



the
UNIVERSITY
of
GREENWICH

ZALOKOSTAS MANOS

Supervisor: **DR. IOANNIS PANDIDAS**

JUNE 2012

NY COLLEGE SCHEDULE HANDLER

A dissertation submitted in partial fulfilment of the University of Greenwich's

INTERNET ENGINEERING & WEB MANAGEMENT

ACKNOWLEDGEMENTS

I would like to thank my supervisor Dr. Ioannis Pandithas for supporting me during the design and development of the project and has stood right next to me, clearing out my thinking and showing me a linear path whenever I found my self confused.

I would additionally like to thank Dr. Petros Bofakos, instructor of the college, that even though not a supervisor of mine, has admitted to examine the database when I ask him to, and has also injected me with ideas for the development phase.

At this section I would also like to state that the system that is deployed on the following pages is totally designed and developed from scratch. There are no code snippets that have been borrowed by others and therefore the readers will not find referenced materials. Application's PHP functionality urges from studding that comes from the past years in college and only HTML and CSS coding comes from previous years studies. Nevertheless there were books that have been studied while designing and developing the project and those are acknowledged in the bibliography section.

ABSTRACT

The system that is designed and developed in the following pages is a web tool that undertakes to build and maintain a schedule delivery for an academic year. The origin of the idea was proposed by supervisor Dr. Ioannis Pandithas which, as an executive member of the college, has located the need for a component that will successively substitute the way that the procedure is designed, delivered and monitored today. The way that the schedule is running currently primarily relies on invoking Microsoft Excel spreadsheets that iterate a 'Master Schedule' pattern, arranged by administration before the beginning of a semester, for every following day of the semester. It was a belief that this technique is much demanding and needs increased manpower to build and much also to sustain. The intention was to migrate the activity in a database-centric environment that will run on the WEB and will assist distant campuses to implicate a common share instrument. Moreover the implication of a database driven application would additionally diffuse increased usability to the users, especially while the recordings were needed to be examined and meaningful results were about to drawn out.

In the following lines we will cover extensively the process that was taken before we actually come up with a system that could potentially cover the needs of a schedule handler for the college.

CONTENTS

ACKNOWLEDGEMENTS	iii
ABSTRACT	iv
CONTENTS	v
FIGURES	vii
1 INTRODUCTION	1
1.1 AIMS & OBJECTIVES	1
1.2 CHAPTERS OVERVIEW	1
2 BASIC THEORETICAL ELEMENTS	3
2.1.1 XHTML (V. 1.0)	3
2.1.2 CSS (V. 2.0)	3
2.1.3 PHP (V. 5.2.0)	3
2.1.4 MySQL (V. 5.0)	3
2.1.5 APACHE (V. 2.2.4)	4
2.1.6 NOTEPAD++ (V. 6.1.2)	4
3 REQUIREMENTS ANALYSIS	5
3.1 THE PROCEDURE	5
3.2 THE LEGACY SYSTEM	5
3.3 THE PROBLEM	7
3.4 THE NEED	7
4 THE REQUIREMENTS	9
4.1.1 ISSUED REQUIREMENTS	9
4.1.2 USER REQUIREMENTS (ABSTRACT)	10
5 DESIGN	11
5.1 DESIGN CONSIDERATIONS	11
5.2 REQUIRED TECHNOLOGY	12
5.3 CATEGORIZATION OF FUNCTIONS	13
6 DEVELOPMENT	14
6.1 PAGE-CONTAINER STRUCTURE	14
6.2 NAVIGATION MENU	15
6.2.1 LOGIN SCRIPT	15
6.3 USER AUTHENTICATION	16
6.4 MAIN MENU	17

6.5	SUB-NAVIGATION MENU	18
6.6	IFRAME - PAGES CONTAINER.....	19
6.7	PAGE STRUCTURE	20
6.8	PAGES GENERATION AND INTERACTION	21
6.8.1	ORIGINAL PAGES AND DERIVATIVES	21
6.9	THE PAGE DESIGN.....	22
6.10	'ACTION' BUTTONS.....	24
6.11	FORM DESIGNERS.....	27
6.11.1	'HTML FORM DESIGNER'	28
6.11.2	BUILD FILTERS IN A FORM (search/insert)	31
6.12	DATA TYPE VALIDATOR.....	33
6.13	LIST RECORDS AREA.....	35
6.14	FEEDBACK MESSAGES.....	37
6.15	SCHEDULE FLOW	39
6.15.1	COURSES & INSTRUCTORS.....	41
6.15.2	MASTER SCHEDULE.....	43
6.16	DAY SCHEDULE	44
6.17	DATABASE ENTITIES- SCHEDULE.....	46
6.18	DATABASE - SCHEDULE ERD	49
6.19	DATABASE ENTITIES - SYSTEM.....	51
6.20	DATABASE - SYSTEM ERD	53
7	IMPLEMENTATION.....	54
7.1	LOGIN DESCRIPTION	54
7.2	VALID USER 'ADMIN'	56
7.2.1	'FACILITIES' FAMILY OF FUNCTIONS.....	56
7.2.2	'SCHEDULE' FAMILY OF FUNCTIONS.....	68
7.3	ENABLING/DISABLING RECORDS.....	89
7.4	ESTIMATIONS.....	93
7.5	'SYSTEM' FAMILY OF FUNCTIONS.....	100
7.6	VALID 'EXECUTIVE' USER	129
7.7	VALID 'FRONT-DESK' USER	132
7.8	VALID 'PUBLIC' USER	136
7.9	SECURITY	139
7.9.1	SQL CODE INJECTION - LOGIN	139
7.9.2	SQL CODE INJECTION - FORMS.....	140

7.9.3	USER PASSWORD	144
7.9.4	TABLE PERMISSIONS.....	145
7.9.5	PREDETERMINED DATA	145
HIDE CRITICAL INFORMATION FROM THE SCREEN		146
7.9.6	TABLE & COLUMN NAME ALIASES.....	146
7.9.7	ENCODE 'URL' VARIABLES	147
7.9.8	ERROR HANDLING	148
7.9.9	SECURE PAGE ACCESS.....	150
BIBLIOGRAPHY		154

FIGURES

Figure 4.1- LEGACY SYSTEM - MASTER SCHEDULE	6
Figure 4.2 - LEGACY SYSTEM - DAY SCHEDULE	6
Figure 4.3 - DATABASE RELATIONSHIPS.....	8
Figure 4.4 - DATABASE CASCADE DATA.....	8
Figure 4.5 - PROJECT VISUALIZATION	9
Figure 4.6 - FUNCTIONAL REQUIREMENTS.....	10
Figure 5.1 PROJECT PLANNING.....	12
Figure 5.2 REQUIRED TECHNOLOGIES	13
Figure 5.3 FUNCTION CATEGORIES	14
Figure 6.1 DISPLAY LAYOUT	15
Figure 6.2 LOGIN ALTERNATIVE VIEWS	16
Figure 6.3 USER AUTHENTICATION PROCESS.....	17
Figure 6.4 MAIN-NAVIGATION BUTTONS DESIGN.....	18
Figure 6.5 SUB-NAVIGATION BUTTONS DESIGN	19
Figure 6.6 IFRAME CONTAINER	20
Figure 6.7 LOADED PAGES LAYOUT	21
Figure 6.8 MAIN PAGES OF THE SYSTEM.....	22
Figure 6.9 REAL-TIME PAGE DESIGN	23
Figure 6.10 BUTTONS DESIGN PROCESS.....	24
Figure 6.11 BUTTONS DETERMINE PAGE DESIGN	26
Figure 6.12 PAGES FORMS.....	27
Figure 6.13 CLI 'DESCRIBE TABLE' FEEDBACK	28

Figure 6.14 FORM INPUTS DESIGN PROCESS	30
Figure 6.15 CLI 'SHOW CREATE TABLE' PROCESS	31
Figure 6.16 MECHANISM THAT 'COURSE-INSTRUCTOR' & 'MASTER SCHEDULE' USE TO STORE DATA TEMPORARILY BEFORE SAVED	33
Figure 6.17 SUBMITTING DATA	34
Figure 6.18 VALIDATE SUBMITTED DATA	35
Figure 6.19 DISPLAY DATA ON SCREEN PROCESS.....	36
Figure 6.20 DESIGN THE LISTING OF RECORDS	37
Figure 6.21 FORM DATA SUBMISSION GENERATE MESSAGES.....	38
Figure 6.22 'QUERY RUN' ASSIGNMENTS	39
Figure 6.23 FORMAL SCHEDULE FLOW	40
Figure 6.24 DB DESIGN - COURSES & INSTRUCTORS ERD	42
Figure 6.25 DB DESIGN - MASTER SCHEDULE ERD	44
Figure 6.26 DB DESIGN - DAY SCHEDULE ERD	45
Figure 6.27 DB ENTITY - ACEDMIC SEMESTER.....	46
Figure 6.28 DB ENTITY - ACADEMIC YEAR	46
Figure 6.29 DB ENTITY - ACTIVE.....	46
Figure 6.30 DB ENTITY - CLASS	46
Figure 6.31 DB ENTITY - CLASS TYPE.....	46
Figure 6.32 DB ENTITY - CAMPUS.....	46
Figure 6.33 DB ENTITY - COURSE	47
Figure 6.34 DB ENTITY - COURSE INSTRUCTOR	47
Figure 6.35 DB ENTITY - DAY SCHEDULE	47
Figure 6.36 DB ENTITY - DEGREE ASSOCIATION	47
Figure 6.37 DB ENTITY - DEGREE DOMAIN	47
Figure 6.38 DEGREE LEVEL.....	48
Figure 6.39 DB ENTITY - DEGREE TITLE.....	48
Figure 6.40 DB ENTITY - DEGREE YEAR.....	48
Figure 6.41 DB ENTITY - INSTRUCTOR	48
Figure 6.42 DB ENTITY - MASTER SCHEDULE.....	48
Figure 6.43 DB ENTITY - TIME SHIFT	49
Figure 6.44 DB DESIGN - SCHEDULE ERD.....	50
Figure 6.45 DB ENTITY - ABBREVIATIONS.....	51
Figure 6.46 DB ENTITY - FUNCTIONS	51
Figure 6.47 DB ENTITY - FUNCTION CATEGORY	51

Figure 6.48 DB ENTITY - PERMISSION.....	51
Figure 6.49 DB ENTITY - PRIVILEGE.....	51
Figure 6.50 DB ENTITY - STATEMENT	51
Figure 6.51 DB ENTITY - SYSTEM ACTIVITY.....	52
Figure 6.52 DB ENTITY - USER.....	52
Figure 6.53 DB ENTITY - USER ACTIVITY	52
Figure 6.54 DB DESIGN - BACK-END SYSTEM ERD	53
Figure 7.1 WELCOME PAGE	54
Figure 7.2 USER LOGIN	54
Figure 7.3 REJECT LOGIN CREDENTIALS	55
Figure 7.4 VALID USER LOGIN.....	55
Figure 7.5 ENDORSE VALID USER	56
Figure 7.6 USER 'ADMIN'	57
Figure 7.7 'ADMIN' ACTION BUTTONS	57
Figure 7.8 'LIST ALL' PREVIEW	58
Figure 7.9 CLICK LINK HEADERS.....	58
Figure 7.10 RECORDS ASCENDING ORDER	59
Figure 7.11APPLY SEARCH WITHOUT VALID ARGUMENTS	59
Figure 7.12 SEARCHING FILTERS.....	60
Figure 7.13 SEARCHING RESULTS	60
Figure 7.14 EDIT RECORD	61
Figure 7.15 UPDATE RECORDS	61
Figure 7.16 SYSTEM RESPONCE ON UPDATES.....	62
Figure 7.17 PREVIEW UPDATED RECORDS	62
Figure 7.18 INSERT NEW DATA.....	63
Figure 7.19 SYSTEM RESPONCE ON REQUEST OF INSERTING INVALID DATA.....	63
Figure 7.20 INSERT VALID DATA	64
Figure 7.21 SYSTEM RESPONCE ON VALID ENTRIES.....	65
Figure 7.22 VERIFY SUBMITTED DATA.....	65
Figure 7.23 DELETE DATA USING 'X' BUTTON	66
Figure 7.24 PREVIEW 'ASSOCIATION' TABLE DATA	66
Figure 7.25 SYSTEM PREVENTS THE DELETION OF REFERENCED DATA.....	67
Figure 7.26 TABLES WITHOUT DATA OPTIONS ON 'INSERT' MODE.....	67
Figure 7.27 SYSTEM RESPONCE WHILE SEARCHING FILTERS ARE MISSING.....	68
Figure 7.28 'COURSES & INSTRUCTORS' TABLE	69

Figure 7.29 'COURSES & INSTRUCTORS' ALL DATA LISTING	69
Figure 7.30 'COURSES & INSTRUCTORS' FILTERED SEARCH	69
Figure 7.31 FILTERED SEARCH RESULTS	70
Figure 7.32 OMIT TO APPLY DATA FOR 'REQUIRED' INPUTS.....	71
Figure 7.33 SYSTEM RESPONSE ON APPLYING INVALID DATA.....	71
Figure 7.34 APPLY VALID RECORDS TO FORM INPUTS.....	72
Figure 7.35 'COURSES & INSTRUCTORS' USING SESSION TO REGISTER DATA	72
Figure 7.36 DELETE RECORDSETS FROM SESSIONS ENTRIES	73
Figure 7.37 SYSTEM RESPONSE WHILE SUBMITTING SESSION ENTRIES.....	73
Figure 7.38 VERIFY THAT SESSION DATA ARE SUBMITTED	74
Figure 7.39 PREVIEW THE 'MASTER SCHEDULE' PAGE	75
Figure 7.40 PREVIEW 'MASTER SCHEDULE' ALL DATA	75
Figure 7.41 'MASTER SCHEDULE' FILTERS FORM	76
Figure 7.42 'MASTER SCHEDULE' INSERT FORM	76
Figure 7.43 'STORE' DATA USING SESSION	77
Figure 7.44 VERIFY SUBMITTED DATA ACTUALLY SAVED.....	77
Figure 7.45 USING MASTER SCHEDULE'S 'RETRIEVE' BUTTON TO COLLECT DATA FROM 'COURSES & INSTRUCTORS' TABLE	78
Figure 7.46 ENHANCE THE 'RETRIEVED' RECORDSETS	79
Figure 7.47 'RETRIEVED' RECORDS UPDATED.....	79
Figure 7.48 SYSTEM ESCAPES TO 'RETRIEVE' FILLED IN RECORDSETS.....	80
Figure 7.49 PREVIEW DATA IN 'COURSES & INSTRUCTORS' TABLE BEFORE RETRIEVE.....	81
Figure 7.50 PREVIEW DATA IN THE 'COURSES & INSTRUCTORS' TABLE AFTER THE 'RETRIVE' BUTTON WAS CLICKED	81
Figure 7.51 'TODAY SCHEDULE'	82
Figure 7.52 'DAY SCHEDULE' ALL RECORDS LIST	83
Figure 7.53 'DAY SCHEDULE' SEARCH FILTERS.....	84
Figure 7.54 APPLY 'DAY SCHEDULE' FILTERS.....	85
Figure 7.55 'DAY SCHEDULE' REPLY ON SEARCH	85
Figure 7.56 'DAY SCHEDULE' INSERT FORM	86
Figure 7.57 INSERT A 'COURSE' FOR THE CURRENT DAY.....	87
Figure 7.58 SAME DAY 'INSERTED' RECORDSET IS PREVIEWED	87
Figure 7.59 'RECORD' THE DAILY ACTIVITY.....	88
Figure 7.60 THE DAILY ACTIVITY IS REGISTERED	88
Figure 7.61 DAILY ACTIVITY WILL BE PREVIEWED EVEN AFTER REGISTERED.....	89

Figure 7.62 DEACTIVE A COURSE.....	90
Figure 7.63 DISABLED COURSE WILL NOT PREVIEW ON 'COURSES & INSTRUCTORS'	90
Figure 7.64 DEACTIVATED COURSES WILL NOT PREVIEWED ON 'MASTER SCHEDULE'	91
Figure 7.65 DEACTIVATE INSTRUCTOR	91
Figure 7.66 DISABLED INSTRUCTORS ARE NOT PREVIEWED IN 'COURSES & INSTRUCTORS'	92
Figure 7.67 DISABLED INSTRUCTORS ARE NOT PREVIEWED IN MASTER SCHEDULE'	92
Figure 7.68 DISABLED INSTRUCTORS ARE NOT PREVIEWED IN 'DAY SCHEDULE'	93
Figure 7.69 'ESTIMATIONS' PAGE	94
Figure 7.70 'ESTIMATIONS' FILTERS	94
Figure 7.71 SYSTEM RESPONCE ON INVALID SEARCH INPUTS	95
Figure 7.72 SELECT AN INSTRUCTOR FOR SEARCHING IN 'ESTIMATIONS' PAGE	96
Figure 7.73 SELECT MANY FILTERS IN 'ESTIMATIONS' PAGE.....	96
Figure 7.74 SYSTEM RESPONCE ON MULTIPLE FILTERS	97
Figure 7.75 'ESTIMATIONS' SELECT MOST OF THE FILTERS.....	97
Figure 7.76 SYSTEM RESPOND WITH DETAILED INFORMATION	98
Figure 7.77 'ESTIMATIONS' CHOOSE THE DATE FILTERS	98
Figure 7.78 SYSTEM RESPONCE ON 'ESTIMATION' DATE FILTERS.....	99
Figure 7.79 'ESTIMATIONS' SELECT TO PREVIEW FALLBACK LESSONS.....	99
Figure 7.80 'ESTIMATIONS' RESULTS TO FALLBACK LESSONS SEARCH	100
Figure 0.1 'USERS' TABLE DATA	101
Figure 0.2 ADD A NEW USER WIH ADMINISTRATION PRIVILEGES	102
Figure 0.3 DISPLAY A REGISTERED USER ON SCREEN.....	103
Figure 0.4 EDIT A REGISTERED USER	103
Figure 0.5 CHANGE A USER PASSWORD	104
Figure 0.6 SYSTEM RESPONCE ON SUCCESSFUL PASSWORD CHANGE	104
Figure 0.7 VERIFY USER DATA UPDATED	105
Figure 0.8 LOGIN WITH THE A NEW USER'S CREDENTIALS	105
Figure 0.9 A NEW USERS WITH ADMINISTRATION ACCESS GRANTS PRIVILEGED ACCESS TO THE SYSTEM	106
Figure 0.10 DISPLAY ALL THE 'ROLES' THAT EXIST IN THE SYSTEM	107
Figure 0.11 CREATE A NEW ROLE FOR THE SYSTEM	107
Figure 0.12 A NEW 'ROLE' IS CREATED	108
Figure 0.13 ASSIGN A NEW ROLE TO A NEW USER	108
Figure 0.14 A NEW ROLE IS SUCCEFULLY ATTACHED TO A NEW USER	109
Figure 0.15 A NEW USER WITH NEW ROLE ATTACHED LOGS IN THE SYSTEM.....	109

Figure 0.16 A USER WITH A NEW ROLE ATTACHED SUCCESSFULLY LOGS IN THE SYSTEM.....	110
Figure 0.17 ATTACH PERMISSIONS TO A NEW ROLE.....	110
Figure 0.18 TABLE PERMISSION IS ATTACHED TO A NEW ROLE	111
Figure 0.19 A USER WITH A NEW ROLE ATTACHED ONLY HAS CERTAIN PERMISSIONS	111
Figure 0.20 A NEW ROLE REVEALS THE ACCESS THAT HAS BEEN ASSIGNED	112
Figure 0.21 THE SYSTEM WILL NOT REVEAL 'ACTION' BUTTONS TO ROLES THAT DO NOT HAVE EXPLICIT PERMISSION	112
Figure 0.22 THE SYSTEM WILL ALLOW TO A NEW ROLE ONLY THE ACCESS THAT IS ASSIGNED. THE 'EDIT' ACTION IS NOT REVEALED	113
Figure 0.23 A NEW ROLE ACQUIRES MORE PERMISSIONS ON THE SYSTEM TABLES.....	114
Figure 0.24 THE ACCESS PRIVILEGES OF A NEW ROLE ARE VERIFIED ON THE 'TABLE PERMISSIONS' PAGE	115
Figure 0.25 A NEW ROLE ENTERS THE SYSTEM AND VERIFIES THE ACCESS PERMISSIONS THAT IS GRANTING. MORE BUTTONS ARE AVAILABLE	115
Figure 0.26 A NEW ROLE REVEALS THE SUB-BUTTONS ACCESS THAT IS GRANTING	116
Figure 0.27 ALL 'FUNCTION-CATEGORY' DATA.....	117
Figure 0.28 CREATE A NEW CATEGORY OF FUNCTIONS.....	118
Figure 0.29 VERIFY A NEW CATEGORY IS CREATED.....	118
Figure 0.30 CREATE A NEW FUNCTION	119
Figure 0.31 VERIFY A NEW FUNCTION IS CREATED	120
Figure 0.32 ATTACH USER PERMISSION TO A NEW FUNCTION	120
Figure 0.33 A USER HAS PRIVILEGES TO A NEW CATEGORY OF FUNCTIONS (MAIN BUTTON).....	121
Figure 0.34 A USER IS DELETED FROM THE SYSTEM	122
Figure 0.35 ALL DATA OF THE 'NAME ALIASES' TABLE	122
Figure 0.36 A TABLE ALIAS NAME IS ALTERED.....	123
Figure 0.37 A TABLE ALIAS HEADER NAME IS CHANGED	123
Figure 0.38 THE 'USER-LOGS' TABLE. 'LIST ALL' BUTTON IS DELIBERATELY HIDDEN	124
Figure 0.39 SEARCH THROUGH THE LOGS THE LOGIN ATTEMPTS.....	125
Figure 0.40 SEARCH THE 'DELETE' STATEMENTS THAT HAVE BEEN APPLIED TO THE SYSTEM	125
Figure 0.41 BACKUP PAGE	126
Figure 0.42 BACKUP PAGE FILTERS.....	126
Figure 0.43 BACKUP ENTIRE DATABASE	127
Figure 0.44 DELETE USER BEFORE BACKUP	127
Figure 0.45 RESTORE DATABASE	128
Figure 0.46 SELECT DATABASE FILE TO RESTORE	128

Figure 0.47 DELETED USER REVEALS AFTER RESTORE.....	129
Figure 0.48 VALID USER 'EXECUTIVE' LOGS IN THE SYSTEM	129
Figure 0.49 THE SYSTEM REVEALS ONLY THE BUTTONS THAT 'EXECUTIVE' USER HAS ACCESS	130
Figure 0.50 'EXECUTIVE' USER MAINTAINS FULL ACCESS PERMISSION ON 'FACILITIES' DOMAIN	130
Figure 0.51 'EXECUTIVE' USER MAINTAINS FULL ACCESS TO 'DAY SCHEDULE' PAGE	131
Figure 0.52 'EXECUTIVE' MAINTAINS FULL ACCESS UNDER 'MASTER SCHEDULE' PAGE	131
Figure 0.53 'EXECUTIVE' MAINTAINS FULL ACCESS UNDER THE 'ESTIMATIONS' PAGE	132
Figure 0.54 'FRONT DESK' USER GRANTS LOGIN ACCESS	133
Figure 0.55 'FRONT DESK' USER MAINTAINS FULL PRIVILEGES UNDER THE 'FACILITIES' DOMAIN	133
Figure 0.56 'FRONT DESK' USER MAINTAINS FULL ACCESS ON THE 'COURSE INSTRUCTOR' PAGE	134
Figure 0.57 'FRONT DESK' USER IS RESTRAINED TO 'READ' ACCESS UNDER THE 'MASTER SCHEDULE' PAGE	134
Figure 0.58 'FRONT DESK' USER CAN SEARCH THROUGH 'MASTER SCHEDULE' BUT ONLY WITH 'READ' PERMISSION	135
Figure 0.59 'FRONT DESK' USER GRANTS FULL ACCESS PRIVILEGES UNDER THE 'DAY SCHEDULE' PAGE	135
Figure 0.60 'FRONT DESK' USER IS NOT ALLOWED TO ACCESS THE 'ESTIMATIONS' PAGE	136
Figure 0.61 'PUBLIC' USER MAINTAINS ONLY 'READ' ACCESS PERMISSION UNDER THE 'FACILITIES' DOMAIN.....	136
Figure 0.62 'PUBLIC' USER CAN USE THE SEARCHING MECHANISM BUT ONLY WITH 'READ' ACCESS PERMISSION	137
Figure 0.63 'PUBLIC' USER MAINTAINS ONLY 'READ' ACCESS PERMISSIONS ON THE 'COURSE INSTRUCTOR' PAGE	137
Figure 0.64 'PUBLIC' USER MAINTAINS ONLY 'READ' ACCESS PERMISSION ON THE 'MASTER SCHEDULE' TABLE	138
Figure 0.65 'PUBLIC' USER MAINTAINS FULL ACCESS PRIVILEGE ON THE 'DAY SCHEDULE' TABLE....	138
Figure 0.66 'PUBLIC' USER IS NOT ALLOWED TO ENTER THE 'ESTIMATIONS' PAGE	139
Figure 0.67 INJECT SQL STATEMENTS TO LOGIN.....	139
Figure 0.68 LOGIN OMITS TO GENERATE SQL INJECTED CODE	140
Figure 0.69 INJECT SQL CODE TO THE FORM INPUTS.....	140
Figure 0.70 THE SYSTEM WILL AVOID TO GENERATE SQL INJECTED CODE	141
Figure 0.71 THE INJECTION OF SQL CODE TO THE 'DESCRIPTION' INPUTS WILL TREAT THE CODE LIKE COMMON STRING INPUT	141
Figure 0.72 INJECTION OF SQL CODE IN THE 'EDIT' FORM WILL TREAT THE CODE AS COMMON STRING INPUT	142

Figure 0.73 INJECTION OF HTML CODE ON THE FORM INPUTS	142
Figure 0.74 HTML VALID CODE IS DELIBERATELY SET TO OPERATE BY DEFAULT	143
Figure 0.75 HTML CODE INJECTION IS HANDLED PROPERLY IN THE DATABASE	144
Figure 0.76 USER PASSWORDS ARE BY DEFAULY ENCODED USING SHA1	144
Figure 0.77 USER PERMISSIONS ON THE DATABASE ARE ASSIGNED BY THE ADMINISTRATION.....	145
Figure 0.78 MOST OF THE DATA USED ARE ALREADY EXISTING IN THE DATABASE	146
Figure 0.79 NAME ALIASES WILL HIDE THE SCHEMA OF THE DATABASE	147
Figure 0.80 THE URL VARIBALES AND VARIBALE CONTAINERS ARE ENCODED	148
Figure 0.81 ERROR HANDLLING IS PROVIDED BY THE SYSTEM. DETAILED MESSAGES ARE ESCAPED	148
Figure 0.82 ALL SYSTEM ERRORS ARE HANDLED BY AN INTERNAL MECHANISM.....	149
Figure 0.83 USER HAS LOGIN THE SYSTEM.....	150
Figure 0.84 USER LOGS OUT THE SYSTEM.....	150
Figure 0.85 USER TRIES TO EXPLOIT THE BROWSERS 'BACK' FUNCTION BUT IS RESTRAINED ACCESS	151
Figure 0.86 USER REACHES TO THE DIRECTORY OF THE SYSTEM	152
Figure 0.87 SYSTEM WILL PROHIBIT USERS FROM FOLLOWING THE LINKS AND ENTER THE SYSTEM	152
Figure 0.88 SYSTEM WILL LOAD BLANK PAGES FOR USERS TRYING TO EXPLOIT THE LINKS OF THE DIRECTORY.....	153

1 INTRODUCTION

1.1 AIMS & OBJECTIVES

The aim of the report is to cover the design and development of a WEB application that is intended to cover the objectives for planning and running an academic year schedule. The project adds value for a wide perspective of procedures that will help the schedule keep up and running. Initially the system will need to store all the properties that are offered by the organization such as 'courses', 'instructors', 'campuses', 'classes' etc. In a next level the system will need to provide facilities only for highly-authorized users to make matches for the organization's properties and thus design a matrix pattern that resolves as the "Master Schedule". While the 'Master Schedule' is arranged the system will need to autonomously recreated every day and offer it to lower-authorized user that will need to record the daily activity in the organization. Based on true facts and not estimations, highly-authorized users will be empowered to draw out consistent and effective results, almost instantly. Additionally the system will keep recording autonomously the activity that the users are generating while using it, and highly-authorized users are enabled to display the activity. Last but not least, the system provides interfaces for highly-authorized users to control and modify it from its back-end interface.

1.2 CHAPTERS OVERVIEW

Chapter 2: We will make a reference to the technology that needed to be used in the new system

Chapter 3: We will undertake an analysis of the legacy system in order to elicit the requirements that that are issued by the legacy system and will need to be transmitted in the new system. Additionally the elicitation of legacy system requirements will reveal its weaknesses to deal with certain objectives, and those will need to be succeeded in the new system.

Chapter 4: We will address the methodology that was used, with cascading steps, in order to design the tool and provide excessive feedback for some of the most complex routines that were implemented. Those will be analyzed with diagrams and discussed so that they can be

comprehended by the reader. Additionally the core of the application, its database, will be exhibited and analyzed along with the entities that constitute it.

Chapter 5: We will travel through all the operations that are supported by the system, empowered visually with screenshots and we will inherit the confidence to the reader that the user requirements that were contracted have actually been granted. Also in this section there are informative notes for the system operation that without the assistance of visual context could hardly been supported. Last but not least this chapter provides feedback for how a prospect extension of the system could add value to some of its routines.

Chapter 6: We will find aligned all the resources that were studied for the development of the system, in the bibliography section.

Appendix-A: We will find the code that for the entities that were designed in the database of the system.

Appendix-B: We will find the development code that brings the system to life, along with verbose comments for each step that help the reader to realize why and how the system functions. Additionally the comments within the application's source code describe some of the weakness that were attached to the system at the development phase, while also they ways that those could be covered in future versions.

2 BASIC THEORETICAL ELEMENTS

The technologies that were implemented and integrated for delivering the project to life are listed below:

2.1.1 XHTML (V. 1.0)

“XHTML (eXtensible HyperText Markup Language) is a family of XML markup languages that mirror or extend versions of the widely-used Hypertext Markup Language (HTML), the language in which web pages are written.

While HTML was defined as an application of Standard Generalized Markup Language (SGML), a very flexible markup language framework, XHTML is an application of XML, a more restrictive subset of SGML. Because XHTML documents need to be well-formed, they can be parsed using standard XML parsers—unlike HTML, which requires a lenient HTML-specific parser.”

WIKIPEDIA, 2012 [ONLINE AT <http://en.wikipedia.org/wiki/HTML>, RETRIEVED ON 25/5/2012]

2.1.2 CSS (V. 2.0)

“Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML, but the language can also be applied to any kind of XML document, including plain XML, SVG and XUL.”

WIKIPEDIA, 2012 [ONLINE AT http://en.wikipedia.org/wiki/Cascading_Style_Sheets, RETRIEVED ON 25/5/2012]

2.1.3 PHP (V. 5.2.0)

“PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. If you are new to PHP and want to get some idea of how it works, try the introductory tutorial. After that, check out the online manual, and the example archive sites and some of the other resources available in the links section.”

PHP.NET, 2012 [ONLINE AT <http://www.php.net/>, RETRIEVED ON 25/5/2012]

2.1.4 MySQL (V. 5.0)

“The MySQL database has become the world's most popular open source database because of its high performance, high reliability and ease of use. It is also the database of choice for a new generation of applications built on the LAMP stack (Linux, Apache, MySQL, PHP / Perl / Python.) Many