Dissertation

# MARE: e**M**powering **A**rchitectural heritage **RE**search in the Eastern Mediterranean

**GEORGIOS ANTONIOU**

**UNIVERSITY OF CYPRUS**

**DEPARTMENT OF COMPUTER SCIENCE**

**May 2015**

# UNIVERSITY OF CYPRUS

## DEPARTMENT OF COMPUTER SCIENCE

**MARE: eMpowering Architectural heritage REsearch in the Eastern Mediterranean**

**Georgios Antoniou**

Supervisor
Dr. Georgios Samaras
Co-Supervisor
Dr. Panayiotis Andreou

The individual thesis submitted for partial fulfillment of the requirements for obtaining the degree of computer science, department of computer science, University of Cyprus

May 2015

## Acknowledgements

I would like to express my sincere thanks to Dr. George Samaras, the supervisor of my dissertation, for believing in me and giving me the opportunity to undertake this project. By giving me this opportunity, during the course of the dissertation, I was able to study and acquire Internet technology knowledge; knowledge that is essential for my academic and professional career.

I place on record, my sincere thank you to Dr. Panayiotis Andreou, the co-supervisor of my dissertation, for all the assistance and guidance he provided me during the whole year of my dissertation and for his overall trust in me.

I am also grateful to Marios Belk, PhD Candidate of Computer Science department, University of Cyprus for his continuous support and assistance in the design and development of the dissertation project.

Finally, I want to thank my parents, Antonis and Maria, for the unceasing encouragement, support and attention.

# Abstract

The Mesarch laboratory is a research laboratory of the Department of Architecture, University of Cyprus focusing on scholarly research on the history and theory of modern architecture in the eastern Mediterranean. Mesarch has established collaboration with faculty from the Department of Computer Science, University of Cyprus, experimenting with the development of innovative information models for interpreting, presenting, and evaluating 20th century architectural heritage

The purpose of this dissertation is the design and the implementation of an interactive, usable, user friendly, online tool capable of allowing authorized users to manage the contents of their website. The software was developed for the Mesarch laboratory of Architecture's department of University of Cyprus and allows the laboratory's personnel to insert, update and delete information regarding their laboratory's researches, like projects, media and actors. Additionally, the software provides a platform where they can present this information in a user friendly way to architecture researchers, academic people and the public in general. The users are able to create accounts and obtain roles, such as data entry user, webmaster and simple user.

The system was deployed on the Department of architecture of University of Cyprus server followed by a testing period of over three months, where the end-users could provide feedback regarding system errors and suggestions for improvement. In order to evaluate, a variety of different criteria and quality metrics like ease of learning, ease of use etc were used that showed overall user satisfaction and acceptance of the system.

# Περίληψη

Το εργαστήριο Mesarch είναι ερευνητικό εργαστήριο του τμήματος αρχιτεκτονικής του Πανεπιστημίου Κύπρου, το οποίο έχει ως αντικείμενο μελέτης την ιστορία και θεωρία της μοντέρνας αρχιτεκτονικής στην ανατολική μεσόγειο. Το εργαστήριο συνεργάζεται με ακαδημαϊκό προσωπικό του τμήματος πληροφορικής του Πανεπιστημίου Κύπρου, με στόχο την ανάπτυξη ενός πρωτοποριακού πληροφοριακού μοντέλου μέσω του οποίου θα μπορεί να παρουσιάζει και να αξιολογεί την αρχιτεκτονική κληρονομιά του εικοστού αιώνα.

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η σχεδίαση και υλοποίηση ενός διαδραστικού, εύχρηστου, φιλικού προς τον χρήστη, διαδικτυακού εργαλείου το οποίο θα παρέχει σε εξουσιοδοτημένους χρήστες την δυνατότητα να διαχειριστούν τα δεδομένα της ιστοσελίδας τους. Το λογισμικό αναπτύχθηκε για το εργαστήριο Mesarch του τμήματος αρχιτεκτονικής του Πανεπιστημίου Κύπρου και επιτρέπει στα μέλη του εργαστηρίου την εισαγωγή, αναβάθμιση ή και διαγραφή δεδομένων σχετικά με την έρευνα του εργαστηρίου, όπως projects, media και actors. Επιπλέον, το λογισμικό παρέχει μία πλατφόρμα όπου τα μέλη του εργαστηρίου μπορούν να παρουσιάσουν αυτές τις πληροφορίες σε ακαδημαϊκούς αλλά και το απλό κοινό με ένα φιλικό προς τον χρήστη τρόπο. Οι χρήστες μπορούν να δημιουργήσουν λογαριασμό και να αποκτήσουν διάφορους ρόλους στο σύστημα, όπως data entry user, webmaster και simple user.

Το σύστημα εγκαταστάθηκε στον web server του τμήματος αρχιτεκτονικής του Πανεπιστημίου Κύπρου. Ακολούθως, ξεκίνησε μία περίοδος ελέγχου του λογισμικού που διήρκησε λίγο περισσότερο από τρεις μήνες και κατά τη διάρκεια της οποίας οι τελικοί χρήστες μπόρεσαν να αξιολογήσουν το σύστημα και να δώσουν εισηγήσεις για βελτίωση. Στο τέλος της περιόδου ελέγχου, διάφορα κριτήρια και μετρικές όπως ευκολία χρήσης και εκμάθησης χρησιμοποιήθηκαν ώστε να αξιολογηθεί το σύστημα.

## Table of Contents

CHAPTER 1 **Introduction** …..…………………………………………………………… **8**

1.1 Thesis overview 8

1.2 Dissertation's contribution 10

1.3 Chapters overview 11

1.4 Acronyms and abbreviations 11

CHAPTER 2 **Related work** ………………..…………………………………………. **18**

2.1 Classification of current systems 19

2.2 Comparison of current systems 23

2.3 Research tools for architectural research 30

CHAPTER 3 **Development methodology**…………………..…………………………… **34**

3.1 Dynamic systems development 34

3.2 Design objectives 36

CHAPTER 4 **Feasibility and foundations**………………..…………………………… **37**

4.1 Stakeholders and roles 37

4.2 Information models 39

4.3 Functional requirements 40

4.4 Prioritization of functionalities 45

4.5 Delivery plan 46

CHAPTER 5 **System design**…………………….……………………….…………… **47**

5.1 System architecture 47

5.2 Database structure and schema 49

5.2.1 ER Diagrams 50

5.2.2 Relational diagrams 54

5.3.1 Interface design 56

5.3.1 Prototype user interface 56

5.3.2 Graphical user interface 60

5.4 Security infrastructure 66

CHAPTER **6** **System development** …….…………………………………………… **68**

6.1 Software design 68

6.2 Implementation tools 70

6.3 Database implementation 73

6.4 User interface implementation 74

6.5 System integration 77

CHAPTER **7** **Prototype deployment and evaluation** ………………………………… **78**

7.1 Experimental methodology 78

7.2 Deployment design 79

7.3 Evaluation 80

CHAPTER **8** **Conclusions and future work** …………………………………………… **83**

**Bibliography** ………..……………………………………………………………. **86**

**Appendix A** **A-1**

# Chapter 1

# Introduction

1.1 Thesis overview 8
1.2 Dissertation's contribution 10
1.3 Chapters Overview 11
1.4 Acronyms and abbreviations 11

## 1.1 Thesis overview

### The mesarch laboratory

Mesarch is a research laboratory focusing on scholarly research on the history and theory of modern architecture in the eastern Mediterranean, with the goal to expand the critical understanding of modernism in the region. The mesarch lab is engaged in research programs and the organization of scholarly activities such as workshops, academic conferences and exhibitions.

The term mesarch is an amalgam of different meanings: *Mesogeios* (Greek word for Mediterranean), *Middle East, Architecture* and *Archive*. The term also exists in Botanology as "the maturing of the centre toward both the interior and exterior," and in many ways, this is also in tune with the mesarch's premise that its research should steer clear of centre-periphery models to treat Eastern Mediterranean's histories of modernism as complexly entangled with histories of European and other modernisms.

Mesarch is conducting critical research on the interdisciplinary fields of:

- Architecture/urbanism and politics in 20th Century Eastern Mediterranean
- Histories of urban development and environmental sustainability, cultural heritage, memory and conflict

- Digital humanities and architectural archives

Mesarch has launched programs of historical and archival research that focus on the Architecture and Modernization Histories of Cyprus and situate Cyprus and its modern architecture within international scholarly debates.

Mesarch's overall commitment to historical research and historiography is promoted through a well-equipped laboratory including state-of-the-art equipment for the documentation and digitization of archival material and a research team experienced in archival research, documentation, and theoretical analysis.

**Current practices**

The laboratory manages information regarding projects, media (photographs, newspaper articles etc that describe the projects) and actors (architects that are involved in the creation of projects) that contribute in the modern architecture of the eastern Mediterranean. Projects involve basic information like title, building type, description, location, construction dates etc, and more advanced information like field reports, references, related media and actors. In a few words, projects provide vital information to any researchers that may want to study modern architecture in the eastern Mediterranean.  Media involves photographs, magazine or newspaper articles, drawings and some further information like media name, creator, creation date, location, type, format etc and in general, they describe the projects in a visual way. Finally, actors involve information about architects that are involved in the creation of projects like actor name, type, photograph, short bio, formal education, geographies of practice etc. Actors can be related with projects or other actors they collaborated with through their career.  The laboratory until recently did not have an integrated system where they could manage and present all this information and the need of the digitization of the data emerged.

**Dissertation's purpose**

Mesarch has established collaboration with faculty from the Department of Computer Science, University of Cyprus experimenting with the development of innovative information models for interpreting, presenting, and evaluating 20th century architectural heritage.

For this reason, this dissertation focuses on the development of a web application that aims to offer a research tool for students and academics studying modern architecture in Cyprus and the larger region of the Eastern Mediterranean and showcases some of the Lab's activities.

The site's organization around Projects, Actors and Media and its various search tools have been designed specifically to facilitate architectural history research, that situates built artifacts within a larger social, cultural and political context. It aspires to become a hub of scholarly exchange on broader issues on the history-theory of Modern architecture in the region. Its intended audience includes architecture students, practicing architects, and academics interested in the history and culture of modernity in the region.

### System evaluation

The system was developed according to the requirements and goals set in meetings with the mesarch lab's personnel. After it was fully implemented, it was deployed on the Architecture's department of University of Cyprus server and the users had the opportunity to interact with it and test it. During this time enough feedback was received from the users regarding functional and designing issues and the proper functionalities were developed to meet the users' needs. At the end of the testing phase, a questionnaire was provided to the users to fill it so the overall usability of the system could be evaluated.

## 1.2 Dissertation's contribution

The current dissertation serves the needs of university of Cyprus architecture's department mesarch laboratory through the use of a web application/system. An innovative database system was designed and implemented to contribute in the management and presentation of laboratory's data and researches. A web interface was developed as well, to present all the data of the database to the architecture world and to the public in general. Additionally, the web graphical interface allows the personnel of the laboratory to manage the data stored in the database in a user friendly way since the users will be people without any programming skills that cannot interact directly with the database.

The system was deployed on the server of the department of architecture of University of Cyprus and real users from the laboratory had the opportunity to interact with it and insert and manage data through the interface to test usability, ease of use and effectiveness of the system. The evaluation process showed positive results, the users were very satisfied with the system, they found it user friendly, simple and easy to use, to learn and to understand.

The website/system will be public for everyone who wants to gain information about modern architecture of eastern Mediterranean and it can be used to affect the architectural society and people in general.

### 1.3 Chapters overview

The first chapter of the dissertation is introductive and its purpose is to inform the readers about the mesarch lab activities and philosophy as well as the purpose of the dissertation and its general contribution to the labs' activities and the general architectural society. In the second chapter a research on current functional architectural web applications/systems is presented. In the third chapter the system's requirements analysis and specifications are discussed regarding the users of the system, the constraints of the system and the information models. In the fourth chapter the architecture, the design tools, the database and the interface of the system are presented. In the fifth chapter the implementations tools are presented as well as the database and the interface of the system regarding the actual implementation. In the sixth chapter the deployment of the system and the evaluation methods used are presented. Finally, in the seventh chapter the conclusions about the system and future opportunities are stated.

### 1.4 Acronyms and abbreviations

- DB: Database
- WA: Web application
- HTML (Hyper Text Markup Language): The programming language used in the WA's interface implementation
- CSS (Cascade Style Sheets): is a style sheet language used for describing the look and formatting of the interface

- SQL (Structured Query Language): Programming language used to manage the data of the system

- CMS (Contend Management System): A CMS is a computer application that allows publishing, editing and modifying content, organizing, deleting as well as maintenance from a central interface

- JavaScript: Scripting programming language to add dynamic behavior to the WA

# Κεφάλαιο 1

## Εισαγωγή

1.1 Επισκόπηση διπλωματικής εργασίας 13

1.2 Συνεισφορά διπλωματικής εργασίας 15

1.3 Επισκόπηση κεφαλαίων 16

1.4 Ακρώνυμα και συντομογραφίες 17

### 1.1 Επισκόπηση διπλωματικής εργασίας

**Το εργαστήριο Mesarch**

Το εργαστήριο mesarch, είναι ένα εργαστήριο που μελετά την ιστορία και θεωρία της μοντέρνας αρχιτεκτονικής της ανατολικής μεσογείου και έχει ως στόχο να διευρύνει την ουσιαστική κατανόηση του μοντερνισμού στην περιοχή. Το εργαστήριο ασχολείται με ερευνητικά προγράμματα και την οργάνωση ακαδημαϊκών δραστηριοτήτων όπως εκθέσεις και ακαδημαϊκά συνέδρια.

Ο όρος Mesarch είναι ένα κράμα διαφορετικών εννοιών: Mesogeios, Middle East, Architecture και Archive. Ο όρος υπάρχει επίσης και στη βοτανολογία ως «η ωρίμανση του κέντρου τόσο προς το εσωτερικό όσο και ως προς το εξωτερικό», και αυτό ισχύει ως προς την γενική ιδέα του εργαστηρίου όπου οι έρευνες του πρέπει να μένουν μακριά από κέντρο-περιφερειακά μοντέλα, έτσι ώστε το εργαστήριο να μπορεί να διαχειριστεί την ιστορία του αρχιτεκτονικού μοντερνισμού της ανατολικής μεσογείου ως πολύπλοκα εμπλεκόμενη με τις ιστορίες Ευρωπαϊκών και άλλων μοντερνισμών.

Το εργαστήριο διεξάγει ουσιαστικές έρευνες σχετικά με τα εξής διεπιστημονικά πεδία:

- Αρχιτεκτονική και πολιτική του 20$^{ου}$ αιώνα στην ανατολική μεσόγειο

- Ιστορία της αστικής ανάπτυξης και της περιβαλλοντικής βιωσιμότητας και της πολιτιστικής κληρονομιάς
- Ψηφιακές ανθρωπιστικές επιστήμες και αρχιτεκτονικά αρχεία

Το εργαστήριο Mesarch εμπλέκεται σε προγράμματα ιστορικής έρευνας που επικεντρώνονται στην αρχιτεκτονική και την μοντέρνα ιστορία της Κύπρου και τοποθετούν την Κύπρο και την μοντέρνα της αρχιτεκτονική σε διεθνή ακαδημαϊκά συνέδρια.

Η δέσμευση του εργαστηρίου στην ιστορική έρευνα και ιστοριογραφία προωθείται μέσω ενός άρτια εξοπλισμένου εργαστηρίου, το οποίο περιλαμβάνει εξοπλισμό τελευταίας τεχνολογίας για την τεκμηρίωση και ψηφιοποίηση αρχειακού υλικού, και μέσω μίας ομάδας έμπειρης στην αρχειακή έρευνα, τεκμηρίωση και θεωρητική ανάλυση.

**Τρέχουσες πρακτικές**

Το εργαστήριο διαχειρίζεται πληροφορίες που αφορούν projects, media (φωτογραφίες, άρθρα εφημερίδων κτλ που περιγράφουν τα projects) και actors (αρχιτέκτονες οι οποίοι συνέβαλαν στη δημιουργία των projects) που συνεισφέρουν στη μοντέρνα αρχιτεκτονική της ανατολικής μεσογείου. Τα projects περιλαμβάνουν βασικές πληροφορίες όπως τίτλο, τύπο κτιρίου, περιγραφή, τοποθεσία, ημερομηνία κατασκευής κτλ, αλλά και πληροφορίες όπως ελέγχους πεδίου, αναφορές/παραπομπές, σχετικά media και actors. Με λίγα λόγια, τα project παρέχουν ουσιαστικές πληροφορίες σε πιθανούς μελετητές που ενδιαφέρονται να μελετήσουν την αρχιτεκτονική του 20$^{ου}$ αιώνα στην ανατολική μεσόγειο. Τα media, περιλαμβάνουν φωτογραφίες, άρθρα περιοδικών και εφημερίδων κτλ αλλά και κάποιες επιπλέον πληροφορίες όπως όνομα, δημιουργό, ημερομηνία δημιουργίας, τοποθεσία, τύπο κτλ και σε γενικές γραμμές, περιγράφουν οπτικά τα projects. Εν τέλει, οι actors περιλαμβάνουν πληροφορίες σχετικά με αρχιτέκτονες που συνεισέφεραν στη δημιουργία των projects, όπως όνομα actor, φωτογραφία, βιογραφικό, εκπαίδευση, τοποθεσίες δραστηριότητας κτλ. Οι actors μπορούν να συσχετίζονται με projects ή και άλλους actors με τους οποίους συνεργάστηκαν κατά τη διάρκεια της καριέρας τους. Το εργαστήριο μέχρι πρόσφατα δεν είχε ένα ολοκληρωμένο σύστημα όπου τα μέλη του θα μπορούσαν να διαχειριστούν και να παρουσιάζουν αυτές τις πληροφορίες και έτσι προέκυψε η ανάγκη δημιουργίας ενός ψηφιακού συστήματος.

### Σκοπός διπλωματικής εργασίας

Το εργαστήριο συνεργάζεται με ακαδημαϊκό προσωπικό του τμήματος πληροφορικής του Πανεπιστημίου Κύπρου, με στόχο την ανάπτυξη ενός πρωτοποριακού πληροφοριακού μοντέλου μέσω του οποίου θα μπορεί να παρουσιάζει και να αξιολογεί την αρχιτεκτονική κληρονομιά του εικοστού αιώνα.

Για αυτό το λόγο, η τρέχουσα διπλωματική εργασία επικεντρώνεται στη δημιουργία μιας διαδικτυακής εφαρμογής η οποία θα εξυπηρετεί ως ερευνητικό εργαλείο για ακαδημαϊκούς και φοιτητές που ενδιαφέρονται να μελετήσουν την μοντέρνα αρχιτεκτονική της ανατολικής μεσογείου, ως επίσης θα είναι σε θέση να παρουσιάζει τις δραστηριότητες του εργαστηρίου.

Η οργάνωση της ιστοσελίδας γύρω από projects, media και actors, αλλά και η υλοποίηση διαφόρων εργαλείων αναζήτησης έγιναν με σκοπό να ευκολύνουν την μελέτη της μοντέρνας αρχιτεκτονικής της ανατολικής μεσογείου. Η ιστοσελίδα στοχεύει να γίνει το κέντρο ανταλλαγής ακαδημαϊκών γνώσεων γύρω από την ιστορία-θεωρία της αρχιτεκτονικής στην περιοχή. Η ιστοσελίδα προορίζεται για χρήση από αρχιτέκτονες, φοιτητές αρχιτεκτονικής, αλλά και ακαδημαϊκούς.

### Αξιολόγηση συστήματος

Το σύστημα υλοποιήθηκε σύμφωνα με της ανάγκες και τους στόχους που καθορίστηκαν κατά τη διάρκεια συναντήσεων με τα μέλη του εργαστηρίου Mesarch. Αφού τελείωσε η υλοποίηση, εγκαταστάθηκε στον server του τμήματος αρχιτεκτονικής του Πανεπιστημίου Κύπρου και έτσι δόθηκε η ευκαιρία στους τελικούς χρήστες να αλληλεπιδράσουν με αυτό. Κατά τη διάρκεια αυτής της περιόδου, οι χρήστες μπόρεσαν να αποστείλουν εισηγήσεις για βελτίωση όσο αφορά λειτουργικά και σχεδιαστικά θέματα και οι κατάλληλες λειτουργίες μπόρεσαν να υλοποιηθούν και να βελτιωθούν έτσι ώστε να εξυπηρετήσουν τις ανάγκες των χρηστών. Στο τέλος αυτής της περιόδου, ένα ερωτηματολόγιο χορηγήθηκε στους χρήστες έτσι ώστε να καταστεί δυνατή η γενική αξιολόγηση του συστήματος.

## 1.2 Συνεισφορά εργασίας

Η τρέχουσα διπλωματική εργασία εξυπηρετεί τις ανάγκες του εργαστηρίου Mesarch διαμέσου της χρήσης μιας διαδικτυακής εφαρμογής. Ένα πρωτοποριακό σύστημα βάσης δεδομένων σχεδιάστηκε και υλοποιήθηκε για να συνεισφέρει στη διαχείριση και παρουσίαση των δεδομένων του εργαστηρίου, και μία διαδικτυακή διεπαφή σχεδιάστηκε για να επιτρέπει την παρουσίαση των δεδομένων στην αρχιτεκτονική κοινότητα και στο κοινό. Επιπλέον, η διαδικτυακή διεπαφή, διαμέσου γραφικού περιβάλλοντος, επιτρέπει στα μέλη του εργαστηρίου να διαχειριστούν όλες τις πληροφορίες του συστήματος με ένα φιλικό προς τον χρήστη τρόπο, αφού οι χρήστες είναι άτομα χωρίς ιδιαίτερες προγραμματιστικές ικανότητες που δεν έχουν την απαραίτητη τεχνογνωσία ώστε να αλληλεπιδρούν απευθείας με τη βάση δεδομένων.

Το σύστημα εγκαταστάθηκε στον server του τμήματος αρχιτεκτονικής του Πανεπιστημίου Κύπρου και τα μέλη του εργαστηρίου είχαν την ευκαιρία να αλληλεπιδράσουν με αυτό. Με αυτό τον τρόπο, μπόρεσε να ελεγχθεί η χρησιμότητα, η ευκολία χρήσης αλλά και η αποδοτικότητα του συστήματος. Η διαδικασία αξιολόγησης έδειξε θετικά αποτελέσματα, αφού οι χρήστες ήταν ικανοποιημένοι με το σύστημα, το βρήκαν φιλικό προς το χρήστη, εύκολο και απλό στη χρήση, και εύκολο στην εκμάθηση και κατανόηση.

Η ιστοσελίδα/διαδυκτιακή εφαρμογή θα είναι δημόσια προς χρήση από τον οποιονδήποτε που ενδιαφέρεται να αποκτήσει πληροφορίες σχετικά με την μοντέρνα αρχιτεκτονική της ανατολικής μεσογείου και μπορεί να χρησιμοποιηθεί για να επηρεάσει την αρχιτεκτονική κοινότητα.

### 1.3 Επισκόπηση κεφαλαίων

Το πρώτο κεφάλαιο της διπλωματικής εργασίας είναι εισαγωγικό και έχει ως σκοπό να ενημερώσει τους αναγνώστες σχετικά με την φιλοσοφία και τις δραστηριότητες του εργαστηρίου Mesarch, ως επίσης και για το σκοπό της διπλωματικής εργασίας και τη συνεισφορά της προς το εργαστήριο και την αρχιτεκτονική κοινότητα. Στο δεύτερο κεφάλαιο παρουσιάζεται μια έρευνα σχετικά με τρέχοντα αρχιτεκτονικά διαδικτυακά συστήματα. Στο τρίτο κεφάλαιο αναλύονται οι ανάγκες και οι προδιαγραφές του συστήματος όσο αφορά τους χρήστες, τους περιορισμούς και τα πληροφοριακά μοντέλα. Στο τέταρτο κεφάλαιο, παρουσιάζεται η αρχιτεκτονική του συστήματος, τα εργαλεία σχεδίασης, η βάση δεδομένων και η διεπαφή του συστήματος. Στο πέμπτο κεφάλαιο, παρουσιάζονται τα εργαλεία υλοποίησης ως επίσης και η βάση δεδομένων αλλά και η διαδικτυακή διεπαφή όσο αφορά την υλοποίηση. Στο έκτο κεφάλαιο παρουσιάζονται η εγκατάσταση και η υλοποίηση του συστήματος. Τέλος, στο έβδομο κεφάλαιο αναλύονται τα συμπεράσματα της εργασίας και παρουσιάζονται μελλοντικές προοπτικές.

## 1.4 Ακρώνυμα και συντομογραφίες

- DB: Database

- WA: Web application

- HTML (Hyper Text Markup Language): Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την υλοποίηση της διαδικτυακής διεπαφής

- CSS (Cascade Style Sheets): Γλώσσα που χρησιμοποιήθηκε για την μορφοποίηση των επιμέρους στοιχείων της διεπαφής

- SQL (Structured Query Language): Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την διαχείριση των δεδομένων του συστήματος

- CMS (Contend Management System): Ένα CMS είναι μια εφαρμογή η οποία επιτρέπει την παρουσίαση, επεξεργασία, οργάνωση, διαχείριση, διαγραφή και συντήρηση πληροφοριακού περιεχομένου διαμέσου μιας κεντρικής διεπαφής

- JavaScript: Προγραμματιστική γλώσσα που προσθέτει δυναμική συμπεριφορά στην εφαρμογή

# Chapter 2

## Related work

2.1 Classification of current systems 19

2.2 Comparison of current systems 23

2.3 Research tools for architectural research 30

### Introduction

At the beginning of this chapter the differences between a static, a dynamic and a content management system (CMS) webpage are distinguished and then some current static dynamic and CMS pages will be examined and compared regarding their functionalities and their technical characteristics. After that, the pros and cons of every type of page will be clarified and then I will address the reasons that led me in selecting the current type of page for the implementation of MARE WA.

Additionally, there is a reference on previous work done on the MARE WA by another student of the computer science department in the past year that was very important for me as I could see that as an extended prototype of what I was supposed to deliver.

Finally, I present some previous work related to the current system like documentation and cataloguing models and digital libraries, a universal system for documentation that is used in MARE system and I explain why MARE will be more innovative than the current systems.

Web page implementation systems can be classified in two main categories: static and dynamic pages. Dynamic pages are those that have a dynamic content, which can be changed very easily and frequently. It allows the owner of the page to insert, delete or update the information of the site without any knowledge in web programming. The dynamic sites use a DB system to store permanently their data and every time a specific page of the site is loaded, the data are retrieved from the DB system; they are formatted and presented on the browser. The owner of the site can use a web interface to insert, delete or update the database. When a specific page loads, it connects to the DB on the server and retrieves the necessary information. The information retrieved is then structured using HTML and formatted using CSS.

An example of a dynamic system is the CMS. CMSs can be used for a wide variety of purposes like news sites, retail sales companies' sites, and advertisements in web pages. CMSs are WAs that allow the online modification of websites content. However, contrary to the dynamic pages, the design and the update or modification of the outline of the website can be done without any HTML/CSS knowledge, but using the CMS system alone, which simplifies the web design for everyone.

Static pages are the ones that their content is predefined by the programmer and cannot change via an interface, but only by updating the source code. These types of pages present the same information to all users. The administrator must possess web programming skills and web technologies knowledge in order to insert, delete or update content or apply any changes in general on the page for the source code must be modified. However, static pages can have simple dynamic behavior that does not change the actual information that is presented, but uses JavaScript code to add interaction to the page. For example, using JavaScript to show or hide different elements of the page.

As a conclusion, websites that are stored on a DB system but do not implement an update interface and the same content is presented to all users, can be labeled as static too.

## 2.1 Classification of current systems

The following are examples of static websites:

Schott glass made of ideas website (http://www.schott.com) is a worldwide company of German origin that produces various glass products. It inspires people in a wide span of areas like electronics, optics, automotive and many other industries. Projects of this company are organized by category (architectural, optics, lightning etc).  A lot of other media are also listed, like various publications, videos, magazines, information about the company and even research information about certain types of glass. In general, the website is organized around five categories  (Products, About, Events, News, Worldwide addresses). Even though it is a large website containing a lot of information, it is implemented only in HTML and so it is static.

Hiroshi Sugimoto website (http://www.sugimotohiroshi.com) is a personal website of Hiroshi Sugimoto gives information about his portfolio and the projects he participated in.

Additionally, his biography, bibliography and all the events that relate to him are presented as well. It is a relatively small website with little information. It is also implemented in HTML only and therefore it is static.

Roadarch website (http://www.roadarch.com) is a personal site of Debra Jae Seltzer that presents images related with roadside architecture. This architecture is subcategorized in buildings, statues, signs and others. It is a huge website (2,300 HTML files and 60,000 photos) and is implemented only in HTML.

Great Buildings website (http://www.greatbuildings.com) is a site that lists the best buildings, architects, architecture books, and 3D models categorized by types of architectural buildings and areas. It is a relatively large website providing much information. It is as well implemented only in HTML, thus it is static.

Artlex (http://www.artlex.com) website is a side that serves as a dictionary of art terms, on a worldwide scale and throughout the years. It includes an index of terms and names' starting letters that help the users navigate and ease their search. It is a rather large website but implemented only in HTML and so it is static.

The following are examples of dynamic websites:

Archigram (http://archigram.westminster.ac.uk/) website is a site of an architectural group which showcases information regarding its activity (projects, drawings, ephemera, magazines, shows, bibliography-publications) to the public for research purposes. It is implemented in PHP and MySQL. However, the website is static because even though the data are stored in the DB, they are presented in a predefined way; the users cannot change these data but only the administrator of the website can.

Marcel Breuer digital archive website (http://breuer.syr.edu/) is a personal site of the architect Marcel Breuer, where he presents his work. The site is organized around finished and undergoing projects and provides access to drawings, letters and photographs. It is implemented in PHP and MySql.

Sadler Brown Architecture website (http://www.sadlerbrown.co.uk/) is a site of an architectural company in England in which its projects are presented. There is also information about the personnel, news and the company profile. It is implemented in MS Visual C# with an SQL Server Database.

The following are examples of CMS websites:

Let Us Serve you website (http://www.letusserveyou.com/) is a personal site of Joel Harner, who offers house related services such as house renovation and maintenance. This site is organized around six categories: Services offered, most famous houses he worked on (portfolio), custom houses, information about him and contact details. It is a Joomla-based page implemented in PHP and MySQL. It is a dynamic page that the administrators can change its content.

Group Home Deal website (http://www.grouphomedeal.com/) is a site of a company that makes offers to its' customers on current projects in India depending on the number of individuals that will join each project. This site stores the projects, its' customers that register on the projects (investors) the land owners and the brokers. Additionally, it keeps an archive of all the deals of the past as well as the past offer groups that were created. It is also a Joomla-based implemented in PHP and MySQL. It is a dynamic page that the administrators can change its content.

Arch Daily website (http://www.archdaily.com/ ) is a site that presents the latest news in the architectural society concerning projects, products, events, interviews and competitions. It offers knowledge and inspiration to its visitors, who can register and upload their own projects, news, awards and events. It is organized around six main categories: projects, news, articles, software, materials and interviews. It is also a Joomla-based page implemented in PHP and MySQL. It is a dynamic page and the administrators can change its content.

Architizer website (http://www.architizer.com/) is a social network that connects numerous architects all over the world. Architects, as well as architectural companies, can register and upload and share their work with other architects and possible customers. This site is divided in five main categories: news, projects, products, jobs and awards. This is another case of a Joomla-based page implemented in PHP and MySQL. It is a dynamic page that the administrators can change its content.

Homedit website (http://www.homedit.com/) is a site that provides architectural articles and presents the latest news on interior design and furniture. There are over than 10,000 articles and 60,000 images and ideas listed in this page. There are 9 main categories: Best of, apartments, architecture, bathroom, DIY, kitchen, office, ideas, living, how to and furniture. The users can submit an architectural idea by sending an e-mail to the site administrator accompanied with a description and a few images. It is a Word Press-based website implemented in PHP and MySQL. It is a dynamic page that the administrators can change its content.

Archello website (http://www.archello.com/) allows individuals and companies to share their knowledge in architecture. Individuals and companies can establish collaborations on projects, products or materials; and they can exhibit their profile to the architectural industry. There are six main categories: projects, products, materials, collections, companies, events. It is also a Word Press-based website implemented in PHP and MySQL. It is a dynamic page that the administrators can change its content.

Contemporist website (http://www.contemporist.com) is a contemporary world site focusing on architecture, design, art and travels. This is one more case of a Word Press-based website implemented in PHP and MySQL. It is a dynamic page that the administrators can change its content.

## 2.2 Comparison between current systems

In the following table, I present the programming language and SQL technologies used to implement the web pages explained before. The web pages are grouped by type (static/dynamic/CMS). Observing the table, one can conclude that static web pages use plain HTML as programming language and they don't use a database at all. On the other hand, all but one dynamic/CMS web pages use PHP server side scripting language in addition to HTML, and MySQL database. The only dynamic web page that does not use PHP/MySQL is Sadler Brown Architecture which uses ASP.Net, Microsoft's server side language. To sum up, five pages are static, three are dynamic and 7 are dynamic/CMS. Nine pages are implemented in PHP/MySQL and one page in ASP.Net.

| Application type | Project | Web page | Prog.Language | Database |
|---|---|---|---|---|
| Static | **P1.Schott** | http://www.schott.com | html | - |
| | **P2.Hiroshi Sugimoto** | http://www.sugimotohiroshi.com | html | - |
| | **P3.Agilitynut** | http://www.agilitynut.com/roadside.html | html | - |
| | **P4.Great Buildings** | http://www.greatbuildings.com/gbc.html | html | - |
| | **P5.Artlex** | http://www.artlex.com/ | html | - |
| | | | | |
| Dynamic | **P6.Archigram** | http://archigram.westminster.ac.uk | php | MySQL |
| | **P7.Marcel Breurer** | http://breuer.syr.edu/ | php | MySQL |
| | **P8.Sadler Brown Architecture** | http://www.sadlerbrown.co.uk/ | ASP.NET | - |
| | | | | |
| Dynamic - CMS | **P9.Let us design/build** | http://www.letusserveyou.com/ | php | MySQL |
| | **P10.GroupHomeDeal** | http://www.grouphomedeal.com/ | php | MySQL |

| | | | | |
|---|---|---|---|---|
| | **P11.Arch daily** | http://www.archdaily.com/ | php | MySQL |
| | **P12.Architizer** | http://www.architizer.com/ | php | MySQL |
| | **P13.Homedit** | http://www.homedit.com/ | php | MySQL |
| | **P14.Archello** | http://www.archello.com/ | php | MySQL |
| | **P15.Contemporist** | http://www.contemporist.com | php | MySQL |

The following table presents the main entities that exist in the web pages mentioned before. Two entities are popular amongst the 15 sites, projects and people. Regarding the other entities, every site has its own dedicated ones.

| **Webpage** | **Projects (10)** | **People (6)** | **Products (3)** | **Events (3)** | **Photographs/ Drawing (3)** | **Bibliography (2)** | **Architecture (2)** | **Other entities** |
|---|---|---|---|---|---|---|---|---|
| **P1.Schott** | x | | | x | | x | x | x |
| **P2.Hiroshi Sugimoto** | | | | | | | | |
| **P3.Agilitynut** | x | | | | | | | |
| **P4.Great Buildings** | x | x | | | | | | |
| **P5.Artlex** | x | | | | | | | |
| **P6.Archigram** | x | x | x | | | | x | |
| **P7.Marcel Breurer** | x | x | x | | | x | | |
| **P8.Sadler Brown** | | | | | | | | |
| **P9.Let us design/build** | | | | | x | | x | |
| **P10.GroupHomeDeal** | x | x | | x | x | | | x |
| **P11.Arch daily** | x | x | | x | x | x | | x |
| **P12.Architizer** | x | x | x | x | x | | | |
| **P13.Homedit** | x | | | | x | | | |
| **P14.Archello** | x | x | x | x | x | | | |
| **P15.Contemporist** | x | | | | x | | | |

In the following table, I present some of the functionalities encountered in the web pages mentioned before. One can observe that the majority of pages offer a search tool and an advanced search tool to help users find faster the information they want. However only four pages offer result filtering option. Almost half of the pages offer login and admin panel facilities. Few pages contain a sitemap or a version in another language. Some pages offer the option to follow them or share on a social network. We can observe that no web page offers all of the listed functionalities.

| Webpage | search | adv search | filters | login | admin panel | sitemap on page | sitemap | other languages | follow on network | share on network |
|---|---|---|---|---|---|---|---|---|---|---|
| **P1.Schott** | x |  |  | x |  | x | x | x | x | x |
| **P2.Hiroshi Sugimoto** |  |  |  |  |  |  |  |  |  |  |
| **P3.Agilitynut** | x |  |  |  |  |  |  |  |  |  |
| **P4.Great Buildings** | x | x |  |  |  |  |  |  |  |  |
| **P5.Artlex** | x |  |  |  |  |  |  |  |  |  |
| **P6.Archigram** | x | x | x |  |  |  | x |  |  |  |
| **P7.Marcel Breurer** | x | x | x |  |  | x |  |  |  |  |
| **P8.Sadler Brown** |  |  |  |  |  |  |  |  |  |  |
| **P9.Let us design/build** |  |  |  |  | x |  | x |  | x | x |
| **P10.GroupHomeDeal** | x | x |  | x | x |  |  | x |  |  |
| **P11.Arch daily** | x | x |  | x | x | x |  | x |  | x |
| **P12.Architizer** | x | x | x | x | x |  |  |  |  |  |
| **P13.Homedit** | x |  |  |  | x |  |  |  | x |  |
| **P14.Archello** | x | x | x | x | x |  |  |  |  | x |
| **P15.Contemporist** | x |  |  |  | x |  |  |  | x | x |

The following table presents some additional functionalities. Almost half of the pages use AJAX and email technologies but only one offers the option of printing page, downloading content. Very few pages contain Google maps, play videos or display a network of actors. Four web pages offer the user a work place and two web pages allow users to comment on projects. Finally, only two pages implement a recommender system.

| Webpage | | AJAX | print<br>page | videos | download | google<br>map | network of<br>architects | workplace | send email | comments on projects | project<br>recommend |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Static | **P1.Schott** | x | x | x | x | | | | | | |
| | **P2.Hiroshi Sugimoto** | | | | | | | | | | |
| | **P3.Agilitynut** | | | | | | | | | | |
| | **P4.Great Buildings** | | | | | x | | | x | | |
| | **P5.Artlex** | | | | | | | | | | |
| | | | | | | | | | | | |
| Dynamic | **P6.Archigram** | x | | | | | | | x | | |
| | **P7.Marcel Breurer** | | | | | | | | | | |
| | **P8.Sadler Brown** | x | | | | | | | | | |
| | | | | | | | | | | | |
| Dynamic-CMS | **P9.Let us design/build** | | | | | | | | x | | |
| | **P10.GroupHomeDeal** | x | | | | | x | x | | | |
| | **P11.Arch daily** | x | | x | | | | x | x | x | x |
| | **P12.Architizer** | | | | | | x | x | x | | x |
| | **P13.Homedit** | | | | | | | | x | x | |
| | **P14.Archello** | x | | | | x | x | x | | x | |
| | **P15.Contemporist** | | | | | | | | | x | |

To sum up, both types of websites, dynamic or static, are viable and can be widely used, depending on the needs of the person who is interested in developing a WA.

Static websites are in general cheaper and simpler than the dynamic ones due to the fact that only HTML technologies are indispensable to function properly, and no intense maintenance is necessary since they do not implement a database. Its' contents are constant and do not change until a new version of the side is uploaded on the server. The biggest con of these types of sites is that the initial version and any subsequent ones must be implemented by the programmers or by people with relevant knowledge. So, updating or creating a new functionality is difficult because the programmer must do it. However they have a relatively small loading time because they are simple and no additional server costs are needed. To conclude on static websites, it is the best option if the content of it will rarely change.

On the other hand, dynamic websites offer way more functionalities than static ones, provide updates or inserts of new content anytime and anywhere by different users and computers, it is easiest to modify them and in general they provide an integrated system which can be used by users without any skills of web programming. They are database driven web pages. For example, when a user books a ticket for a travel abroad via a web page, the DB on the server is informed and this ticket is not shown as available for the next user who seeks to book it. However dynamic websites are slower and more expensive to implement and maintain due to the fact that a server is indispensable for the DB of the site to be hosted. When the size of the website is relatively large it is better to have a dynamic page because the content of the site can be a few dozens of HTML files that change dynamically instead of a few hundreds or even thousands of static HTML files.

There are two main frameworks for implementing dynamic websites:

ASP.NET: Microsoft's platform for web design/programming. HTML files and server side programming is done in MS visual studio using either VB.net or C#.net and the DB used is MS SQL server. It is expensive because these technologies must be bought.

APACHE (PHP): It’s an open source platform that uses PHP scripting language embedded in the HTML files to add dynamic behavior to a website and the DB used id MySQL. The greatest benefit of these technologies is that are free because they are open source.

### System prototype

A website for mesarch lab was developed in the past in another thesis of the Computer Science Department of University of Cyprus. However, the requirements of the website changed since then and additionally it wasn’t a full functional WA; parts of it were underdeveloped or not developed at all. For the purposes of my own dissertation I kept the old one as an extended prototype, an example that could help me to develop the functionalities specified by the Mesarch lab team, and in general, I used the old one as a guide.

I used the old DB of the previous system developed as a backbone for the new version of the system. Although almost identical, the new version DB is altered slightly in order to provide new functionalities.

However, the web interface was developed again from scratch. No old HTML files were used for several reasons:

- The requirement of responsive design of the site was added, so I had to use different CSS technologies that the ones used in the past
- The interface should be changed to be more user friendly and more modern
- The old one was not fully implemented and key functionalities were missing
- The source code should be written again more clearly

So, the new MARE WA should be dynamic as was the old one due to the fact that the content of the site should be frequently updated by the lab personnel and that in general, the users should manage the WA. As long as the mesarch lab users posses no web development or programming skills, the dynamic type was used.

The framework used was the ASP.net one for several reasons:

- I was familiar with MS SQL Server and the MS SQL Server management studio

- I had some prior experience with the C# programming language (used in ASP.net) but PHP was completely unknown to me

- C# is a strong-typed compiled language and I personally prefer that instead of a scripting language such as PHP.

### 2.3 Research tools for architectural research

State-of-the-art methodologies of architectural history and historiography focus on architecture as profoundly social and cultural project, having decidedly move the field away from stylistic analyses, or single-minded narratives of evolutionary progress, to uncover the complexities of architectural history, engaging into an in depth analysis from multiple perspectives that have forged new and more complex alignments between the history/theory and criticism of architecture, and the scholarly domains of the social sciences and the humanities. These interdisciplinary methodologies employ multiple theoretical tools spanning the disciplines of architectural history, postcolonial theory, environmental history, and cultural studies as demonstrated by the work of various scholars. (Gaonkar, 2001; Appadurai, 2000 and 1996, Hardt and Negri, 2000; Escobar, 1995).

Among the main research topics that prevail in the above field, is the scholarship on Modern architecture, which aims to uncover the fascinating interconnections between architecture and larger sociopolitical and cultural processes such as modernization, decolonization, nation-building, and development in the second half of 20th century. (e.g., McLaren, 2006; Pyla, 2007 and 2006; Cohen and Eleb, 2002; AlSayyad, 2001; Wharton, 2001; Bozdogan, 2001; Vale, 1992; Nabantoglou & Wong, 1997, Scott, 1998; Rabinow, 1995 and 1992). Particularly, up-to-date scholarship aims to situate built works within the larger sociopolitical context that influenced their design and implementation, and to reflect on their social, cultural, and environmental impact (Pyla, 2013) as well as to map transnational flows of ideas, people and capital, and investigate how the practices of architects and/or their patrons (individuals, states, or corporations) have been intertwined with visions of social change, practices of economic development, or even agendas of political power. (Stanek, 2012, Pyla and Phokaides, 2011, Healy and Upton, 2010)

Even though architectural historians have taken grade strides in demonstrating the complex entanglement of modern architecture with cultural and social values, economic and political processes in Europe and the rest of the world, neither heritage studies nor their alignment with digital technologies have incorporated these insights into the increased understanding, interpretation of cultural heritage. MARE system will draw from the above reference work in order to investigate theoretical questions, research challenges and methodological insights of current interdisciplinary research approaches on architectural history, and will translate these insights to create an information model sensitive to the complexities of historiographical research and to put forward new research enquires tailored to the 20th century architectural heritage.

Documentation and cataloguing models are developed through and become the basis of various research activities that include the conservation and documentation of architecture, archival and museum practices (Kent, 2006, Maygene, 2000) as well as in the most recent developments of digital libraries and online collections that have aided the communication and dissemination of data to researchers and wider audiences. A selective evaluation of different cases of information models and online research tools is performed below in order to present their value and their limitations, against which MARE system will exhibit its innovation.

Besides a few common information that are used when cataloguing architectural heritage (name of architect, date of construction, location info etc.), information models are largely customized to meet the needs of their indented use (research, documentation, conservation etc.), the context (e.g. regional, national, transnational) they address and the larger scope of the research project they are developed in. This variability is of great importance as it affects the historical research and the study of architectural heritage, which depend on the documentation, and organization of data as much as on the study of physical objects (buildings, drawings etc.). This in effect has an impact also on the understanding and evaluation of the built environment and its representations.

In order to develop a universal system for the documentation and the evaluation of Modern Architecture in various contexts, DoCoMoMo fabricated a cataloguing schema called Full Documentation Fiche (DoCoMoMo, 2003). Besides the standard description of a building or a landscape (e.g. identification, location info, important dates, original and current condition), the Fiche, promotes the evaluation of a building through a technological, social, aesthetical and cultural perspective. These evaluations aim directly to demonstrate a project’s overall historical value (in a particular, regional or national context) and to justify the importance of its protection and conservation. Additionally, specific fields are provided to support information on people and organizations involved in the design and realization of a project as well as to document the projects' development over time (changes, alterations etc.). In the documentation and evaluation on architectural heritage advanced by DoCoMoMo's Fiche, small value is given to the representations of a project and to the cultural background.

The Marcel Breuer Digital Archive (MBDA) and the Archigram Archival Project (AAP), launched in 2012 and 2010 accordingly, acknowledge the research and cultural value of all forms of tangible architectural heritage, providing access to digital versions of traditional architectural material such as buildings' drawings, and photographs together with other documents such as personal writings, letters, manuscripts, papers, slides (MBDA, 2012); as well as magazines, articles, slides and multi-media material (AAP, 2010). In effect both projects are developed around more complex information models with multiple entities, while using tools (e.g. filtering) to facilitate easier searching and alternative presentations of data by allowing the cross-referencing of digital material with projects and people, thus provide additional information on the network of collaborators. Although both projects exhibit the current state of the art in digital libraries they appear as "boutique projects" (Friedlander, 2009) that do not eventually promote the association of architecture with larger cultural processes and values; as well as larger collaborative research projects in the larger field of digital scholarship.

Contrary to The Marcel Breuer Digital Archive and The Archigram Archive, Archnet (one of the first examples of online digital collections and databases on architectural history launched in 2002) is not focused on the work of a single author or the collaborative project of one group, but it is rather centred on the sharing and dissemination of knowledge on the built environment of a larger cultural group, that of the Muslim societies. Archnet's main elements are the Archnet Digital Library, a resource that consists of both historic archives and documentation on contemporary building trends, that provides access and sharing of visual and textual material; and an online international community of experts (scholars, students, and professionals) in the study of the built environment in Muslim societies. Archnet's weak feature is the Digital Library, which is developed on a poor information model, that does not provide information on the cultural background or the larger network of actors related to the design, realization and use of architecture and its representations. In effect the whole project promotes the traditional understanding of architectural heritage by focusing mostly on built artefacts. However, a strong feature of Archnet is the creation of a common online platform that allows sharing of data and research knowledge while continuously mapping a broad architectural culture that is transnational and inexhaustible in nature. While this is an exciting joint research project of transnational level, Archnet does not reach wider audiences to promote the broader understanding of values associated with the architectural heritage of Muslim cultures.

MARE system fabricating a holistic information model, with multiple entities along the lines of MBDA and AAP, which will be more innovative as it will support autonomous entities (Projects, Actors and Media) that can be directly associated to each other. The relations between these entities will also carry qualitative information that will explain the social, professional or other nature of these relations. In effect this advanced information model will facilitate a more effective mapping of a wider network of actors (i.e., who the actors are, where they practiced/collaborated, etc.) as well as overlapping project's network (i.e., how the project itself created networks of expertise, know-how, etc.) Furthermore the system will support autonomous entities indirectly associated (e.g. through tagging) to the rest of the entities. This feature will offer a way to document and present data not directly related to architectural production, but can inform the cultural background as well as intangible features of architectural heritage. Additionally the project will implement an open and expandable architecture that will allow the accommodation of more entities and different forms of information.

The realized information model will support answering multi-dimensional enquiries from multiple perspectives/disciplines by utilizing the "relationships" and mapping of seamlessly different contexts between artifacts (i.e., Projects, Actors, Media). In order to support this analysis, a number of data transformation mechanisms such as aggregation, filtering or slicing will have to be developed.

This system is promoting digital scholarship at a critical time when cultural heritage is migrating to digital formats, significantly reshaping our relationship with knowledge and culture. In this context, this database is a solid step into the effective preparation of digital humanities and historical research, as well as heritage management and study, in addressing a foreseeable challenge: that of the proliferation of digital cultural content, which will render the examination and analysis possible only through the use of computational technology and methods and will also transform our understanding of social and cultural values. (Borgman, 2007)

# Chapter 3

## Development methodology

3.1 Dynamic systems development 34
3.2 Design objectives 36

### Introduction

In this chapter the software development methodology (DSDM) that was used is explained; Its' principles and are cited, its' benefits are presented, and its' phases are described in detail and the design objectives. Finally, the design objectives are stated.

### 3.1 Dynamic systems development

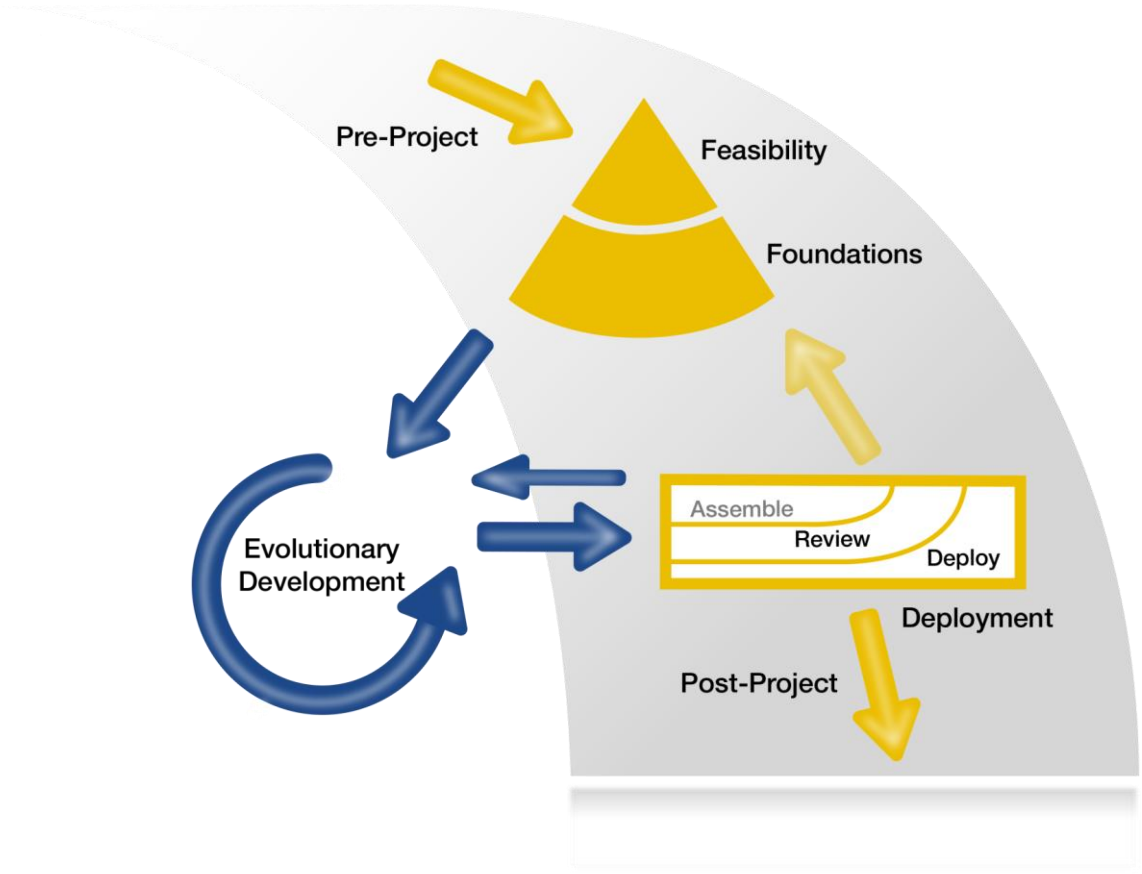
The software development methodology used is the Dynamic Systems Development Method (DSDM). DSDM consortium was formed in 1994 and its purpose was to bring together the best practices for developing systems from the combined experience of Analysts and Developers. This study was concluded in DSDM Agile project framework that was launched in 2014 and it aimed to produce a robust, but agile framework which can be successfully applied across a variety of organizations and project contexts.

There are four main principles that the Agile project focuses on:

- Individuals and interactions: self-organization and motivation are important, as are interactions like co-location and pair programming
- Working software: working software is more useful and welcome than just presenting documents to clients in meetings
- Customer collaboration: requirements cannot be fully collected at the beginning of the software development cycle, therefore continuous customer or stakeholder involvement is very important

- Responding to change: agile methods are focused on quick responses to change and continuous development

There are various benefits of using the DSDM methodology, instead of the traditional ones. Giving customers the option to collaborate interactively during the software development cycle, the risk of building the wrong system is greatly reduced; the final system is more likely to meet the real business requirements. Furthermore, having an active role, user will be better trained and they will be more likely to claim ownership of the system. Finally, the system implementation goes more smoothly than it would go, using traditional methods.



The DSDM methodology can be summarized in four main stages; feasibility, foundations, evolutionary development and deployment of the system. During the feasibility study, the costs, the timescales and the technical solutions are considered and it is decided whether the project should continue or not. During the foundations phase, the system and the functionality scope are defined, and the team is committed to the project. The types/roles of users are defined and the functionalities are prioritized according to the MoSCoW protocol, which summarizes functionalities in four groups; must have, should have, could have, want but won't have. After that, the outline delivery plan is developed. During the evolutionary development, prototypes are developed iteratively to investigate and define functional requirements and to demonstrate that the solution fits the purpose. Additionally, non-functional requirements like security, user access and network facilities are established and the costs/benefits/risks are analyzed. The system is developed iteratively and incrementally to full operational readiness, so the outcome is a tested and documented system. Finally, during the deployment phase the tested system from the previous phase is set up in the user environment and users are trained. User manuals and guides are produced and the developers receive user feedback and views about the system; in this phase, the project is reviewed.

## 3.2 Design objectives

The primary design objective is the simplicity of the system; it will be used by people outside the field of computer science so they must be able to use it with ease, and it must be user friendly. Additionally, the learning time should be short and once the users learn how to use it, they must be able to reuse it with no difficulty; No need of written instruction should be provided to help the users, they must be able to remember how it works. The successful usage rate should be kept high, but in any case error recoverability techniques must be implemented. It must be useful and integrated to help users be more productive and effective; it must satisfy users and fulfill all of their needs. Finally, the system will be used on a variety of devices; computers, tablets and smart phones. The interface design should be responsive in order to adapt to the various screen sizes and provide satisfying user experience.

# Chapter 4

## Feasibility and foundations

4.1 Stakeholders and roles 37

4.2 Information models 39

4.3 Functional requirements 40

4.4 Prioritization of functionalities 45

4.5 Delivery plan 46

### Introduction

In this chapter the MARE system is defined. Specifically, the feasibility and the foundations of the system are viewed in detail, regarding Stakeholders and roles, information models, functionalities, prioritization of functionalities and the delivery plan. Additionally, some use cases of the users are presented.

During the early stages of the requirements analysis multiple meetings with the Mesarch laboratory personnel were set up to discuss the goals of the new system. During the meetings, interviews took place to discuss basic data entities and information models that needed to be stored, managed and presented. Furthermore, the functional requirements that the system should provide were discussed and analyzed. In this chapter, the outcome of the meetings is presented.

### 4.1 Stakeholders and roles

There are two main user profiles that must be implemented, registered user and unregistered one. Registered users are further split into seven categories: administrator, data administrator, webmaster, data entry level 1, data entry level 2, collaborator data entry and simple user.

### Unregistered users

This category includes the visitors of the website that have not created an account. They can navigate in the contents of the page (frontend) and view information and images about projects, media and actors. They can also do an advanced search on the site and send a review/comment on the contact us page. However they cannot see data flagged as private.

### Registered users

Simple user: This category includes the visitors of the website that have created an account but are not given any special permission yet. They can navigate in the contents of the page (frontend) and view information and images about projects, media and actors as well. They can also do an advanced search on the site and send a review/comment on the contact us page. They cannot either see data flagged as private. However they can view a list of recently seen items on certain pages

Administrators: Members of the administrator team are responsible for inserting, deleting and updating the information shown to the public, as well as marking items as private or not. They are in general responsible for the data maintenance of the system. They have access to another site, the backend one, from where they can perform all these actions. They have also the access to the private fields in the frontend site. They can approve data inserted by other users and they can backup/restore the system. However, they have no access to the DB of the system neither they can modify the source code of the WA.

Data administrators, data entry users level 1, webmasters: They have the same permissions as the administrators except that they cannot delete data inserted by other users, neither can they backup nor restore the system.

Data entry users level 2, collaborators data entry: They have the same permissions as the previous three categories except they cannot delete any data and furthermore, the data they insert must be approved by an administrator, a data administrator or a data entry user level 1 before being inserted into the database. Additionally, regarding housekeep data, they can only manage main and secondary construction system and media format.

### 4.2 Information models

There are three major categories/entities that constitute the system: projects, media and actors.

Projects: Architectural projects such as buildings, building complexes and master plans. For each project the system needs to store a lot of information: project ID, title and additional titles, project type and project type details, description, other info and sources, proposal construction and completion dates, address city and state, additional information, main contributor and main contributor reference, GPS longitude and magnitude, mesarch notes and original uses, main and secondary construction systems. Additionally, other critical information about the projects include field reports, media that describe the project, any possible building alterations that took place in the past, any possible changes of use, references on the project and actors that are related with the project. Regarding project type details, original uses, construction systems, field reports, references, alterations and changes of use a project may have more than one entry. A project may have many media that describe it, multiple photographs or drawings etc. Actors can be related with projects if they had been involved in the development of the project in some way, for example architect of the project, designer, patron etc. Multiple actors that contribute in the development of a project can establish a collaboration on that project.

Media: Drawings, photographs, films, newspaper clippings etc that explain the projects. The information that needs to be stored about media is the following: media ID, title, description and other info, actual image, thumbnail image, media type, scale, duration, physical dimensions, digital dimensions, digitization information, condition description, creation and mesarch acquisition dates, format type and other possible formats, city and country, GPS longitude and magnitude, creator, mesarch notes, license, rights and media keywords. A media can be bound on many projects.

Actors: Individuals, teams, firms, companies, government departments etc that had an impact on the creation of the projects. The system needs to store the following information about actors: actor ID, actor name and surname, about the actor and other info, photo and photo caption, birth and death dates, schools (name, year, country) attended, mesarch notes, geographies of practice, collaborators and related projects. Regarding geographies of practice, an actor may have multiple entries and each entry involves city/country, kind of practice and dates of practice. For every project development an actor had an impact on, the project and his role on it are stored. Finally, if he collaborated with another actor during the development of the project, both actor names and the type of collaboration are stored.

### 4.3 Functional requirements

The systems must be capable of presenting all the information stored in database as well as allowing the personnel of the mesarch laboratory to manage this information.

Users that visit the website must be able to view complete listings of projects, media and actors that are stored in the database. They must be allowed to select an individual item from the list and view further details about it. Additionally, they will be able to select specific options that filter the result list. In any case, an additional general search tool must be provided where users can type keywords to find immediately the item they are looking for, so they can skip the listing and filtering process. Users must also be able to view information about the mesarch lab, its purposes, its members, its contact info and location and send a comment/review about the website or about the laboratory in general.

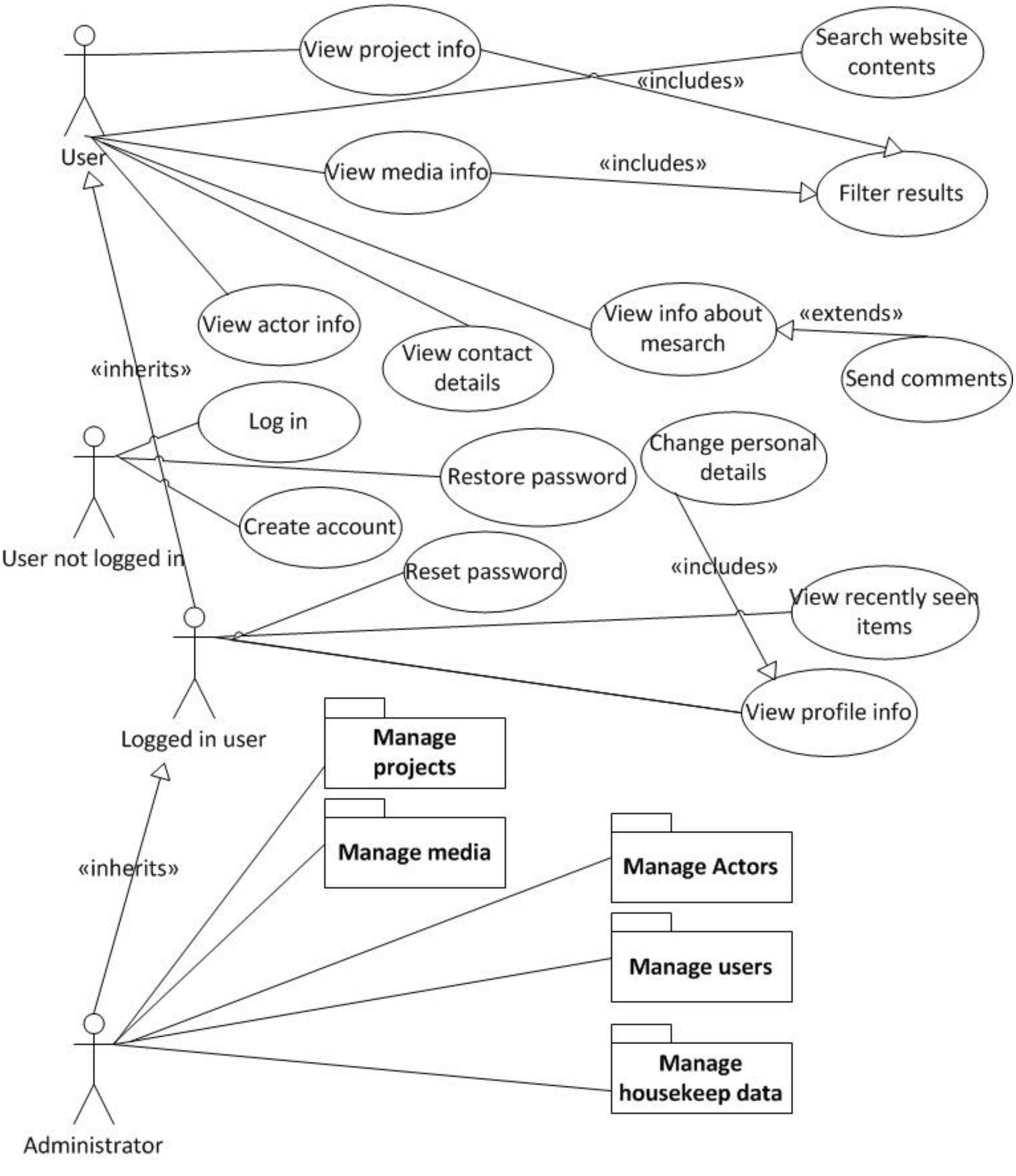
Apart from that users must be able to create an account (if they are not already members), login, view and change personal information, restore and reset password. A list of most popular items must be presented to all the users, and another list of recently seen items must be available for logged in users.

Functional requirements regarding the administrator team include all the above as well as further functionalities explained below.

Administrators must be able to manage all the information stored in the database. This includes adding new items, updating or deleting them, uploading pictures and banner images and managing the housekeep lists (lists that describe the three major entities such as countries list, project type list etc.). They must be able also to manage the users and their respective roles. For example view information for all users, delete a user and promote or demote a user to a different role.

**Use cases**

General use case scenario



The preceding diagram describes the general functionalities of the system. Logged in and not logged in users both inherit user properties and additionally the administrator inherits the logged in user properties.

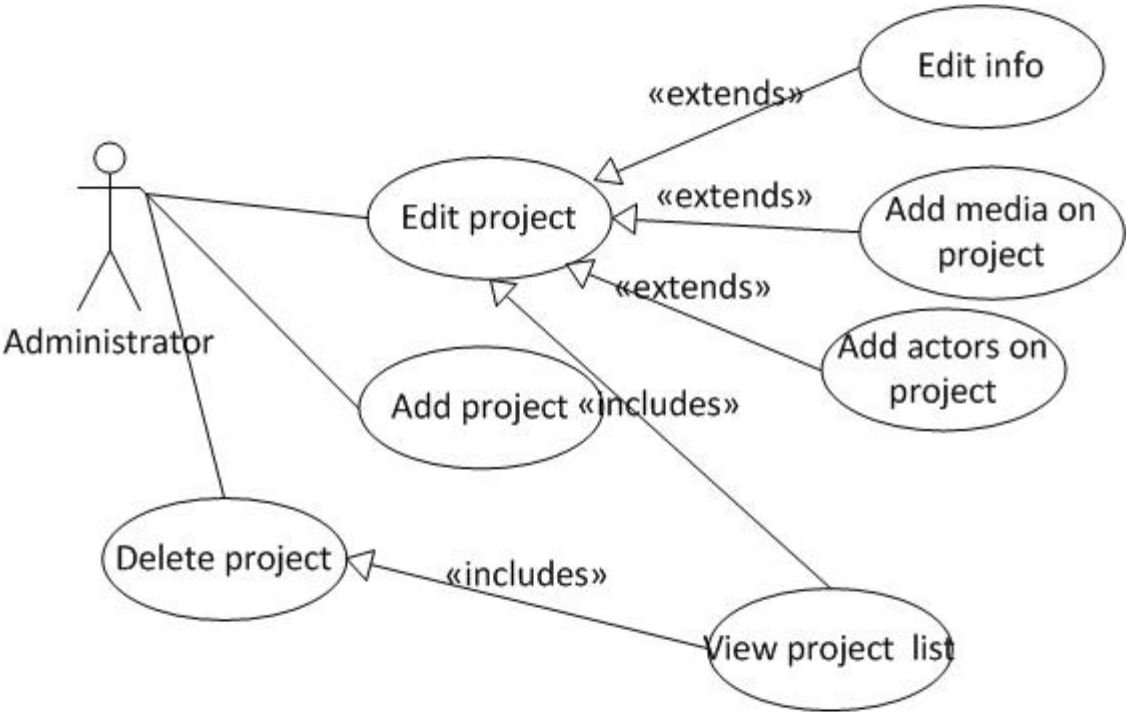
A user can view details of projects media and actors; the main purpose of the website. Additionally it can filter the results on media and projects by selecting certain criteria. Also, he can perform an advanced search by typing just a keyword and selecting an item from the list that is shown. Furthermore, he can view details about the mesarch lab (about mesarch, terms of use, mesarch team) and contact details. In the contact details page he can also send comments/reviews about the site.

A not logged in user can login if he had already signed up in the past, create an account or restore password in case he has forgotten it.

A logged in user can view and change his personal details and reset his password. Additionally he can view a list of recently seen items.

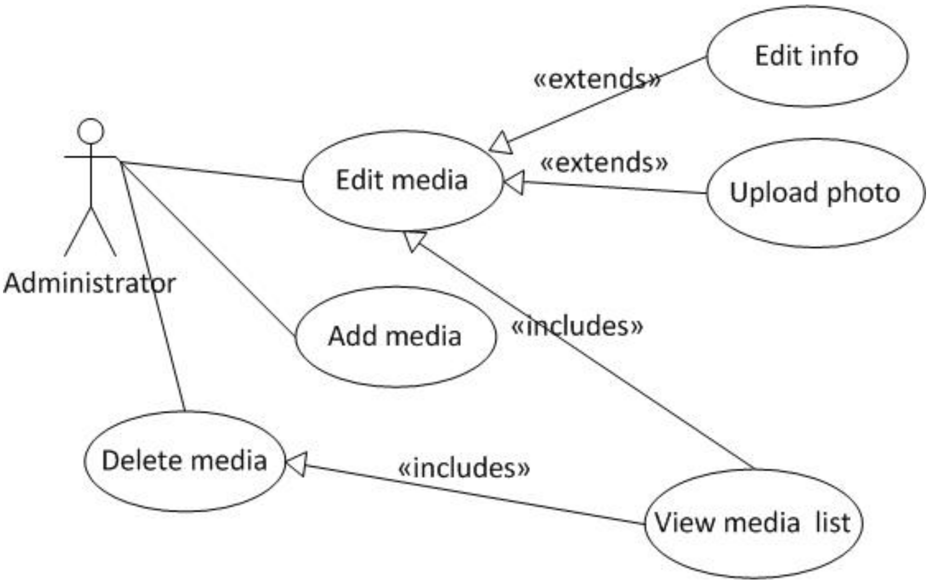
Finally an administrator can manage users, projects, media, actors and housekeep data.

Manage projects:



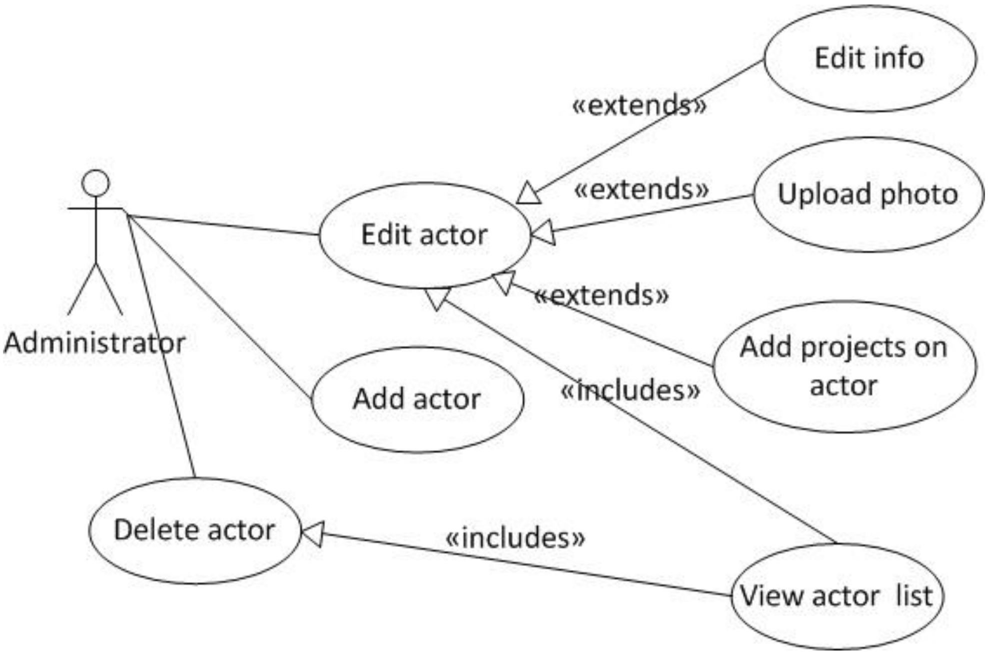
The administrator initially views a list of current projects in the system. He can edit the info of the project and add media and actors to it. Additionally he can delete a project or add a new one.

Manage Media:



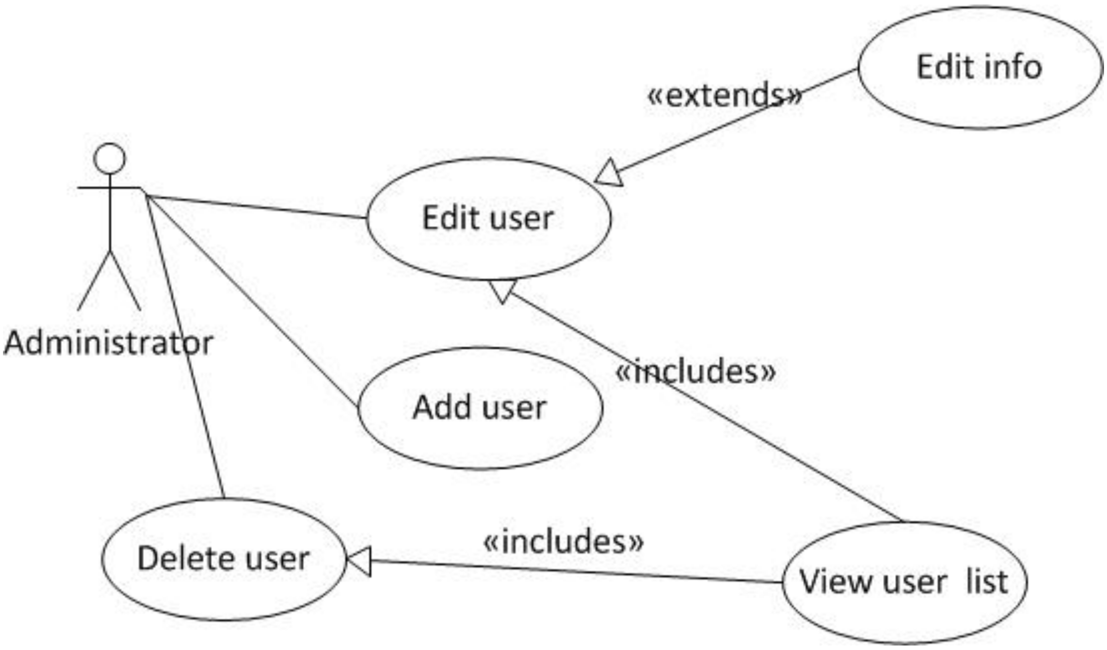
The administrator initially views a list of current media in the system. He can edit the info of the media and upload a photo on this media. Additionally he can delete a media or add a new one.

Manage Actors:



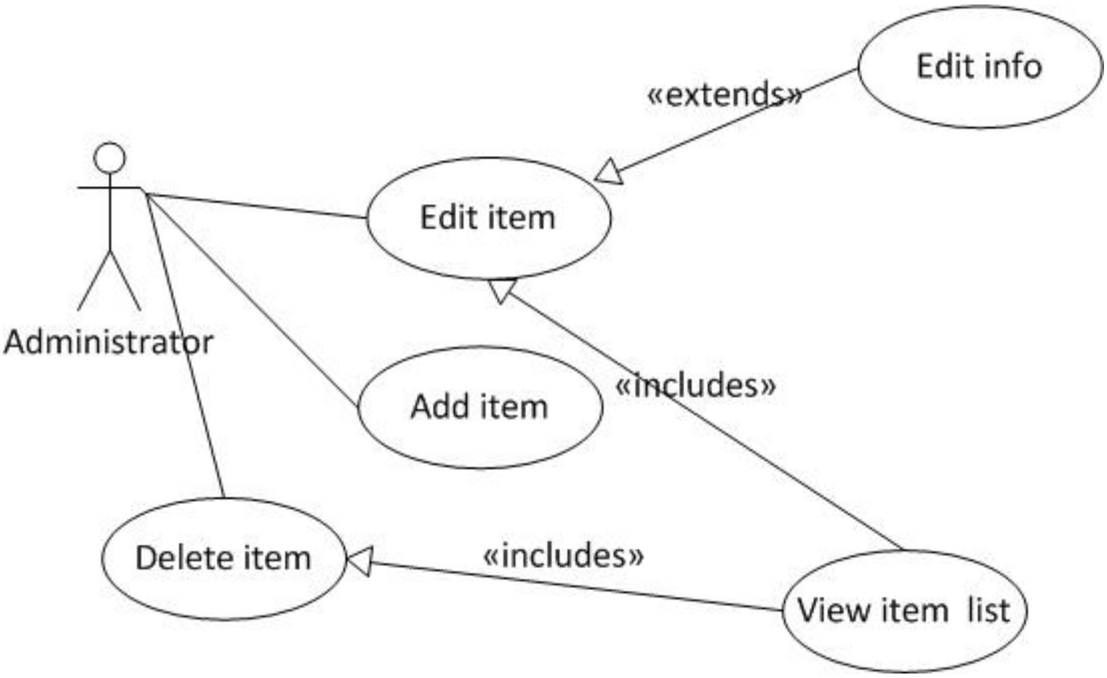
The administrator initially views a list of current actors in the system. He can edit the info of the actor and add projects to it or upload a photo. Additionally he can delete an actor or add a new one.

Manage users:



The administrator initially views a list of current users in the system. He can edit the info of the user. Additionally he can delete a user or add a new one.

Manage items:



The administrator initially views a list of current items in the system. He can edit the info of the item. Additionally he can delete an item or add a new one.

### 4.4 Prioritization of functional requirements

The functional requirements were prioritized and grouped in four categories based on the MoSCoW model. “Must have” functionalities that are indispensable for the system, “should have” functionalities that are not top priority, but are needed, “could have” functionalities that are not indispensable but they improve user experience and “want but won’t have” functionalities that could not be developed.

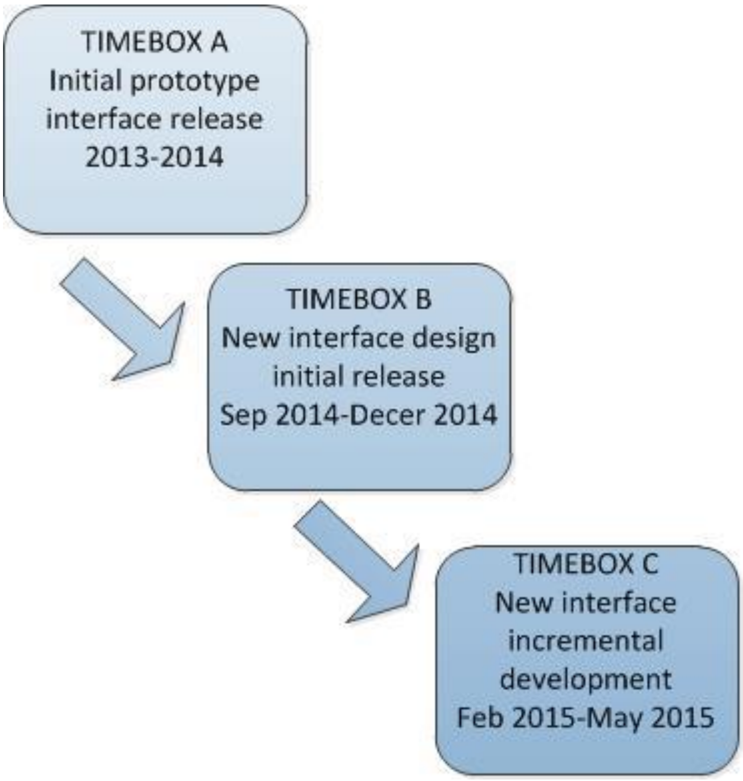
Must have functionalities include projects, media and actor listing as well as dedicated details pages for each individual one; create account/login/logout/restore password option to allow the users to obtain roles in the system; manage project, media, actors, and users to allow the administrator to perform the maintenance of the system; search contents facility to allow users direct access to information.

Should have functionalities include project filtering based on project type, project type details, program type, program use and actors involved in projects on the project listing page; related projects, media type location and creators on the media listing page. Additionally, advanced search facility where users can decide whether they want to search for projects, media and actors by their name/title or by the places they were built/created at.

Could have functionalities include sorting projects, media and actors results and lists, use paging to group results, show results/pages of different sizes, provide contact form, provide contact/about/terms of use information, use JQuery/AJAX technologies to retrieve data asynchronously and a simple recommender system which will present recently seen items.

Finally, two functionalities are included in the last category and won’t be implemented, due to the complexity of the implementation and to the lack of time: A graph, summarizing geographies of practice/collaboration of an actor through the years where his career can be studied and evaluated, and an advanced security protocol, where multiple roles are created to support different user types with different authorizations and data accesses.

### 4.5 Delivery plan



In 2013-2014, TIMEBOX A, an initial prototype interface was released and the users could interact with it. The initial interface didn’t provide responsive design; it did not adapt well to the various screen sizes (smart phones, tablets, computers). Furthermore, it wasn’t as user friendly as it should be hence the need for a new interface design emerged. The new interface design development, TIMEBOX B, begun in September and finished in December 2014. It was user friendlier and adopted a responsive design theme.

In February 2015 a meeting with the users took place where the new interface designed was presented and explained. After a brief period of deployment on the Architecture Department’s server, the incremental development began, TIMEBOX C. The users had the opportunity to interact with the system and insert data. The incremental development procedure was simple: The users interacted with the system for one week; they filled a list of feedback, functional errors and suggestions for improvement, and they forwarded it to me. The following week, considering the feedback received, code improvements and updates took place and the lists were forwarded back with comments; whether and how the issues were solved. This procedure ended in early May, when the final updates were delivered.

# Chapter 5

## System design

5.1 System architecture 47

5.2 Database structure and schema 49

5.2.1 ER Diagrams 50

5.2.2 Relational diagrams 54

5.3 Interface design 56

5.3.1 Prototype interface used 56

5.3.2 New interface design 60

5.4 Security Infrastructure 66

### Introduction

In this chapter the architecture of the system and its main components are presented and the ER and the relational diagrams that explain the database are provided. Additionally, the interface design, the old prototype used and finally the security infrastructure of the system are presented as well.

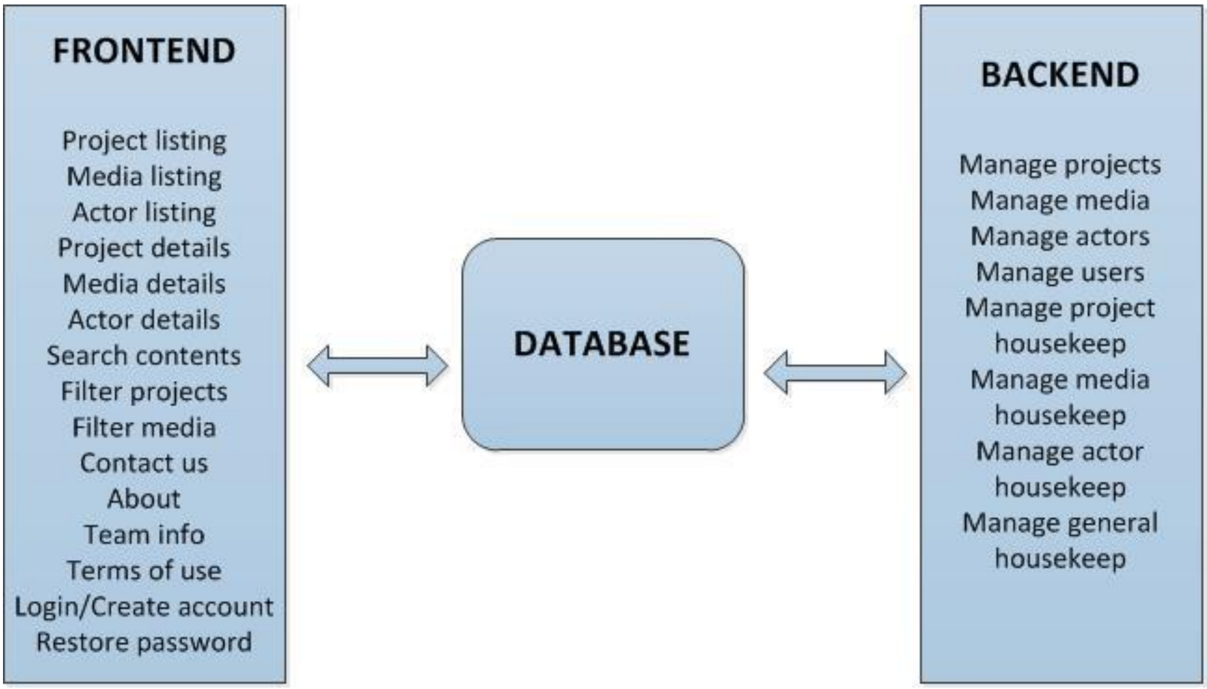
### 5.1 System architecture

The system consists of two major subsystems, the database and the website. The website can be further split in two subsystems, the data presentation system (frontend, the one the public can see) and the data management system (backend, the one the administrator uses). The architecture model used is the client-server where clients are the two website subsystems, the data presentation and the data management system and where server are the database server and the web server that hosts the website systems.

Data presentation system (front end): It is the software that manages the presentation of the information of the system to the simple users and the public in general.

Data management system (back end): It is the software that manages the information of the system. Only administrators have access to this system and it's the system where they can insert, update or delete information.

Relational database system: It is the software that stores all the information of the system. Administrators can manage its contents via the data management system, but they have no direct access to it, they cannot interact directly with the database.



In the preceding figure the frontend and backend system components are presented. The fronted requests and retrieves data from the database for presentation purposes, and the backend requests and retrieves data from the database for management and storing purposes.

There are many components that constitute the fronted site; the most important ones are the project, media and actors listings, where the main information of the system is presented. Users can see a list of pictures, project titles and project types in the project listing page, a list of pictures, media titles and media types in the media listing page and finally a list of photographs, actor names and types in the actor listing page. Additionally, filter bars exists in the project and media listing pages to facilitate easier searching. Project filtering options include project type, project type details, program type, project location and actors involved in the creation of the project; media filtering options include media type, related projects, location and media creator. When on a listing page, users can click a result to navigate to the details page where extensive description of projects, media or actors is presented. A basic component of the frontend system is the advanced search button; users can type a keyword, and a direct search in the contents of the website will be done. Furthermore, users can select an additional option between projects, media, actors and places, to improve search results. Also, general information pages such as terms of use, contact details, about the lab and team information exists. Finally, login/create account facility is provided, to allow users obtain a role in the management of the system.

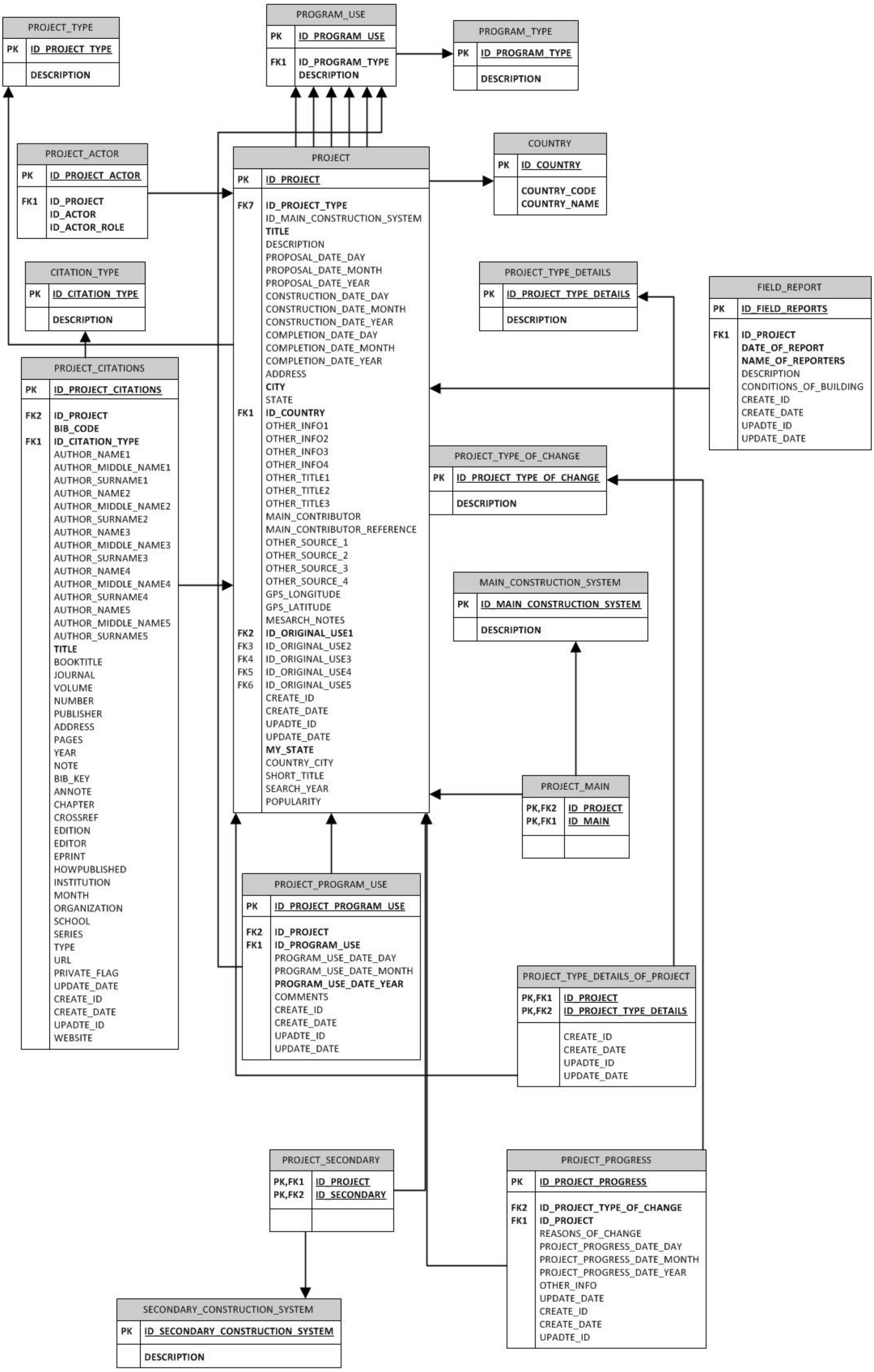
The backend system is responsible for the management of the information of the system. Only administrators can login to this site, and they can manage information regarding projects, media, actors, users and user roles. Furthermore, each main category has housekeep data that can be managed. Project housekeep include project type, project type details, program type, program type details, project type of change, main construction system, and secondary construction system.  Media housekeep include media type, media keywords, media format, and media format category. Actor housekeep include actor type and actor role, and finally, general housekeep data include countries, banner images and citation type.

### 5.2 Database structure and schema

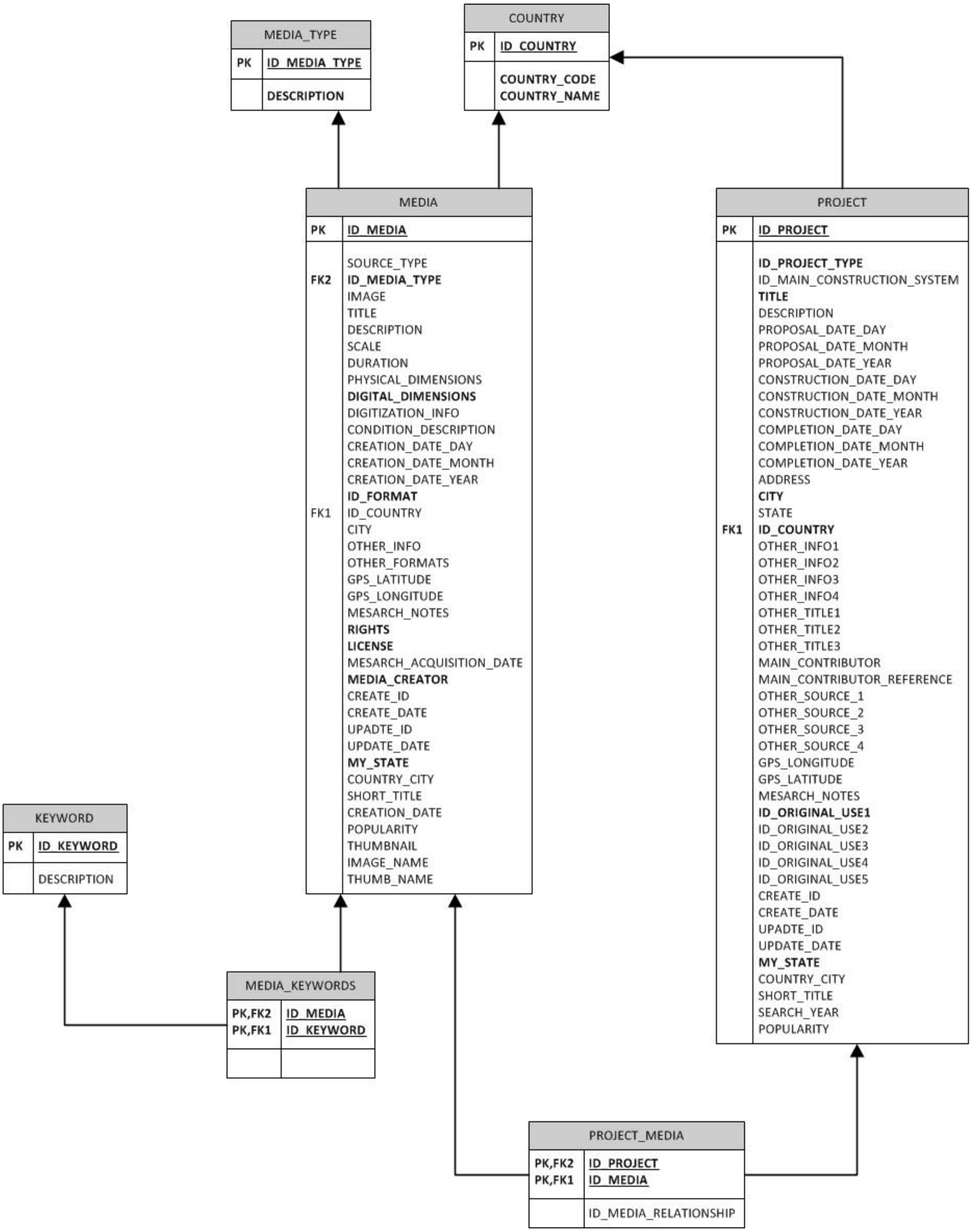
The following figures are the ER and Relational diagrams of the database. The ER diagram describes the database schema and data, and the relationships between the entities. Due to the fact that there are too many tables in the database, the ER diagram is split in 4 pages: projects relations, media relations, actor relations and user relations. The relational diagram contains only the keys (due to space issues) and it indicates how the entities are related, in which primary keys the foreign keys point at.

### 5.2.1 ER diagrams

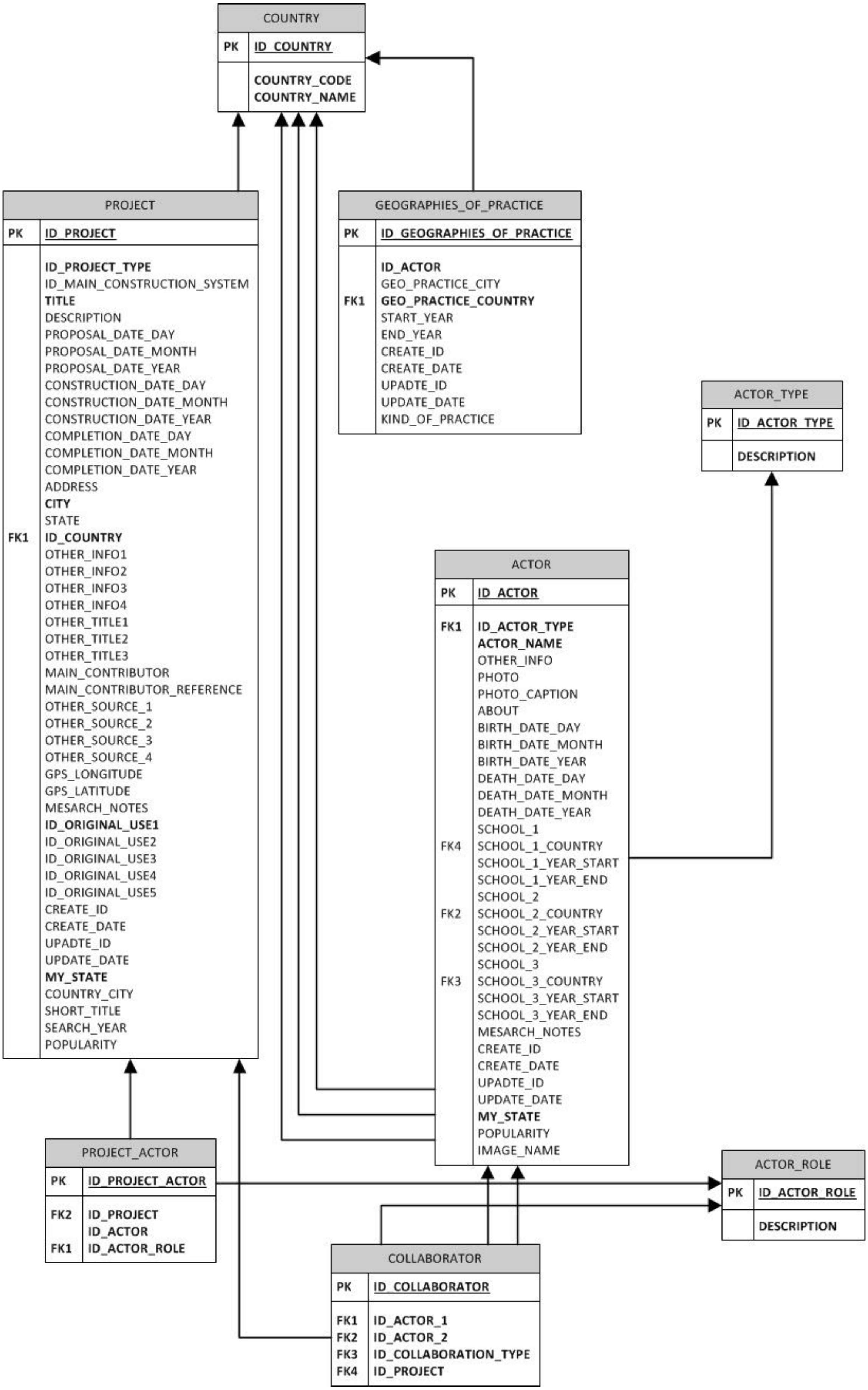
Project relations



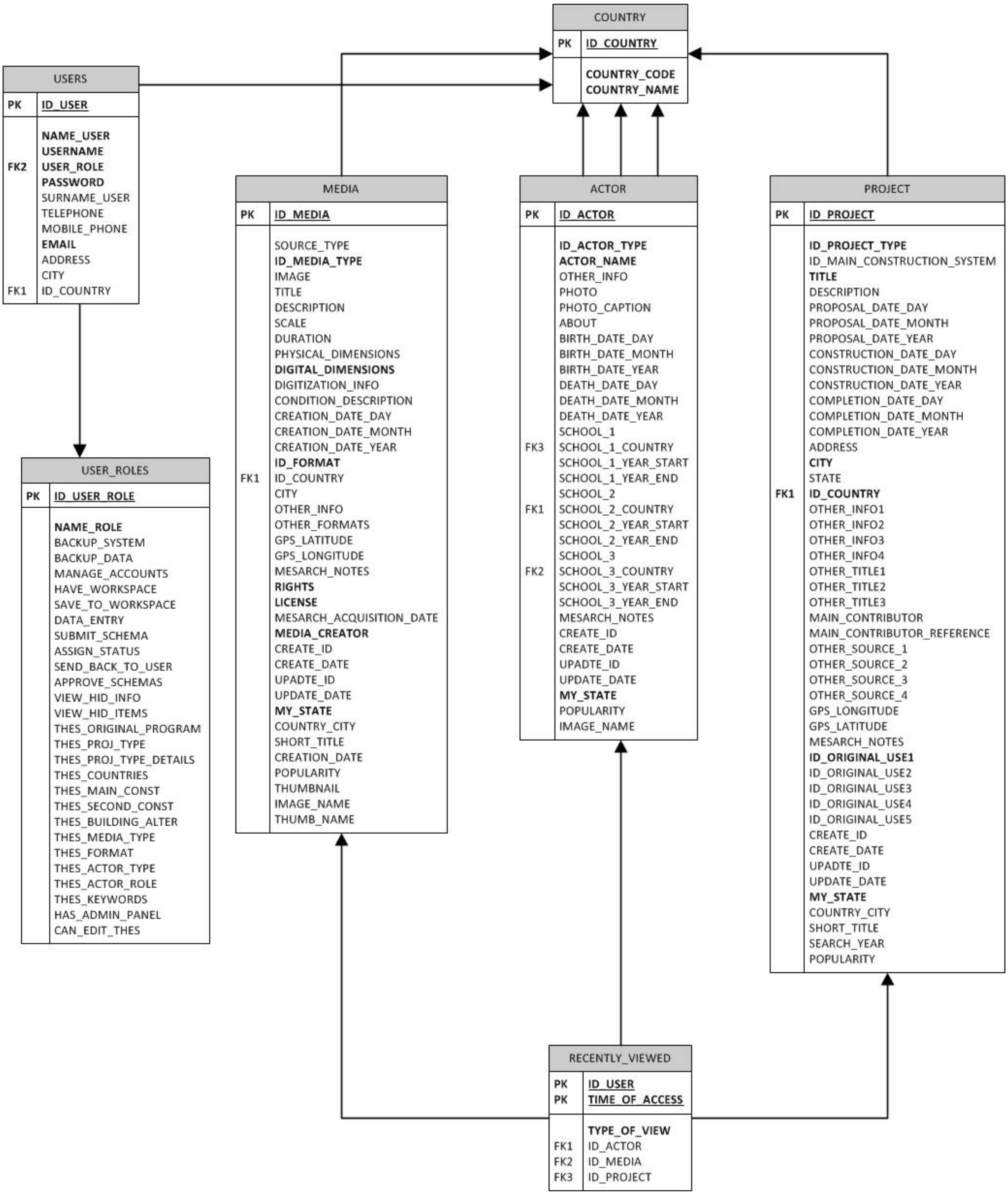
## Media relations



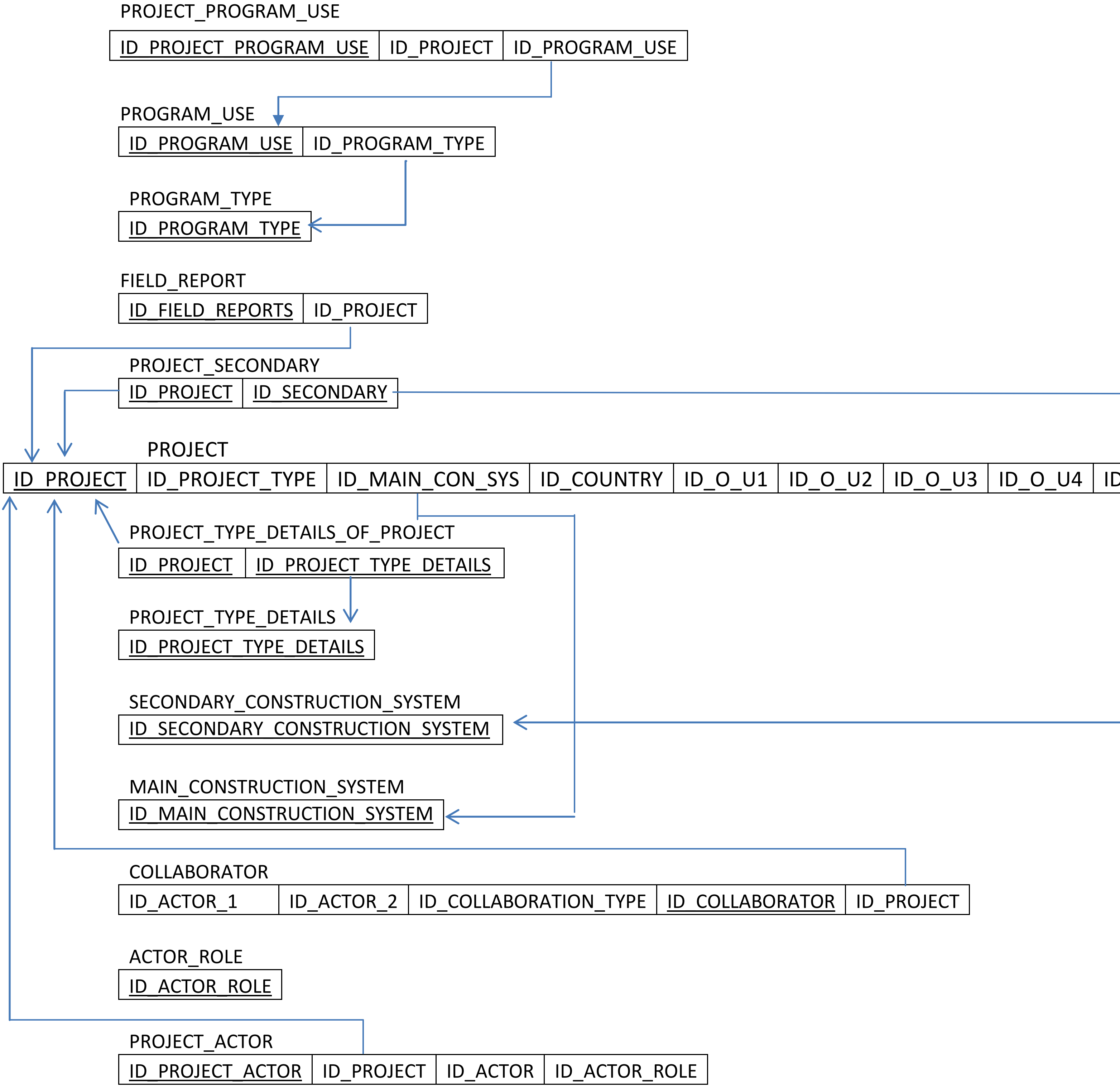
## Actor relations

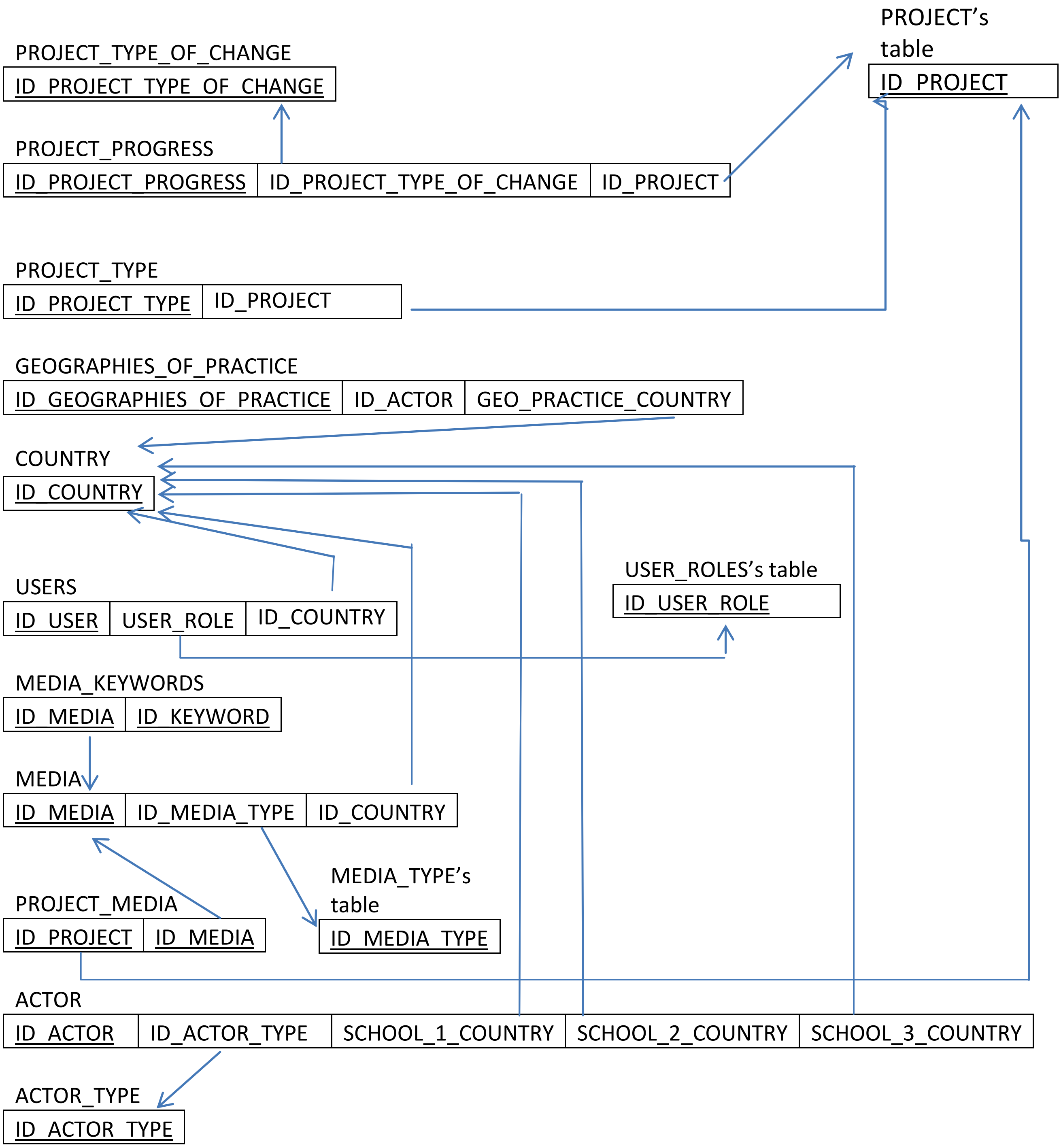


## User relations



### 5.2.2 Relational diagrams (only keys)





PROJECT's table

ID_PROJECT

PROJECT_TYPE_OF_CHANGE

ID_PROJECT_TYPE_OF_CHANGE

PROJECT_PROGRESS

ID_PROJECT_PROGRESS | ID_PROJECT_TYPE_OF_CHANGE | ID_PROJECT

PROJECT_TYPE

ID_PROJECT_TYPE | ID_PROJECT

GEOGRAPHIES_OF_PRACTICE

ID_GEOGRAPHIES_OF_PRACTICE | ID_ACTOR | GEO_PRACTICE_COUNTRY

COUNTRY

ID_COUNTRY

USER_ROLES's table

ID_USER_ROLE

USERS

ID_USER | USER_ROLE | ID_COUNTRY

MEDIA_KEYWORDS

ID_MEDIA | ID_KEYWORD

MEDIA

ID_MEDIA | ID_MEDIA_TYPE | ID_COUNTRY

MEDIA_TYPE's table

ID_MEDIA_TYPE

PROJECT_MEDIA

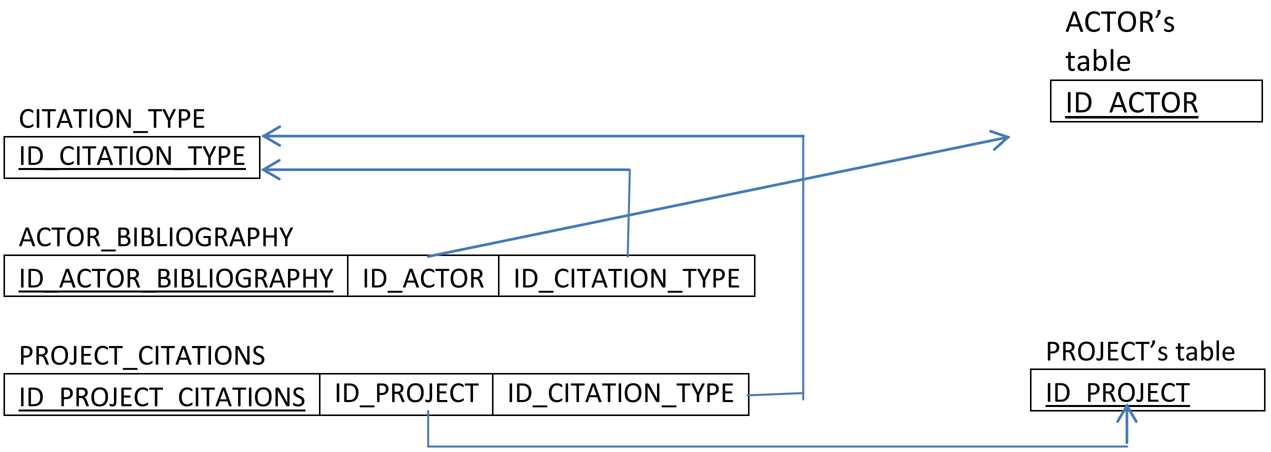
ID_PROJECT | ID_MEDIA

ACTOR

ID_ACTOR | ID_ACTOR_TYPE | SCHOOL_1_COUNTRY | SCHOOL_2_COUNTRY | SCHOOL_3_COUNTRY

ACTOR_TYPE

ID_ACTOR_TYPE



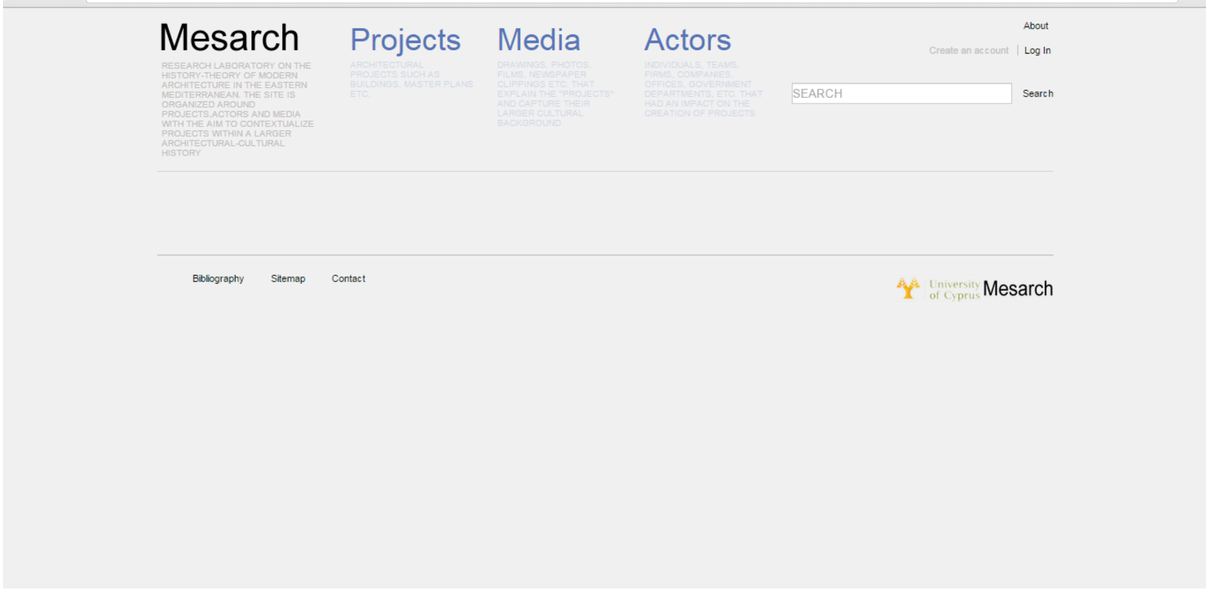
## 5.3 Interface design

### 5.3.1 Prototype used

For the purposes of my dissertation, I had full access to an interface prototype developed and used by another student in the past. This old prototype is naïve and it lacks state-of-the-art technology such as bootstrap, so I used this only as a tool to guide me in what I have to develop and deliver.

Screenshots:

*Homepage*



RESEARCH LABORATORY ON THE HISTORY-THEORY OF MODERN ARCHITECTURE IN THE EASTERN MEDITERRANEAN. THE SITE IS ORGANIZED AROUND PROJECTS,ACTORS AND MEDIA WITH THE AIM TO CONTEXTUALIZE PROJECTS WITHIN A LARGER ARCHITECTURAL-CULTURAL HISTORY

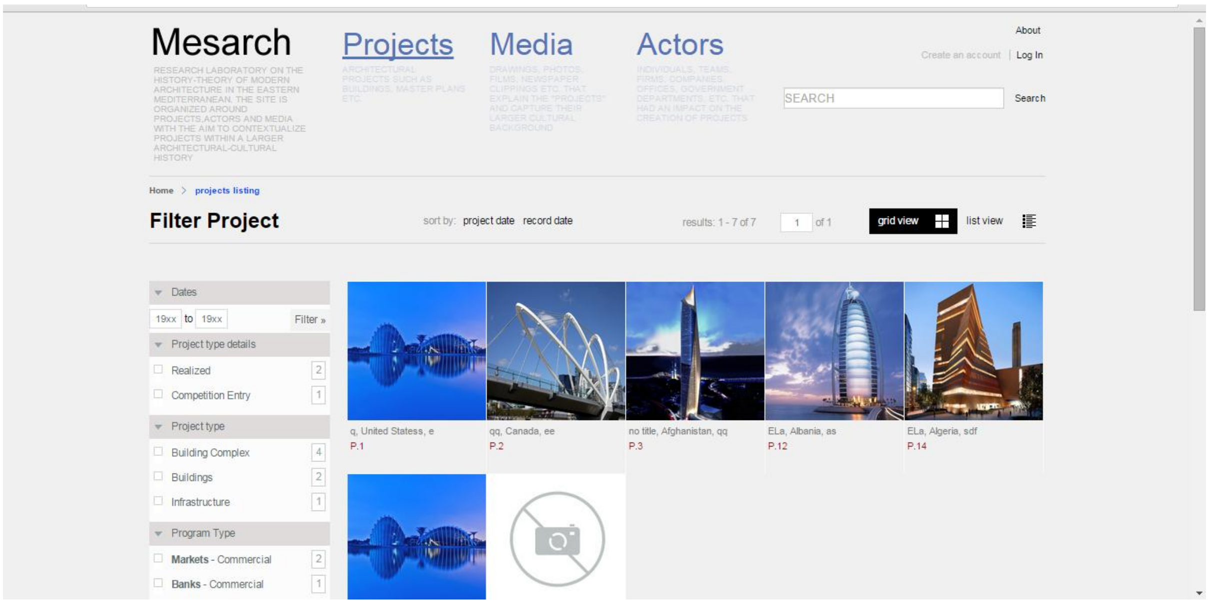
ARCHITECTURAL PROJECTS SUCH AS BUILDINGS, MASTER PLANS ETC.

DRAWINGS, PHOTOS, FILMS, NEWSPAPER CLIPPINGS ETC. THAT EXPLAIN THE "PROJECTS" AND CAPTURE THEIR LARGER CULTURAL BACKGROUND

INDIVIDUALS, TEAMS, FIRMS, COMPANIES, OFFICES, GOVERNMENT DEPARTMENTS, ETC. THAT HAD AN IMPACT ON THE CREATION OF PROJECTS

In the homepage figure, we can see the main categories, project, media and actors, with some description on the top of the page. On the top right corner we can see the create account, login, about links and search facility. Finally, on the footer of the page we can see the bibliography, the sitemap, the contact and the University of Cyprus links.

*Projects listing*



RESEARCH LABORATORY ON THE HISTORY-THEORY OF MODERN ARCHITECTURE IN THE EASTERN MEDITERRANEAN. THE SITE IS ORGANIZED AROUND PROJECTS,ACTORS AND MEDIA WITH THE AIM TO CONTEXTUALIZE PROJECTS WITHIN A LARGER ARCHITECTURAL-CULTURAL HISTORY

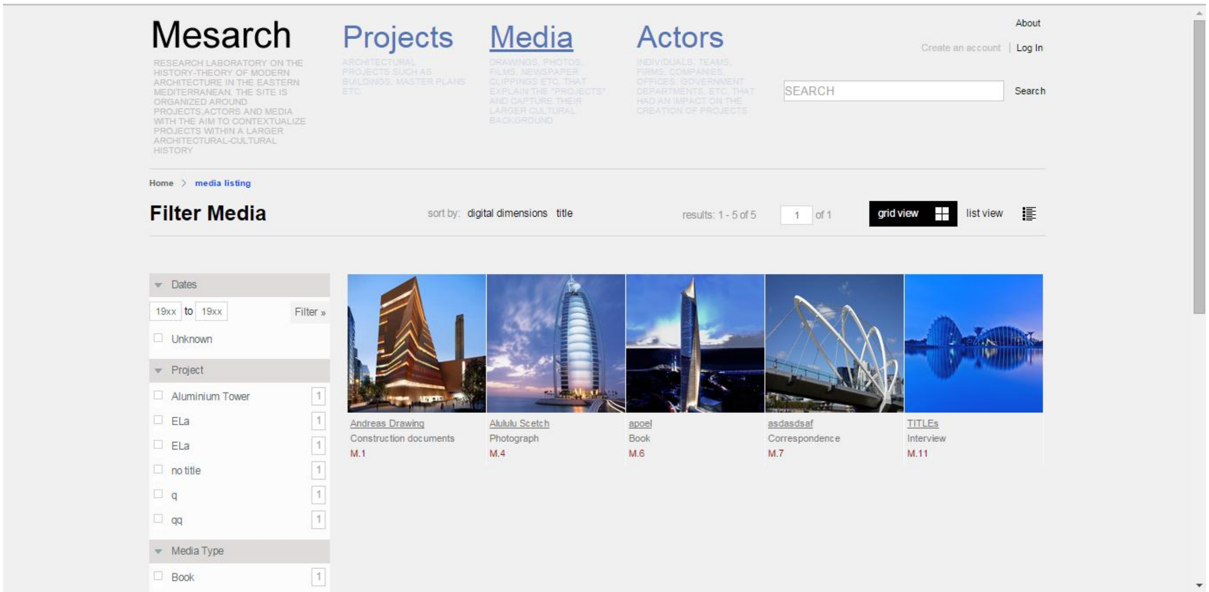
ARCHITECTURAL PROJECTS SUCH AS BUILDINGS, MASTER PLANS ETC.

DRAWINGS, PHOTOS, FILMS, NEWSPAPER CLIPPINGS ETC. THAT EXPLAIN THE "PROJECTS" AND CAPTURE THEIR LARGER CULTURAL BACKGROUND

INDIVIDUALS, TEAMS, FIRMS, COMPANIES, OFFICES, GOVERNMENT DEPARTMENTS, ETC. THAT HAD AN IMPACT ON THE CREATION OF PROJECTS

In this figure we can see the project listing. Information about projects includes picture, project ID, project title, project type and project location. On top of the page, below the main menu, we can see the navigation bar, the result sorting options, the result pages and finally the list type, grid or list view. Finally, on the left of the page we can see the filtering bar.

*Media listing*



RESEARCH LABORATORY ON THE HISTORY-THEORY OF MODERN ARCHITECTURE IN THE EASTERN MEDITERRANEAN. THE SITE IS ORGANIZED AROUND PROJECTS,ACTORS AND MEDIA WITH THE AIM TO CONTEXTUALIZE PROJECTS WITHIN A LARGER ARCHITECTURAL-CULTURAL HISTORY

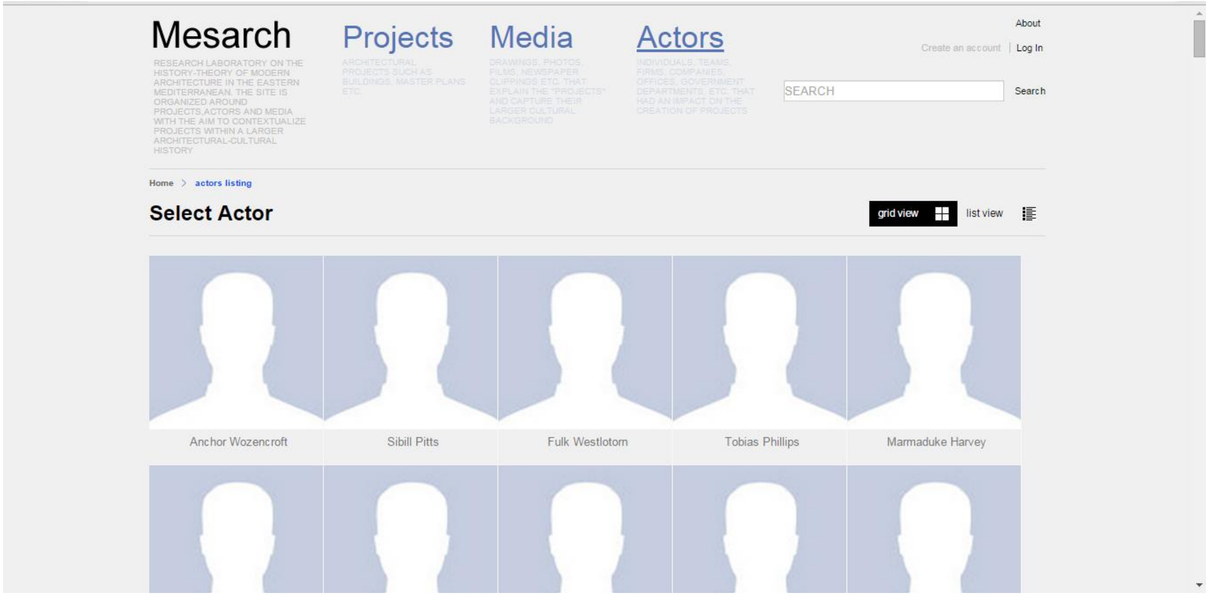
ARCHITECTURAL PROJECTS SUCH AS BUILDINGS, MASTER PLANS ETC.

DRAWINGS, PHOTOS, FILMS, NEWSPAPER CLIPPINGS ETC. THAT EXPLAIN THE "PROJECTS" AND CAPTURE THEIR LARGER CULTURAL BACKGROUND

INDIVIDUALS, TEAMS, FIRMS, COMPANIES, OFFICES, GOVERNMENT DEPARTMENTS, ETC. THAT HAD AN IMPACT ON THE CREATION OF PROJECTS

In this figure we can see the media listing. Information about media includes picture, media ID, project title, and media type. On top of the page, below the main menu, we can see the navigation bar, the result sorting options, the result pages and finally the list type, grid or list view. Finally, on the left of the page we can see the filtering bar.

*Actors listing*



RESEARCH LABORATORY ON THE HISTORY-THEORY OF MODERN ARCHITECTURE IN THE EASTERN MEDITERRANEAN. THE SITE IS ORGANIZED AROUND PROJECTS,ACTORS AND MEDIA WITH THE AIM TO CONTEXTUALIZE PROJECTS WITHIN A LARGER ARCHITECTURAL-CULTURAL HISTORY

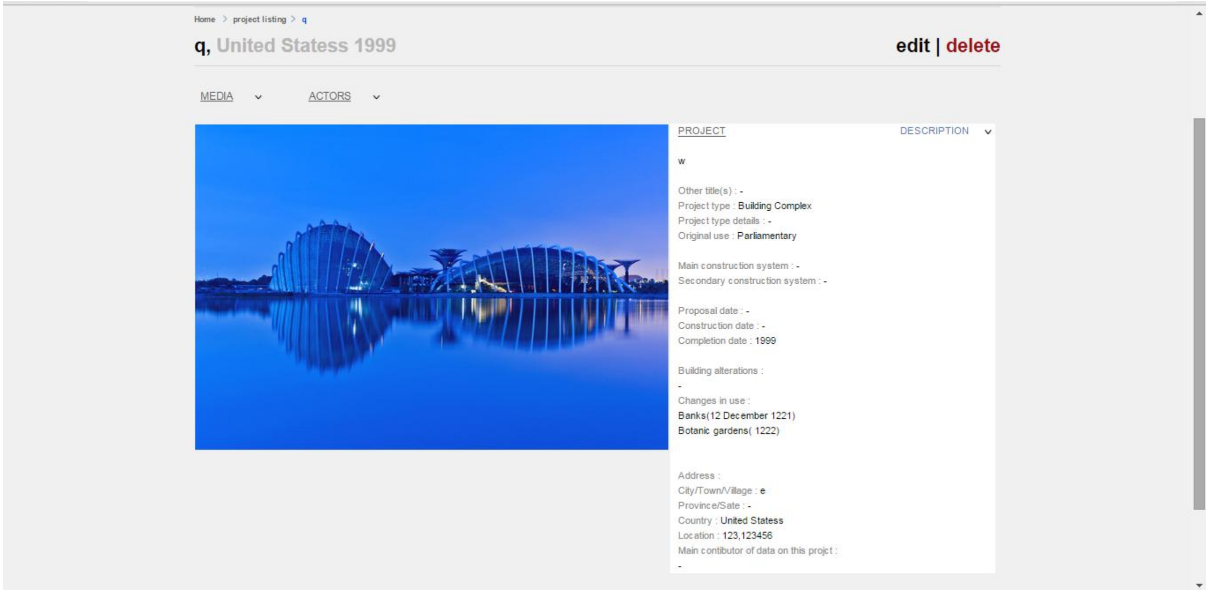
ARCHITECTURAL PROJECTS SUCH AS BUILDINGS, MASTER PLANS ETC.

DRAWINGS, PHOTOS, FILMS, NEWSPAPER CLIPPINGS ETC. THAT EXPLAIN THE "PROJECTS" AND CAPTURE THEIR LARGER CULTURAL BACKGROUND

INDIVIDUALS, TEAMS, FIRMS, COMPANIES, OFFICES, GOVERNMENT DEPARTMENTS, ETC. THAT HAD AN IMPACT ON THE CREATION OF PROJECTS

In this figure we can see the actor listing. Information about actor includes picture and actor name. On top of the page, below the main menu, we can see the navigation bar and the type of view, grid or list.

*Project details*



q, United Statess 1999

w

Other title(s) : -
Project type : Building Complex
Project type details : -
Original use : Parliamentary

Main construction system : -
Secondary construction system : -

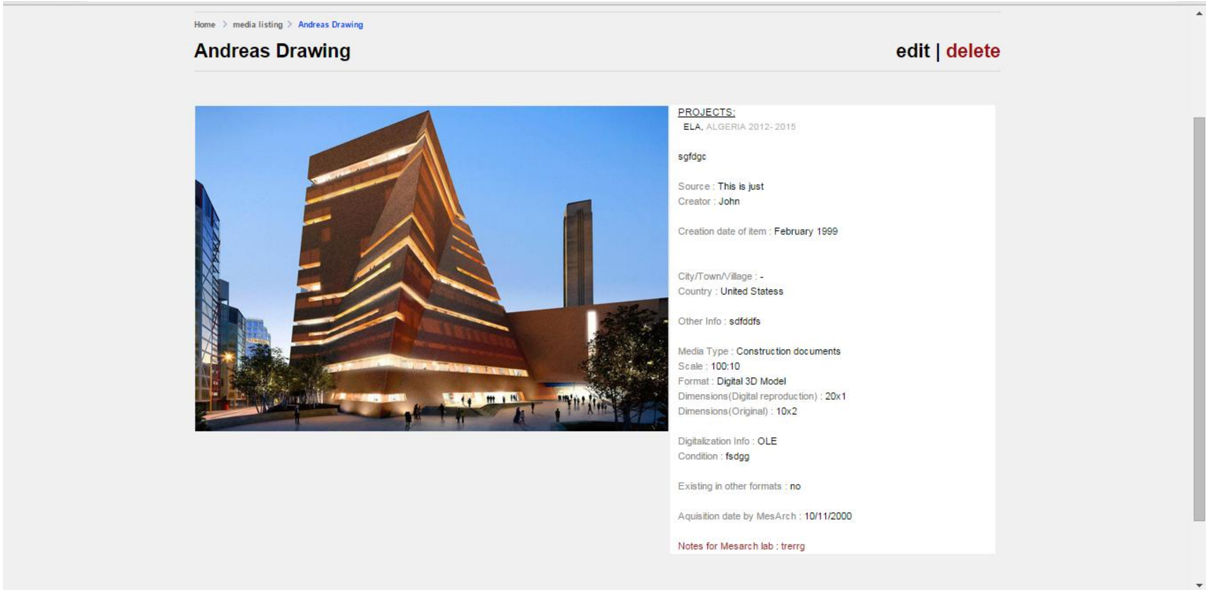
Proposal date : -
Construction date : -
Completion date : 1999

Building alterations :
-
Changes in use :
Banks(12 December 1221)
Botanic gardens( 1222)

Address :
City/Town/Village : e
Province/Sate : -
Country : United Statess
Location : 123,123456
Main contibutor of data on this projct :
-

In the preceding figure we can see the project details page. The picture of the project is presented on the left of the page and the main content info is presented on the right. The user can choose what kind of information he wants to see, from the upper right side. On the top left side, below the navigation bar, the related media and actors are shown, in a drop down list.

*Media details*



Home > media listing > Andreas Drawing

Andreas Drawing

edit | delete

PROJECTS:
ELA, ALGERIA 2012- 2015

sgfdgc

Source : This is just
Creator : John

Creation date of item : February 1999

City/Town/Village : -
Country : United Statess

Other Info : sdfddfs

Media Type : Construction documents
Scale : 100:10
Format : Digital 3D Model
Dimensions(Digital reproduction) : 20x1
Dimensions(Original) : 10x2

Digitalization Info : OLE
Condition : fsdgg

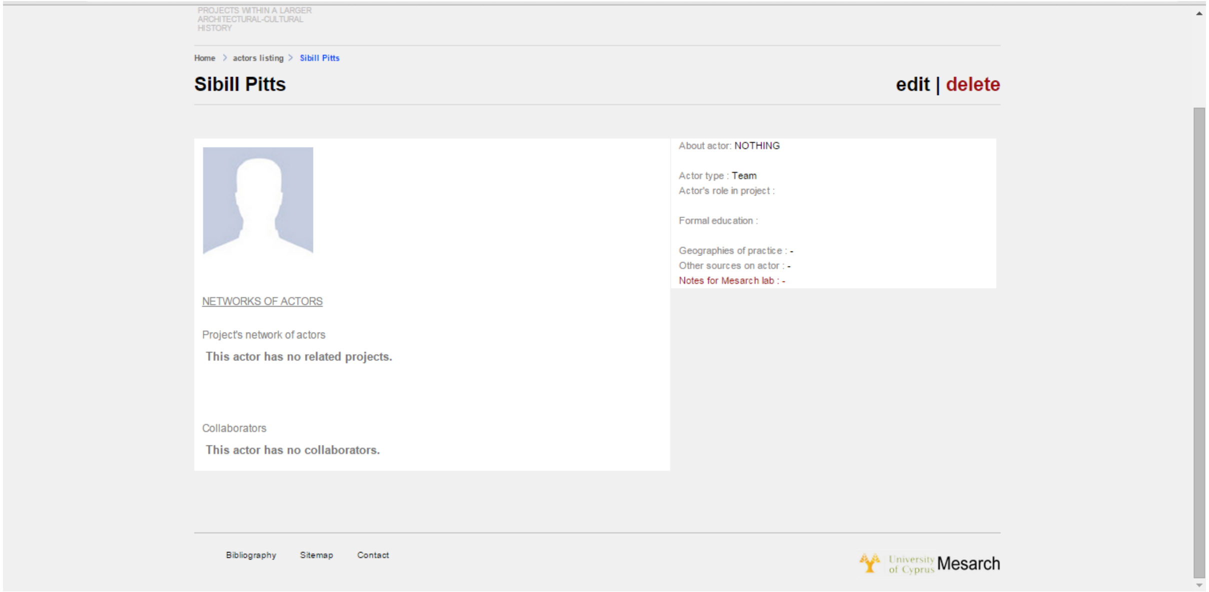
Existing in other formats : no

Aquisition date by MesArch : 10/11/2000

Notes for Mesarch lab : trerrg

In this figure we can see the media details page. The picture of the media is presented on the left of the page and the main content info is presented on the right. There is a navigation bar above the picture.

*Actor details*



PROJECTS WITHIN A LARGER ARCHITECTURAL-CULTURAL HISTORY

Home > actors listing > Sibill Pitts

Sibill Pitts

edit | delete

NETWORKS OF ACTORS

Project's network of actors

This actor has no related projects.

Collaborators

This actor has no collaborators.

About actor: NOTHING

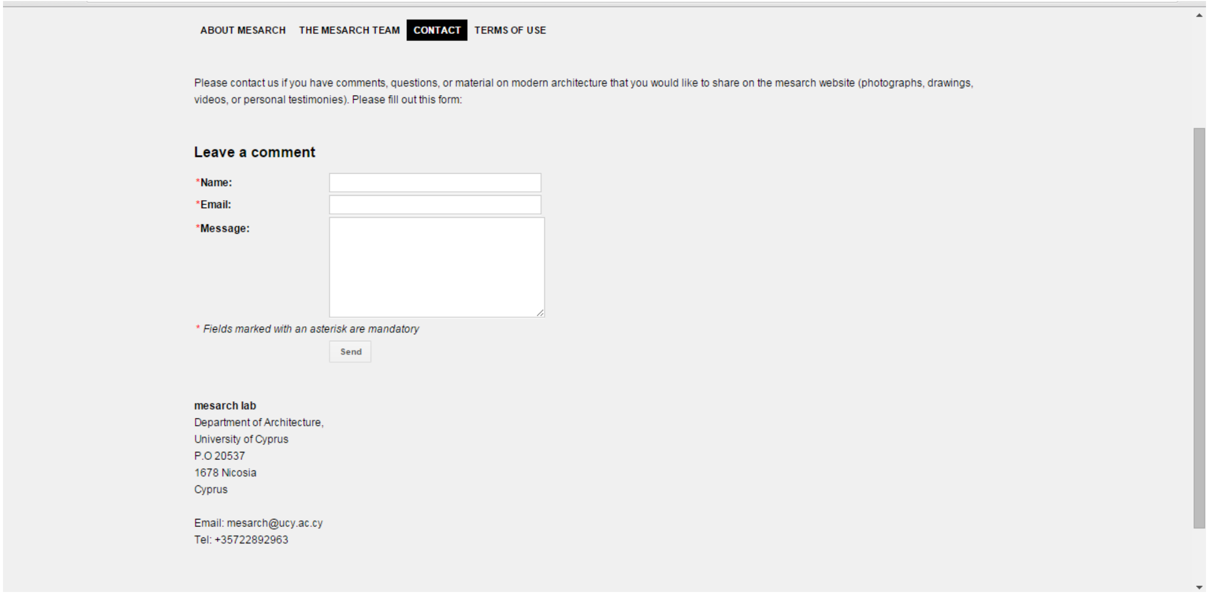
Actor type : Team
Actor's role in project :

Formal education :

Geographies of practice : -
Other sources on actor : -
Notes for Mesarch lab : -

In the preceding figure we can see the actor details page. The picture of the actor is presented on the left of the page and the main content info is presented on the right. Below the actor image, the network of actors is presented, which includes the projects he was involved in, the other actors that were involved in these projects and his collaborators. There is a navigation bar above the picture.

*Contact info*



Please contact us if you have comments, questions, or material on modern architecture that you would like to share on the mesarch website (photographs, drawings, videos, or personal testimonies). Please fill out this form:

**Leave a comment**

*Name:

*Email:

*Message:

* *Fields marked with an asterisk are mandatory*

**mesarch lab**
Department of Architecture,
University of Cyprus
P.O 20537
1678 Nicosia
Cyprus

Email: mesarch@ucy.ac.cy
Tel: +35722892963

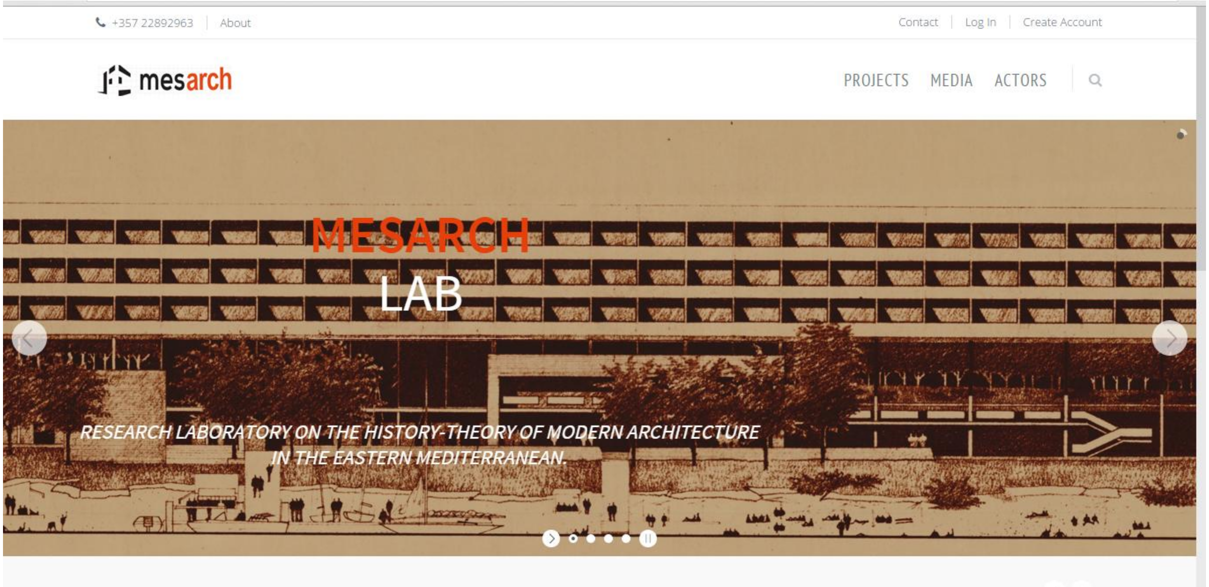
In this figure we can see the contact form page and the contact details. On top of the page, the user can choose the type of information to be presented, between about mesarch, the mesarch team, contact, and terms of use. Comments can be sent from this page, by entering name, email and message.

### 5.3.2 New interface design

For the reasons explained before (especially the lack of bootstrap technology) I used a different template, the metronic theme from themeforest.net website ([http://themeforest.net/item/metronic-responsive-admin-dashboard-template/4021469](http://themeforest.net/item/metronic-responsive-admin-dashboard-template/4021469)) and I build the new version of the interface around this theme. The new template supports a responsive design website that is mobile friendly; the interface provides an optimal viewing experience, easy reading and navigation with a minimum of resizing, panning, and scrolling across a wide range of devices and adapts the layout to the viewing environment.

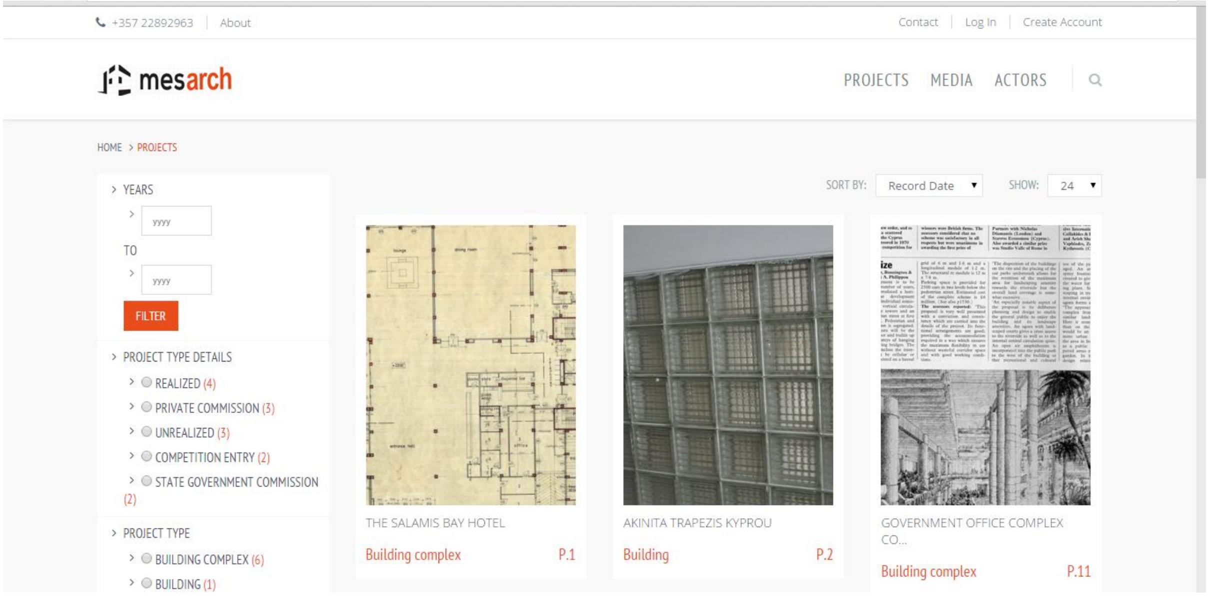
Screenshots:

*Homepage*


+357 22892963
About
Contact
Log In
Create Account
mesarch
PROJECTS
MEDIA
ACTORS
MESARCH
LAB
RESEARCH LABORATORY ON THE HISTORY-THEORY OF MODERN ARCHITECTURE
IN THE EASTERN MEDITERRANEAN.

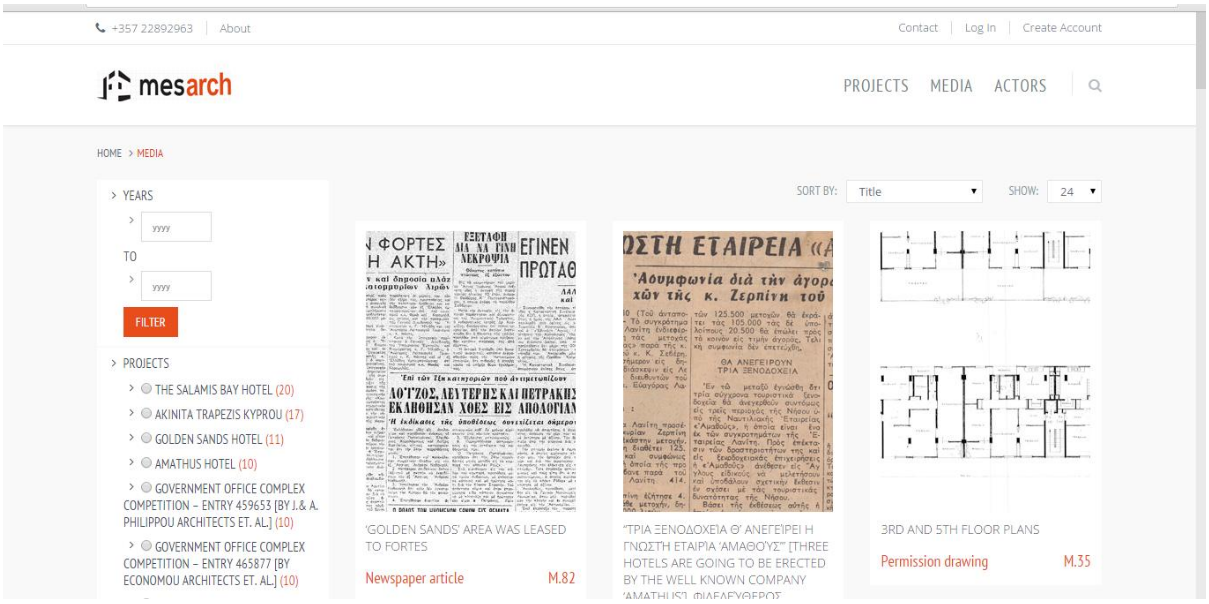
In the homepage figure, we can see the main categories, project, media and actors, the top right of the page, below the utility bar. The utility bar includes links for the about, and the contact pages, as well as login and create account options. Below the create account link, there is the search facility that pops out after the user click on the icon. There are some banner images, presented in the background that alternate automatically.

*Projects listing*



In the preceding figure we can see the project listing page. Information about projects includes picture, project ID, project title, and project type. On top of the page, below the main menu on the right, we can see the navigation bar, and on the left the result sorting options (project date newest-oldest, record date newest-oldest) and the result size. Finally, on the left of the page we can see the filtering bar.

*Media listing*



THE SALAMIS BAY HOTEL (20)
AKINITA TRAPEZIS KYPROU (17)
GOLDEN SANDS HOTEL (11)
AMATHUS HOTEL (10)
GOVERNMENT OFFICE COMPLEX COMPETITION – ENTRY 459653 [BY J.& A. PHILIPPOU ARCHITECTS ET. AL.] (10)
GOVERNMENT OFFICE COMPLEX COMPETITION – ENTRY 465877 [BY ECONOMOU ARCHITECTS ET. AL.] (10)

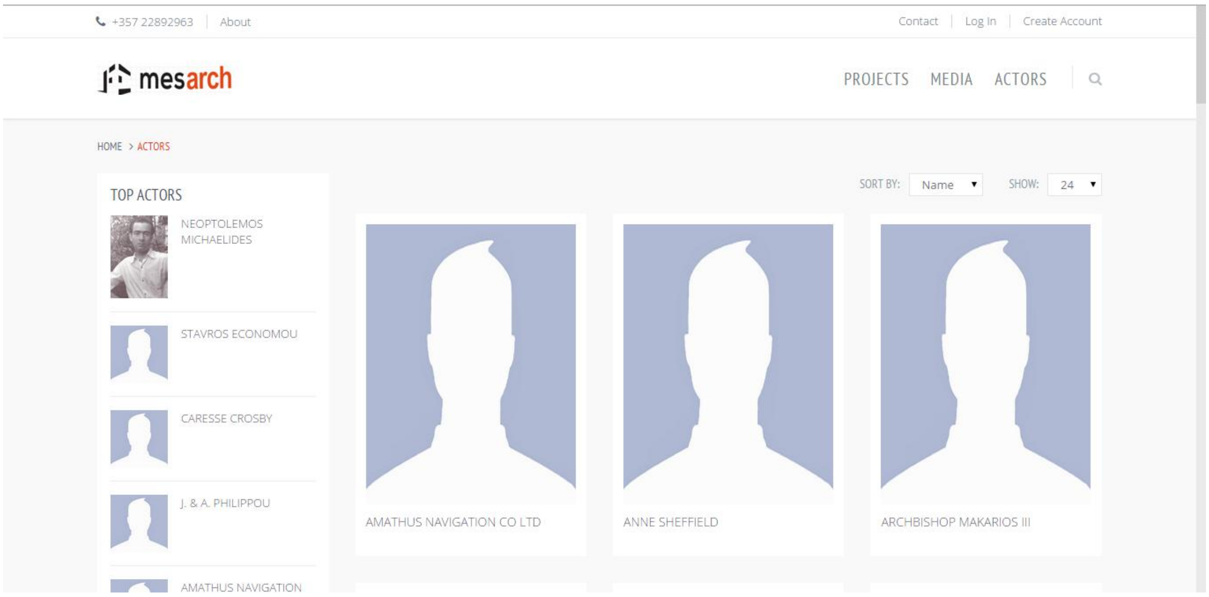
'GOLDEN SANDS' AREA WAS LEASED TO FORTES
Newspaper article M.82

"ΤΡΙΑ ΞΕΝΟΔΟΧΕΊΑ Θ' ΑΝΕΓΕΊΡΕΙ Η ΓΝΩΣΤΗ ΕΤΑΙΡΊΑ 'ΑΜΑΘΟΎΣ'" [THREE HOTELS ARE GOING TO BE ERECTED BY THE WELL KNOWN COMPANY 'AMATHUS'] ΦΙΛΕΛΕΥΘΕΡΟΣ

3RD AND 5TH FLOOR PLANS
Permission drawing M.35

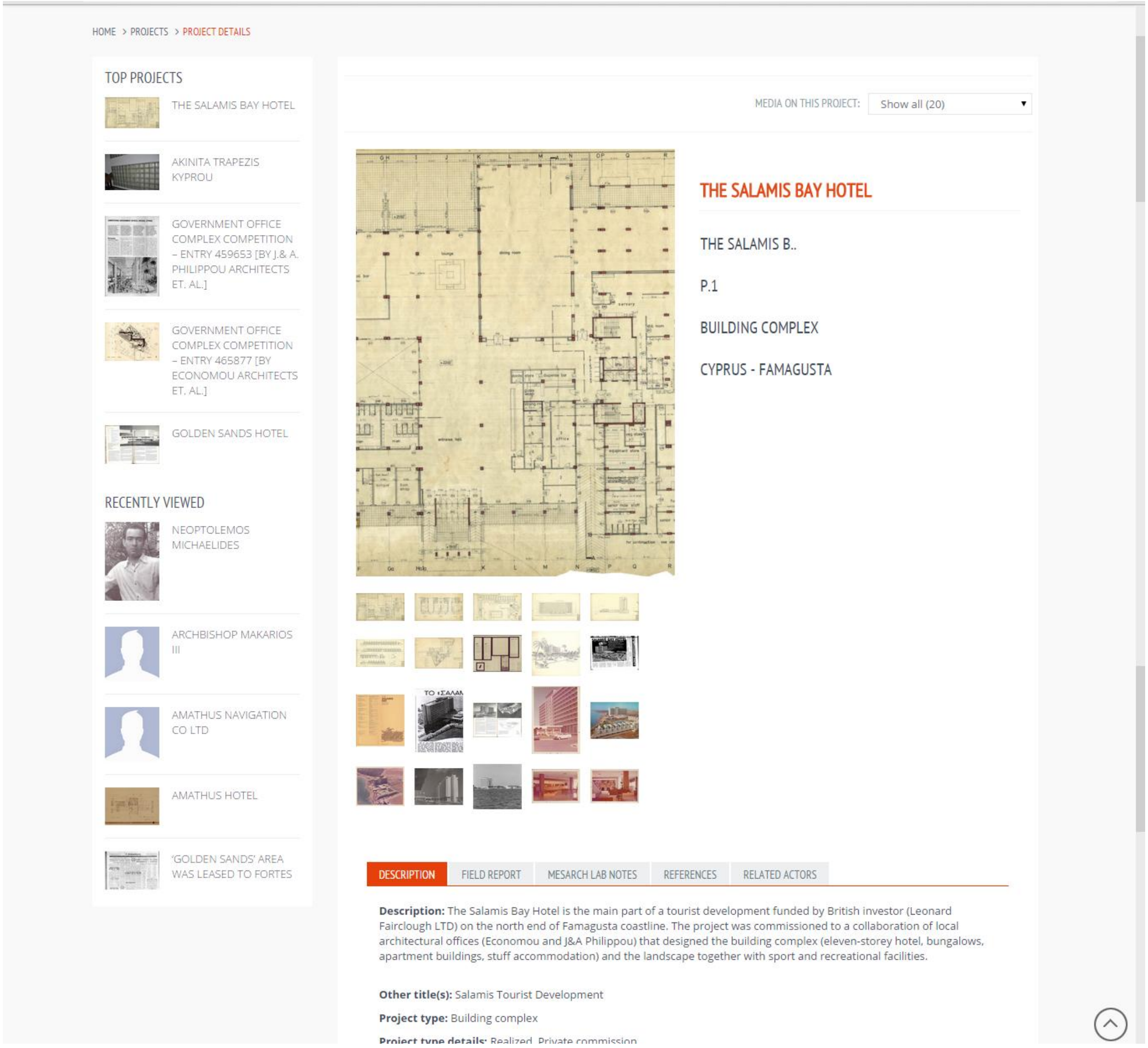
In this figure we can see the media listing. Information about media includes picture, media ID, project title, and media type. On top of the page, below the main menu, we can see the navigation bar, the result sorting options, the result pages and finally the list type, grid or list view. Finally, on the left of the page we can see the filtering bar.

*Actors listing*



In this figure we can see the actor listing. Information about actor includes picture and actor name. On top of the page, below the main menu on the left, we can see the navigation bar and on the right the sorting of the results as well as the result size. Finally, on the left we can see the top actors based on popularity.

*Project details*



TOP PROJECTS

- THE SALAMIS BAY HOTEL
- AKINITA TRAPEZIS KYPROU
- GOVERNMENT OFFICE COMPLEX COMPETITION – ENTRY 459653 [BY J.& A. PHILIPPOU ARCHITECTS ET. AL.]
- GOVERNMENT OFFICE COMPLEX COMPETITION – ENTRY 465877 [BY ECONOMOU ARCHITECTS ET. AL.]
- GOLDEN SANDS HOTEL

RECENTLY VIEWED

- NEOPTOLEMOS MICHAELIDES
- ARCHBISHOP MAKARIOS III
- AMATHUS NAVIGATION CO LTD
- AMATHUS HOTEL
- 'GOLDEN SANDS' AREA WAS LEASED TO FORTES

THE SALAMIS BAY HOTEL

THE SALAMIS B..

P.1

BUILDING COMPLEX

CYPRUS - FAMAGUSTA

**Description:** The Salamis Bay Hotel is the main part of a tourist development funded by British investor (Leonard Fairclough LTD) on the north end of Famagusta coastline. The project was commissioned to a collaboration of local architectural offices (Economou and J&A Philippou) that designed the building complex (eleven-storey hotel, bungalows, apartment buildings, stuff accommodation) and the landscape together with sport and recreational facilities.

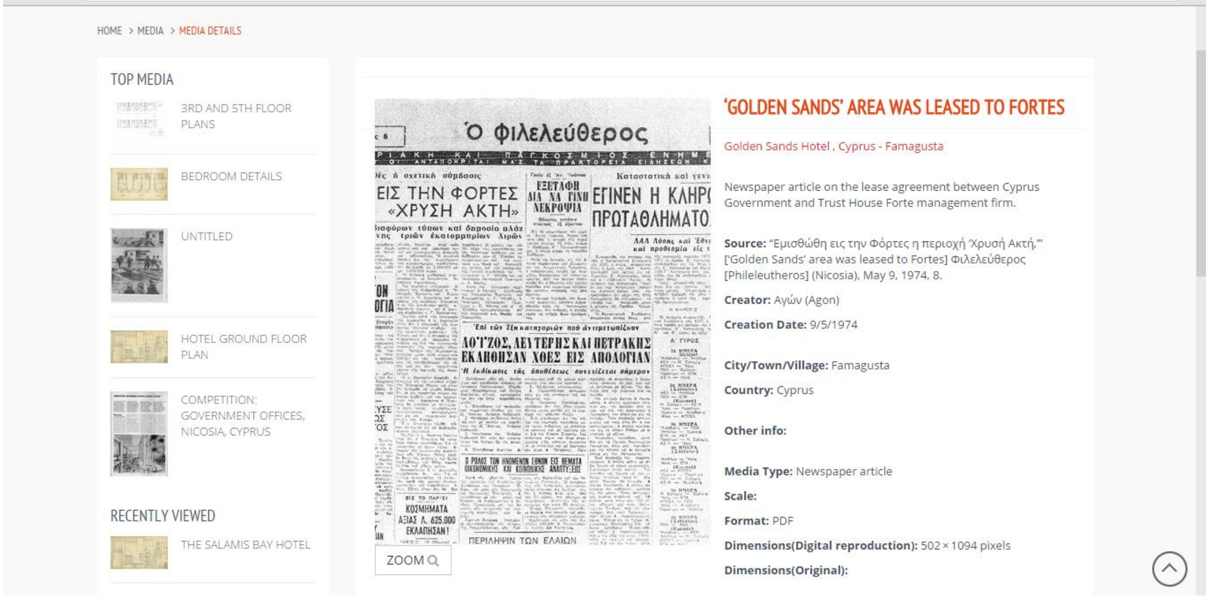
**Other title(s):** Salamis Tourist Development

**Project type:** Building complex

**Project type details:** Realized, Private commission

In this figure we can see the project details page. On the top of the page there is the navigation bar, and on the left side bars we can see the top projects based on popularity and recently seen items. Beside the recommender bars, the main info is presented. The picture is on the right; there is some basic info on the left and related media of the project below the main image. Below the related media pictures we can see the detailed information of the project, grouped by description, field report, mesarch notes, references and related actors.

*Media details*



'GOLDEN SANDS' AREA WAS LEASED TO FORTES

Golden Sands Hotel , Cyprus - Famagusta

Newspaper article on the lease agreement between Cyprus Government and Trust House Forte management firm.

**Source:** "Εμισθώθη εις την Φόρτες η περιοχή 'Χρυσή Ακτή,'" ['Golden Sands' area was leased to Fortes] Φιλελεύθερος [Phileleutheros] (Nicosia), May 9, 1974, 8.

**Creator:** Αγών (Agon)

**Creation Date:** 9/5/1974

**City/Town/Village:** Famagusta

**Country:** Cyprus

**Other info:**

**Media Type:** Newspaper article

**Scale:**

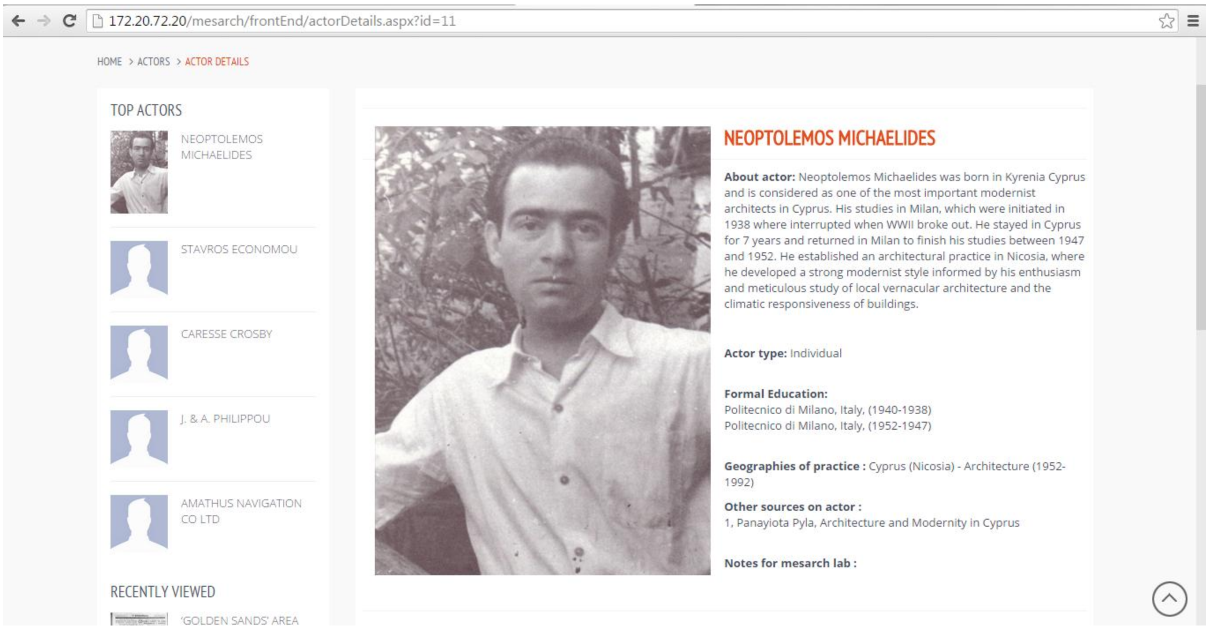
**Format:** PDF

**Dimensions(Digital reproduction):** 502 × 1094 pixels

**Dimensions(Original):**

In this figure we can see the media details page. On the top of the page there is the navigation bar, and on the left side bar we can see the top media based on popularity. Beside the recommender bar, the main info is presented. The picture is on the right and the detailed information is on the left.

*Actor details*



NEOPTOLEMOS MICHAELIDES

**About actor:** Neoptolemos Michaelides was born in Kyrenia Cyprus and is considered as one of the most important modernist architects in Cyprus. His studies in Milan, which were initiated in 1938 where interrupted when WWII broke out. He stayed in Cyprus for 7 years and returned in Milan to finish his studies between 1947 and 1952. He established an architectural practice in Nicosia, where he developed a strong modernist style informed by his enthusiasm and meticulous study of local vernacular architecture and the climatic responsiveness of buildings.

**Actor type:** Individual

**Formal Education:**
Politecnico di Milano, Italy, (1940-1938)
Politecnico di Milano, Italy, (1952-1947)

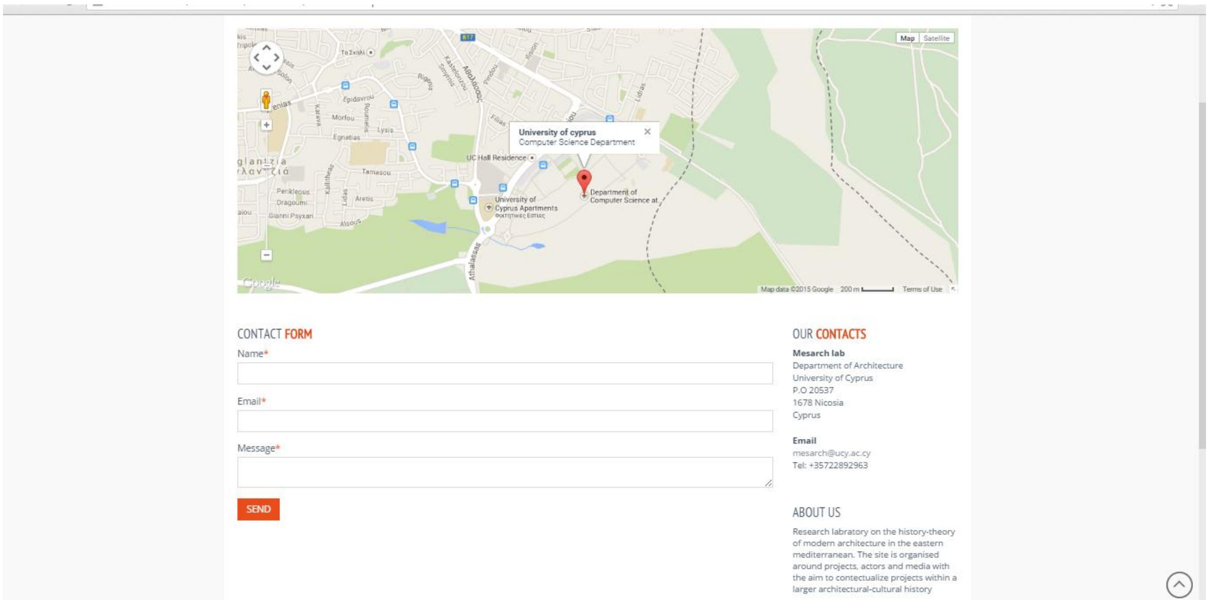
**Geographies of practice :** Cyprus (Nicosia) - Architecture (1952-1992)

**Other sources on actor :**
1, Panayiota Pyla, Architecture and Modernity in Cyprus

**Notes for mesarch lab :**

In the preceding figure we can see the actor details page. On the top of the page there is the navigation bar, and on the left side bar we can see the top actors based on popularity. Beside the recommender bar, the main info is presented. The picture is on the right and the detailed information is on the left.

*Contact info*



CONTACT FORM

Name*

Email*

Message*

SEND

OUR CONTACTS

**Mesarch lab**
Department of Architecture
University of Cyprus
P.O 20537
1678 Nicosia
Cyprus

**Email**
mesarch@ucy.ac.cy
Tel: +35722892963

ABOUT US

Research labratory on the history-theory of modern architecture in the eastern mediterranean. The site is organised around projects, actors and media with the aim to contectualize projects within a larger architectural-cultural history

In this figure we can see the contact details page. A google map is presented on the top, showing mesarch laboratory’s position on the map. Below the map, on the left, there is a contact form where the user can send comments by entering name, email and message. Finally, on the right, the contact details and some information about the laboratory is presented.

### 5.4 Security Infrastructure

The source code of the website and of the database includes the necessary checks to prevent SQL injection attacks. Such attacks seek to exploit SQL weaknesses and through website queries (for example login form queries) or URL query strings may try to retrieve private data and/or destroy the system.

Additionally, the system prevents unauthorized access to the administrator panel. If a user knows the exact URL to navigate to the backend site, he can type that and navigate there. Once being there, all the data of the system are exposed to him; he can add, delete and update whatever he wants to.  The system checks every time a user tries to navigate to the backend site and if he is not authorized to do so, it redirects him to the frontend site automatically.

Furthermore, the system prevents cross-site scripting (XSS) that can happen in places where user inputs data (for example login form username text box). XSS enables users inject client side script into web pages viewed by other users.  When the system detects such scripts, it redirects the user to an error page instead of executing any code.

# Chapter 6

## System implementation

6.1 Software design 68

6.2 Implementation tools 70

6.3 Database implementation 73

6.4 User interface implementation 74

6.5 System integration 77

### Introduction

In this chapter, the implementation of the system and its individual parts are discussed. Initially, the software design and the class diagrams of the system are presented and after that the implementation tools used, the database and user interface implementation and finally the system integration.

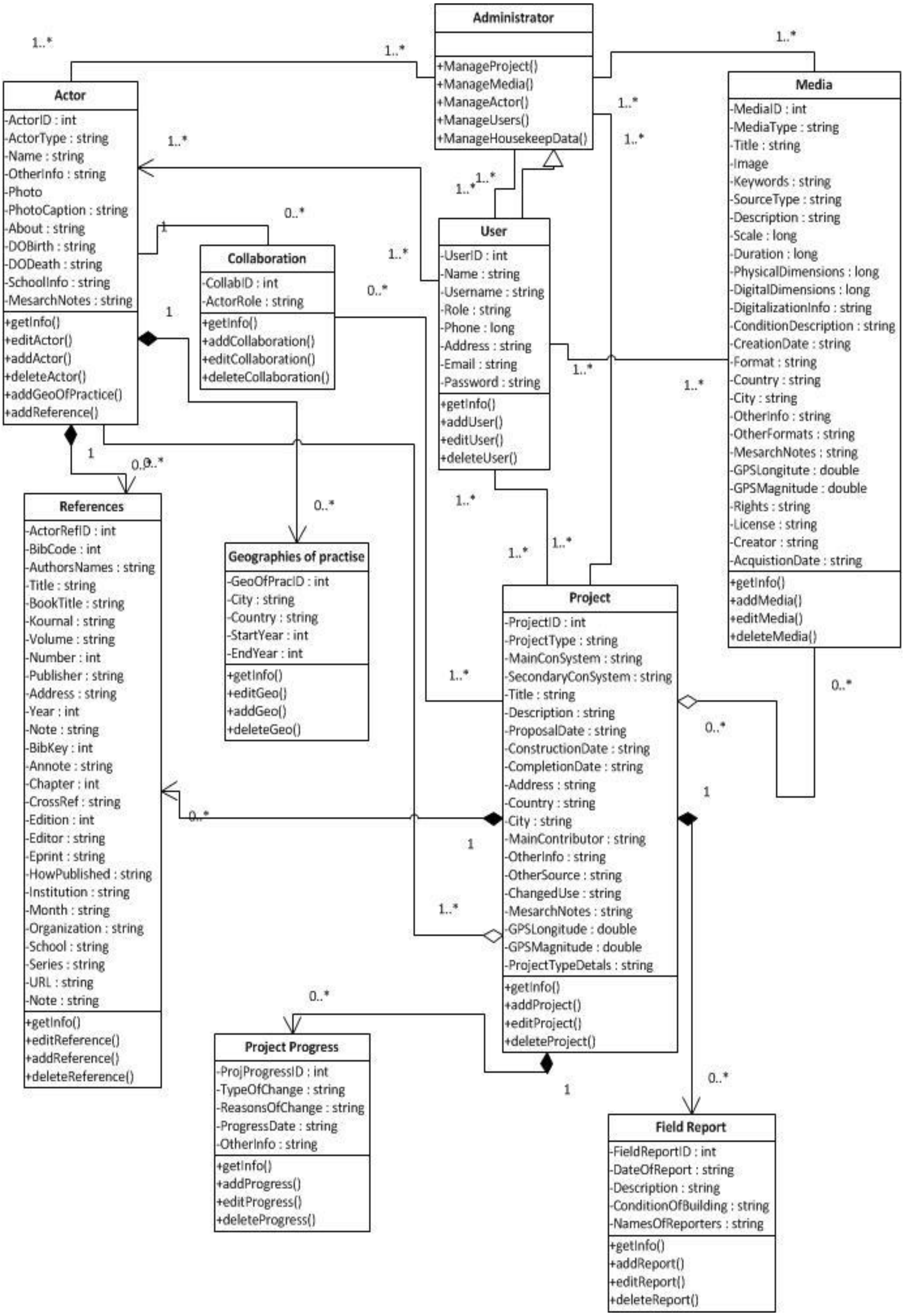
### 6.1 Software design

The model used for the MARE system is the client-server model where the client is the web interface and server is the database. Users and administrators query the database through the graphical user interface and the database responds with the information requested.

There are three major data structures that reside in the database, as explained and before and all the queries are built around them: projects, media and actors. There are further relationships between these structures like actors on a specific project and media on a specific project.

Additionally, there are two main user interfaces designed, one for the users where they can view the information of the system and one for the administrators where they can manage it.

## Class diagrams



The preceding diagram is the class diagram of the MARE system. The abstract classes that constitute the system along their characteristics, their functions and their dependencies are presented.

The relationships are explained below:

- A project can have many field reports but an individual field report belongs to an individual project and its owner it’s an individual actor

- A project can have many actors and an actor can work on many projects as well

- An actor can have many geographies of practice but an individual geography is on an individual actor

- An actor can have many references but an individual reference belongs to only one actor

- Collaboration consists of two actors, a project and a collaboration type. Individual actors can have as many collaborations as they want with different actors

- A project can have many references but an individual reference belongs to only one reference project

- A project can have many media and a media can belong to many projects as well

- An administrator can process all the classes of the system

- A user can view the information of the system

### 6.2 Implementation tools

Technologies:

**ASP.Net:** It stands for active server pages and it is included in the Microsoft’s .Net framework. It is used for the creation of dynamic and/or static websites and it is implemented over common language runtime so it allows the programmer to code in asp using any .Net supported programming language.

**JQuery:** It is a cross-browser JavaScript library that allows the integration of script code with the html code for simplifying client-side scripting. Some of the JQuery functions are the easy management of DOM elements, the creation of animations, event management (ex. Click events, mouse hovering events etc) and the implementation of AJAX functions.

Tools:

Microsoft Visual Studio 2010:

- Product name: Microsoft Visual studio 2010
- Version: 2010
- Company: Microsoft
- Documentation: http://msdn.microsoft.com/en-us/vcsharp/default.aspx
- Purpose: It is the framework used in the implementation of the WA. The programming language used is C# because I am much more familiar with it than with Visual Basic. There are many tools included in this suite for the implementation of web forms and web pages, error detection tools and it is event-driven based. Additionally, the connection with MS SQL Server is very easily established

SQL Server:

- Product name: MS SQL Server.
- Version: 2008 R2
- Company: Microsoft
- Documentation: http://www.microsoft.com/sqlserver/en/us/default.aspx
- Purpose: It is the DB management system that the DB of the system was implemented in. It was chosen because of the compatibility with the .net framework used for the implementation of the WA.

Adobe fireworks CS4:

- Product name: Adobe fireworks
- Version: CS4
- Company: Adobe
- Documentation: http://www.adobe.com/cfusion/tdrc/index.cfm?loc=en&product=fireworks
- Purpose: It was used for the creation of the logos and custom images

Operating System:

- Product name: Windows 7
- Version: Home Edition
- Company: Microsoft

I chose the Microsoft tools and technologies (ASP.Net, SQL Server, Visual Studio) instead of the corresponding LAMP ones for many reasons. Visual studio provides an excellent interface and plenty of tools for web developing, and it is easily used. It supports object-oriented programming and it provides error handling techniques that are very helpful during the debugging procedure. The programming language used is C#, a language that I am familiar with. Finally, ASP.Net and SQL Server are compatible and can be easily connected.

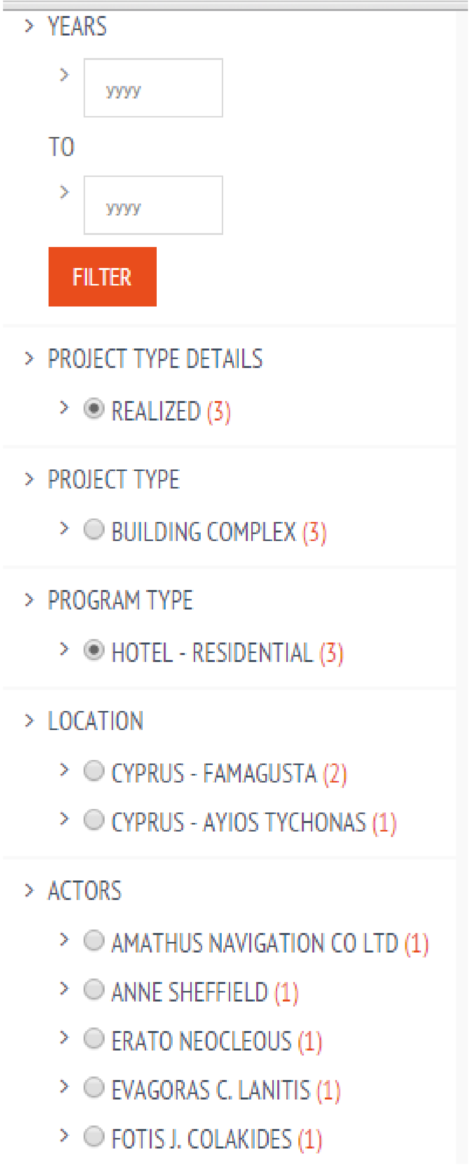
I preferred the JQuery library over the plain JavaScript because JQuery met most of my needs and required much less coding than conventional JavaScript might require. JQuery has been optimized to perform many common scripting functions and it does so while using fewer lines of code. Additionally, JQuery has been optimized to work with a variety of browsers automatically while JavaScript may face cross-browser compatibility issues.

### 6.3 Database implementation

Studying the user requirements presented in chapter 3, the corresponding diagrams presented in chapter 4 were developed and served as the guideline for the database implementation. For every entity presented in the ER diagram, a database table was created and for every relation expressed in the relational diagram, integrity constraints (Primary keys, foreign keys) were checked. Furthermore, additional domain constraints like data type and length were checked as the database schema took its final form.

However, some functionalities of the WA were complex enough they couldn’t be developed in visual studio hence a database API was developed that I could use through the WA to automate various complex functionalities of the system. For example, the results filtering and the advanced search functionalities needed to be developed in the database and use an API call to execute them. Additionally, having queries developed in SQL instead of an external programming language and using an API call to execute those speeds up the overall execution time because the database can build several indexes based on those queries.

*Result filter sidebar*



> YEARS

> yyyy

TO

> yyyy

FILTER

> PROJECT TYPE DETAILS

> ● REALIZED (3)

> PROJECT TYPE

> ○ BUILDING COMPLEX (3)

> PROGRAM TYPE

> ● HOTEL - RESIDENTIAL (3)

> LOCATION

> ○ CYPRUS - FAMAGUSTA (2)

> ○ CYPRUS - AYIOS TYCHONAS (1)

> ACTORS

> ○ AMATHUS NAVIGATION CO LTD (1)

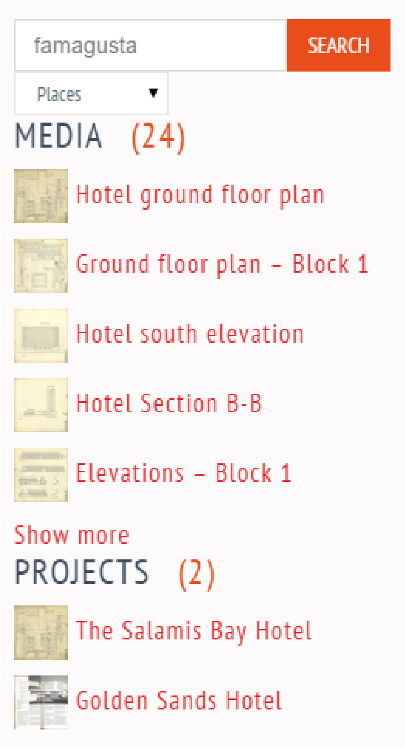
> ○ ANNE SHEFFIELD (1)

> ○ ERATO NEOCLEOUS (1)

> ○ EVAGORAS C. LANITIS (1)

> ○ FOTIS J. COLAKIDES (1)

*Advanced search window*



famagusta SEARCH

Places ▼

MEDIA (24)

Hotel ground floor plan

Ground floor plan – Block 1

Hotel south elevation

Hotel Section B-B

Elevations – Block 1

Show more

PROJECTS (2)

The Salamis Bay Hotel

Golden Sands Hotel

The API is used to asynchronously filter the results of each subsection of the filter sidebar that exists in the projects and media listing pages, as well as the content of these pages. When a user clicks on an option of the sidebar an API call is made and the contents of the page and the results of the subsections are updated. Furthermore, it is used to dynamically update the result-suggestions that appear in the advanced search window and to list the complete results if the user clicks on the "search" button or the "show more" option.

### 6.4 User interface implementation

The MARE WA is a dynamic website implemented in Microsoft Server technology, ASP.Net with SQL Server database. It consists of HTML code, AJAX, JavaScript, JQuery, and C# for the server-side code behind of ASP.Net.

The design of the system must be consistent in all the individual web pages so I created two basic templates, one for the frontend pages and another one for the backend ones. All the pages inherit their corresponding template's characteristics

The individual web pages are classified into different classes depending on their purpose. So, all the hyperlinks used for intra-system navigation navigate one folder backwards before the actual destination.

In the code behind, the C# code establishes connection with the database and it dynamically creates the content of the website or manages its content.

There are some helper classes that help in the transformation of the binary files stored in the database to image files in order to be able to show them on the web interface. Additionally, there are some classes that contribute in the asynchronous load (AJAX) of a web page. Those classes are called through JQuery functions, they take some parameters as input and they change asynchronously the content of a page, without loading the page again.

In general, in the WA I used a lot of JQuery and JavaScript code to improve the user's interaction with the system. Show and hide of tabs in data presentation of project details and interactive validation of user inputs are some examples of such code used.

### User interfaces

There are two main user interfaces implemented, explained below:

Unregistered user/Registered simple user/Collaborator: They can navigate only in the frontend site and use only that site's features as explained above. They cannot modify any data and the main differences between these three classes of users is the recently viewed list shown to the registered users as well as the permission to view private data that is given to the collaborators.

Administrators: They have the same permissions as the collaborators on the frontend site. However they have access to the backend site where they can modify all the data of the DB of the system. They can insert/delete/update projects, media and actors. They can also modify all the housekeep data like construction systems and project types, change the frontend homepage banner images and set an item to private/public. Additionally they can promote or demote registered members.

### The classes in detail

The MARE system interface consists of many classes as each individual html page is implemented in a different class. There are dozens of different classes but they are summarized in five categories:

- Frontend classes: All the html pages that a simple user (logged in or not) can see and navigate belong to this category. In a few words, the classes of this category constitute the image of the website that the public can see. For example the media listing, media details, contact details pages etc belong to this category. All the frontend classes inherit the frontend master page and so they all follow the same pattern and design.

- Backend classes: In this category there are all the html pages that the administrator uses for the maintenance of the system (inserts, updates and deletes of system data). For example, the new project, new media and new actor pages belong to this category. Similar to the frontend classes, they inherit the backend master page and so they all follow the same pattern and design. The backend master page follows a different pattern and design than the corresponding frontend one.

- User classes: All the pages that are related to the user login belong to this category, for example the create account, restore password and login pages. These classes inherit either from backend, either from frontend master page, depending on the type of the user (admin or not).

- Helper classes: This category consists from classes that don’t correspond to html pages but help other classes/pages manipulate their content. For example, the image handler class that belongs in this category transforms the binary format of the image retrieved from the database to the appropriate one needed for showing it.

- Global classes: Similar to the previous category, neither these classes correspond to html pages but they also help other classes/pages manipulate their content. For example, the AJAX calls that are called in some frontend pages are implemented in these classes. Additionally, some static fields that are used throughout the system (for example username and role) are kept as static in a class of this category.

## 6.5 System integration

### Software interfaces

There are two major software interfaces, the database interface and the web interface, developed independently. The website development started as a static website that showed the outline of the user interface, the way the information will be shown and managed. On the other hand, the database was developed from the beginning to be fully-functional: Tables, constraints, stored procedures and functions were created and tested to function properly.

When the design of the web interface finished, the two sub-systems were integrated to function together. The website adopted a dynamic behavior instead of static as it retrieved the information from the database and allowed the management of the database data through its interface.

In ASP.Net the formal way to connect a website with a database is through a connection string defined in the web.config (the main settings and configuration file) file. Later on, in the code behind of each individual web page the programmer can connect to the database to retrieve or store essential information using this connection string.

### Communication interfaces

The application is web base so an internet connection is indispensable for it to function. It is a cross-browser WA so the type of the browser doesn't matter. Additionally, the operating system must support any network technology like Wi-Fi (IEEE 802.11) or Ethernet (IEEE 802.3).

# Chapter 7

## Prototype deployment and evaluation

7.1 Experimental methodology 78

7.2 Deployment design 79

7.3 Evaluation 80

### Introduction

In this chapter the deployment of the system on the Architecture departments' server is discussed as well as the experimental methodology used and evaluation of the system.

### 7.1 Experimental methodology

After the system was deployed and explained to the end users, the test phase began. During the testing phase the users inserted real data into the system and we could find flaws of the system that were not visible during the developers testing. For example, some images couldn not be uploaded because they exceeded the ASP.Net default 4 MB request size restriction and some fields were not practical and needed to change, like the calendar method to add dates had to change, because it restricted the users to enter full dates only (dd-mm-yyyy) where in some cases the day or the month was missing.

The location of the deployment was the architecture departments building in Ledras Street, downtown Nicosia. The users were the mesarch lab personnel, the actual users who will use the system and for whom the system was built for. The server that the system was migrated on, was the Architectures departments server.

<u>Server characteristics:</u>

- Server edition: Windows server 2008 R2 standard
- Server manufacturer: IBM

- Model: System x3550 M3 -[7944ZF7]-
- Processor: Intel Xeon CPU E 5620 @2.40 GHz 2.53 GHz
- System type: 64 bit operating system
- SQL server edition:  MS SQL Server 2008 R2
- Features installed: web server (IIS) tools, .Net framework 3.5.1
- Hard disk storage: 4 TB

## 7.2 Deloyment design

The system was implemented using the 3-tier architecture model, presentation, application and data tiers.

Presentation tier: This tier is the topmost one and is responsible for displaying information to the users.  It communicates with the application tier by which it puts out the results to the browser/client.

Application tier: This tier controls the application's functionality by performing detailed processing. It consists of the web server and the ASP.Net technologies. It communicates with the data tier by which it retrieves data to send to the presentation tier.

Data tier: This tier is the lowermost one and it includes data persistence mechanism, the database server that stores the information of the system. It communicates with the application tier through its API and sends the requested data.

### System deployment

The system was deployed on the Department of architecture of university of Cyprus server. At the beginning, a meeting was arranged with the IT personnel for the migration of the individual components of the system on the server.

The database (data tier) was migrated on their SQL Server 2008 R2 and the ASP.Net source code on the IIS server (application tier) in order for the system to be functional on the actual location that was implemented for and to provide users access to the website contents (presentation tier).

A training meeting was set as well with the mesarch lab personnel to discuss the system functionality, to explain to them how it works, how data are inserted and how they can interact with the system and in general solve any possible questions.

Finally, for a testing period of over three months, I was in touch with the testers of the system who were constantly giving me feedback regarding system errors and suggestions for improvement. The users interacted with the system for one week; they filled a list of feedback, functional errors and suggestions for improvement, and they forwarded it to me. The following week, considering the feedback received, code improvements and updates took place and the lists were forwarded back with comments; whether and how the issues were solved. This procedure ended in early May, when the final updates were delivered.

## 7.3 Evaluation

The main purpose of the system was to provide the mesarch lab's personnel an interactive, usable, user friendly, online tool to manage and present their lab's information and data. In order to meet these goals, I now proceed with the definition of the criteria that will be used for the user acceptance evaluation process.

Quality metrics used:

1. Learning time
2. Ease of reusability
3. Ease of learning
4. User friendliness
5. System simplicity
6. Ease of use
7. Need of written instructions
8. Ease of error recoverability
9. Successful usage rate

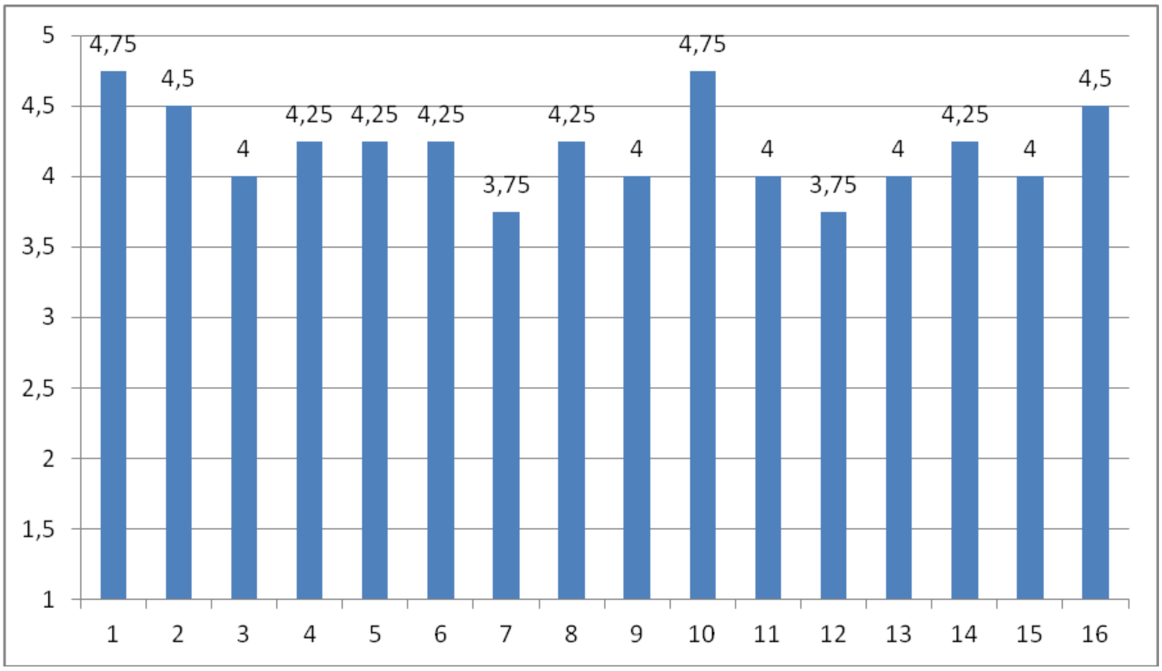
10. Usefulness
11. Integration level

12. Effectiveness
13. Productivity
14. User satisfaction
15. User needs fulfillment
16. User agreeableness

<u>Questions asked</u>

1. I learned to use the system quickly
2. I easily remember how to use it
3. It’s easy to learn to use it
4. The system is user friendly
5. It is simple to use
6. It is easy to use
7. I can use the system without written instructions
8. I can recover from mistakes quickly and easily
9. I can use it successfully every time
10. The system is useful
11. It does everything I expect it to do
12. It helps me be more effective
13. It helps me be more productive
14. I am satisfied with it
15. It works the way I want it to work
16. It is pleasant to use

Results:



The Y axis of the chart represents user answers based on the Likert scale (1-Strongly disagree, 5-Strongly agree) and the X axis represents the questions asked, presented by their corresponding number. A total of 16 system quality metrics were used to evaluate the system. Regarding time spent to learn the system ease of reusability and ease of learning, the majority of the users answered they could learn to use the system very quickly. Additionally, once they learned it, it was easy to reuse it and finally the users found it easy to learn. Regarding user friendliness, interface simplicity and ease of use evaluation results were satisfying as well. Overall, users found it user friendly and very simple and easy to use. Furthermore, very few written instructions were needed for the users to be able to use it and this is satisfying as well. Regarding ease of error recoverability and successful usage rate, the majority of users answered positively. Users can recover from errors easily and overall the system has a high successful usage rate. Regarding usefulness, integration level, effectiveness and productivity the evaluation results were satisfying as well. Users found the system very useful and complete. Additionally, it let them users be more productive and effective while they used it. Finally, regarding user satisfaction, user needs fulfillment and user agreeableness we got very positive results. In general, users were very satisfied by the system, it fulfilled the majority of their needs and they were happy about it. To sup up, the evaluation results were very satisfying. This indicates that the system was well adapted to the users and fulfilled their needs.

# Chapter 8

## Conclusions and future work

### Conclusions

The purpose of this dissertation was the design and the implementation of an interactive, usable, user friendly, online tool to meet the needs of the Mesarch laboratory of Architecture Department, University of Cyprus. The system should provide a platform where the members of the lab could manage and present academic information in a user friendly way to architecture researchers, academic people and the public in general. The information should be organized around three major entities, projects, media and actors and additional features like result filtering and advanced search should make the search and navigation easier.

In order to meet the laboratory's needs, a 3-tier web application was designed and developed around the requirements specified by the users. An innovative, responsive, user-friendly interface was developed and served as the presentation tier of the application. Through the interface the information was presented to the users organized around the three major entities and supporting navigation facilities. Additionally, the interface provided data management facilities allowing authorized users to have full control over the data of the system. ASP.Net technologies and Architecture Department's web server served as the application tier, that controlled the application's functionality by performing detailed processing and communicating with the data tier. Finally, a database was designed to store the information of the application and a database API was developed as it was needed to be used for many functionalities supported by the application. The database served as the data tier that sends the requested data to the application tier. Comparing the current project with the projects studied in the related work chapter, we can conclude it provides many functionalities; major and additional ones: search, advanced search, filters, login/create account, administrator panel, ajax technologies, google map, network of actors, email sending, comments sending and finally a recently seen recommender system.

| search | advanced search | filters | login | Admin panel | sitemap | other languages | follow on network | share on network |
|---|---|---|---|---|---|---|---|---|
| x | x | x | x | x | | | | |

| Ajax | Print page | videos | download | google map | network of actors | workplace | send email | comments | recommender system |
|---|---|---|---|---|---|---|---|---|---|
| x | | | | x | x | | x | x | x |

The system was deployed on the Department of architecture of university of Cyprus server followed by a testing period of over three months where the end-users could provide feedback regarding system errors and suggestions for improvement. After the testing period was over, a questionnaire that checked various quality metrics was answered by the end-users and the system could be evaluated as a whole. The results of the evaluation were quite positive; the system is easy to learn and use, it is simple and user friendly, it has high successful usage rate, it promotes productivity and effectiveness and it achieved high overall user satisfaction and agreeableness. In other words, the system was well adapted to the users and fulfilled their needs.

**Future work**

There are a few things and features that can be added to the system to improve it and make it more interactive. For example, a Google map showing the position of each project can be added in the future to help the users view its actual location. Project locations can be summarized in a global map and users will be able to navigate to each project via the map. This feature may help researchers who are interested in a particular sub area of the eastern Mediterranean and want to study the projects of a specific location.

Additionally an “exhibition” feature can be added to make the system more interactive. Specific users (with the appropriate role) will have the option to gather project, media and actors info and organize exhibition sessions in the website where they can present information around a specific topic. For example, they can gather all the projects of an individual actor that is involved in and present them, or they can gather all the projects around a specific location.

The system lists dozens of different actors many of whom established different collaborations with each other through their career. A researcher may be interested in studying these collaborations and find out the results and the effects of a collaboration in the architecture of different projects. This can lead us to the addition of another feature which will summarize and present the collaborations between different actors as a graph where nodes can be the actors and the collaborations and edges the relationships between them.

# Bibliography

1. Aletras, N., Stevenson, M., & Clough, P. (2012). Computing similarity between items in a digital library of cultural heritage.
2. *Arch daily*. (n.d.). Retrieved from Arch daily web site: www.archdaily.com/
3. *Architizer*. (n.d.). Retrieved from Architizer web site: www.architizer.com/en_us
4. *Artlex*. (n.d.). Retrieved August 1996, from Artlex web site: www.artlex.com/
5. *Contemporist*. (n.d.). Retrieved from Contemporist web site: www.contemporist.com/
6. *Great buildings collections*. (n.d.). Retrieved 1997, from Great buildings collections web site: www.greatbuildings.com/gbc.html
7. *GroupHomeDeal*. (n.d.). Retrieved from GroupHomeDeal web site: www.grouphomedeal.com/
8. Harris, T., & Morgan, J. (2011). *Marcel Breurer digital archive*. Retrieved from Marcel Breurer digital archive web site: breuer.syr.edu/
9. *Hiroshi Sugimoto*. (n.d.). Retrieved from Hiroshi Sugimoto web site: www.sugimotohiroshi.com
10. *Let us design/build*. (n.d.). Retrieved from Let us design/build web site: www.letusserveyou.com/
11. Lucian, G. S. (n.d.). *Homedit*. Retrieved from Homedit web site: http://www.homedit.com/
12. Matthews, P., & Aston, J. (2012). Interactive multimedia ethnography: Archiving workflow, interface aesthetics and metadata.
13. Mimno, D. (2012). Computational historiography: Data mining in a century of classics journals.
14. Postmus, M. (2009). *Archello*. Retrieved from Archello web site: www.archello.com/en
15. *Sadler Brown Architecture*. (n.d.). Retrieved from Sadler Brown Architecture web site: www.sadlerbrown.co.uk/default.aspx
16. *Schott glass made of ideas*. (n.d.). Retrieved from Schott web site: www.schott.com
17. Seltzer, D. J. (n.d.). *Agilitynut*. Retrieved 2000, from Agilitynut web site: www.agilitynut.com/roadside.html
18. Visnjic, F., & Panfilo, P. (n.d.). *Archigram*. Retrieved from Archigram web site: archigram.westminster.ac.uk

# Appendix A

## Database API

### Stored procedures

1. fn_GET_FILTERED_CREATOR(@FROMcreation_date_year INT,@TOcreation_date_year INT,@isUnknown BIT,@project_id INT ,@location NVARCHAR(MAX),@IDMediaType INT,@searchString NVARCHAR(MAX))

   This procedure returns the media creators and the number of the creators returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel. For example a user may select a location and a project to reduce the result size of this function.

2. fn_GET_FILTERED_LOCATION(@FROMcreation_date_year INT,@TOcreation_date_year INT,@isUnknown BIT,@project_id INT ,@creators NVARCHAR(MAX),@IDMediaType INT,@searchString NVARCHAR(MAX))

   This procedure returns the media location and the number of the locations returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel.

3. fn_GET_FILTERED_MEDIA(@FROMcreation_date_year INT, @TOcreation_date_year INT,@isUnknown BIT,@project_id INT ,@location NVARCHAR(MAX),@creators NVARCHAR(MAX),@IDMediaType INT,@startRow INT,@maxShownItems INT,@sortBy NVARCHAR(MAX),@searchString NVARCHAR(MAX))

   This procedure returns the media names and the number of the names returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel.

4. fn_GET_FILTERED_MEDIA_PROJECT(@FROMcreation_date_year INT,@TOcreation_date_year INT,@isUnknown BIT,@location NVARCHAR(MAX),@creators NVARCHAR(MAX),@IDMediaType INT,@searchString NVARCHAR(MAX))

This procedure returns the projects and the number of the projects returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel.

5. fn_GET_FILTERED_MEDIA_TYPE(@FROMcreation_date_year INT,@TOcreation_date_year INT,@isUnknown BIT,@project_id INT ,@location NVARCHAR(MAX),@creators NVARCHAR(MAX),@searchString NVARCHAR(MAX))

This procedure returns the media types and the number of the media types returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel.

6. fn_GET_FILTERED_PROGRAM_TYPE(@FROMdate_year INT,@TOdate_year INT,@project_type_id INT ,@project_type_details_id INT ,@location NVARCHAR(MAX),@actor_id INT,@searchString NVARCHAR(MAX))

This procedure returns the program types and the number of the program types returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel.

7. fn_GET_FILTERED_PROJECT_ACTOR(@FROMdate_year INT,@TOdate_year INT,@project_type_id INT ,@project_type_details_id INT ,@program_type_id INT ,@location NVARCHAR(MAX),@searchString NVARCHAR(MAX))

This procedure returns the actors and the number of the actors returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel.

8. fn_GET_FILTERED_PROJECT_LOCATION(`@FROMdate_year INT,@TOdate_year INT,@project_type_id INT ,@project_type_details_id INT ,@program_type_id INT ,@actor_id INT,@searchString NVARCHAR(MAX)`)

This procedure returns the project location and the number of the locations returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel.

9. fn_GET_FILTERED_PROJECT_TYPE(`@FROMdate_year INT,@TOdate_year INT,@project_type_details_id INT ,@program_type_id INT ,@location NVARCHAR(MAX),@actor_id INT,@searchString NVARCHAR(MAX)`)

This procedure returns the project type and the number of the project types returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel.

10. fn_GET_FILTERED_PROJECT_TYPE_DETAILS(`@FROMdate_year INT,@TOdate_year INT,@project_type_id INT ,@program_type_id INT ,@location NVARCHAR(MAX),@actor_id INT,@searchString NVARCHAR(MAX)`)

This procedure returns the project type details and the number of the project type details returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel.

11. fn_GET_FILTERED_PROJECTS(`@FROMdate_year INT,@TOdate_year INT,@location NVARCHAR(MAX),@project_type_id INT ,@project_type_details_id INT ,@program_type_id INT ,@actor_id INT,@startRow INT,@maxShownItems INT,@sortBy NVARCHAR(MAX),@searchString NVARCHAR(MAX)`)

This procedure returns the projects and the number of the projects returned. The results are filtered by the parameters of the function that represent the user options in the media filter panel.

12. fn_GET_MEDIA_COUNT(`@FROMcreation_date_year INT,@TOcreation_date_year INT,@isUnknown BIT,@project_id INT ,@location NVARCHAR(MAX),@creators NVARCHAR(MAX),@IDMediaType INT,@searchString NVARCHAR(MAX)`)

This procedure returns the number of the medias returned based on the filters selected by the user.

13. fn_GET_PROJECTS_COUNT(`@FROMdate_year INT,@TOdate_year INT,@location NVARCHAR(MAX),@project_type_id INT ,@project_type_details_id INT ,@program_type_id INT ,@actor_id INT,@searchString NVARCHAR(MAX)`)

This procedure returns the number of the projects returned based on the filters selected by the user.

### Scalar-valued functions

1. fn_COUNTRY_CITY(`@ID_COUNTRY INT,@CITY NVARCHAR(MAX)`)

This function is used for the computed column COUNTRY_CITY (project, media) and returns unknown in cases where neither city nor country is available. It returns only city or only country when one of the two is available and both of them separated by a dash when they are both  available.

2. fn_PROJECT_SEARCH_DATE(`@PROPOSAL_DATE_YEAR SMALLINT,@COMPLETION_DATE_YEAR SMALLINT,@CONSTRUCTION_DATE_YEAR SMALLINT`)

This function is used for the computed column SEARCH_YEAR (project) and defines which year will be used for the search in project filtering. If PROPOSAL_DATE_YEAR is available, this is used, after that we check whether the COMPLETION_DATE_YEAR is available and we return this and if it doesn’t exist either, it returns the CONSTRUCTION_DATE_YEAR. If neither of the above exists, it returns null.

3. fn_SHORTEN_MEDIA_TITLE(`@ID_MEDIA INT,@MAX_CHARS INT`)

It is used for the computed column SHORT_TITLE (media). If the title has more characters than the @MAX_CHARS variable, it returns the @MAX_CHARS of the title and it adds three dots at the end.

4. fn_SHORTEN_PROJECT_TITLE(`@ID_PROJECT INT,@MAX_CHARS INT`)

It is used for the computed column SHORT_TITLE (project). If the title has more characters than the @MAX_CHARS variable, it returns the @MAX_CHARS of the title and it adds three dots at the end.

5. fn_CREATION_DATE(`@creationDateDay INT,@creationDateMonth INT,@creationDateYear INT`)

It is used for the computed column CREATION_DATE (media) and it returns the date in the form: dd-mm-yyyy. If a field is not available, it doesn't show it.