience. The evaluation results are considered promising, confirming the usefulness of the system and encouraging further research and development work in this area.



# Ενίσχυση της εμπειρίας των επισκεπτών μουσείων μέσω προσωποποιημένης διαδραστικής πολυμεσικής πληροφόρησης

## Περίληψη

Τα τελευταία χρόνια, η διαθεσιμότητα και ο ρόλος των κινητών τηλεφώνων έχουν αυξηθεί σημαντικά. Στο παρελθόν, τα κινητά τηλέφωνα, χρησιμοποιήθηκαν αποκλειστικά με σκοπό την επικοινωνία. Στη σημερινή όμως εποχή, τα σύγχρονα έξυπνα κινητά τηλέφωνα έχουν μετατραπεί σε φορητούς υπολογιστές υψηλών επιδόσεων. Τα τεχνικά χαρακτηριστικά που διαθέτουν, επιτρέπουν στους χρήστες να τα χρησιμοποιούν για ποικίλες εργασίες σε διάφορους τομείς της καθημερινότητας, όπως η εργασία, η εκπαίδευση, η ψυχαγωγία, κ.λπ. Ωστόσο, τα πιθανά οφέλη και η καινοτόμος χρήση τους σε άλλους τομείς, όπως τα μουσειακά περιβάλλοντα, αποτελούν ακόμα αντικείμενο συνεχούς έρευνας και ανάπτυξης.

Αυτή η εργασία παρουσιάζει ένα σύστημα που υποστηρίζει τη διαχείριση μουσειακών εκθεμάτων και παρέχει την δυνατότητα σύνταξης και παρουσίασης πολυμεσικής πληροφορίας στους επισκέπτες των μουσείων. Το περιεχόμενο της παρουσίασης δημιουργείται δυναμικά σύμφωνα με προκαθορισμένες συνθήκες αλλά και με βάση τα χαρακτηριστικά και τις προτιμήσεις του χρήστη. Το προτεινόμενο σύστημα βασίζεται στην αξιοποίηση των «έξυπνων» κινητών τηλεφώνων (smartphones) σε περιβάλλοντα Διάχυτης Νοημοσύνης και στη χρήση τεχνολογιών του Σημασιολογικού Ιστού (Semantic Web) με σκοπό την επαύξηση της εμπειρίας επίσκεψης μέσω της εξατομικευμένης πληροφόρησης.

Ποικίλες τεχνολογίες από διαφορετικούς ερευνητικούς τομείς, έχουν χρησιμοποιηθεί για τον σχεδιασμό του παρόντος συστήματος. Συγκεκριμένα, η εφαρμογή για κινητά τηλέφωνα χρησιμοποιεί την μηχανή κατασκευής 3D γραφικών Mobile Unity3D και το σύστημα διαχείρισης περιεχομένου είναι υλοποιημένο με τεχνολογία ReactJS. Τα παραπάνω πλασιώνουν το .NET Core Web Server καθώς και το σύστημα μοντελοποίησης δεδομένων που χρησιμοποιεί σύνταξη RDF για την αναπαράσταση γνώσης. Για την ανάπτυξη του συστήματος, υιοθετήθηκε μία αρχιτεκτονική τύπου Model-View-Controller, σε συνδυασμό με άλλα υποσυστήματα. Τα επιμέρους υποσυστήματα υποστηρίζουν την αναπαράσταση και εξαγωγή γνώσης, τον σχολιασμό (annotation), την πολυτροπική αλληλεπίδραση, τη δυνατότητα εξατομίκευσης της πληροφορίας (content personalization) καθώς και τη δυνατότητα προσαρμογής της διεπαφής (UI adaptation). Επιπλέον, το προτεινόμενο σύστημα διευκολύνει την εξαγωγή σημασιολογικών γνώσεων σύμφωνα με τις προτιμήσεις των χρηστών και το ιστορικό χρήσης. Συμπερασματικά, αυτές οι πληροφορίες μπορούν να βοηθήσουν το σύστημα να δημιουργήσει πιο σχετικές και εξατομικευμένες πολυμεσικές παρουσιάσεις.

Για την αξιολόγηση της ευχρηστίας και της εμπειρίας των χρηστών, διεξήχθη μία ευρετική αξιολόγηση με τη συμμετοχή πέντε εμπειρογνομόνων. Η αξιολόγηση είχε θετική έκβαση, επιβεβαιώνοντας τη χρησιμότητα του συστήματος και ενθαρρύνοντας περαιτέρω ερευνητικές και αναπτυξιακές εργασίες στον τομέα αυτόν.





## Acknowledgements

I would like to thank my supervisor Prof. Constantine Stephanidis for the support and encouragement throughout the conduction of this thesis.

Furthermore, I would like to thank my colleagues at the Human-Computer Interaction laboratory of ICS-FORTH, especially Dr. Emmanouil Zidianakis and Dr. Nikolaos Partarakis for their guidance throughout this thesis.

The work reported in this thesis has been conducted in the context of a scholarship provided by the Institute of Computer Science (ICS) of the Foundation for Research and Technology – Hellas (FORTH), including financial support and the use of the necessary equipment and facilities.

Additionally, I would like to thank my friends Vladimir and Konstantinos for their support throughout the conduction of this thesis. Last but not least, I would like to thank my parents, Grigori and Tatiana, as well as my brother Manolis, who despite the distance between us stood by me at all times.



# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>15</b>
<b>2</b>	<b>Background and Related Work</b>	<b>17</b>
2.1	Ambient Intelligence	17
2.2	Ambient Intelligence environments	19
2.3	Semantic Web	20
2.3.1	RDF	21
2.3.2	OWL Web Ontology Language	22
2.3.3	CIDOC-CRM	22
2.4	Related Work	24
2.4.1	Indonesian museum information system	24
2.4.2	The iMuse Mobile Tour	26
2.4.3	A Multimodal Mobile Museum Guide for All	27
2.4.4	Location-Aware Tour Guide Systems in Museum	28
2.4.5	Interactive Museum Guide	28
2.5	Discussion and Motivation	29
<b>3</b>	<b>System Design</b>	<b>30</b>
3.1	User groups and requirements	30
3.1.1	User groups	31
3.1.2	User requirements	31
3.2	Design	32
3.2.1	Schema / High level architecture	32
3.2.2	Design of the Internal Components	33
3.2.3	Design of the Curator Authoring Tool	36
3.2.4	Design of the Mobile App	37
<b>4</b>	<b>Implementation</b>	<b>39</b>
4.1	Backend Service	39
4.1.1	Application Provided Interface (API)	39
4.1.2	Facebook Client	41
4.1.3	SPARQL Client	41
4.1.4	INSERT	42
4.1.5	DELETE	43
4.1.6	SELECT	43
4.1.7	ASK	44
4.2	Curator Authoring Tool	45
4.2.1	Collections	45
4.2.2	Exhibits	46
4.2.3	Create Exhibit	46
4.2.4	Exhibit Description Builder	48
4.3	Unity Museum Guide	49
4.3.1	Basic Architecture	49
4.3.2	View Components	51
<b>5</b>	<b>Evaluation</b>	<b>57</b>
5.1	Heuristic Evaluation	57

<b>5.2</b>	<b>The Evaluation Process.....</b>	<b>57</b>
<b>5.3</b>	<b>Severity score .....</b>	<b>58</b>
<b>5.4</b>	<b>Results.....</b>	<b>59</b>
5.4.1	Curator Authoring Tool.....	59
5.4.2	Unity Mobile Application for Visitors.....	63
<b>6</b>	<b><i>Lessons Learned</i> .....</b>	<b>64</b>
<b>7</b>	<b><i>Discussion and Future Work</i> .....</b>	<b>65</b>
<b>7.1</b>	<b>Discussion.....</b>	<b>65</b>
<b>7.2</b>	<b>Future Work .....</b>	<b>66</b>
<b>8</b>	<b><i>Bibliography</i> .....</b>	<b>67</b>

# List of Figures

Figure 1: Aml contributing technologies.....	18
Figure 2: Semantic Web Stack .....	20
Figure 3: An RDF Graph Describing Eric Miller .....	21
Figure 4: CIDOC CRM Class Hierarchy .....	23
Figure 5: Android Application .....	25
Figure 6: QR Code Tag on Bicycle Miniature Display .....	26
Figure 7: iMuse Mobile Tour architecture .....	27
Figure 8: Location-Aware Tour Guide System .....	28
Figure 9: Interface of the object recognition application.....	29
Figure 10: System Architecture.....	33
Figure 11: Ontology.....	35
Figure 12: The structure of Aml Middleware .....	36
Figure 13: Class that can be serialized into the triples .....	42
Figure 14: Serialized collection object .....	42
Figure 15: Insert query.....	42
Figure 16: Insert raw query.....	43
Figure 17: Delete query builder .....	43
Figure 18: Delete raw query .....	43
Figure 19: Select query builder .....	44
Figure 20: Select raw query .....	44
Figure 21: Ask query builder .....	44
Figure 22: Ask raw query .....	44
Figure 23: Collections Page.....	45
Figure 24: Exhibits.....	46
Figure 25: Create Exhibit Form .....	47
Figure 26: Add material .....	47
Figure 27: Multimedia Description Builder.....	48
Figure 28: Audience Picker .....	48
Figure 29: Unity App Basic Architecture .....	51
Figure 30: Unity App. View Components Schema .....	51
Figure 31: Unity, Unauthorized User .....	52
Figure 32, Unity App. Dashboard Component.....	52
Figure 33, Unity App, Settings Component .....	53
Figure 34, Unity App, Target Group.....	53
Figure 35, Unity App, Font Settings .....	54
Figure 36, Unity App, Voices Component.....	54
Figure 37, Unity App, QR Code Scanner .....	55
Figure 38, Unity App, Information Component.....	55
Figure 39, Unity App, Virtual Narrator.....	56

# List of Tables

Table 1, System ontology user components.....	34
Table 2, System ontology, exhibit description components .....	35
Table 3, API Methods.....	39
Table 4, Severity Score.....	58
Table 5, Curator Authoring Tool Issues.....	59
Table 6, Unity Mobile App Issues.....	63

# 1 Introduction

Smartphones have evolved significantly, over the past 10 years. The release of first iPhone and development of Android platform have changed dramatically the whole market of mobile applications. Due to powerful competition between the main vendors, performance improvements and innovations, smartphones have evolved into a powerful and small minicomputer. Beside the performance improvements, they were also equipped with a wide range of sensors. Along with hardware part, software platforms, such as Android and iOS, also became mature, more advanced and functional. In this context, as the number of the smartphones is esteemed to be billions [1], developers and researchers have a good opportunity to apply new ideas in various domains.

Ambient Intelligence (Aml) environments are characterized by the ubiquitous and unobtrusive presence of electronic equipment in the user's environment, that allows monitoring the user's actions and react accordingly in order to help them achieve their goals. In Aml environments the interface is indistinguishable from the physical setting, as the real world assumes the role of the interface.

The potential benefits and innovative use cases of smartphones and Semantic Web technology, is still under investigation across multiple domains, such as the museum environment. This thesis aims to create a system that will explore the features of aforementioned technologies. Semantic Web, by its nature, provides the shared storage that can be accessed and maintained by museums all around the world. This feature opens new possibilities to data interexchange and personalization features since each museum would be able to access visitor preferences and its history of visits. Moreover, global collaboration between museums would simplify and reduce costs of the data management.

The system in this thesis will have the ability to model, store, retrieve and visualize linked object data. It will be able to work in autonomous or global mode. Moreover, it has to produce the opportunity to maintain museum collections, interactive multimedia description about exhibits and to display them to the potential museum visitors. The museum's curator should be able to have the capability to add new collections and to fill them with exhibits using the CIDOC-CRM ontology and to construct rich multimedia description that respects user preferences in the context of target audience. The personalization feature will be based on ontology that describes connections between various entities. By exploring visitor's history and discovered exhibits, relevant collections can be suggested or more personalized descriptions generated.

The visitor will be provided with a universal mobile application that can be used to explore written multimedia constructions, by scanning the QR code of the exhibit. The application supports multi-modal interaction as it provides multimedia content using sound, text or visual alternatives.

The results of heuristics evaluation of the system were positive regarding the idea of non-centralized data storage that can be shared across museums. However, user interface and functions that were provided by the system require deeper investigation and clarification of its purpose. The results from the feedback were aggregated and summarized for the future work and further improvements of the system.

The rest of the thesis is divided into the following chapters:

- Chapter 2 will provide an up-to-date overview of all the related work, regarding the different aspects of the thesis. Existing approaches that are related to the characteristics of our system are presented, analyzed and discussed.
- Chapter 3 presents the design of the system and examines the requirements and user groups regarding this thesis.
- Chapter 4 is dedicated to the analysis of the system implementation details.
- Chapter 5 discusses the evaluation process and its results regarding the usability and user experience the system.
- Chapter 6 presents the conclusions in respect to the initial objectives, taking into consideration the results of the evaluation process. Furthermore, future work is discussed concerning additions that can improve the system and enhance user experience.



## 2 Background and Related Work

### 2.1 Ambient Intelligence

Ambient intelligence is an emerging discipline that brings intelligence to our everyday environments and makes those environments sensitive to us. Ambient intelligence (AmI) research builds upon advances in sensors and sensor networks, pervasive computing and artificial intelligence [2]. It refers to the future of the information society stemming from the convergence of ubiquitous computing, ubiquitous communication and intelligent user-friendly interfaces as envisaged by the European Information Society Technology Advisory Group [3].

According to [2], several definitions for Ambient Intelligence are presented in literature, including:

1. "A developing technology that will increasingly make our everyday environment sensitive and responsive to our presence.
2. "A potential future in which we will be surrounded by intelligent objects and in which the environment will recognize the presence of persons and will respond to it in an undetectable manner "
3. "Ambient Intelligence implies intelligence that is all around us "
4. "The presence of a digital environment that is sensitive, adaptive, and responsive to the presence of people "
5. "A vision of future daily life... contains the assumption that intelligent technology should disappear into our environment to bring humans an easy and entertaining life "
6. "A new research area for distributed, non-intrusive, and intelligent software systems "
7. "In an AmI environment people are surrounded with networks of embedded intelligent devices that can sense their state, anticipate, and perhaps adapt to their needs "
8. "A digital environment that supports people in their daily lives in a nonintrusive way

All the above highlight the features that are expected in AmI environments and technologies: sensitivity, adaptation, transparency, ubiquity and intelligence.

From these definitions and features that are used to characterize Ambient Intelligence, we can see how the discipline compares and contrasts with areas such as pervasive computing, ubiquitous computing, and artificial intelligence. The fact that AmI systems must be sensitive, responsive, and adaptive highlights the dependence that AmI research has on context-aware computing. Similarly, the AmI characteristic of transparency is certainly aligned with the concept of the disappearing computer. This methodological trend was envisioned by Weiser [4], who stated: "The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it".

As a conclusion of the above features and definitions comes one general definition which states that an Ambient Intelligence system is a digital environment that proactively, but sensibly, supports people in their daily lives.

From its definition, Aml has a decisive relationship with many areas in computer science. The contributing technologies are organized into five areas, shown in the Figure 1. A key factor in Aml research is the presence of intelligence. The Aml algorithm perceives the state of the environment and users with sensors, reasons about the data using a variety of AI techniques, and acts upon the environment using controllers in such a way that the algorithm achieves its intended goal.



Figure 1: Aml contributing technologies

## 2.2 Ambient Intelligence environments

In Ambient Intelligence environments, interaction techniques are user-centric. The environment learns how to react to the actions and needs of the user, and this is achieved in an unobtrusive and transparent way. Thus, the ability to learn patterns of behavior becomes an essential aspect for the successful implementation of Aml systems [5].

One of the key aspects of Ami environment is multimodality. Multimodal interfaces describe interactive system that seek to leverage natural human capabilities to communicate via speech, gesture, touch, facial expression, and other modalities, bringing more sophisticated pattern recognition and classification methods to human-computer interaction [6].

According to [7], advantages of multimodal interfaces include the following:

1. They permit the flexible use of input modes, including alternation and integrated use,
2. They support improved efficiency, especially when manipulating graphical information,
3. They can support shorter and simpler speech utterances than a speech-only interface, which results in fewer disfluencies and more robust speech recognition,
4. They can support greater precision of spatial information than a speech-only interface, since pen input can be quite precise,
5. They give users alternatives in their interaction techniques,
6. They lead to enhanced error avoidance and ease of error resolution,
7. They accommodate a wider range of users, tasks, and environmental situations,
8. They are adaptable during continuously changing environmental conditions,
9. They accommodate individual differences, such as permanent or temporary handicaps,
10. They can help prevent overuse of any individual mode during extended computer usage.

Multimodal systems enhance information visualization, replacing original WIMP (Windows, Icons, Menu and Pointers) interfaces with richer applications and accommodating users' perceptual capabilities.

### ➤ Speech

Speech was one of the first extensively used input methods. Originally, speech recognition was used to enable computer control over voice, helping impaired persons interact with their environment [8].

### ➤ Gesture

Gesture recognition is an emerging field of computer science which focuses on the problem of segmentation and interpretation of the human gestures. It can be seen as a process in which receiver will recognizes the gesture made by a user. In that sense gesture recognition provide a natural mean for human computer interaction.

Gesture recognition can be seen as identifying the motion or posture of a body part for example hands, fingers, face expressions etc. Gesture can be static or dynamic or can have the elements of both. [9]

### ➤ Context Awareness

Context aware (location-aware) systems track users and offer a wide range of information on demand. Context-aware systems are able to adapt their operations to

the current context without explicit user intervention and thus aim at increasing usability and effectiveness by taking environmental context into account. [10].

### ➤ Eye-tracking, Head Tracking and Gaze Detection

Eye-tracking and head tracking are types of interaction where the system monitors a user's eyes and head respectively in order to identify the direction pointed by the user. It is the most natural interface because users are not forced to perform any other activity except from looking at what is of his interest

## 2.3 Semantic Web

The main purpose of the Semantic Web is to drive the evolution of the current Web by enabling users to find, share, and combine information more easily. The Semantic Web is an information space used by machines rather than humans [11].

Often the terms "semantics", "metadata", "ontologies" and "Semantic Web" are used inconsistently. In particular, these terms are used as everyday terminology by researchers and practitioners, spanning a vast landscape of different fields, technologies, concepts and application areas.

The architectural model proposed by Tim Berners-Lee is used as basis to present a status model that reflects current and emerging technologies (see Figure 2) [12].

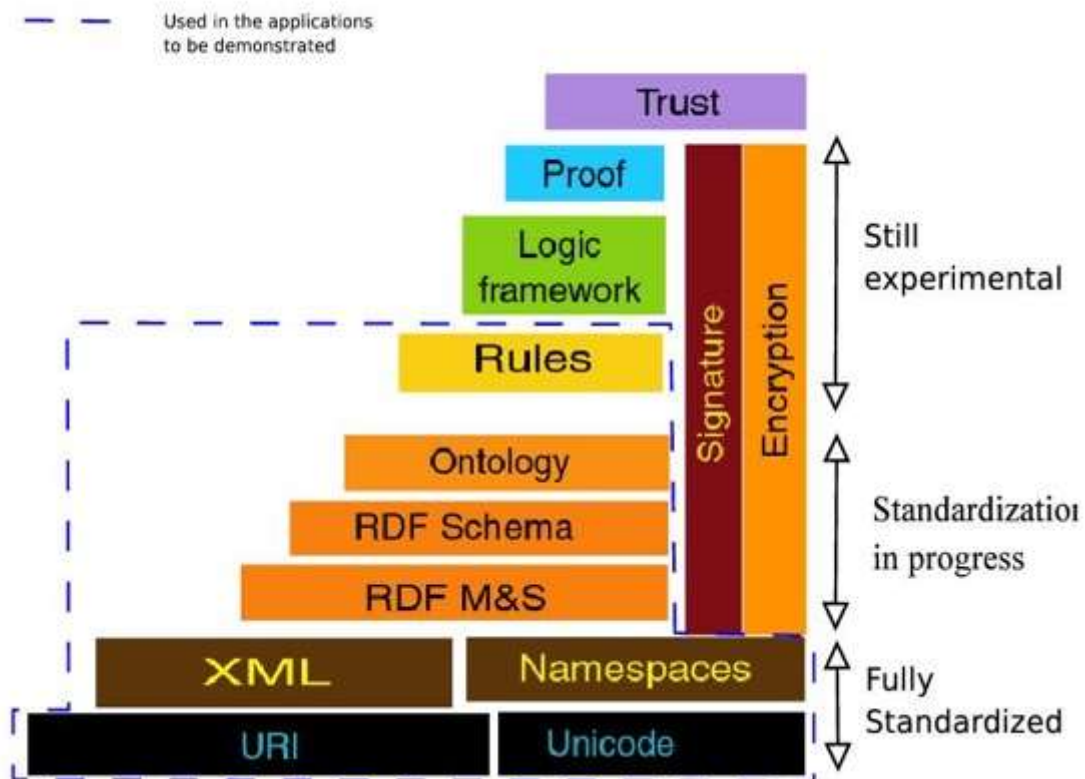


Figure 2: Semantic Web Stack

## 2.3.1 RDF

In 1997, the W3C defined the first Resource Description Framework specification. The purpose of RDF (Resource Description Framework) is to declare metadata that is machine-processable. RDF provides a mechanism to declare statements that describe resources by means of a basic data model. A statement describes an entity (resource) in terms of properties, which have values. Furthermore, an RDF statement is a subject, predicate, object triple [13].

The subject is the resource of the statement. The predicate is the property or characteristic of the subject specified by the statement (examples include creator, creation-date, or language), and the value of the property is the object.

A collection of RDF statements intrinsically represents a labeled, directed multigraph. As such, an RDF-based data model is more naturally suited to certain kinds of knowledge representation than the relational model and other ontological models.

Example of RDF graph is presented in Figure 3 [14].

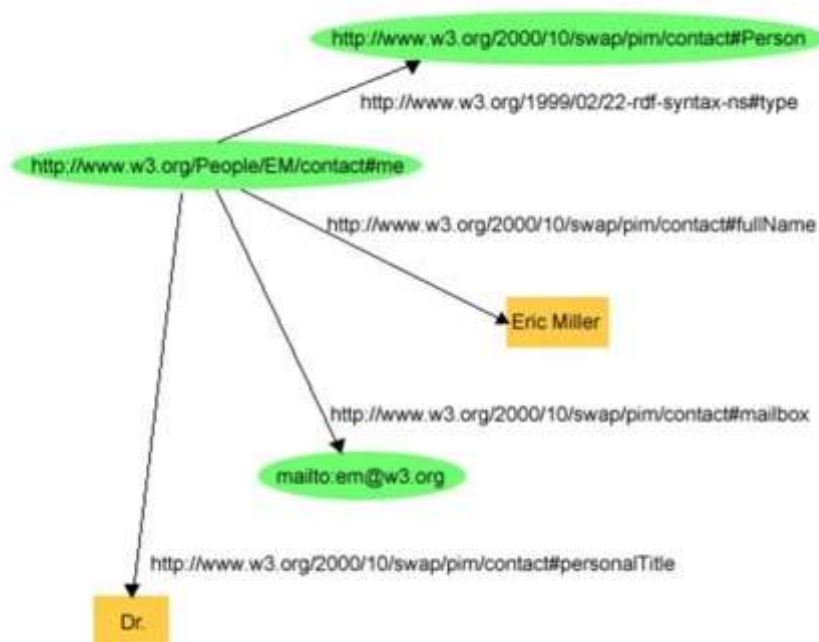


Figure 3: An RDF Graph Describing Eric Miller

## 2.3.2 OWL Web Ontology Language

According to [15] the Web Ontology Language OWL is a semantic markup language for publishing and sharing ontologies on the World Wide Web. OWL's core idea is to enable efficient representation of ontologies that are also amenable to decision procedures. It checks an ontology to see whether it's logically consistent or to determine whether a particular concept falls within the ontology. [16]

OWL uses the linking provided by RDF to allow ontologies to be distributed across systems. In this way OWL is specifically engineered for the Web and Semantic Web. It facilitates greater machine interpretability of Web content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics.

OWL provides three increasingly expressive sublanguages designed for use by specific communities of users [15]: OWL Lite, OWL DL, and OWL Full [17] provides the following description of each OWL type:

- OWL Lite: supports those users primarily needing a classification hierarchy and simple constraint features. For example, while OWL Lite supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and provide a quick migration path for thesauri and other taxonomies.
- OWL DL: supports those users who want the maximum expressiveness without losing computational completeness and decidability of reasoning systems. OWL DL includes all OWL language constructs with restrictions such as type separation. OWL DL was designed to support the existing Description Logic business segment and has desirable computational properties for reasoning systems.
- OWL Full: is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right. OWL Full allows an ontology to augment the meaning of the predefined (RDF or OWL) vocabulary. It is unlikely that any reasoning software will be able to support every feature of OWL Full.

## 2.3.3 CIDOC-CRM

CIDOC-CRM [18] is a formal ontology intended to facilitate the integration, mediation and interchange of heterogeneous cultural heritage information. It was developed by interdisciplinary teams of experts, coming from fields such as computer science, archaeology, museum documentation, history of arts, natural history, library science, physics and philosophy, under the aegis of the International Committee for Documentation (CIDOC) of the International Council of Museums (ICOM).

CIDOC-CRM is defined in an object-oriented formalism which allow for a compact definition with abstraction and generalization. The model is event centric, that is, actors, places and objects are connected via events. CIDOC-CRM is a core ontology in the sense that the model does not have classes for all particulars, as depicted in Figure 4. According to the latest specification [19] from 17/05/2018, CIDOC-CRM has more than 90 classes and 188 properties.

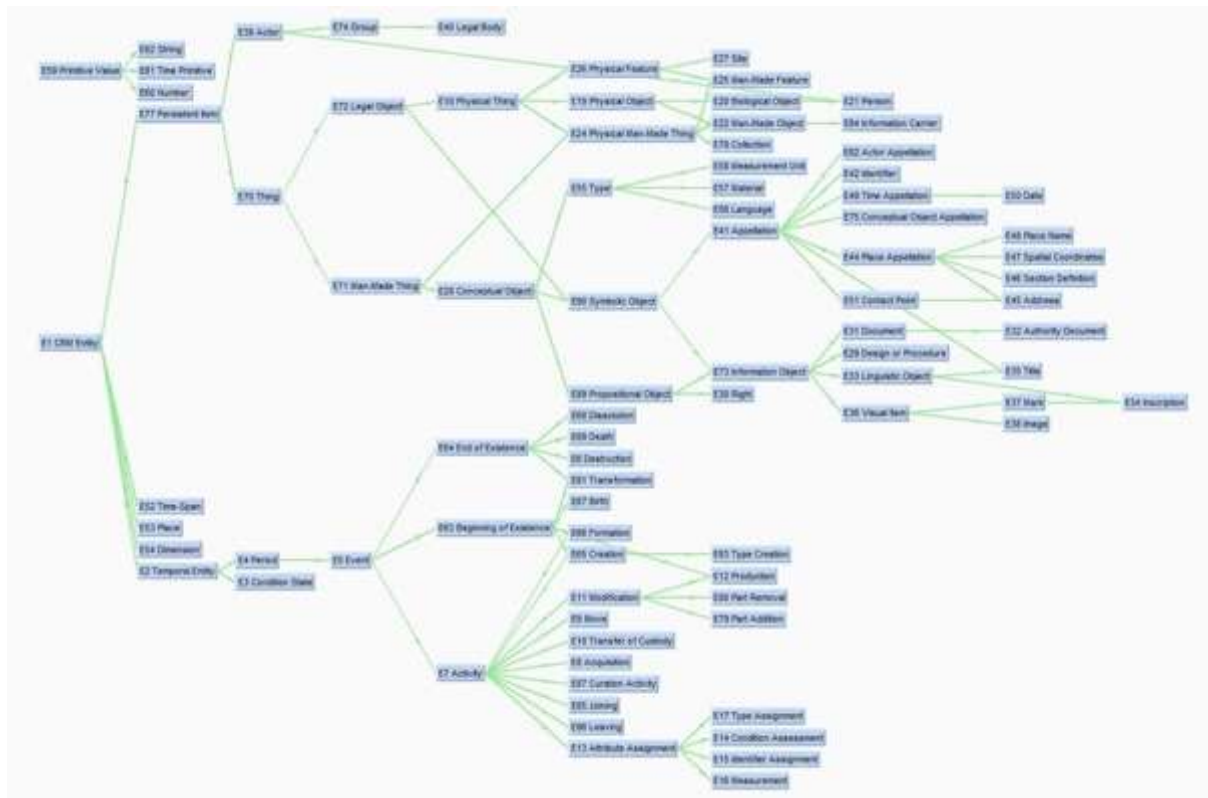


Figure 4: CIDOC CRM Class Hierarchy

The primary role of the CRM is to enable information exchange and integration between heterogeneous sources of cultural heritage information. It aims at providing the semantic definitions and clarifications needed to transform disparate, localized information sources into a coherent global resource, be it within a larger institution, in intranets or on the Internet. Its perspective is supra-institutional and abstracted from any specific local context. This goal determines the constructs and level of detail of the CRM [20].

According to [20], CRM aims to support the following specific functionalities:

1. Inform developers of information systems as a guide to good practice in conceptual modelling, in order to effectively structure and relate information assets of cultural documentation,
2. Serve as a common language for domain experts and IT developers to formulate requirements and to agree on system functionalities with respect to the correct handling of cultural contents,

3. To serve as a formal language for the identification of common information contents in different data formats; in particular to support the implementation of automatic data transformation algorithms from local to global data structures without loss of meaning. The latter being useful for data exchange, data migration from legacy systems, data information integration and mediation of heterogeneous sources,
4. To support associative queries against integrated resources by providing a global model of the basic classes and their associations to formulate such queries,
5. It is further believed, that advanced natural language algorithms and case-specific heuristics can take significant advantage of the CRM to resolve free text information into a formal logical form, if that is regarded beneficial. The CRM is however not thought to be a means to replace scholarly text, rich in meaning, by logical forms, but only a means to identify related data.

## 2.4 Related Work

At current moment, there are many related efforts focusing on how modern technologies can improve user experience in a museum context. In many of those, researchers used to analyze the impact of sensors, augmented reality and virtual assistantship on the quality of user experience (UX) and perception. Currently, many museums already offer some of those technologies in order to improve user experience by using some special devices or sensors. In the following section, there will be an overview of already existing solutions.

### 2.4.1 Indonesian museum information system

The system presented in [21] was developed in order to provide an interactive museum information for visitors, especially museums in Indonesia. It provides an Android application (Figure 5) that can compose information from all museums in Indonesia.

The user can see the complete information about many museums in Indonesia. Information that provided such as general information about museums, opening/closing hours, museum location using Google Map, image gallery from the objects in the museum, etc.



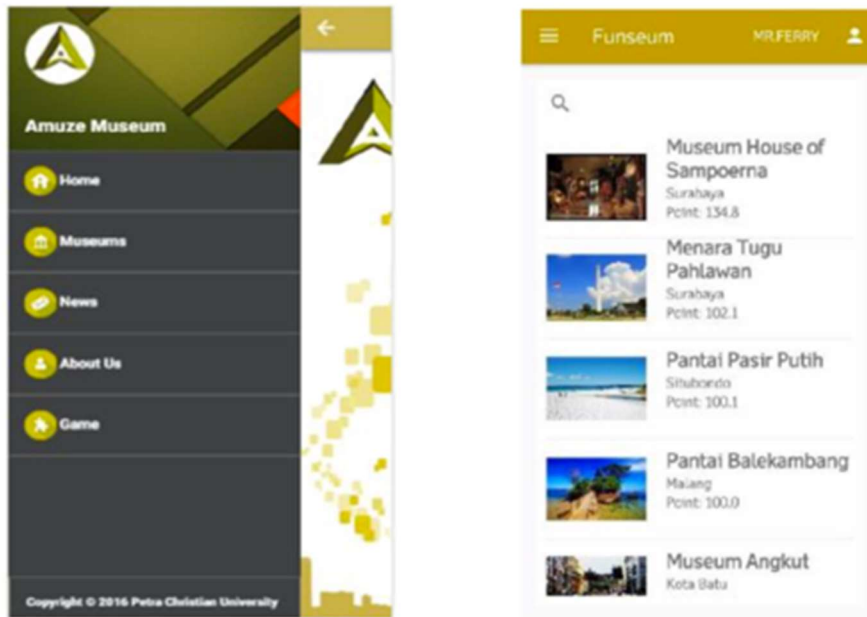


Figure 5: Android Application

In addition, the application also tries to use QR code to provide additional information while increasing the interaction for museum visitors. QR code tags also placed on the objects or in the exhibition room in the museum (as depicted in Figure 6). Visitors can scan the QR code tag using camera on the mobile phone and QR code reader. Then the application will display additional information that provided (according the code on the QR code tag) such as details information article about the object or additional images, and also sound or video that could enrich the existing information. By this way visitors are expected to get more detailed information from objects or rooms, where this information can be displayed in a more interactive.

Museum administrator also could place as many as detail information not only on text description like usual but also in sound, video, animation, website link, games, etc. Museum administrator could entry any details of museum information such as a museum general information, building and objects picture, opening/closing hour information, etc. from website administrator.



Figure 6: QR Code Tag on Bicycle Miniature Display

## 2.4.2 The iMuse Mobile Tour

The iMuse Mobile Tour uses UHF RFID tags attached to museum's showcases to link exhibits to multimedia information [22]. It supports multilingual and multi-audience content for free-style and pre-designed tours. The system is enhanced to support self-defined tours and interactive educational games.

The operation of the system is based on the handheld's integrated pistol grip. On first launch, visitor is provided with a short video presentation that illustrates the handheld's usage. Visitor has to setup his profile by choosing language and audience type. The system can provide simple descriptions that are made for kids and more detail information for experts. In order to get an information about exhibit, visitor has to pull pistol trigger. The information about exhibit includes a photo and a short description. Description can also be perceived using a vocal message.

During free style tours, visitors are able to explore the museum in their pace and retrieve information about any exhibit attracting their interest. Pre-defined tours are composed by the archeologists and specialize to a specific thematic entity of the museum. Visitors can also create their own tours using certain criteria.

In order to support learning in groups, a context-aware service was pro-posed by the following project. This service utilizes iMuse Mobile Tour and visitors' personal smartphones to deliver personalized content to each visitor. Each group member can personalize language and audience type by changing his profile.

The client application consists of three modules, the RFID Tag Identification module, the Request Information module and the User Interaction Control module. Figure 7 depicts the architecture of iMuse Mobile Tour that follows client-server architecture.

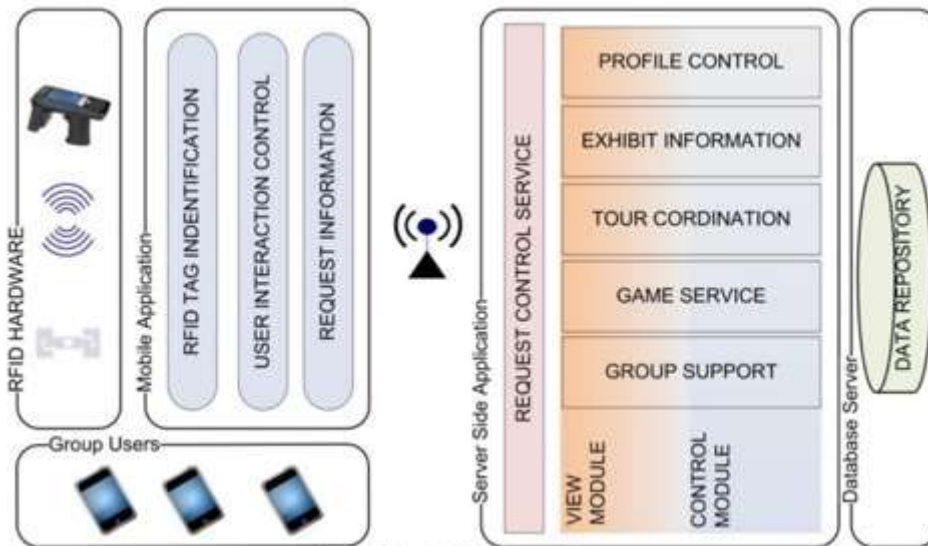


Figure 7: iMuse Mobile Tour architecture

### 2.4.3 A Multimodal Mobile Museum Guide for All

The Multimodal Mobile Museum Guide for All applies scan and tilt paradigms, which combines multiple modalities - gestures, physical selection, location, graphical and voice [23]. It also can support different levels of accessibility such as:

- Basic Navigation: this option is aimed at allowing user to use the application through the classical buttons/pen of the PDA
- Navigation with Audio Feedback: in this option, which represents the minimal level of accessibility and can be considered helpful for both partially-sighted and blind users, the user's navigation is facilitated by a vocal feedback that is activated when necessary.
- Navigation using Tilt: in this modality only, the tilt modality is active (the voice is not used). Therefore, there might be tilt movements to which no action is associated anymore. The navigation is carried out by using small inclination movements of the PDA, which substitute the use of PDA pen and buttons.
- Navigation with Audio and Tilt: in this option the tilt is used together with the voice. This type of navigation allows the users to use the museum guide and navigate within it using the tilting options and, at the same time, to have a vocal feedback from the application. This configuration option is especially aimed at supporting the blind.

The client device is equipped with a Compact Flash (CF) RFID reader with small-sized antenna to read tags and a 2D accelerometer able to recognize small movements of the handheld device.

## 2.4.4 Location-Aware Tour Guide Systems in Museum

This system [24] combine wireless networking, established content in digital archives for museums, interior locating technology and geographical information system. It consists of four components, the client (PDA), the Location Position Agent, the Context-Aware Agent and the Content Server.

The client is responsible to collect the power signature information and to transmit it to the location agent. Additionally, the client acts as a terminal to display the received messages.

The location position agent estimates the position of the PDA by combining the strengths of 802.11b/g wireless access signals. Afterwards, neural networks are used to estimate position from RSSI signature information received from the access points. The location position agent stores results in the content server.

The Context-aware agent is responsible to send data to the appropriate recipients. The information about exhibit can be presented in a various multimedia formats such as images, audio, video or text. The usage example is displayed in Figure 8.

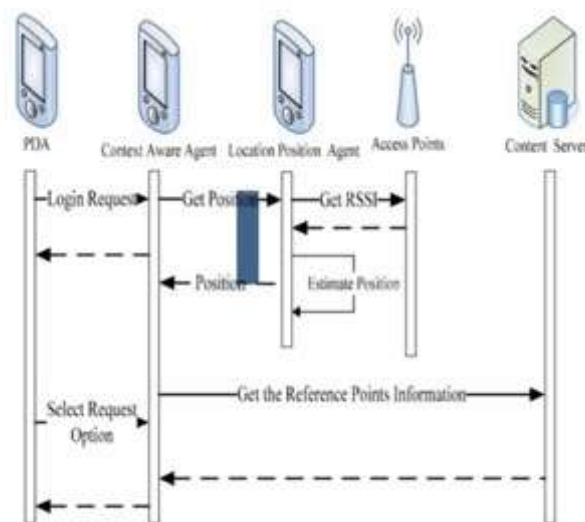


Figure 8: Location-Aware Tour Guide System

## 2.4.5 Interactive Museum Guide

The following system [25] is an interactive, image-based museum guide. Visitor is also provided with hand-held device that allows the visitor to take a picture of an object of interest from any position. The device features a touchscreen, a webcam and a Bluetooth receiver. The system can provide voice description of exhibits.

The system contains two different modules: image-based object-recognition module and automatic exposition room detector using Bluetooth. It is using object recognition with Bluetooth to identify possible exhibit and to provide basic information about it. Example of object recognition process is displayed in Figure 9.



Figure 9: Interface of the object recognition application

## 2.5 Discussion and Motivation

All provided solutions are context-aware systems and most of them share common input. The sources are either provided by smartphone or other external sensors. PDA devices and RFID tags were used almost in all previously reviewed solutions.

Important issue is that these solutions provide information about exhibits with a limited personalization support or its absence at all. Some of previously mentioned solutions such as **Indonesian museum information system 2.4.1** and **Interactive Museum Guide 2.4.5** provide multimedia information about exhibits using interactive approach and modern smart devices, but the provided information doesn't take into account users' preferences or past experience in current or other museums.

Museums are the environments where visitors receive a wealth variety of information from many sources. New data and correspondence innovations encourage that the attributes and related data elements about fine arts can be offered in a more justifiable, appealing and simple approach to the visitors. In this sense, the context information ends up with the goal to customize the experience for each visitor at each minute.

The goal of this thesis is the creation of a system that attempts to provide the ability to maintain museum's collections, interactive multimedia information about exhibits and to present it into multimodal form according to personal preferences of each user. To achieve

this goal, system integrates best practices and technologies from **Ambient Intelligence** and **Semantic Web**.

The designed system extends available semantic knowledge about artifacts/exhibits by implementing a subset of CIDOC CRM ontology model. This knowledge is used to provide personalized and interactive multimedia description of exhibit in accordance with the previous experience of the user.

## 3 System Design

In order for the system to be designed, several procedures were followed to define basic construction requirements. First, brainstorming sessions were performed to develop the exact characteristics of the system. Brainstorming sessions were the first step to produce ideas regarding to the scope of the designed system and moreover, to create a general overview of it. Furthermore, focus groups were scheduled in order to discuss in depth the detailed properties that the system would contain.

Appointments with the end users took place to define the basic attributes that are necessary to offer for the presentation of the system irrespective of its context. The interviews also defined the properties and features that an end user would expect to have in the system. The age of the people who participated in the interview varied from 25 to 50 years old, which is the target group that the system is intended to be used by. The results of the interviews were truly helpful, as a wide variety of features that users would expect system to have.

Once an approximate image of the system was formed, mockups were designed to determine an initial look of the system. This approach was particularly helpful in order to identify weaknesses to the surface and allow the refinement of the design.

Finally, throughout the development process of the system each new component was constantly evaluated.

### 3.1 User groups and requirements

This chapter describes user groups and the requirements that were formed during the design of the system. Firstly, user groups were identified according to the context of use. Secondly, functional and non-functional requirements were determined for each user group.

### 3.1.1 User groups

After analysis of the system, two user groups were formed as the potential users of the system. The users were grouped as museum visitors and curators. Each of this group has its own goals and requirements to the system that are described in the current chapter.

#### 3.1.1.1 UG1 – Museum Visitors

The museum visitor is a typical user that sometimes visits the museum. User can be of any age or education. It is expected that user is familiar with a smartphone and is using it in his everyday life.

#### 3.1.1.2 UG2 – Museum Curators

The user from the following group is expected to work in a museum and to know in details the structure of the museum and location of the exhibits

### 3.1.2 User requirements

Functional requirements are related to the clarification of the functionalities that the system must do, whereas non-functional requirements express the quality of the system. Functional requirements are presented according to the user group.

#### 3.1.2.1 Functional requirements

##### 3.1.2.1.1 User group – Museum Visitors

- Visitor should be able to authorize using his social network account
- Visitor should be able to specify his preferences
- Visitor should be able to identify exhibit
- The multimedia description of identified exhibit should be available to the visitor
- Visitor should be able to explore the provided multimedia description in various modalities

##### 3.1.2.1.2 User group – Museum Curators

- Curator should be able to register and authorize with a system
- The list of available collections and exhibits should be available to the visitor
- System should provide functions to add, remove, update museum collections
- System should allow curator to add, remove or update exhibits for each collection
- Curator should to be able to create, update and remove interactive multimedia descriptions of exhibit or group of exhibits

### 3.1.2.2 Interactive Requirements

The interactive requirements of the system include:

- The ability to retrieve input from a wide set of sources that describe semantic information
- Interaction should be achieved through the use of natural means of interaction.
- Compatibility with a variety of means of interaction and support of multimodal interaction, using different interaction techniques

### 3.1.2.3 Non-Functional Requirements

The non-functional requirements refer to the characteristics of the system regarding the accomplishment of the user's goals and include the following:

- The system should be self-explaining. Ideally, it should be easy to use for the first time without any special training or at least after three minutes of training
- The interaction techniques should be intuitive and easy to memorize
- The system should be responsive to user actions
- The Interaction with system should be precise
- The overall system settings should be stable and robust
- Mobile component of the system should be compatible with all modern smartphone based operating systems
- System web panel for curators should be compatible with all modern desktop and mobile browsers
- System should provide authorized and secure access

## 3.2 Design

### 3.2.1 Schema / High level architecture

The designed system represents itself as a convergence of smartphones with Ambient Intelligence, Semantic Web and Interactive Multimedia Presentation of the information. Various technologies stemming from different research domains such as RDF, .NET Core, ReactJS and Unity 3D, are employed by the designed system. Regarding the system implementation, a service-oriented architecture was conceived, along with the reciprocity of various subsystems. The various components were designed and built based on a generic architecture, in order to support knowledge management annotation and extraction, alternative ambient interaction techniques, personalization and adaptation.

An overview of the applied technologies will be provided during the current chapter. The architecture schema of the developed system is illustrated in Figure 10.



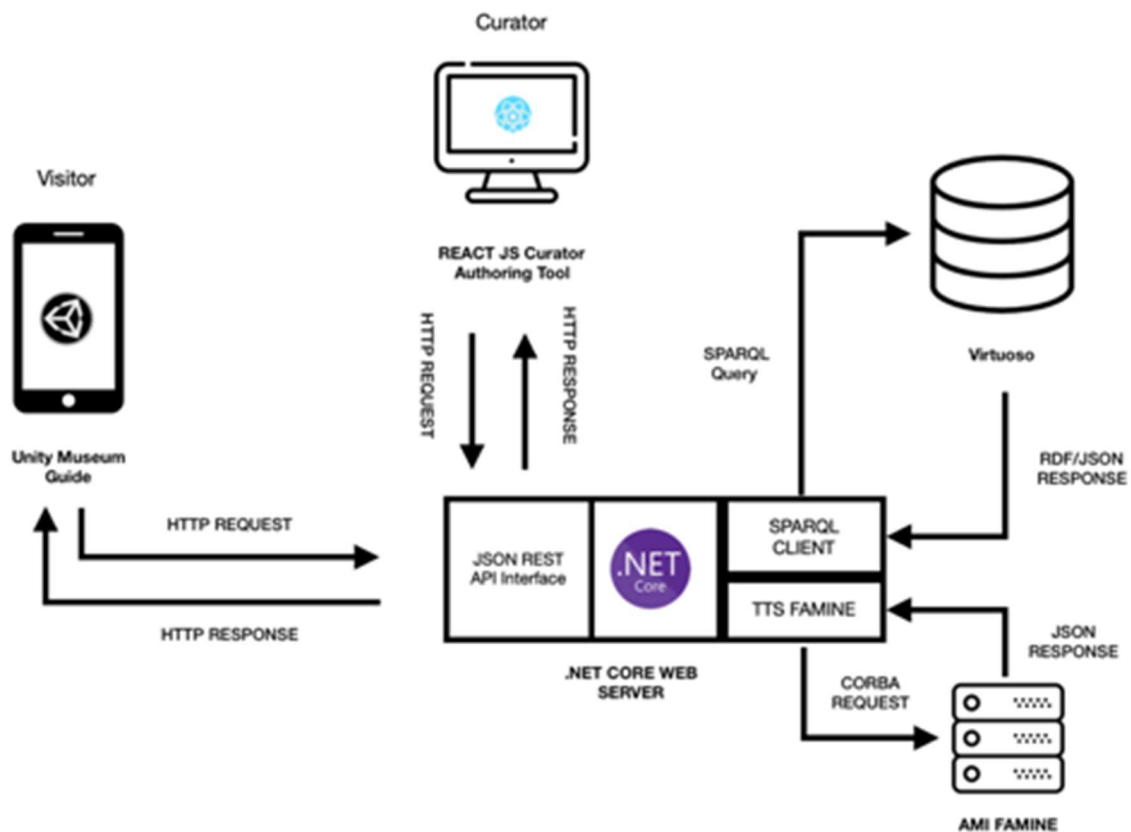


Figure 10: System Architecture

### 3.2.2 Design of the Internal Components

Internal components are the most important parts of the system. They are responsible for low level business logic and data processing. The basic description of each internal component is presented in the following section.

#### 3.2.2.1 ASP.NET

ASP.NET [26] is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

It is the most important core component of the system that is responsible for business logic of the system. It handles all the communication processes with Virtuoso and Famine TTS components using their own specific API and provides unified REST API endpoint. The provided endpoint is using JSON format for data interchange.

### 3.2.2.2 OpenLink Virtuoso

OpenLink Virtuoso is a cross-platform universal server that acts as a virtual database engine that combines the functionality of a traditional RDBMS, ORDBMS, virtual database, RDF, XML, free-text, web application server and file server functionality in a single system. The software has been developed by OpenLink Software with Kingsley Uyi Idehen and Orri Erling as the chief software architects. It also has a built-in HTTP server that provides a DAV repository, SOAP and WS\* protocol end points and dynamic web pages in a variety of scripting languages [27].

Virtuoso is used as a default RDF storage, but system doesn't rely on any Virtuoso specific feature. Therefore, any RDF 1.1 compatible storage provider can be used as a transparent replacement.

### 3.2.2.3 Ontology

Current subsection contains information about the designed ontology schema. Protégé [28] was used as a tool to design the ontology. It is an open source ontology and knowledge base editor. Entities inside Protégé are organized in classes containing properties. CIDOC-CRM ontology was used to maintain data about collections and exhibits.

For the current system, ontology was created based on **CidocCRM** subset. This schema provides information about classes, properties and connections that are used in the system. Connections between visitor and exhibits are used to identify similar exhibits and relevant collections. By using ontology schema and correctly established connections, SPARQL query language can fetch data using relational algebra operators to infer nested properties or subclasses.

The system is using the default reasoner, provided by OpenLink Virtuoso (3.2.2.2), to resolve direct or nested connections between entities using inference rules and SPIN vocabulary. Classes and properties that were designed for the system are listed in Table 1 and Table 2.

Table 1, System ontology user components

Curator	Visitor	Preferences	History
<i>email</i>	<i>facebookId</i>	<i>targetGroup</i>	<i>did_explored</i>
<i>password</i>	<i>has_preferences</i>	<i>fontSize</i>	
<i>firstName</i>	<i>has_history</i>	<i>voice</i>	
<i>lastName</i>		<i>lang</i>	

Table 2, System ontology, exhibit description components

Description Flow	Description	Block	Condition
<i>label</i>	<i>label</i>	<i>position</i>	<i>type</i>
<i>cover</i>	<i>cover</i>	<i>contentType</i>	<i>value</i>
<i>date</i>	<i>intro</i>	<i>content</i>	
<i>hasDescription</i>	<i>date</i>	<i>hasCondition</i>	
	<i>jsonStruct</i>	<i>hasExhibit</i>	
	<i>hasBlock</i>		

The ontology schema of the system can be explored in Figure 11.

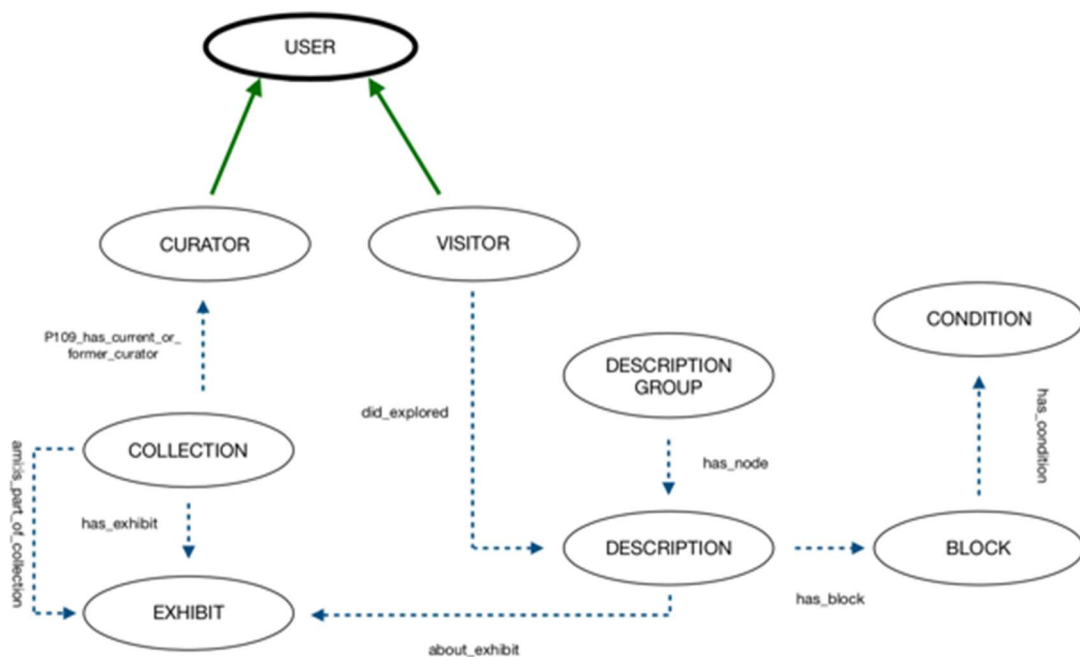


Figure 11: Ontology

### 3.2.2.4 Famine TTS Service

**Famine TTS Service** is a proxy service that was developed as a subcomponent of the web server system. Main purpose of this component is to provide Text-To-Speech functionality using system REST API interface.

Text-To-Speech service is a part of **Aml Middleware service** and it is developed and hosted by Ambient Intelligence Lab of the Institute of Computer Science (ICS) of the Foundation for Research and Technology Hellas (FORTH). The structure of Aml Middleware can be explored Figure 12.



Figure 12: The structure of Aml Middleware

The designed component takes responsibility to establish and maintain connection with **Aml Middleware service** using CORBA interface. It encapsulated the work with CORBA interfaces and exposes two methods:

- GetVoices, method doesn't take any arguments and returns back the list of available voices
- TransformTextToSpeech, method has to be provided with a text argument. Voice argument is optional. It returns an audio byte stream that is also accompanied with phonemes and visemes.

By using these methods, any client application can access Text-To-Speech service without implementing any CORBA specific logic.

### 3.2.3 Design of the Curator Authoring Tool

Curator Authoring Tool is an advanced web application that should provide curator basic functionality that is described in 3.1.1.2. It is also required that it should be fast, efficient and capable to work on any modern browser. Moreover, it also has to be adaptive across various displays, including the mobile.

In order to fulfill these functional and non-functional requirements the React JS library was selected as a modern and advanced technology for creating web applications.

Besides, in order to provide modern and attractive user interface controls, Material UI library [29] was used on top of ReactJS.

#### 3.2.3.1 React JS

React JS [30] - an open source library for building rich user interfaces. It is maintained by Facebook and the community of individual developers and companies. React can be used

as a base in the development of single-page or mobile applications. As written in [31], React provides a technology called the virtual DOM.

The virtual DOM is a fast, in-memory representation of the real DOM, and it's an abstraction that allows developers to treat JavaScript and DOM as if they were reactive. Shortly, the way virtual DOM works can be explained as:

- Whenever the state of our data model changes, the virtual DOM and React will rerender our UI to a virtual DOM representation.
- React calculates the difference between the two virtual DOM representations: the previous virtual DOM representation that was computed before the data was changed and the current virtual DOM representation that was computed after the data was changed. This difference between the two virtual DOM representations is what actually needs to be changed in the real DOM.
- React updates only what needs to be updated in the real DOM.

Aside from React JS, Redux state container was used to maintain global application store. Redux is a predictable state container. This means that the cause of state change can easily be identified. Redux uses a single store instead of many different stores like other flux inspired libraries do. This is often called “single source of truth”, as the store is the only place where state can be accessed from [32].

As applications become more complex, they have to manage more state. Redux makes state changes predictable. It achieves this by setting several rules. The first of which being that state must be immutable. Secondly the information about state changes must be contained by actions. And thirdly the use of reducers, which are pure functions, specifying how actions transform the state. References to multiple states can be stored, as the state in a Redux application is immutable [33].

### 3.2.3.2 Material UI

Material Design is a design language developed in 2014 by Google and is very popular for web and mobile applications. Material Design is inspired by the physical world and its textures, including how they reflect light and cast shadows. Material surfaces reimagine the mediums of paper and ink.

### 3.2.4 Design of the Mobile App

Mobile application has to support all modern mobile operating systems such as iOS, Android, Windows Phone and etc. Moreover, it also should provide rich multimedia and 3D capabilities.

The only platform that could satisfy all these requirements was Unity 3D. It was developed by Unity Technologies [34] and it is a multi-platform integrated game development tool. It allows developers to create 3D video games, visual architectures, real-time 3D animations and other types of interactive content. It is a fully integrated professional game engine. Unity supports deployment to multiple platforms. By starting a project in Unity3D (Unity) developers are in control of application delivery to mobile devices, web browsers, desktops, consoles and have a set of tools to outline the specification of texture compression as well as screen resolution settings. It currently provides development for iOS, Android, Windows, Linux, Flash, web browsers and others. The Unity supports the developers with features such as rendering, scripting, asset tracking, platforms and physics.

# 4 Implementation

This chapter contains details regarding to the system implementation. It provides description of each subcomponent.

## 4.1 Backend Service

Backend service is the central component of the implemented system. It was created using ASP.NET core, see section 3.2.2.1. It consists from various components that are responsible to retrieve, process and expose the data. The core components of this service are: a) Application Provided Interface, b) Facebook Client and c) SPARQL client. The description of each subcomponent is given in the current section.

### 4.1.1 Application Provided Interface (API)

API is the most important part of any modern Web Service system. It is a set of subroutine definitions, communication protocols, and tools for building software. In general terms, it is a set of clearly defined methods of communication among various components.

In the following thesis, an **API** interface was created to provide data exchange across various parts of the system. Various data formats can be used to exchange the data like **JSON** (JavaScript Object Notation) or **XML** (Extended Markup Language).

The designed system is using **JSON** for data exchange because it is faster and more efficient for data transfer across the network.

The list of available API methods is presented below (Table 3). Each of this method can be accessed as **HOST: PORT / API / Controller / Method**.

Table 3, API Methods

Controller	Request Method	Path
Curator Controller		
	POST	Authorize
	POST	Authorize
Exhibits Controller		
	GET	Get
	POST	Create
	GET	Search
	Delete	Delete

	PUT	Update
Collections Controller		
	GET	Get
	POST	Create
	DELETE	Delete
	PUT	Update
Description Management Controller		
	GET	GetDescription
	POST	CreateDescription
	DELETE	DeleteDescription
	PUT	UpdateDescription
	GET	GetFlows
	POST	CreateFlow
	DELETE	DeleteFlow
	PUT	UpdateFlow
	GET	GetFlowStructure
	PUT	UpdateFlowStructure
Visitor Controller		
	POST	Login
	GET	Preferences
CidocCRM Controller		
	GET	Get
	POST	Create
	PUT	Update
	DELETE	Delete
	GET	Search
Text-To-Speech Controller		
	GET	Voices
	POST	TransformTextToSpeech
Visitor History Controller		
	GET	GetDescription
	GET	GetSimilar



Tags Controller		
	GET	Get
	POST	Create

### 4.1.2 Facebook Client

In order to store the usage history of the visitor, the system has to define connections between the user and the collected usage data. In order to establish this connection, Facebook ID of the user was used.

Facebook client is an important component of a backend service that is responsible to authorize the visitor. System is using this component to confirm user identity via provided Facebook access token. This token is used only once during the authorization process and isn't stored by the developed system. The user privacy is concern and the system doesn't request, fetch or store any personal information except Facebook identifier that it is public and available to anyone.

This component is a C# singleton class that provides access to various Facebook Graph API methods. Facebook's Graph API presents a simple, consistent view of the Facebook social graph, uniformly representing objects in the graph (e.g., people, photos, events, and pages) and the connections between them (e.g., friend relationships, shared content, and photo tags) [35].

### 4.1.3 SPARQL Client

As was mentioned in [27], the designed system stores its data using Resource Description Framework format (RDF). In order to retrieve or manipulate this data, system is using a semantic query language that is called SPARQL. According to [36], SPARQL is an official W3C Recommendation to work with the RDF data.

SPARQL allows for query to consist of triple patterns, conjunctions, disjunctions and optional patterns.

One of the most important components that was developed for the system is the SPARQL Client. It was created as a response to increased complexity of performed queries. SPARQL provides great flexibility and very powerful features, but it is verbose, hard to read and error-prone. Developer has to take care not only about the valid query structure, but also to escape invalid symbols inside the triple nodes.

This component provides various classes that solve abovementioned problems. One of the most useful features is the class named **TripleSerializer**. It can be used to convert a C# object into the list of triples or vice versa. This feature is implemented on top of C#

reflection and meta-programming capabilities. An example of a valid class that can be used with abovementioned serializer is presented in Figure 13

```
[OwlClass("crm:E78_Collection")]
public class Collection: OwlObject
{
    [Predicate("crm:P102_has_title")]
    public string Title;

    [Predicate("crm:P3_has_note")]
    public string Note;

    [Predicate("ami:cover")]
    public string Cover;

    [Predicate("ami:date")]
    public string Date;

    [Predicate("crm:P109_has_current_or_former_curator", typeof(UriNode))]
    public string CuratorUri;
}
```

Figure 13: Class that can be serialized into the triples

The output of **TripleSerialized** can be examined in Figure 14.

```
{
  _:blank_Collection a crm:E78_Collection.
  _:blank_Collection crm:P102_has_title ?Collection_Title.
  _:blank_Collection crm:P3_has_note ?Collection_Note.
  _:blank_Collection ami:cover ?Collection_Cover.
  _:blank_Collection ami:date ?Collection_Date.
  _:blank_Collection crm:P109_has_current_or_former_curator ?Collection_CuratorUri.
  _:blank_Collection rdf:resource ?Collection_Uri.
}
```

Figure 14: Serialized collection object

Moreover, classes that simplify creation of basic queries such as **INSERT**, **DELETE**, **SELECT** and **ASK** were also implemented. Example of each query usage and its raw representation is presented furthermore.

#### 4.1.4 INSERT

In Figure 15 is displayed an example of **Insert** query. Object is automatically serialized into the triples which are inserted into the specified graph.

```
var curator = new Curator {
    Password = Crypto.HashPassword(model.Password),
    Email = model.Email,
    FirstName = model.FirstName,
    LastName = model.LastName
};

var status = await _sparqlClient.Insert(curator, GraphUri.Curators);

if (status == false) {
    return this.GenerateError("unknown", "Failed to create new curator record");
}
```

Figure 15: Insert query

The generated insert query can be explored in Figure 16. The instance URI is automatically generated based on the target graph and URL dangerous symbols are escaped.

```
WITH <https://ami.ics.forth.gr/museum/ima:curators/> Insert {
  <https://ami.ics.forth.gr/museum/ima:curators/CURATOR/07632812-52ce-4001-a23c-5a015d5b5afe> a ami:Curator.
  <https://ami.ics.forth.gr/museum/ima:curators/CURATOR/07632812-52ce-4001-a23c-5a015d5b5afe> ami:email "curator%40mail.com".
  <https://ami.ics.forth.gr/museum/ima:curators/CURATOR/07632812-52ce-4001-a23c-5a015d5b5afe> ami:password "AQAAAAEAACcQAA".
  <https://ami.ics.forth.gr/museum/ima:curators/CURATOR/07632812-52ce-4001-a23c-5a015d5b5afe> foaf:firstName "Denis".
  <https://ami.ics.forth.gr/museum/ima:curators/CURATOR/07632812-52ce-4001-a23c-5a015d5b5afe> foaf:lastName "Sancov".
}
```

Figure 16: Insert raw query

## 4.1.5 DELETE

Delete construction can be visualized in Figure 17. The following code shows the declaration of triple components and the way how filters can be applied to the condition clause.

```
var uri = new UriNode(model.Uri);
var property = new Var("property");
var obj = new Var("object");

var status = await _sparql.Delete(query => query
  .With(GraphUri.NarrativeFlow)
  .Quads(pattern => pattern.PushTriple(uri, property, obj))
  .Where(pattern =>
    {
      pattern
        .PushTriple(uri, property, obj)
        .PushPattern(Filter.In(property, new[] {"ami:struct", "ami:hasNode"}));
    }
  ));
```

Figure 17: Delete query builder

The raw query is displayed in Figure 18.

```
WITH <https://ami.ics.forth.gr/museum/ima:narrative-flow/> Delete
{
  ?NodeVar ?predicate ?value.
}
WHERE
{
  ?NodeVar ?predicate ?value.
  ?NodeVar ami:tree <https://ami.ics.forth.gr/museum/ima:narrative-flow/NARRATIVE/3b59df5b-d933-496b-a6d8-101a41f19940>.
}
```

Figure 18: Delete raw query

## 4.1.6 SELECT

Select construction is presented in Figure 19. It shows an example of how the data can be fetched and filtered. Moreover, the returned data is automatically transformed into the list of objects.

```

return await _sparqlClient.SelectAll<Exhibit>(
    query => query
        .FromGraphs(GraphUri.Collections)
        .Where(pattern => {
            var triples = TripleSerializer.TriplesFromObject<Exhibit>();
            pattern.PushTriples(triples);

            if (!string.IsNullOrEmpty(queryString))
            {
                var title = OwlObjectUtils.VariableFor<Exhibit>(nameof(Exhibit.Title));
                pattern.PushPattern(Filter.StringMatch(title, queryString));
            }
        })
        .Limit(limit)
        .Offset(offset)
);

```

Figure 19: Select query builder

The raw query representation can be examined in Figure 20.

```

SELECT * FROM <https://ami.ics.forth.gr/ima:collections/> WHERE {
  {
    _:blank_Exhibit a crm:E22_Man-Made_Object.
    _:blank_Exhibit crm:P102_has_title ?Exhibit_Title.
    _:blank_Exhibit crm:p3_has_note ?Exhibit_Note.
    _:blank_Exhibit ami:date ?Exhibit_Date.
    _:blank_Exhibit crm:P138i_has_representation ?Exhibit_Cover.
    _:blank_Exhibit ami:is_part_of_collection ?Exhibit_CollectionUri.
    _:blank_Exhibit rdf:resource ?Exhibit_Uri.
    FILTER(regex(str(?Exhibit_Title), "a", "i"))
  }
} LIMIT 20

```

Figure 20: Select raw query

#### 4.1.7 ASK

Ask query is used to check the existence of a particular resource by specifying condition clause. The query that checks the existence of the curator by email is presented in Figure 21. The generated raw query can be explored in Figure 22.

```

var exists = await _sparqlClient.Ask<Curator>(GraphUri.Curators,
    obj => obj.Email = model.Email
);

```

Figure 21: Ask query builder

```

ASK WHERE {
  GRAPH <https://ami.ics.forth.gr/museum/ima:curators/> {
    _:blank_Curator a ami:Curator. _:blank_Curator ami:email "denis.sancov%40gmail.com".
  }
}

```

Figure 22: Ask raw query

## 4.2 Curator Authoring Tool

This component was written using React JS framework and provides a tool that can be used by a museum curator to maintain data about museum collections and to create multimedia descriptions about them. An overview of provided functionality is provided in further subsections.

### 4.2.1 Collections

The following page (Figure 23) allows curator to explore the list of already created collections. The collections are represented using grid layout and are capable to layout according to the size of screen. Each collection is represented with a cover image, a title and other properties. Collections could be filtered according to filter panel. Curator can present a form to create a new collection by pressing “CREATE NEW” button.

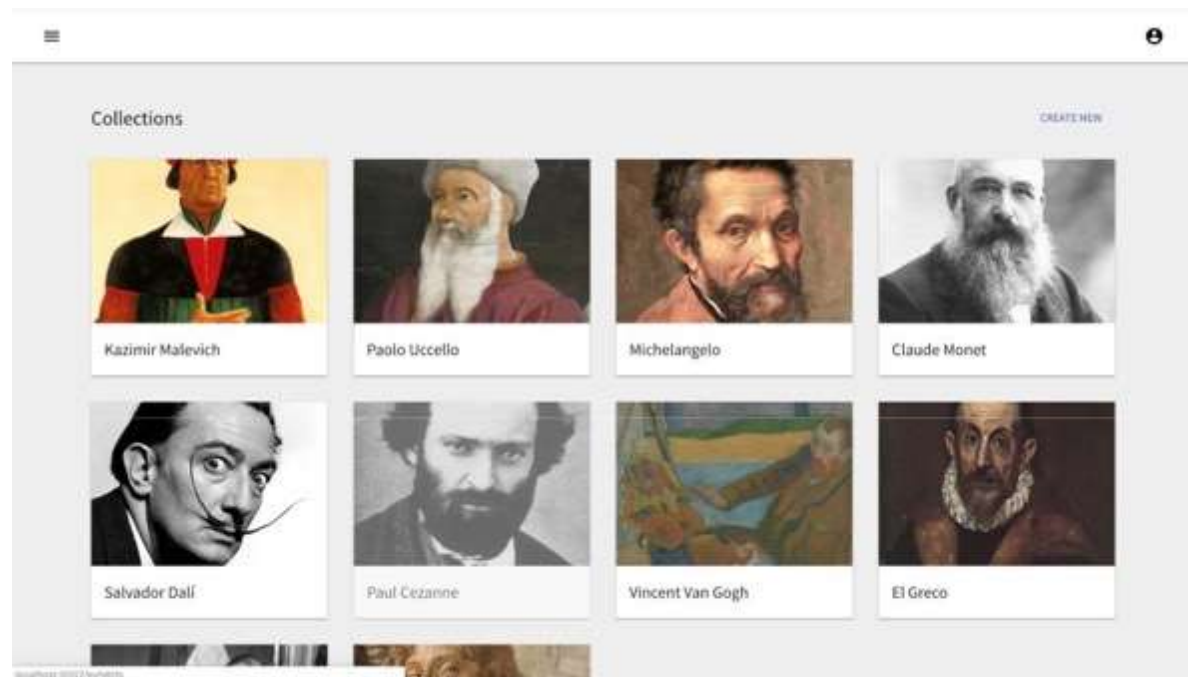


Figure 23: Collections Page

## 4.2.2 Exhibits

The following page (Figure 24) allows the curator to explore the list of available exhibits. Exhibits are grouped by their collection. The title of collection is displayed below the page title.

The exhibits are visualized using grid layout and are capable to layout according to the size of screen. Each exhibit is represented with a cover image and a short description. Curator can add a new exhibit to collection by pressing “CREATE NEW” button.

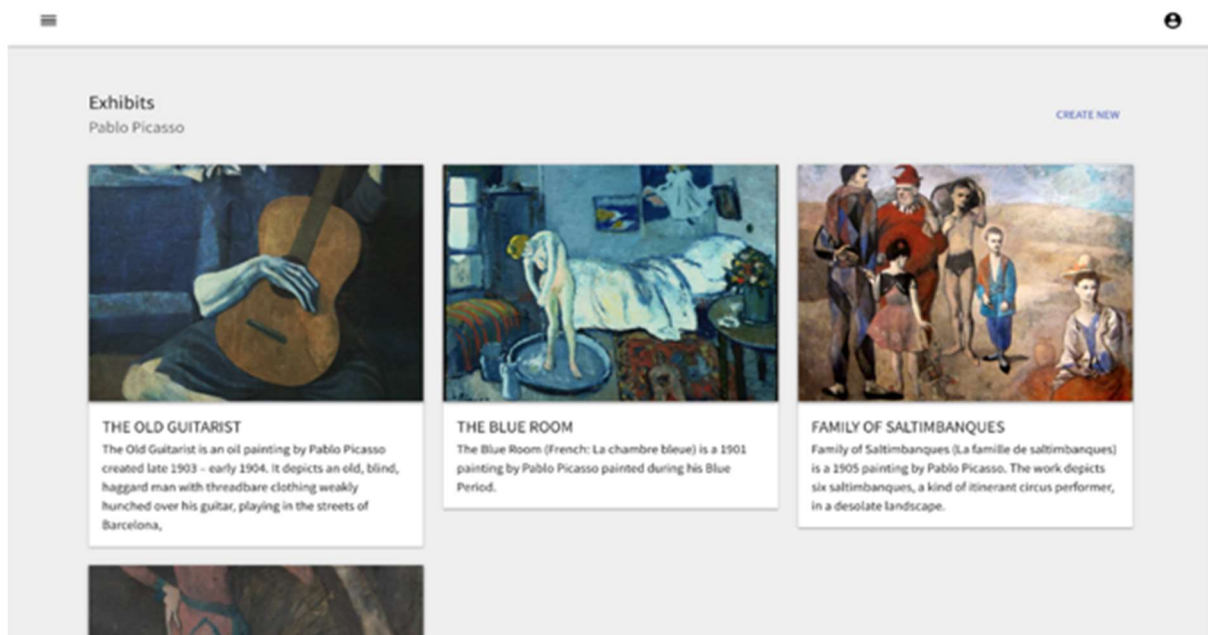


Figure 24: Exhibits

## 4.2.3 Create Exhibit

The following page (Figure 25) allows the curator to add new exhibits to the collection. The form provides inputs to following fields:

- Cover
- Title
- Multiline note
- Creator
- Type of exhibit
- Created At
- Creation Date

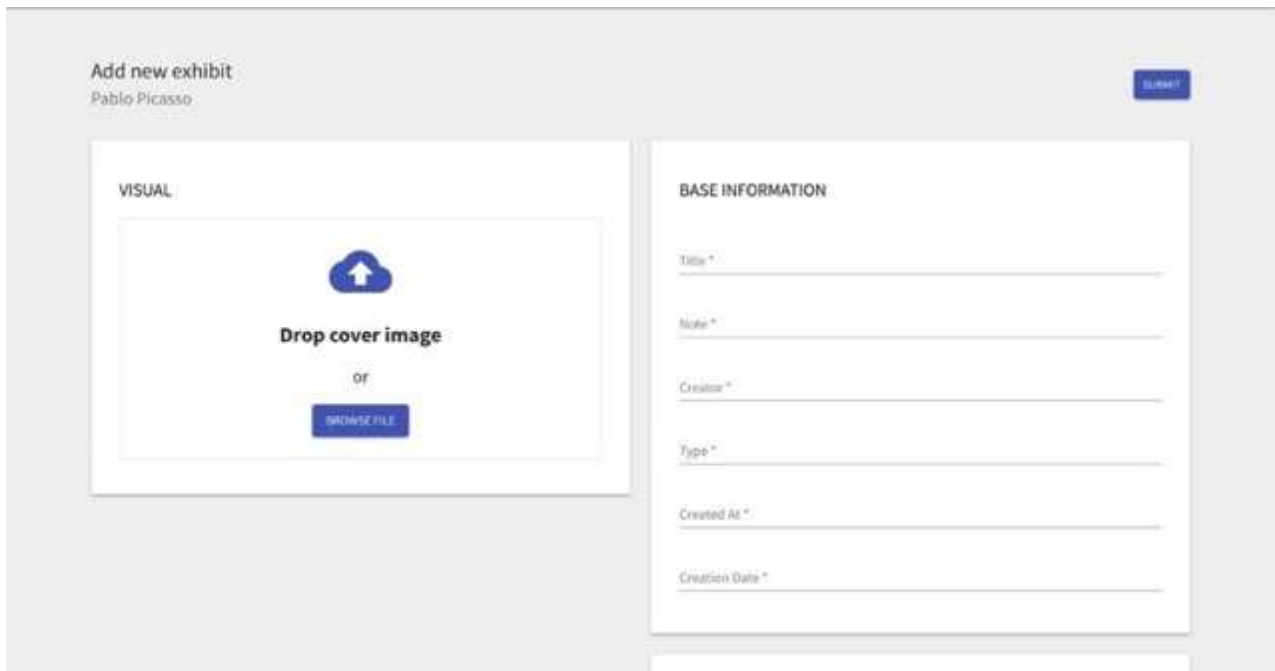


Figure 25: Create Exhibit Form

Moreover, as depicted in Figure 26, Curator can also specify the list of materials. Creation of the exhibit is performed by pressing “**SUBMIT**” button.

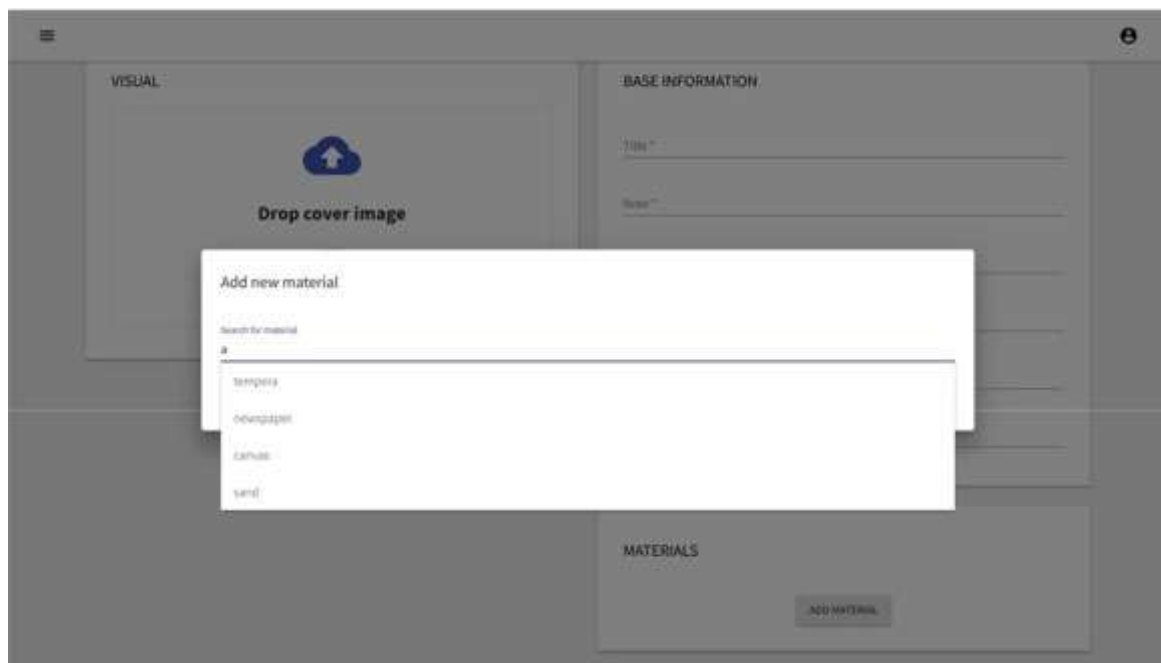


Figure 26: Add material



## 4.2.4 Exhibit Description Builder

Figure 27 depicts one of the most advanced features of the system. The exhibit description builder provides flexible way to maintain description about an exhibit or a group of exhibits. It was created on top of slate-react, a completely customizable framework for building rich text editors. Each paragraph can contain conditions regarding to its target audience. At current level of implementation, curator is limited to the specific enumeration of visitor categories. Curator can select between: **Expert**, **Adult**, **Children** and **Everyone** (as depicted in Figure 28).

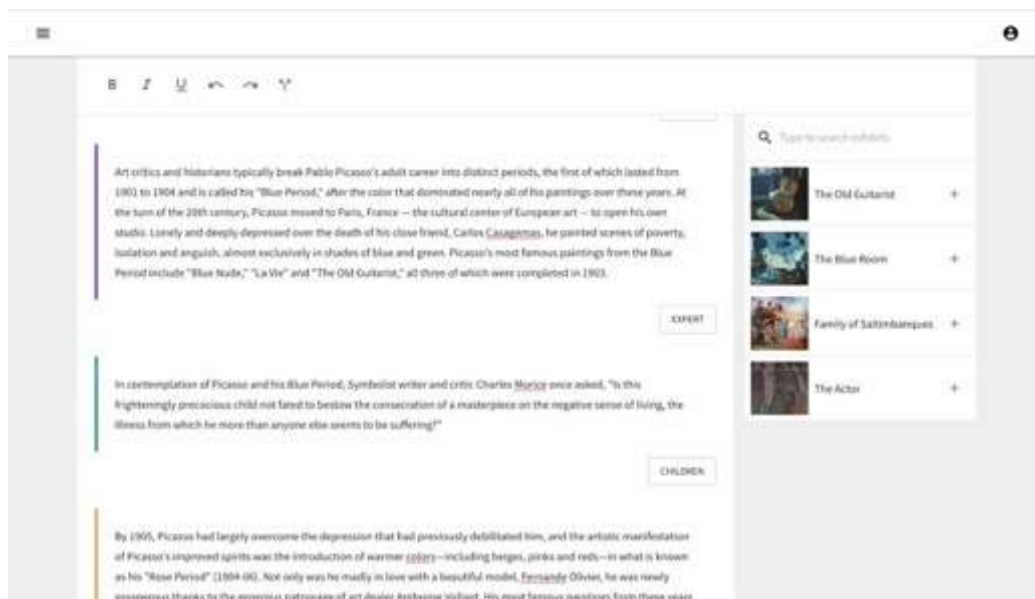


Figure 27: Multimedia Description Builder

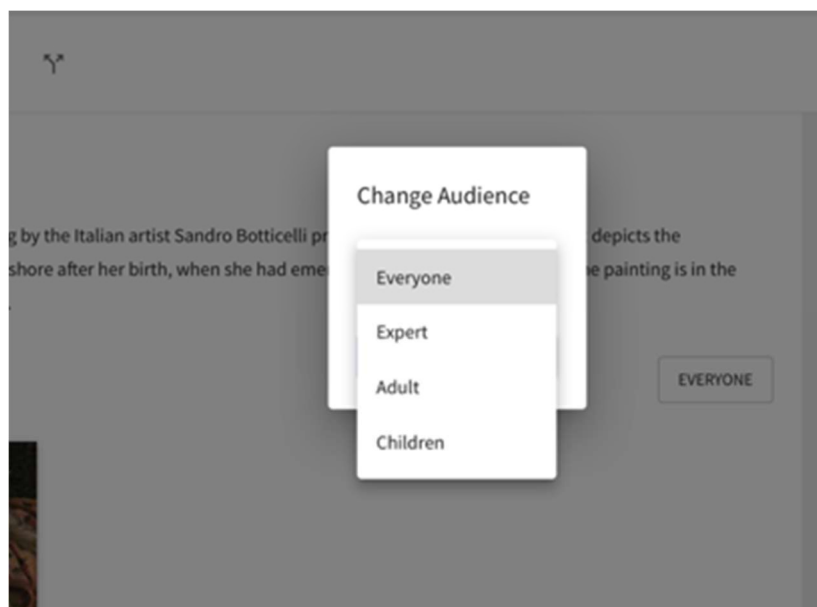


Figure 28: Audience Picker



## 4.3 Unity Museum Guide

Unity museum guide is a mobile application that provides visitor the possibility to explore interactive multimedia exhibit descriptions that are created by museum curators. Application is written in C# and follows Model-View-Controller architecture.

### 4.3.1 Basic Architecture

Core component of each Unity application is a **scene**. A Unity scene is created by arranging and moving objects in a three-dimensional space. It is the topmost object that holds the environment of the application. The developed system has three direct child objects of the scene: **Camera**, **App Delegate** and **EventSystem**.

A **Camera** is an object that defines a view in scene space. The object's position defines the viewpoint, while the forward (Z) and upward (Y) axes of the object define the view direction and the top of the screen, respectively. The **Camera** component also defines the size and shape of the region that falls within the view. With these parameters set up, the camera can display what it currently "sees" to the screen. As the camera object moves and rotates, the displayed view will also move and rotate accordingly. A camera in the real world, or indeed a human eye, sees the world in a way that makes objects look smaller the farther they are from the point of view. This well-known perspective effect is widely used in art and computer graphics and is important for creating a realistic scene. A camera that does not diminish the size of objects with distance is referred to as orthographic and Unity cameras also have an option for this. The perspective and orthographic modes of viewing a scene are known as camera projections [34].

**AppDelegate** is a component that provides an entry point of the designed application. It holds **Canvas**, **AudioSource** and **ThreadDispatcher**.

The **Canvas** is the area that holds all UI elements. UI elements in the Canvas are drawn in the same order they appear in the Hierarchy. The first child is drawn first, the second child next, and so on. If two UI elements overlap, the later one will appear on top of the earlier one. The Canvas has a Render Mode setting which can be used to make it render in screen space or world space.

**Screen Space – Overlay** render mode places UI elements on the screen rendered on top of the scene. If the screen is resized or changes resolution, the Canvas will automatically change size to match this.

**Screen Space - Camera** is similar to Screen Space - Overlay, but in this render mode the Canvas is placed a given distance in front of a specified Camera. The UI elements are rendered by this camera, which means that the Camera settings affect the appearance of the UI. If the Camera is set to Perspective, the UI elements will be rendered with perspective, and the amount of perspective distortion can be controlled by the Camera Field of View. If the screen is resized, changes resolution, or the camera frustum changes, the Canvas will automatically change size to match as well.

In **World Space** render mode, the Canvas will behave as any other object in the scene. The size of the Canvas can be set manually using its Rect Transform, and UI elements will

render in front of or behind other objects in the scene based on 3D placement. This is useful for UIs that are meant to be a part of the world. This is also known as a “diegetic interface”.

The developed system is using **screen space – camera** render approach.

The **AudioSource** plays back an Audio Clip in the scene. The clip can be played to an audio listener or through an audio mixer. The audio source can play any type of Audio Clip and can be configured to play these as 2D, 3D, or as a mixture (SpatialBlend). The audio can be spread out between speakers (stereo to 7.1) (Spread) and morphed between 3D and 2D (SpatialBlend). This can be controlled over distance with falloff curves. Also, if the listener is within one or multiple Reverb Zones, reverberation is applied to the source. Individual filters can be applied to each audio source for an even richer audio experience.

**ThreadDispatcher** is a component that was developed to interact with Unity from the background thread. Unity doesn't support multithread interaction and doesn't provide any dispatcher that can be used to pass the data between various threads.

The **EventSystem** is responsible for processing and handling events in a Unity scene. A scene should only contain one EventSystem. The EventSystem works in conjunction with a number of modules and mostly just holds state and delegates functionality to specific, overrideable components. When the EventSystem is started it searches for any BaseInputModule attached to the same GameObject and adds them to an internal list. On update each attached module receives an UpdateModules call, where the module can modify internal state. After each module has been updated the active module has the Process call executed. This is where custom module processing can take place.

As was previously mentioned **App Delegate** is a starting point of application and by itself contains two child game objects **Transition Context** and **View Components Stack**.

**Transition Context** is a component that maintains transitions between view components. It contains **Animator Controller** and two layers that can visually simulate animated transitions.

An **Animator Controller** allows to arrange and maintain a set of Animation Clips and associated Animation Transitions for a character or object. In most cases it has references to the Animation clips used within it, and manages the various Animation Clips and the Transitions between them using a State Machine, which could be thought of as a flow-chart of Animation Clips and Transitions, or a simple program written in a visual programming language within Unity.

**View Components Stack** is a component that is used to maintain and hold the stack of various view components. It provides methods to push/pop additional view components and triggers appearance methods accordingly.

Representation of the described above architecture is presented in Figure 29.

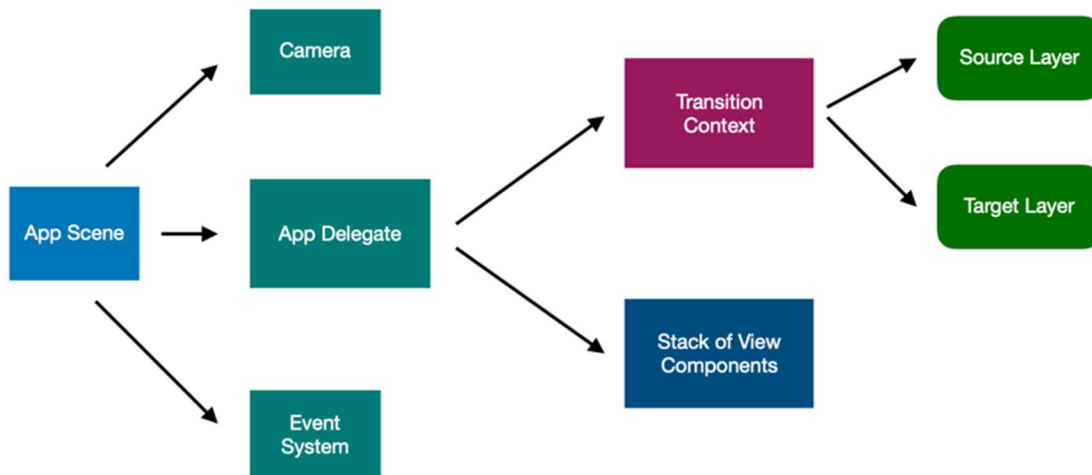


Figure 29: Unity App Basic Architecture

### 4.3.2 View Components

View Components or “Controllers” are responsible to maintain business logic and are responsible to transfer data from models to UI elements. They also form application screen hierarchy by providing **Modal** and **Navigation Stack**. The main difference between **Modal** and **Navigation Stack** is that **Navigation Stack** cannot be used as a parent for other hierarchies, it can contain only other View Components. Each view component is stored in a Unity as a **prefab** container and it is lazily instantiated when required. The view components hierarchy schema can be explored in Figure 30.

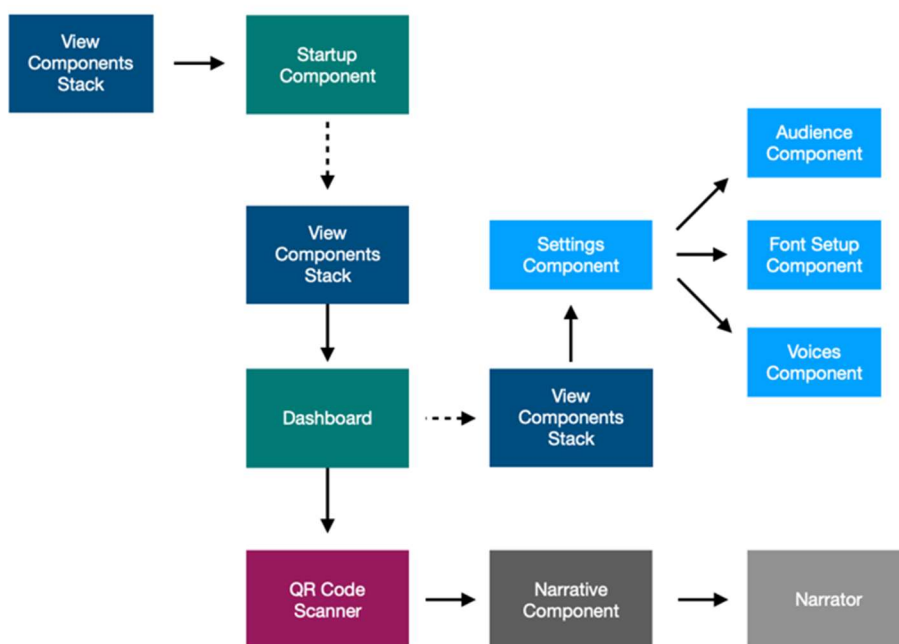


Figure 30: Unity App. View Components Schema

### 4.3.2.1 Startup Component

As depicted in Figure 31, the first screen is displayed to an unauthorized user.

Authorization flow involves two steps:

- User performs authorization using standard mechanisms that are provided by Facebook SDK for Unity
- Received access token is used to authorize with a system using the provided API interface.

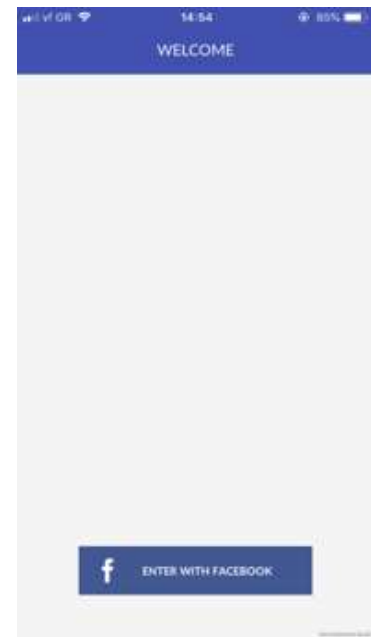


Figure 31: Unity, Unauthorized User

### 4.3.2.2 Dashboard Component

At current level of implementation, the following component, see Figure 32, provides a bridge between different view component entries, such as **Options** and **QR Code scanner**.

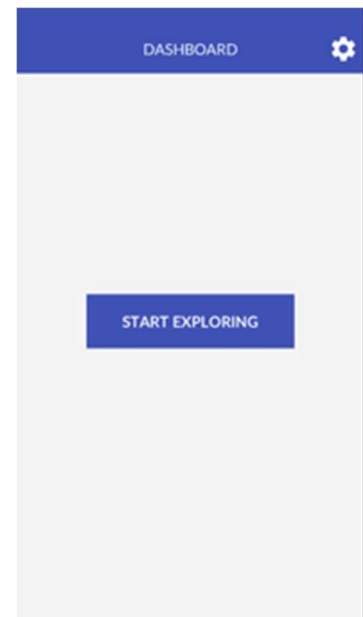


Figure 32: Unity App. Dashboard Component

### 4.3.2.3 Settings Component

Settings component (Figure 33) provides access to various options, such as **Narration Level**, **Font Setup** and **Narrator Voice**. It also contains button to perform logout operation.

The table layout was simulated using standard Button component that is provided by Unity 2D UI module.

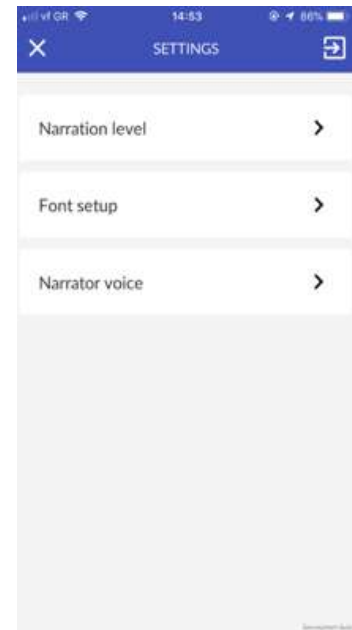


Figure 33: Unity App, Settings Component

### 4.3.2.4 Target Group Component

This component allows user to specify his target audience. System will use this value to provide more appropriate content. User can select between **Child**, **Adult** and **Expert** options. Selected option is marked with a tick, as depicted in Figure 34.

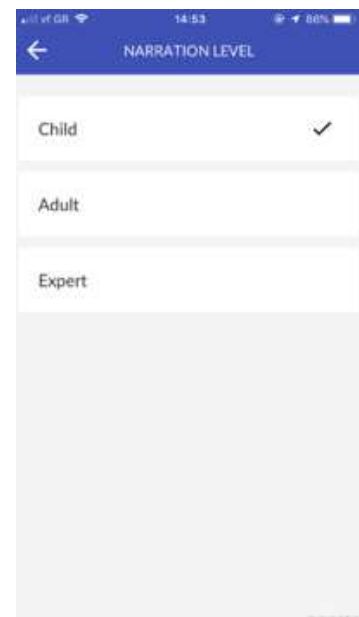


Figure 34: Unity App, Target Group

### 4.3.2.5 Font Setup Component

**Font Setup Component** allows user to dynamically adjust preferred font size. User is provided with a slider and text preview. Following changes would be applied to the text size of a description. Screen implementation is presented in Figure 35.

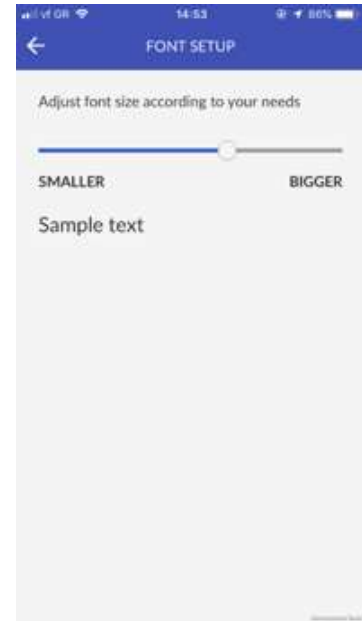


Figure 35: Unity App, Font Settings

### 4.3.2.6 Voices Component

This component (Figure 36) provides list of available voices. These voices are retrieved using system API interface. Each of it can be previewed by the user on tap selection.

Each voice cell display information about the voice title and its language. Voice preview is performed via Unity **AudioSource** component.

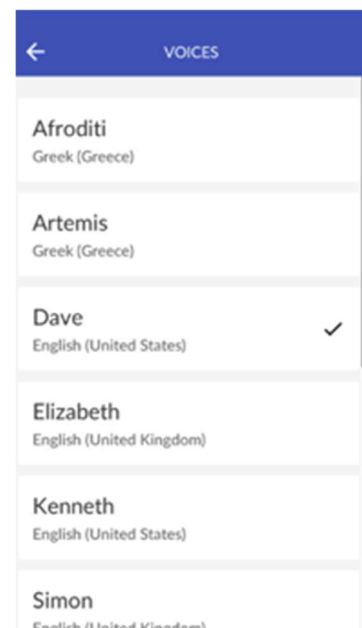


Figure 36: Unity App, Voices Component

#### 4.3.2.7 QR Core Scanner

This view component (Figure 37) is built on top of ZXing library and Unity **WebCamTexture**. It is capable to display and process video stream from rear camera in order to identify QR codes.

As soon as a valid QR code is detected, description about exhibit is presented to the user. Title of detected exhibit is displayed in the bottom part of the screen.

The decoding process is performed in a background thread using standard C# features. Moreover, in order to optimize this process, only area below transparent white rectangle is used. This approach made it possible to perform the real time decoding without lousing frames.



Figure 37: Unity App, QR Code Scanner

#### 4.3.2.8 Information Component

Information component it used to display descriptions about exhibits. Visitor can explore text blocks and images according to his type of audience. The UI elements and their order are automatically generated during the runtime because the structure of a description is unknown and can be different for each user. User can explore description text using audio modally by pressing "Play" button, as depicted in Figure 38.



Figure 38: Unity App, Information Component

#### 4.3.2.9 Narrator Component

This component allows user to explore the content using Max, a virtual 3D assistant, as depicted in Figure 39.

Max is a six-year-old low polygon three-dimensional character that supports lip synchronization and can show emotions through facial expressions [37].



Figure 39: Unity App, Virtual Narrator



# 5 Evaluation

Current chapter discusses the evaluation method, process and its results regarding the usability of the system.

## 5.1 Heuristic Evaluation

Heuristic [38] is a usability engineering method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process. Heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the "heuristics").

Heuristic evaluation is performed by having each individual evaluator inspect the interface alone. The results of the evaluation can be recorded either as written reports from each evaluator or by having the evaluators verbalize their comments to an observer as they go through the interface.

The output from using the heuristic evaluation method is a list of usability problems in the interface with references to those usability principles that were violated by the design in each case in the opinion of the evaluator.

## 5.2 The Evaluation Process

The evaluation process took place inside FORTH's Aml Facilities. A personal computer was used to evaluate **Curator Administration Web Panel**. An iPhone 7 running iOS 12 was used to evaluate **Unity Mobile Application** for a visitor.

A total of 5 volunteers participated in the evaluation, 3 women and 2 men. Their age varied from 25-40. All of them were experienced or expert computer users and each one them had a smartphone before the evaluation.

The evaluation process started with the users being informed about the evaluation process purpose. The short introduction to the concept of the evaluation and the functionality that system aims to provide was explained to each user. Users were asked to perform heuristic evaluation of curator **Administration Web Panel** and **Unity Mobile Application** for visitor.

Apart from the heuristic evaluation, all participants were encouraged to express their thoughts throughout the evaluation process, which were written down using notes and further processed through the examination of the recordings.

The evaluation process was designed to cover the needs of this thesis, however, further assessment of the system would be interesting in order to reach extra conclusions. An

additional evaluation could involve real museum curators and non-expert users to measure their reaction and needs regarding to the system.

## 5.3 Severity score

Severity ratings can be used to allocate the most resources to fix the most serious problems and can also provide a rough estimate of the need for additional usability efforts. If the severity ratings indicate that several disastrous usability problems remain in an interface, it will probably be unadvisable to release it. But one might decide to go ahead with the release of a system with several usability problems if they are all judged as being cosmetic in nature.

According to [39] the 0 to 4 rating scale can be used to rate the severity of usability problems. Table 4 explains each score value.

Table 4, Severity Score

<b>SEVERINITY SCORE</b>
<b>0</b> = I don't agree that this is a usability problem at all
<b>1</b> = Cosmetic problem only: need not be fixed unless extra time is available on project
<b>2</b> = Minor usability problem: fixing this should be given low priority
<b>3</b> = Major usability problem: important to fix, so should be given high priority
<b>4</b> = Usability catastrophe: imperative to fix this before product can be released

## 5.4 Results

The results that were obtained during the heuristic evaluation were promising. Evaluators were satisfied with the concept of the system. Non-centralized storage system based on the Semantic Web technologies that supports global collaboration between museums.

The user interface part does require improvements in terms of clarification and concreteness. Functions regarding the description flow were not clear to all evaluators and possible solutions to this problem require further investigation.

Less serious issues are listed and discussed in the next subsections.

### 5.4.1 Curator Authoring Tool

During the heuristic evaluation, most of the users had problems due to ambiguous titles and controls. Generic nature of the **Curator Authoring Tool** provides high flexibility, but in the same time makes it difficult to use even by an experienced evaluator.

The total number of discovered issues is 47. The most critical issues that were discovered during the evaluation are relevant to **Description Builder** screen:

- The backspace button should not merge text blocks.
- Text blocks should be equipped with reposition and remove buttons.
- Feedback indicators are required on save action or received error.
- Exhibits that are presented in description should not be displayed on the right list of exhibits
- "CREATE BLOCK" button should always be visible to the user. The text block should be inserted at cursor position. Inserted block should be automatically focused.

The total list of problems and their severity score is presented in Table 5.

Table 5, Curator Authoring Tool Issues

SCREEN	ISSUE	SCORE
COLLECTIONS		
	The page should provide controls for categorization, search and filtering.	2,75
	The collection row should include the description of collection, as well as information about the amount of exhibits, the creation date and the curator name.	2,25

	The collection row should provide buttons to edit or remove functionality.	2
CREATE COLLECTION		
	The selected cover is not expected to be removed by dragging it.	2,5
	The purpose of "NOTE" field should be explicitly mentioned.	2,75
CREATE EXHIBIT		
	"SUBMIT" button should be renamed to "CREATE".	1,25
	The selected cover is not expected to be removed by dragging it.	2,25
	Feedback indication is required due to a creation or error event.	2,25
	The purpose of each form field should be presented to the user.	2,25
	Some form fields have bad labels. "CREATOR" should be renamed to "ARTIST" and the "CreatedAt" field to "Origin country". The "Creation Date" field should provide information about the expected date format or an appropriate date picker control.	2,25
	Dropdown fields should contain an indicator image on the right side. The available options should be presented on cursor focus.	2,75
EXHIBITS		
	"CREATE NEW" button should be renamed to "CREATE EXHIBIT"	1,25
	The exhibit row should provide buttons to edit or remove functionality.	2
	A user should have the ability to create exhibits without collection.	3
NEW DESCRIPTION FLOW		

	Cover should be marked as an optional field.	1,25
EXHIBIT DESCRIPTION FLOW		
	Loading indicators should be presented on long operations.	2,25
	The purpose of the QR code should be explicitly mentioned.	1,5
	Each description node should have a short preview and tags.	2,5
	"BUILDER" button should be renamed to "Edit".	1,5
	The description node should be clickable.	1,5
	Flow should provide functionality to reposition or remove nodes	2,5
	Curator should be able to create description for one specific exhibit.	2
	The description flow should allow curators to categorize it according to the museum environment.	2,25
	Description flow preview and simulation tool should be provided to curator.	1,75
	Curators should be provided with a tool that would guide them during the flow creation. Guide should include various dialogs with questions and options.	2,75
DESCRIPTION EDITOR		
	Each toolbar item should be provided with a tooltip that would contain short explanation regarding to the item purpose.	1,25
	The backspace button should not merge text blocks.	3,75
	The description flow title should be provided at the top.	2,25
	"CREATE BLOCK" button should always be visible to the user. The text block should be inserted at cursor position. Inserted block should be automatically focused.	2,75
	The purpose of the list of exhibits should be provided with an explanation mark.	1

List of exhibits purpose should be categorized by common properties.	2,25
Empty text blocks should be ignored on submit.	2,25
Text blocks should be equipped with reposition and remove buttons.	3,25
The text block area should be visualized with a border.	2
Audience picker should contain color indicator according to the audience type.	1,75
Audience picker should be presented as a dropdown	2
It should be possible to select two or more audiences	2,75
Feedback indicators are required on save action or received error.	2,75
Exhibits that are already presented in description flow should not be displayed on the right list of exhibits	2,75
A virtual museum floorplan with the location of each exhibit would be a useful functionality	2,5

## 5.4.2 Unity Mobile Application for Visitors

During the evaluation, no critical issues were discovered. The most popular generic issues that were encountered by all evaluators are:

- Users should be able to specify their area of interest.
- The list of available descriptions should be provided to the user on a Dashboard screen, according to their preferences.
- Player controls should be presented on the Information screen to control audio stream. These controls should always be visible to the user.
- The Virtual Narrator should be presented on the Information screen.

The total number of discovered issues is 13. Their titles and severity scores are represented below (Table 6).

Table 6, Unity Mobile App Issues

ISSUE	Score
A welcome guide should be provided where users will be able to specify basic preferences	2
The application should provide a museum floorplan with the location of exhibits	2
Option screen icons should be replaced with more accurate ones	2
The list of voices should be merged with "Voice Info" screen.	2
"Options" should be available in all screens	3
Users should be able to specify their area of interest.	3
The list of available exhibit descriptions should be provided to the user on a Dashboard screen, according to their preferences.	3
QR code screen has to be provided with hints. Hints should explain which QR codes are valid and where they could be found.	2
Short preview of the detected exhibit description and the rest flow of included exhibits should be presented after QR code recognition.	2

Player controls should be presented on an Information screen to control audio stream. These controls should always be visible to the user.	3
Scroll of a multimedia content should be in sync with audio player progress.	3
The Virtual Narrator should be presented on an Information screen.	3
"SEE MORE INFORMATION" button should be presented on an Information screen. When selected, it should present description of relevant exhibits.	2

## 6 Lessons Learned

Semantic Web is an advanced technology that can be used to solve various range of problems in an efficient and effective way. Ontology schema and **SPARQL** query language provide a flexible way to maintain the data, especially in the context of the museum where objects have multiple properties and connections with each other. Ontology schema provides an efficient way to resolve direct or nested relations.

Non-centralized nature of **Semantic Web** allows to organize the data across multiple storages in totally different places. Each organization or the museum can have its private storage that only accessible by them or to cooperate with others. Cooperation and data exchange will improve the quality of museum content and by having access to the global visitor profile, museums would be able to generate more personalized multimedia descriptions of the exhibit.

The system in this thesis benefits from abovementioned **Semantic Web** features by providing a platform to maintain the museum data and create personalized exhibit descriptions.

Museum data is stored according to the system ontology schema that inherits and extends **CidocCRM** subset. Extended schema includes various classes that can represent objects from museum collections, visitors, curators and multimedia exhibit descriptions. **Semantic Web** provides clean and simple way not only to fetch information about one specific exhibits, but also to discover similar or related objects by using property match or advanced property path expressions. The data can be collected and combined from sources that could be maintained by totally different museums as long as they don't violate basic schema structure.

Personalized multimedia description of the exhibit is automatically generated during the runtime by combining description blocks that are pre-created by the museum curator.



Each block contains its own set of predefined conditions that are used by the system to decide how relevant and interesting its content might be to the potential visitor. Transitive closure and other relational algebra operators are used in the context of **SPARQL** query to fetch, filter and combine potentially relevant description blocks.

## 7 Discussion and Future Work

### 7.1 Discussion

In recent years, the amount and role of mobile phones have significantly increased. Nowadays, modern phones have become portable high-performance computers that allow people to use them for various tasks like work, education, entertainment, etc.

Apart from users, developers and researchers got possibilities to resolve problems in a new, more sophisticated way. However, the potential benefits and innovative use cases of smartphones, is still under investigation across multiple domains, such as the museum environment.

This thesis has presented an approach of integrating technologies from various domains such as **Ambient Intelligence** and **Semantic Web** in the context of the museum environment.

The system reported in this thesis has been designed to research the integration of these technologies in order to build a modular system that is capable to provide exhibit descriptions based on the user profile. The designed system supports the exhibit management, to create its multimedia description and to present it to the end users.

The designed system represents itself as a convergence of smartphones with Ambient Intelligence, Semantic Web. Various technologies such as RDF, .NET Core, ReactJS and Unity 3D were presented and discussed during the current work.

The knowledge representation of the developed system is based on the semantic data through the usage of ontologies. System ontology was created to maintain the structure of the internal objects, while CIDOC CRM was used to resolve and maintain connections between exhibits. Furthermore, the proposed system supports the extraction of semantic knowledge according to the user preferences and his usage history. As a result, such information allowed the system to generate more relevant and personalized exhibit descriptions.

A heuristic evaluation was conducted to analyze the developed system in terms of usability and user experience. Based on the results of the evaluation process and feedback of evaluators, the system provides an interesting concept, but its flexibility makes the

system difficult to use by unprepared user. The system has to provide more limited and concrete set of functions. The further research is required to improve the system in terms of usability and comprehension.

## 7.2 Future Work

Improvements of the system design are also planned based on the conclusions of the evaluation. Various ideas about how to improve the system were mentioned during the evaluation.

Curator Authoring Tool:

- Information about collections should be presented in a more structured manner, categorization and filtering capabilities have to be provided
- Exhibit description flow simulation tool is required in order to understand how flow will be presented to the visitor
- Virtual map with the location of exhibits should be provided

Unity mobile application:

- Visitor should be provided with a welcome guide on first launch
- Virtual Narrator should be presented on an information screen
- Player control should be presented on an information screen in order to control audio stream
- Audio stream progress should be synchronized with a scroll of an information component

Moreover, the Unity mobile application can be extended to present exhibits using augmented reality modality.

## 8 Bibliography

- [1] Statista, "Number of smartphone users worldwide from 2014 to 2020 (in billions, 2018). [Online]. Available: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>.
- [2] J. C. A. b. V. R. J. Diane J. Cook a, "Ambient intelligence: Technologies, applications, and opportunities," vol. 53, no. 4, pp. 277-298, 2009.
- [3] I. A. Group, "Ambient intelligence: From vision to reality for participation—in society and business," 2003.
- [4] I. A. Group, "I.A. Group, Scenarios for ambient intelligence in 2010, 2001".
- [5] A. I. J. C. A. Asier Aztiria, "Learning patterns in ambient intelligence environments: a survey," *Springer Science+Business Media*, pp. 34-50, 2010.
- [6] M. Turk, "Multimodal interaction: A review," *Pattern Recognition Letters*, 2013.
- [7] S. Oviatt, "Ten myths of multimodal interaction," *Commun. ACM*, vol. 42, no. 11, pp. 74 - 81, 1999.
- [8] J. T. Garrett, "Using Speech Recognition Software to Increase Writing Fluency for Individuals with Physical Disabilities," 2011.
- [9] B. Bansal, "Gesture Recognition: A Survey," *International Journal of Computer Applications*, vol. 139, no. 2, pp. 8-10, 2016.
- [10] M. Baldauf, "A survey on context-aware systems", journal = "Int. J. Ad Hoc and Ubiquitous Computing," *Int. J. Ad Hoc and Ubiquitous Computing*, vol. 2, no. 4, pp. 263-277, 2007.
- [11] M. S. DECKER S., "The Semantic Web: The Roles of XML and RDF," *IEEE Internet Computing*, vol. 4, pp. 63-74, 2000.
- [12] A. B. A. v. d. M. A.J. Gerber, "A Semantic Web Status Model," *Integrated Design and Process Technology*, vol. 4, pp. 473-482, 2006.
- [13] L. C. L. H. L. Y. A. W. H. DONG J.S., "Semantic web foundations: A combined approach to checking web ontologies," *ACM Press New York*, p. 714 – 722, 2004.
- [14] W3C, "RDF Primer," 2018. [Online]. Available: <https://www.w3.org/TR/rdf-primer/>.
- [15] F. v. H. J. H. I. H. D. L. M. P. F. P.-S. L. A. S. Sean Bechhofer, "OWL Web Ontology Language Reference," 2004.
- [16] Y. Y. Z. Q. G. X. Y. P. a. S. L. Li Ma, "Towards a Complete OWL Ontology Benchmark," *The Semantic Web: Research and Applications: 3rd European Semantic Web Conference*, pp. 125-139, 2006.
- [17] S. T. K. Z. S. L. T. G. J. Y. D. G. F. M. E. S. I.-Y. M. Wang, "Conceptual Modeling for Advanced Application Domains," *Workshops CoMoGIS, CoMWIM, ECDM, CoMoA, DGOV, and eCOMO*, 2004.
- [18] I. D. S. Group, "CIDOC-CRM," 2018. [Online]. Available: <http://www.cidoc-crm.org>.
- [19] I. D. S. Group, "CIDOC-CRM," 2018. [Online]. Available: [http://www.cidoc-crm.org/sites/default/files/2018-5-16%23CIDOC%20CRM\\_v6.2.3\\_esIP%28XDP%29%28XM%29.pdf](http://www.cidoc-crm.org/sites/default/files/2018-5-16%23CIDOC%20CRM_v6.2.3_esIP%28XDP%29%28XM%29.pdf).

- [20] M. D. C. E. O. S. S. Patrick Le Boeuf, "CIDOC Conceptual Reference Model," 2018.
- [21] A. H. R. Lim2, "Implementing QR Code to Create Indonesia Museum Interactive Information Application," in *Global Conference on Engineering and Technology*, Bangkok, 2017.
- [22] P. T. a. P. B. Athanasios Fevgas, "iMuse Mobile Tour: a personalized multimedia museum guide opens to groups," *IEEE Symposium on Computers and Communications (ISCC)*, pp. 971-975, 2011.
- [23] C. a. P. F. a. R. G. a. L. B. Santoro, "A multimodal mobile museum guide for all," *IEEE Symposium on Computers and Communications (ISCC)*, 2007.
- [24] C.-Y. a. C. S.-Y. a. L. S.-W. Tsai, "Location-Aware Tour Guide Systems in Museum," in *Collaborative Product and Service Life Cycle Management for a Sustainable World*, R. a. C. S. a. T. A. Curran, Ed., London, Springer Londo, 2008, pp. 349--356.
- [25] B. a. V. G. L. Fasel, "Interactive Museum Guide: Accurate Retrieval of Object Descriptions," in *Proceedings of the 4th International Conference on Adaptive Multimedia Retrieval: User, Context, and Feedback*, Geneva, Springer-Verlag, 2007, pp. 179-191.
- [26] M. Corp, "ASP.NET Core," 2018. [Online]. Available: <https://www.asp.net>.
- [27] O. a. M. I. Erling, "RDF Support in the Virtuoso DBMS," in *Networked Knowledge - Networked Media: Integrating Knowledge Management, New Media Technologies and Semantic Systems*, Springer Berlin Heidelberg", address="Berlin, Heidelberg, 2009, pp. 7--24.
- [28] Stanford, "PROTÉGÉ," 2018. [Online]. Available: <https://protege.stanford.edu>.
- [29] MaterialUI, "<https://material-ui.com>," 2018. [Online]. Available: <https://material-ui.com>.
- [30] F. Inc, "ReactJS," 2018. [Online]. Available: <https://reactjs.org/>.
- [31] A. Fedosejev, *React.js Essentials*, PACKT, 2015.
- [32] D. Abramov, "The Redux Journey," React Europe, 2016.
- [33] O. S. Community, "Official Redux Documentation," 2018. [Online]. Available: <https://redux.js.org>.
- [34] U. Technologies, "Unity," 2018. [Online]. Available: <https://unity3d.com>.
- [35] F. Inc, "Graph Api," 2018. [Online]. Available: <https://developers.facebook.com/docs/graph-api/>.
- [36] W3C, "[www.w3.org](http://www.w3.org)," 15 1 2008. [Online]. Available: [https://www.w3.org/blog/SW/2008/01/15/sparql\\_is\\_a\\_recommendation/](https://www.w3.org/blog/SW/2008/01/15/sparql_is_a_recommendation/).
- [37] E. a. P. G. a. S. C. Zidianakis, "A Cross-platform, Remotely-controlled Mobile Avatar Simulation Framework for Aml Environments," 2014.
- [38] J. a. M. R. Nielsen, "Heuristic Evaluation of User Interfaces," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 249--256, 1990.
- [39] J. Nielsen, "Severity Ratings for Usability Problems," 1995.
- [40] D. G. C. S. Chryssi Birliraki, "Employing Virtual Humans for Interaction, Assistance and Information Provision in Ambient Intelligence Environments," *Distributed, Ambient, and Pervasive Interactions*, vol. 9189, pp. 249-261, 2015.

- [41] G. K. G. Jenna Burrell, "E-graffiti: evaluating real-world use of a context-aware system," *Interacting With Computers*, vol. 14, pp. 301-312, 2002.
- [42] J. Dallas, "Museum Practice in the Era of Information in Museums Space and Power," *ISS ICOFOM Study Series*, vol. 22, pp. 17-51, 1993.
- [43] J. Campbell, "A Hero with a Thousand Faces," *London: Fontana Press*, 1993.
- [44] M. Bal, "Narratology: Introduction to the Theory of Narrative," *Univ. of Toronto Press*, 1985.
- [45] M. C. Villaseñor, "Building Echoes. The Role of Storytelling in Museums and Galleries," 2007.
- [46] E. Johnsson, "Telling tales. A guide to developing effective storytelling programmes for museums," *Renaissance London*, 2006.
- [47] G. a. S. K. Bull, "arrDigital storytelling in the language arts classroom," *Learning \& Leading with Technology*, pp. 46-49.
- [48] J. Bruner, "Acts of meaning," *USA, Harvard University Press*, 1990.
- [49] K. K. a. A. B. a. M. Turki, "Voiced/unvoiced speech classification-based adaptive filtering of decomposed empirical modes for speech enhancement," *IET Signal Processing*, vol. 10, pp. 69-80, 2016.
- [50] G. R. S. M. a. R. S. Jadon, "Gesture Recognition: A Survey," *International Journal Of Information Technology and Knowledge*, pp. 8-10, 2009.