A WEB-BASED DATABASE FOR EXPERIMENTAL STRUCTURAL ENGINEERING RESEARCH

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ÖNCEL UMUT TÜRER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CIVIL ENGINEERING

APRIL 2010

Approval of the thesis:

A WEB-BASED DATABASE FOR EXPERIMENTAL STRUCTURAL ENGINEERING RESEARCH

submitted by **ÖNCEL UMUT TÜRER** in partial fulfillment of the requirements for the degree of **Master of Science in Civil Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Güney Özcebe _____
Head of Department, **Civil Engineering**

Asst. Prof. Dr. Özgür Kurç _____
Supervisor, **Civil Engineering Dept., METU**

Examining Committee Members:

Assoc. Prof. Uğur Polat _____
Civil Engineering Dept., METU

Asst. Prof. Dr. Özgür Kurç _____
Civil Engineering Dept., METU

Assoc. Prof. Dr. Barış Binici _____
Civil Engineering Dept., METU

Assoc. Prof. Dr. Tunç Medeni _____
Informatics Institute, METU

Specialist İhsan Tolga Medeni _____
Computer Engineer, METU

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name :

Signature :

ABSTRACT

A WEB-BASED DATABASE FOR EXPERIMENTAL STRUCTURAL ENGINEERING RESEARCH

Türer, Öncel Umut

M.Sc., Department of Civil Engineering

Supervisor : Asst. Prof. Dr. Özgür Kurç

April 2010, 142 pages

This study presents a web-based database application for storing experimental data and related documents at a single location and sharing them among researchers and engineers from all over the world. The database application, accessible from a website was developed for experimental structural engineering researches, and initially tested at Structural and Earthquake Engineering Laboratory of Civil Engineering Department, METU. The application is composed of two parts. The first part is the database that stores information about projects, specimens, experiments, experimental data, documentation, site members, and member groups at the server side. The second part is the website that provides a functional user interface for easy use of application and providing accessibility from everywhere via internet. After the development of the database and the website, these two parts were attached to each other and application functionalities that enabled users to create, modify, search, and delete projects, specimens and experiments; allowed users to upload/download documentation and experimental data; provided abilities to users to plot test data and share their opinions were ensured. ASP.NET

framework and C# programming language was utilized for the web application development. Functionality and usability of the database application was then tested by uploading and sharing various experimental results.

Keywords: Web-Based Database, Experiment Database, Online Database, Database Development, Web Application for Civil Engineers.

ÖZ

DENEYSEL YAPI MÜHENDİSLİĞİ ARAŞTIRMALARI İÇİN WEB TABANLI VERİTABANI

Türer, Öncel Umut
Yüksek Lisans, İnşaat Mühendisliği Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Özgür Kurç

Nisan 2010, 142 sayfa

Bu çalışmada deneysel verilerin ve ilişkili dökümanların tek bir noktada toplanabilmesi ve bunların dünyanın her yerindeki araştırmacılar ve mühendisler arasında paylaşılabilmesi için geliştirilmiş web tabanlı veritabanı uygulaması geliştirilmiştir. Deneysel yapı mühendisliği araştırmaları için internet aracılığı ile erişilebilecek bir veritabanı uygulaması geliştirilmiş ve ilk olarak ODTÜ İnşaat Mühendisliği Bölümü Yapı ve Deprem Mühendisliği laboratuvarlarında test edilmiştir. Uygulama iki parçadan oluşmaktadır. Birinci parça, proje, numune, deney, deney verisi, dökümanlar, site üyeleri ve üye grupları hakkında bilgilerin sunucu tarafında saklanmasını sağlayan veritabanıdır. İkinci parça, uygulamanın dünyanın her yerinden kolayca kullanılabilmesini sağlayan ve aynı zamanda uygulama için fonksiyonel bir kullanıcı arabirimi görevi gören web sitesidir. Veritabanının ve web sitesinin geliştirilmesinden sonra bu iki parça birleştirilmiş ve uygulamaya kullanıcıların proje, numune ve deney oluşturabilmelerini, değiştirebilmelerini, arayabilmelerini ve silebilmelerini sağlayan; onların veritabanına döküman ve

deney verisi gönderip almalarını olanaklı kılan; bu deney verilerinin grafiksel olarak çizdirilebilmesini ve kişisel görüşlerini paylaşabilmelerini sağlayan özellikler kazandırılmıştır. Web uygulaması geliştirilirken ASP.NET ve C# programlama dili kullanılmıştır. Daha sonra, veritabanı uygulamasının fonksiyonelliği ve kullanılabilirliği çeşitli deneylerin sonuçları ve ilgili dökümanları siteye yüklenerek ve paylaşılarak sınanmıştır.

Anahtar Kelimeler: Web Tabanlı Veritabanı, Deney Veritabanı, Online Veritabanı, Veritabanı Geliştirilmesi, İnşaat Mühendisleri için Web Uygulaması.

To My Family

ACKNOWLEDGMENTS

This study was conducted under the supervision of Asst. Prof. Dr. Özgür Kurç. I would like to express my sincere thanks and appreciations for his support, guidance, criticism and encouragement during this study. It was a great honor and pleasure to work with him.

I also would like to thank to M. C. Yücel and Ç. Akyayla for their support and encouragement.

Finally, I am grateful to my parents and my brother. They always supported me for whatever I deal with, without waiting for a counterpart.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

1.1 Problem Definition

Information sharing has become a crucial issue in the enhancement of engineering approaches and methodologies due to the developments in information technologies. In structural and earthquake engineering, experimental data is one of the most important information due to the fact that most of the derived formulations and assumptions are based on experiments. Therefore, in order to fine tune formulations and increase the accuracy of calculations, various experiments must be conducted and shared among scientists and engineers.

In structural and earthquake engineering, experiments are often time consuming and preparing the required test setup is expensive. This situation restricts the variety and the number of experiments to be conducted. Therefore, every experiment is important, and sharing this information is crucial. However, sometimes, due to lack of a common, structured storing and sharing platform, experimental data might get lost or become inscrutable after they have been used in intended studies.

Considering these problems, it is clearly seen that structural and earthquake engineers need a medium which should prevent re-conducting of identical, time consuming and costly experiments by providing an information sharing environment that is not restricted by geographical borders in order to make it

possible for contributors in different regions to share and to improve the pool of experimental data. Also, this medium should provide storing of information in a standardized, structured manner to facilitate finding the required experimental data and improve comprehensibility of the information related to experiments.

Today, having such a medium is made possible with the improvements in the database systems and the Internet. In the presented study, a web-based database application for experimental structural engineering researches was developed as an information storing and sharing platform for Structural and Earthquake Engineering Laboratory of Civil Engineering Department of METU. Here, the database provide a standardized, consistent and secure way of data storage, where the web interface provides an easy way of managing and sharing the data via the Internet. After the development of the web application, the platform was tested by uploading the documentation and the test data of various experiments which were conducted in Structural and Earthquake Engineering Laboratory of Civil Engineering Department of METU.

1.2 Literature Survey

For many years, database systems are being used in different branches of civil engineering for storing experimental data. However, most of the databases are accessed via the Internet by using client-side software programs which can only be used with a specific operating system (OS). Therefore, the data cannot be accessed by everyone. Recently, with the rapid development of web technologies, database systems are started to be combined with World Wide Web in order to be accessible by more people without necessity of a software program. Thus, they become free of

operating system restrictions, which means any device that can browse websites is able to access the stored data.

In this section, several web-based experiment databases are presented. Most of which are developed in order to store experimental data on earthquake performances of various structures. The growing pool of data is later investigated statistically and helps the refinement of the formulations used for solving seismic problems. Some other presented web-based database studies were about determining best stormwater management practices and collecting in situ test result and their locations in a database. These geographic databases provide time efficiency and reduction in future project expenditures. There are also some real-time database management systems which are combined with wireless sensor networks in order to check sensor readings and the condition of the structure remotely via the Internet.

One of the biggest web-based experiment database for civil engineers is NEEScentral which is a national and networked project developed by Network for Earthquake Engineering Simulation (NEES) Consortium, Inc. for broadening the knowledge of experimental and theoretical earthquake engineering. This program was supported by National Science Foundation. NEEScentral is a web-based portal created for establishing communication between NEES community members and its database storage. It makes the contributors to be aware of new researches, new simulation models, and industry practices possible. NEEScentral has a fully user controlled very comprehensive database which enables re-conducting of wide variety of experiments. NEEScentral has tremendous amount of various experimental data and lots of different users managing the database. It stores files related with experiments and projects directly in file system where all other data is stored in the DBMS in order to keep things organized. [12]

Another web-based database that stores experiment results is the Pacific Earthquake Engineering Research Center Structural Performance Database (PEERspd) which was built on the previous database of National Institute of Standards and Technology (NIST) and its purpose is to help the studies on seismic performance models for reinforced concrete columns. This database is expanded in the University of Washington with the support of Earthquake Engineering Research Centers Program of the National Science Foundation. The preceding database was holding 107 tests of rectangular and 92 tests of spiral reinforced concrete columns. With the initialization of PEERspd, these quantities were increased to 274 rectangular, 160 spiral reinforced columns. Also a web based user interface was utilized in order to achieve an interactive database. In the original NIST database, digital top force-displacement history, key material properties and the test geometry description were available. Now, in addition to these WWW accessibility, P-Δ configuration, the maximum column deflection imposed before reaching various damage states, column reinforcing details, axial load information, key drawings and images (where available), comments (e.g., unusual characteristics), references and links for further information can be found.[13,14]

There is also a partnered research program called "Next Generation of Ground Motion Attenuation Models" (NGA) project which is an extension of PEER Strong Motion Database, done by Pacific Earthquake Engineering Research Center-Lifelines Program (PEER-LL), U.S. Geological Survey (USGS), and Southern California Earthquake Center (SCEC). This project includes 173 earthquakes, 1400 recording stations, and about 3500 multi-component recordings with the ground motion information such as source-to-site travel path, and local site conditions of the recording stations. All this information can be accessed via a web based graphical user interface.[15,16]

Kawashima Research Group in Civil Engineering Department of Tokyo Institute of Technology was developed an experimental database which also deals with earthquake loads. The aim of the research group is to economically minimize the damage taken from earthquakes while preserving practicality. Their main tests were on reinforced concrete bridge piers. Experimental data is divided into two categories and can be obtained from their web [17]. These two categories are "Cyclic Loading Test Data of Reinforced Concrete Bridge Piers" and "Cyclic and Hybrid Loading Test on Premature Shear Failure of Reinforced Concrete Bridge Piers with Termination of Main Reinforcements". In the former category, experiments conducted with different column section types and sizes in order to introduce the effects of some variables on other variables like "the effect of a longitudinal reinforcement diameter on a plastic hinge length" or their linear and nonlinear interactions over column performance like "effect of varying axial force on seismic performance". The test result data is presented in three file types: PDF, TXT, and XLS. In latter category, effects of different loading conditions on shear failure type of a specific bridge pier are given. All necessary data about that specific bridge pier is presented and its failure is experimented under unilateral cyclic, bilateral cyclic, static, and unilateral hybrid (pseudodynamic) load tests. As a result no numerical data is given; however, recorded videos and taken pictures during test process are presented.

In order to determine the seismic performance of steel structures under earthquakes, many cyclic loading and pseudo-dynamic tests are conducted on structural steel components in Japan where there is a high risk of seismic activity. In order to maintain and publish the experiment details and results, Numerical Database for Steel Structures (NDSS) has been developed at Nagoya University in Japan [18,19]. NDSS is a distributed collaborative

database system which can store and process both seismic (cyclic loading and pseudo-dynamic experiments) and ultimate strength experiments. The results can be alpha-numeric data, image or video. There exists a central metadata server and many collaborative databases in different servers, connected via WWW. This system has a function of running seismic numerical analyses in the central server and presenting test databases in remote servers. Multi-server database system used in this project which has various advantages over a single-server database system i.e. the work load over the network and server is lessened, server problems are reduced, each researcher is responsible from their own work and researches decide the range of data to be released and if any data change is necessary, data owners can update database by using their own servers [18].

Researchers of the Natural Hazards Modeling Laboratory, which is associated with the Department of Civil Engineering and Geological Sciences of University of Notre Dame and supported by the National Science Foundation, try to calculate loads caused on structures due to different natural hazards, estimate response of the structures and find a way to reduce hazardous effects. NatHaz facilities contain high-bay structural testing laboratory, dynamics and sensors laboratory and an Atmospheric Wind Tunnel [20]. The main research fields of the facilities are damping systems, dynamics of long-span bridges & tall buildings, full-scale monitoring of structures, wind tunnel modeling, dynamic load simulator, system identification, modeling, analysis and simulation of random processes like applications to wind, waves and earthquakes, modeling of transient events such as gust/wave fronts, wind and wave effects on offshore platforms, natural hazard risk analysis and management, codes & standards, IT and networked sensing devices [20]. NatHaz also holds an interactive web-based simulator, "On-line Wind Simulator (NOWS)" and an interactive database system, "Aerodynamic Loads

Database". NOWS provides users an online dynamic structural analysis under wind loads by simulating wind fields through computer network and gives the results as a text file [21]. On the other hand, Aerodynamic Loads Database provides guidance at the preliminary design stages of tall buildings by calculating dynamic wind loads on them. Web interface asks for the geometry, dimensions, and environmental condition of the building that will be simulated and then gives aerodynamic load spectra, wind force components acting on each floor of building model, survivability and serviceability design moments, displacements, and accelerations for the alongwind, acrosswind, and torsional directions of the building according to previously conducted wind tunnel tests over 27 different building models with different shapes and sizes under several wind flow conditions. [22,23]

Web-based databases are also widely used with Geographical Information Systems (GIS). One of them is Resolution Of Site Response Issues from the Northridge Earthquake (ROSRINE) Project. ROSRINE brings together a strongly coordinated group of geologists, geotechnical engineers and seismologists from a number of organizations to address geotechnical site characterization and ground motion response issues resulting from the Northridge earthquake. The objectives of this project are collection, compilation, and rapid dissemination of high-quality site geotechnical and geophysical data to the research community. And its focused analyses limited to determining the extent to which local site and regional wave propagation effects control strong ground motion, including nonlinear site response; evaluating the adequacy of the conventional approach to estimating site effects using measured shear-wave velocities, results of laboratory tests, and one-dimensional equivalent-linear and nonlinear analyses; assessing the degree of model complexity (2D and 3D) required to adequately explain site effects; and determining the uncertainties in measure properties and how

these uncertainties affect ground motions. To date, ROSRINE database has geological and geophysical characterization of approximately 50 strong motion station sites and dynamic laboratory testing of approximately 50 samples from these sites. The user interface of this database is simple and powered by google maps application. Interface allows database querying based on the site location, code geographical location and some other parameters. The resulting report includes all available information about the selected site, including downloadable boring log, geophysical data, results of lab tests and others. It also offers some meta-information about data, such as data references, comments about site investigation, a reference to the site investigator, etc. [24,25]

Another information system for geotechnical data and information exchange is the Consortium of Organizations for Strong Motion Observation Systems (COSMOS) Virtual Data Center (VDC). COSMOS VDC is a comprehensive, unrestricted, on-line, interactive strong ground motion search engine for engineers, seismologists, and other earthquake professionals. It provides easy and free access to strong motion earthquake data from a large and growing number of sources. The data are available for download as text files containing raw acceleration recordings, processed accelerations, velocity, displacement, and Fourier and response spectra. You may also download tables of metadata parameters, such as peak ground acceleration and s-wave velocity for all stations reporting a given earthquake. The VDC has various interfaces, which allow you to select data by earthquake, by recording station, via a map interface or by searching metadata parameters. One may also preview the acceleration and the response spectra before downloading the data files. VDC is restricted to earthquakes of magnitude 5.0 or larger in earthquake prone areas and 4.5 or larger in areas or low seismicity. The database has searchable metadata for hundreds of

earthquakes, thousands of stations and accelerograms and it continuously improving. [25-30]

The Urban Water Resources Research Council of the American Society of Civil Engineers (ASCE) and U.S. Environmental Protection Agency (EPA), together initiated International Stormwater Best Management Practice Database Project in order to gather sufficient techincal design and performance information for improving BMP selection and design, so that local stormwater problems can be cost-effectively addressed. In order to reach this goal a web-based database is developed and the data is open for public use from project website www.bmpdatabase.org. Currently database holds more than 340 BMPs. Application uses MS Access as a centralized DBMS and the data upload to website is done by using pre-formatted excel spreadsheets. Website also has a search engine, thus BMP data can be retrieved based on variety of search criteria. The retrieved data can be printed out or downloaded either as a MS Access file or a comma-delimeted text file. [31,32]

Geotechnical Group from the Civil Engineering Department of the Federal University of Pernambuco – Brazil (UFPE) has been developing geotechnical studies from laboratory and in situ tests in the Receife lowland soils. For this purpose, a geotechnical database was developed using tools of the information technology and presents all in situ tests results (SPT, CPTU, PMT, DMT and Field Vane) from two research sites. The main purposes of the database are: to allow a faster and efficient way for advanced research and statistical analysis of the geotechnical parameters obtained by in situ tests; to give basic information about the equipments used and about the in situ tests performed by the UFPE team; to contribute for spreading the use of others in situ tests in Brazil; to make easier and more efficient the

evaluation/calibration of empirical correlation from literature to be used in Receife soft clays; to be used as a pedagogical tool to help teaching in graduate and undergraduate geotechnical engineering courses. The database can be accessed by using specific software or from a website. [33]

There are also real-time databases beside conventional disk-based databases. National Institute of Standards and Technology (NIST) is currently exploring novel technologies for sensing in buildings to achieve several goals such as: to enable sensor technologies that will lead to energy savings and occupant comfort; to enhance the amount of available data to building control and fault detection systems; to provide new forensic tools for finding and diagnosing problems in buildings; and to promote the interoperability of data from various sources to improve the security of building occupants. One current focus area is on the emerging field of wireless sensor networks. Wireless technology has worked its way into the field of sensing, allowing for the quick deployment of sensors throughout a building that transmit data without the need for wiring. Building Environment Division of NIST developed an online wireless sensor network database in order to collect data from sensors installed on a building, and store it in a database, then, make it possible to monitor the building by using a WWW environment. [34,35]

1.3 Objectives and Scope

The main objective of this study is the development of a web-based database for Structural and Earthquake Engineering Laboratory of Civil Engineering Department in METU to store the results of experiments conducted in the laboratory, plus the information and documentation about

the related experiments, projects and specimens in one location, and to make sharing and management of the stored information possible.

As the scope of the project, the database system is constructed by using the MySQL Database Management System. In order to provide a user interface for the database system, the web application was developed by using the Microsoft's Visual Studio 2008 as the Integrated Development Environment (IDE). ASP.NET and C# were selected as the programming languages.

1.4 Outline

In Chapter 2, detailed information about databases and Database Management Systems (DBMSs) is given. The purpose of an Entity Relationship Diagram and how to read it is explained. Also, structure of a web application, which provides an interface for users to let them manage the data contained in databases remotely, is explained. Additionally, meanings of a Use Case Diagram and a Sequence Diagram, which are used for explaining the website functionalities, are briefly discussed.

In Chapter 3, data abstraction process of the database entities is discussed. Also, all the entities, their attributes and relations between those entities are explained in detail and presented as ER diagrams.

In Chapter 4, the platforms and programming languages used for application development are described together with the development stages of the website. Also, the tasks that the website users can perform are listed as a Use Case Diagram, and the execution processes of those tasks are explained briefly with given Sequence Diagrams.

In Chapter 5, a test scenario is carried out while introducing the user interface of the application with step by step screenshots.

Finally, in Chapter 6, achieved goals are noted, restrictions and limitations of the application is discussed and their solutions are presented. Also, some recommendations for further development of the application are given.

CHAPTER 2

BACKGROUND INFORMATION

2.1 Overview

In this chapter, detailed information about databases and Database Management Systems (DBMSs) is given. The purpose of an Entity Relationship Diagram (ERD) and how to read it is explained. Also, structure of a web application, which provides an interface for users to let them manage the data contained in databases remotely, is explained. Additionally, meanings of a Use Case Diagram and a Sequence Diagram, which are used for diagrammatizing website functionalities, are explained.

2.1.1 Databases and Database Management Systems

Databases and database technology are crucial for almost all areas where computers are used, including business, electronic commerce, engineering, medicine, law, education, and library science etc. A database can simply be defined as a collection of related data, where data means known facts that can be recorded and have implicit meaning. However, common definition of a database includes some restrictions like; a database represents some aspect of the real world where changes are reflected in the database. Also, a database should have been designed, built, and populated with data for a specific purpose and has an audience that is actively interested in the contents of the database [1].

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient [1,2]. By using DBMSs [1],

- Data can be stored in the database by specifying the data types, structures and constraints.
- Data can be updated and deleted to reflect the changes.
- Database can be accessed by multiple users or programs at the same time.
- Protection of data, from hardware or software malfunction (or crashes) and against unauthorized or malicious access is provided.
- Security and maintenance of data is ensured during the life cycle of database.

Typical commercial DBMSs include Oracle, Sybase Adaptive Server Enterprise, Microsoft SQL Server, where PostgreSQL and MySQL are the commonly used free DBMS solutions. These DBMSs can communicate with various desktop and web applications over a network such as the Internet or an intranet. MySQL is preferred as DBMS software for the development of presented thesis project because of its free license, and the communication between the database and the users is achieved through websites which can be displayed by using web browsers.

DBMS software and the database are together called a database system. Components of a database system can be seen in Figure 2.1. In a database system, users or programs communicate with the DBMS via other application programs or by sending queries directly to the DBMS. The queries are written by using several languages such as Data Description Language (DDL)

and Data Manipulation Language (DML). This language types will be explained later in this section. After receiving the query, DBMS software processes the query and then reflects the changes to the database which commonly resides on a hard disk drive. Two types of data is stored in a database. One stores the metadata which contains the definitions about the database structure and the data, the other one stores the data itself.

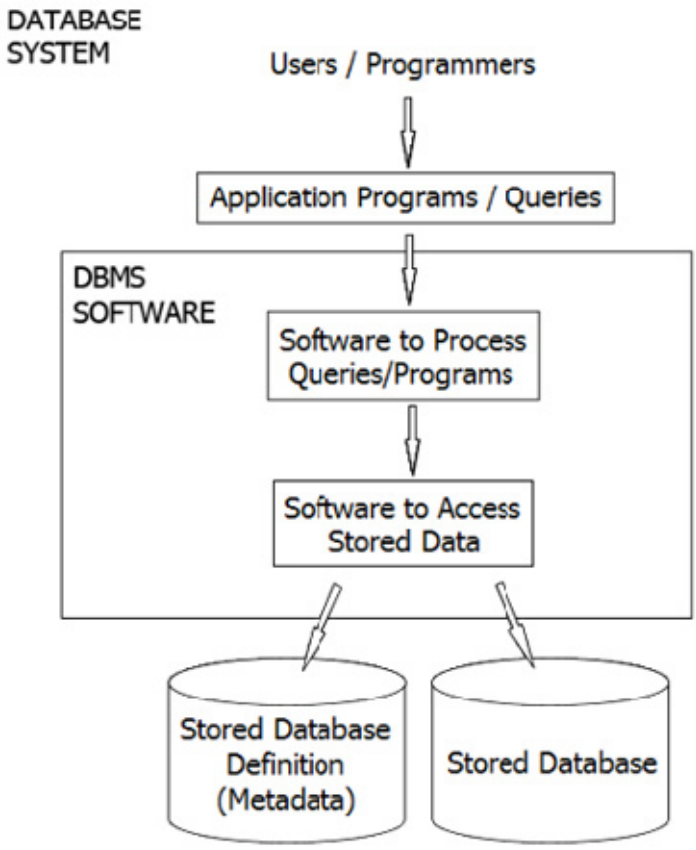


Figure 2.1, A simplified database system environment [1]

A number of characteristics distinguish the database systems from typical file-processing systems. Typical file-processing system is supported by a conventional operating system. The system stores permanent records in

various files, and it needs different application programs to extract records from, and add records to, the appropriate files. Before DBMSs came along, organizations usually stored information in such systems. However, keeping organizational information in such a system has major difficulties in assuring data redundancy, consistency, integrity, security and administration. Also, isolating data and either direct or concurrent accessing it cause serious problems. These difficulties, among others, prompted the development of database systems. [2]

A fundamental characteristic of the database approach is that the database system contains not only the database itself but also a complete definition or description of the database structure and constraints. This definition is stored in the DBMS catalog, which contains information such as the structure of each file, the type and storage format of each data item, and various constraints on the data. The information stored in the catalog is called *metadata*, and it describes the structure of the primary database [1]. Other characteristic properties of DBMS are providing facilities for storing, retrieving and updating data, supporting multiple view of data, providing facilities for specifying integrity constraints, provide facilities for controlling access to data, allowing simultaneous access and update by multiple users, supporting transactions, providing facilities for database recovery, and providing facilities for database maintenance [3].

In order to understand the structure of a database, some fundamental concepts such as entities, attributes and relationships are needed to be known. An *entity* is a "thing" or "object" in the real world that is distinguishable from other objects and shown as tables in a database. A project, a document, members of the website and their groups can be given as examples of entities. *Attributes* are the properties of an entity and entities

are described in a database by a set of attributes. the name of a project and its starting date are the examples for attributes of project entity; name of a file and its size can be an example for attributes of a document entity; username and password can be given as an example for attributes of a member. A *relationship* is an association between two or more entities. Entities are related with each other on their attributes. A relation between documents and projects can be ensured over the "related project ID" attribute of a document where groups and their members are associated by matching their IDs in a separate group membership table.[1,2]

When designing a database, these concepts gain a physical meaning. Entities are represented as tables in a database and attributes become the columns of tables. The data or the information which to be stored is entered into the database as rows of these entity tables. Each row is called as a *tuple* and each tuple in a table must be unique. This means that no two tuples can have the same combination of values for all their attributes; thus, redundancy is prevented in a database. However, this does not mean all the attributes have to be distinct. Mostly, one or two special attribute is enough to uniquely identify a data such as the social security number which can directly identify a citizen by itself. The set of these attributes which make a tuple unique is called a *primary key (PK)* and these attributes cannot be left empty when entering a data into a database. Relations between entities, thus tables, are provided over primary keys. The attributes which point to the primary key of another table is called a *foreign key (FK)*. Referential integrity of a database is ensured by primary and foreign keys.

A Database Management System (DBMS) is an intermediate link between the physical database, computer, operating system, and the users. In order to carry out many different tasks, a DBMS provides specialized programming

languages called database languages. Data Description Language (DDL) and Data Manipulation Language (DML) can be given as main database languages. DDL is used for defining various types of data in the database and their relationships with each other. The basic functions performed by DDL are creating tables, files, databases and data dictionaries, specifying the storage structure of each table on disk, assigning integrity constraints on various tables, setting security and authorization information of each table, specifying the structure of each table and maintaining the overall design of the database. On the other hand, DML enables users to access and manipulate data by providing users the ability to retrieve, insert, update and delete data.

DBMSs can be grouped under 3 different categories according to their data storage and management model, how many computer does the database is distributed and their data storage platform.

Depending on its data storage and management model, DBMSs can be classified under four main structural models [4,5]:

- *Hierarchical Model* where data relationship is maintained in a tree-like structure (parent-child). Therefore entities are mapped using one-to-many (1:N) relationships.
- *Network Model* is an extended Hierarchical Model. If this one-to-many relationship of Hierarchical Model is violated by assigning many-to-many (M:N) relationship, then this model is called as a Network Model.
- *Relational Model* is the most popular DBMS where data is structured within "Tables" and relationships constructed between data entries

and those tables. Database management is controlled by Structured Query Language (SQL).

- *Object-Oriented Data Model* is the next generation of DBMSs. Storing and retrieving data is handled not only by individual data entries but also using objects which are encapsulated data and operations.

Depending on how many computers does the database run on, DBMSs are grouped into two [6]:

- *Centralized DBMSs* run on a single computer. All the data storage and database management can only be handled by using this computer.
- *Distributed DBMSs* run on different computers connected together via some communication network, in which each computer has its own database and management system; however since each computer agreed to work together in a common network, one can access the entire database by using a single computer.

Depending on data storage platform and whether the transactions have timing constraints or not, DBMSs are categorized as conventional disk-based databases and real-time databases.

Modern databases consume large storage area. Therefore, ferromagnetic disks are very suitable platforms for information storage due to their low cost per capacity ratios. Although most of today's databases are disk resident and satisfy user needs with their easy to use management systems and user-friendly interfaces, disk's mechanical nature limits their speed. Even though disk-based databases are fast enough to handle most of the applications which involve human users, their speed is not sufficient for process controlled real-time applications. Access time of disk-based databases can be

improved by caching data in Random Access Memory (RAM). But doing so is still not enough especially for databases whose transactions require intensive write operations [6-9].

RAM-based storages have much higher speed in individual read and write operations compared to that of disk-based storages. But RAM-based storages are as expensive as they are faster. Therefore, the applications of RAM based databases have to be limited to systems where only a small amount of data needs to be stored; for example hand-held personal organizers or high-value high-performance applications where it is economically feasible, both from a space and a cost point of view, to use large banks of RAMs [7]. For those special databases where high performance is necessary, it is a must to store and manage entire data in RAM. In order to do so, real-time data management systems are designed.

The real-time database systems need to [9]:

- respond to very large amounts of simple queries (almost always through a primary or unique secondary key);
- manage simple data with frequent updates;
- be highly available, with downtime of less than half a minute per year; and
- have very short response times, of only a few milliseconds.

In a real-time database system, timing constraints are associated with transactions, and data are valid for specific time intervals. The transaction timing constraints can be completion deadlines, start times, periodic invocations, and so on. It is not necessary that every transaction have a timing constraint, only that some do. In addition to transaction timing

requirements, data has expiration duration as well. This means that the recorded values are valid only for a certain time interval. A real-time database makes this validity interval explicit as part of its database schema. We can define transaction correctness as a transaction meeting its timing constraints and using data that is absolutely and relatively timing-consistent. Absolute time consistency means that individual data items used by a transaction are still temporally valid and reflect the true state of the world to an acceptable degree of accuracy. Relative time consistency means that multiple data items used by a transaction are updated (sensed) within a specified time interval of each other. For example, if a transaction uses temperature and strain data to make a decision regarding a stress calculation process, these two data values must correlate closely in time or the computation will likely make no sense. [10]

In real-time database management systems, RAM-based storage has several negative effects even though its boosts the performance. Reliability is distorted and system crashes may cause fatal errors. One costly preposition for increasing the stability is synchronous replication in separate memory segments. The cost increase is extended by designing the system adoptable to different applications, or transaction patterns. Therefore, for the real time database management requirements, a parallel main memory database system with synchronous replication of data distributed over a number of network nodes is considered as an adequate solution. A "shared-nothing cluster", which is a parallel system of nodes where every node has its own operating system, disk and RAM, all connected by high-speed interconnect, distributes computing on several processors and it automatically handles partial system failures. At the instants of failure, the faulty processor is identified and let out. For a qualified and uninterrupted service, this erroneous processor(s) are identified and put into stand-by mode in about 50

milliseconds. This system can also be used to support online rolling upgrades. With 30 seconds of down-time in a year, which is 99.9999% uptime, it can succeed in occasions where high availability is a must to have.[9]

After investigating all these possibilities, it is decided to build the presented study on a disk-based centralized relational database management system for initial development. The relational model selected because it is the most widely used type today and it is easier to construct and maintain. Since the database is used initially only in METU, having a distributed database is not necessary. Also, currently there is no real-time readings from sensor networks. Therefore, data is stored in ferromagnetic disks rather than RAM. In the future, presented DBMS can be extended according to necessities and may become a real-time distributed relational database.

2.1.2 Entity-Relationship Diagrams

Representation of a real world example as a schematic notation in terms of entities, attributes and relations between them excluding the data itself is called an Entity Relationship Diagram (ERD). An ERD is a data modeling technique which can be considered as a blue-print of a database system. By looking at an ERD, all the entities, their attributes and the relations between those entities can be seen.

Several ERD notations are available today, such as Chen's notation, IDEF1X, Bachman notation, min-max and UML notations and Crow's Foot notation. Main differences between these notations are the way they draw the cardinality and modality of a relation. Cardinality refers to the maximum number of times an instance in one entity can be associated with instances in the related entity, where modality refers to the minimum of that.

Cardinality either be 1 or *Many* and the symbol is placed on the outside ends of a relationship line, closest to the entity. On the other hand, modality can be 1 or 0 and its symbol is placed on the inside, next to the cardinality symbol as shown in Figure 2.2. In this thesis, Crow's Foot notation is used for drawing the ERD of the database, and in Crow's Foot notation, cardinality and modality of 1 is drawn as a straight line, where *Many* is drawn as a three toes foot and 0 is represented by a circle (Figure 2.2). Cardinality and modality are indicated at both ends of a relationship line and that relationship is read as being one to one (1:1), one to many (1:M), or many to many (M:M) according to the indicators. [36]

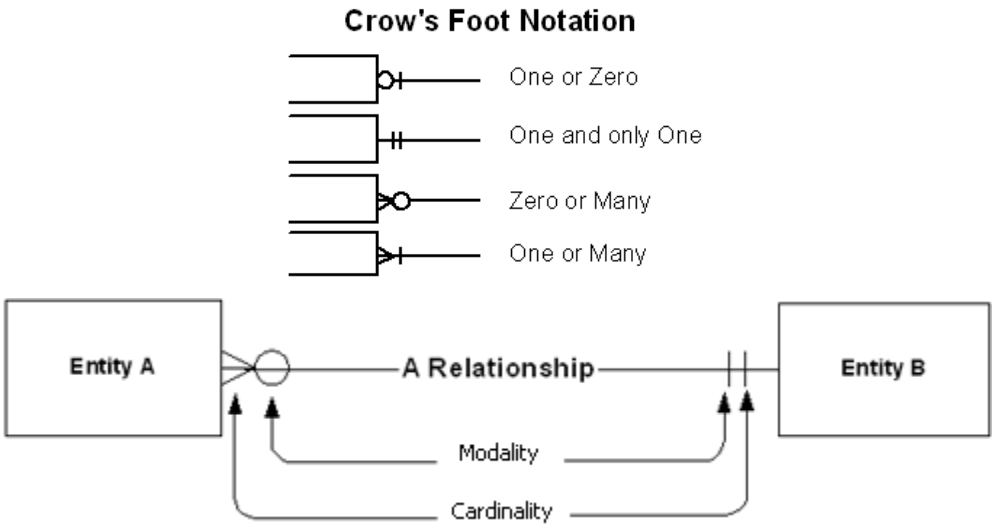


Figure 2.2, Representations of cardinality and modality in terms of crow's foot notation in an entity relationship diagram [36]

Other than cardinality and modality relations between entities, two other relation type are shown in the presented ER diagrams. These are *Identifying*

Relationship (shown as a straight line in ERD) and *non-Identifying Relationship* (shown as a dashed line in ERD). An identifying relationship means that the child table cannot be uniquely identified without the parent. This relationship is commonly seen in many to many relationships. In order to reduce data repetition, a new intersection table is created containing the primary key attributes of entities that are in many to many relation. The data entered in the intersection table cannot exist without having been defined in the parent tables. Therefore many to many relationships are identifying relationships. On the other hand, if the data in both parent table and child table can be identified independently without existing in the other table, this type of relation is called a non-identifying relationship.

The ER Diagrams presented in this study display entities as blue boxes and their names are written at the top section of those boxes. The yellow boxes, on the other hand, are connection tables which are used for showing many to many relationships between entities and are preventing unnecessary data repetition. Entities are represented as tables in the database together with connection tables. The columns of a table represent the attributes of an entity or a connection table. In the presented ERDs, primary key and foreign key attributes are listed inside the boxes below each table name. An attribute does also have several properties which show whether the attribute is a primary key or foreign key; the value of attribute can be left empty or not; the value of the attribute is determined by the database or defined by the user. Also, the value types of all attributes are shown in the ERD.

The yellow key icon in front of an attribute name shows that attribute is a primary key for that entity. Although only one attribute is enough to uniquely identify a tuple for entities, for two connection tables two attributes form a primary key together and for one connection table four attributes are

necessary to form a primary key. Foreign key attributes are marked with red diamond symbol. They provide referential integrity of data in a database. The links between entities are provided by relating foreign keys of a table with a primary key of another table. These relations between tables are shown by using the Crow's Foot notation.

Other than primary key and foreign key attributes, some attributes marked with blue and white diamond symbols. The blue and white diamond symbols shows whether the attribute value can be left empty or not when entering data into database. The attributes with blue diamond symbol are essential attributes and their values cannot be null where the attributes with white diamond symbol can be left empty when entering the data and their values can be filled later in time.

At the right side of an attribute name, the value type of that attribute is defined. In the presented database system, nine different value types are used. These are CHAR(n), VARCHAR(n), LONGTEXT, INT(n), TINYINT(1), DOUBLE, DATETIME, TIMESTAMP and LONGBLOB. Here, *char*, *varchar* and *longtext* means that the attribute is text based and the number in parenthesis is the character count limitation. *Int* and *double* states that the attribute is numerical and specifies whether the number has decimal part or not. *Tinyint(1)* is used in order to show the attribute is a Boolean value. Depending on the value is true or false, 1 or 0 is stored in the database, respectively. *Datetime* and *timestamp* value types are used for storing time in the database. Lastly, *longblob* is defined in order to store binary data in the database like profile picture of a user. This field is currently not used by the application; however, it may be used in the future when it becomes necessary. Some attributes have UN, NN and AI written on the right side of their names. UN refers to 'unsigned' and used for integer value types. The

capacity of an unsigned attribute is doubled but it is not possible to enter negative values for this attribute. NN refers to 'not-null', which means the field cannot be left empty when entering data into database. AI attribute refers to 'auto increment', and shows that the specified attribute value is increasing automatically on each data entry. No value can be assigned to this field manually. Since a new and unique value is entered automatically for auto incremented fields with each data entry, the auto incremented attributes are the perfect candidates for being a primary key.

2.1.3 World Wide Web and Web Applications

A website is consisted of a collection of interlinked, text-based documents called web pages and its content can be viewed with web browsers. Web pages are written using Hypertext Markup Language (HTML) which provides a means to create structured documents by denoting formatting information for text, links to other web pages, and information about embedded images, audio and video objects. It can also contain programming codes called scripts which are written in different programming languages such as javascript, visual basic and C#. All this different type of information is ensured by standardized HTML elements consisting of "tags" which are special text phrases surrounded by angle brackets within the web page content. All websites that are publicly accessible via the Internet constitute the World Wide Web. The World Wide Web is abbreviated as WWW, W3 and commonly known as The Web.

In WWW, web pages and the objects embedded to their contents are usually addressed with human-readable labels called Uniform Resource Locator (URL). Therefore, a web page or a multimedia file contained in a web page can be accessed by using its URL. Accessing and transferring HTML pages is handled through the Hypertext Transfer Protocol (HTTP). HTTP defines a

standard format for specifying the request of resources on the Web and what actions web servers and browsers should take in response to various commands.

Since the creation of first web page in 1990, the Web has evolved from single servers that host simple and static HTML pages to a multi-domain platform, offering support not only for information delivery, but also for application execution. For a web application, the HTML markup defining the presentation of web pages is nothing but the surface of an application which the users interact with, while the actual application is running on a remote web server or, in some cases, on multiple distributed remote servers. [11]

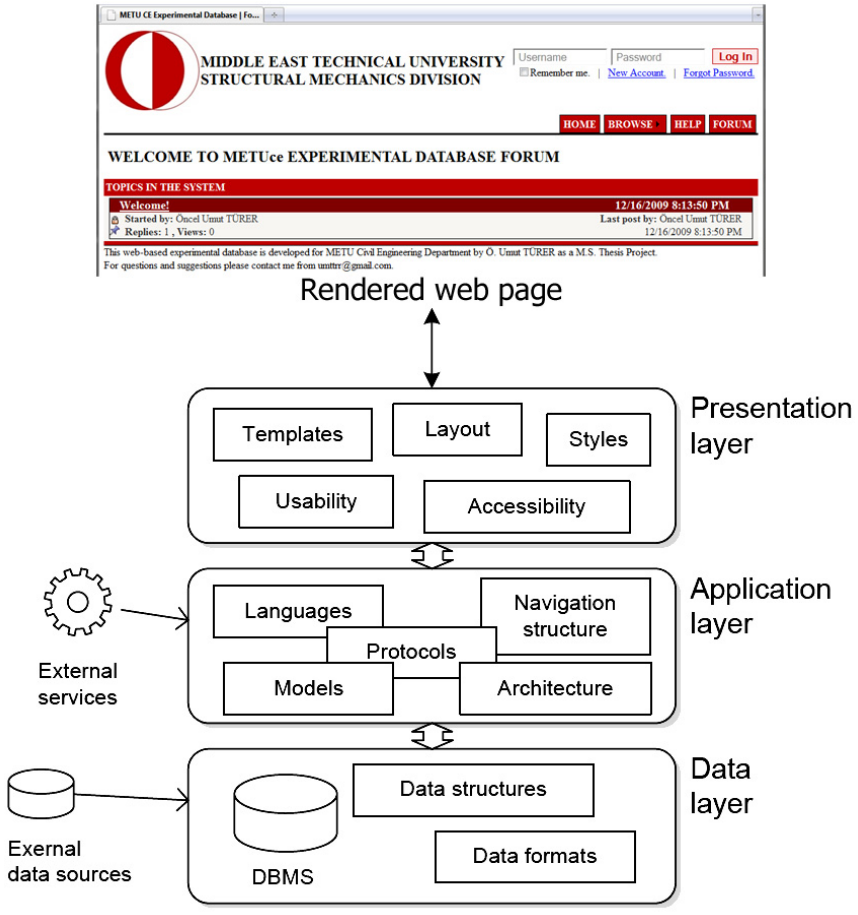


Figure 2.3, Web application development layers and their contents [11]

Commonly, web applications are divided into several development layers, typically a data layer, an application layer, and a presentation layer. These layers and their contents can be seen in Figure 2.3. At the data layer, the developer needs to decide where and how to store the data underlying the application under development, which data formats or database management systems to use, the structure of the database system, and whether to use external data sources as well. At the application layer, the developer decides on the programming and markup languages, models, protocols, and application architectures to be used. Also, the navigation structure between web pages of the application is defined in this layer. If there is a need for third party sources or remote services, their integration process is handled in this layer too. Finally, at the presentation layer, the developer focuses on the user interface of the application such as the layout of the application front ends, HTML templates and styles. All the decisions made should foster appeal, usability, and accessibility, for the end users to "like" the application and to be satisfied, without encountering any hurdles in using it. [11]

2.1.4 Use Case Diagrams

A Use Case Diagram is a type of Unified Modeling Language (UML) diagram which are used for modeling the functional interactions between actors and the system. The system is represented with a rectangle and the name of the system is written in that rectangle. The actors are drawn outside of the system. Every task that the system can perform is presented inside the system rectangle. Each specific task is shown as an oval shape and called a "use case". The names of each task is written inside these oval shapes. Sometimes, some tasks may invoke other tasks during its execution. This type of relation is shown with an "includes" keywords. The direction of the arrow points the invoked task. Also, execution process of some tasks may

include execution of other tasks. In this case, the relation between tasks is shown with an "extends" keyword. The direction of the arrow points the included (extended) task. An "actor" is shown with a stick person symbol, represents an external user, which may be a human user, a representative group of users, a certain role of a person in the organization, or anything external to the system. [1]

2.1.5 Sequence Diagrams

Sequence Diagram is a type of UML diagram which describes the interactions between various objects over time by showing the flow of messages between them. Within a sequence diagram objects are shown at the top of a dashed vertical line which is called the O*bject's Lifeline*. The lifeline represents the existence of object over time. Objects are shown in different shapes according to their types. In presented Sequence Diagrams, four different type of objects are used: *Actor* which represents the website user; *Boundary* representing a user interface such as the website or web pages; *Entity* represents an e-mail message or a file in the server; and *Database* represents the application database. Shapes used for different object types can be seen in Figure 2.4. [1]

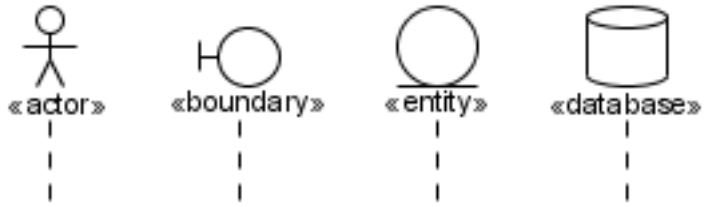


Figure 2.4, Shapes that are used in Sequence Diagrams for various objects

The rectangular boxes on a lifeline are the symbols of *Activation* which indicates that an object is performing an action. Messages between objects are represented as arrows between the lifelines of two objects. A *Message* bears a name and may have arguments and control information to explain the nature of the interaction. The order of message is read from top to bottom. Objects can also message to themselves. In given diagrams a self-message is shown as a 'C' shaped arrow inside an objects activation box. A condition that a message must satisfy is shown in square brackets before the message name. A dashed arrow line shows a return from a message and is optional unless it carries a special meaning. Object deletion is shown with a cross and ends an object's lifeline. [1]

CHAPTER 3

THE DATABASE

3.1 Overview

In this chapter, data abstraction process of the database entities is discussed. Also, all the entities, their attributes and relations between those entities are explained in detail and presented as ER diagrams.

3.2 Data Abstraction

The development process of the application was started with data abstraction. The entities are decided and relations between them were formed. The data abstraction model can be seen in Figure 3.1.

The "Experiment" entity forms the core of the model. Each experiment needs a "Signal Readings" entity to store the test readings and a "Signal Information" entity which provides detailed information about each signal; thus, makes the test readings more meaningful. In order to create an experiment, the test data related with the experiment must be stored in "Signal Readings" entity. Each time a test data inserted into the database, general information about signal readings is automatically inserted into "Signal Information" entity even though no information is specified about signals. Every experiment must be related to one entry of "Signal Information" entity and through that information, every experiment must be related to one set of signal readings data.

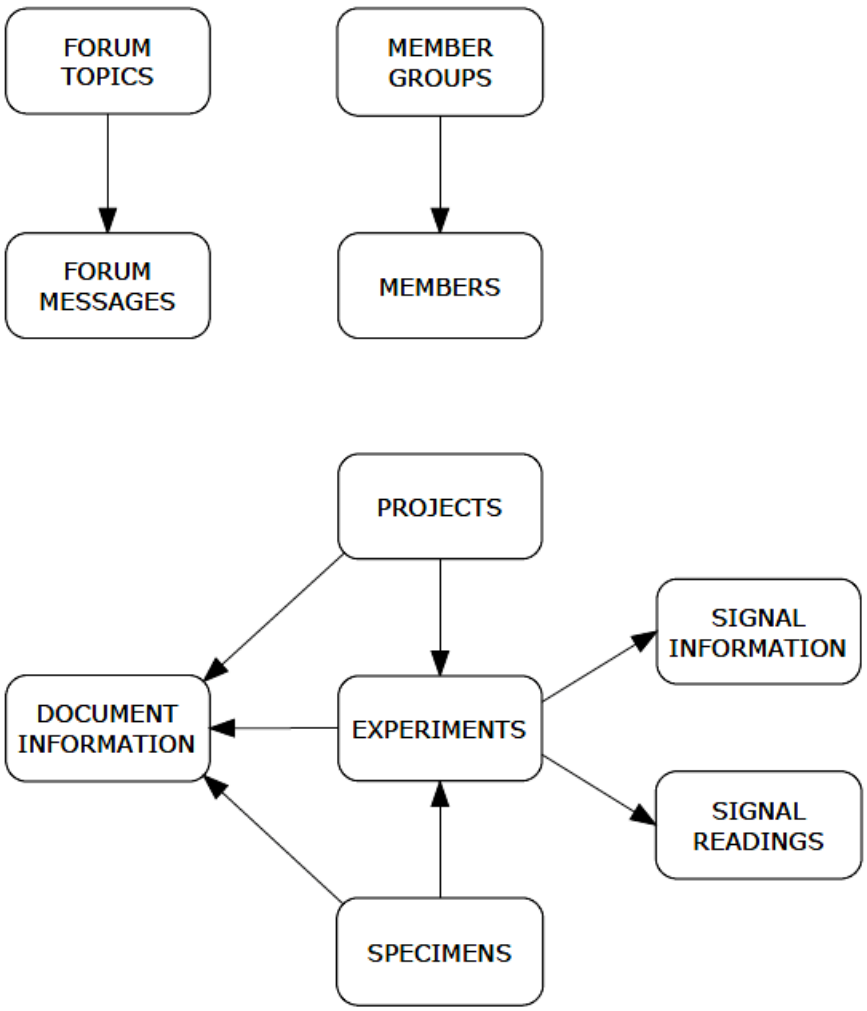


Figure 3.1, Data abstraction model and entity hierarchy

To group experiments that are conducted under same project, "Project" entity is included in the model. Also, in order to group experiments which are conducted on same specimens, "Specimen" entity was introduced. Creation of projects and specimens are not restricted to experiments, so they can be created independently without being related to an experiment. Therefore, projects and specimens can have zero or more experiments where an experiment can only be associated with one project and one specimen in order to be created. Experiments also serve as a connection between

projects and specimens. The specimens that are used in a project can be identified through the experiments of that project.

To enable file uploading to the system, a "Document Information" entity was introduced to the data abstraction model. The information about uploaded files are stored in this entity where the file itself stored outside of the database in order to enable other programs to access these files easily. Any multimedia document such as pictures, charts, audio and video files, PDF documents that are related with projects, specimens and experiments can be uploaded to the system by users.

The entities that are introduced to the data abstraction model up to now are for storing necessary information about experiments, projects, specimens and documentation in a structured manner. However, the development of the website brought new information that needs to be stored in the database about website users, data security and reliability.

Since various users will interact with the web interface of the database, a "Members" entity was included in the data abstraction to store information about website members. This information must include member roles which provide data reliability by allowing only specific, trusted members to create, modify and delete projects, specimens, experiments and documents. Four role types are determined. These are *User*, *HigherRankedUser*, *Moderator* and *Administrator*. With the inclusion of non-member website users which are called as "guests", five different type of website users are defined where guests are the least privileged users and the administrators are the most powerful one. The user types in the system can be seen in Figure 3.2 and the capabilities of these user types are explained in detail in Chapter 5.

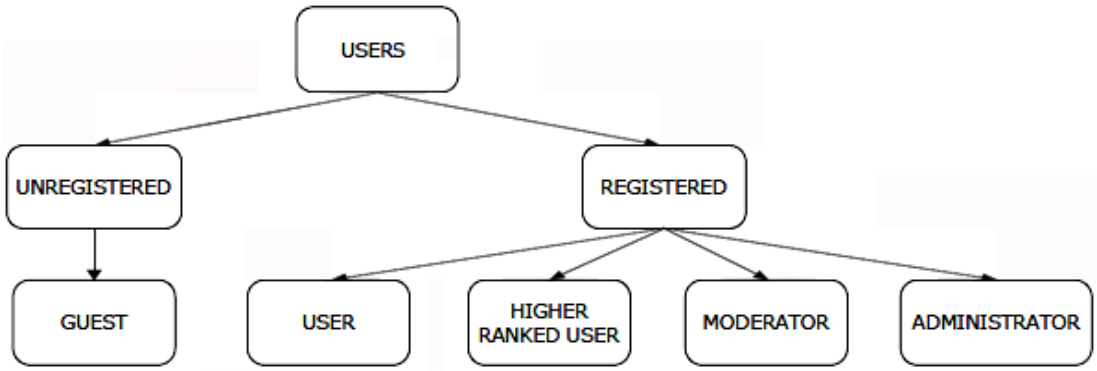


Figure 3.2, User types in the system

Aside from user roles, in order to group website member according to their common properties, "Member Groups" are added to the data abstraction model as a new entity.

In order to ensure data security; project, specimen and experiment creators need to select the members or member groups that can view, modify or delete the entered information. For this purpose, two concepts introduced to the system which are called "Ownership" and "Visibility". With the introduction of these concepts members or member groups can be assigned as an owner of a project, specimen or experiment. Also, confidential projects, specimens and experiment can be set visible to only selected members and member groups. If a member group selected as an owner or a viewer of a project, specimen or experiment, same ownership and visibility settings are applied to the members associated with that group also.

Finally, in order to provide a communication platform among website users, a forum is integrated into the data abstraction model which has two entities: "Topic" which stores information about the topics in the forum and "Messages" which stores messages posted under created topics.

3.3 The Database System

After the data abstraction process, the database system was developed. Entities, their attributes and the relations between entities are formed according to the data abstraction model. The database system of the presented study's is shown in Figure 3.3 as an ER diagram. The presented ER diagram is simplified by removing attributes other than primary and foreign keys in order to reduce the crowdedness. The complete ERD can be found in Appendix B.

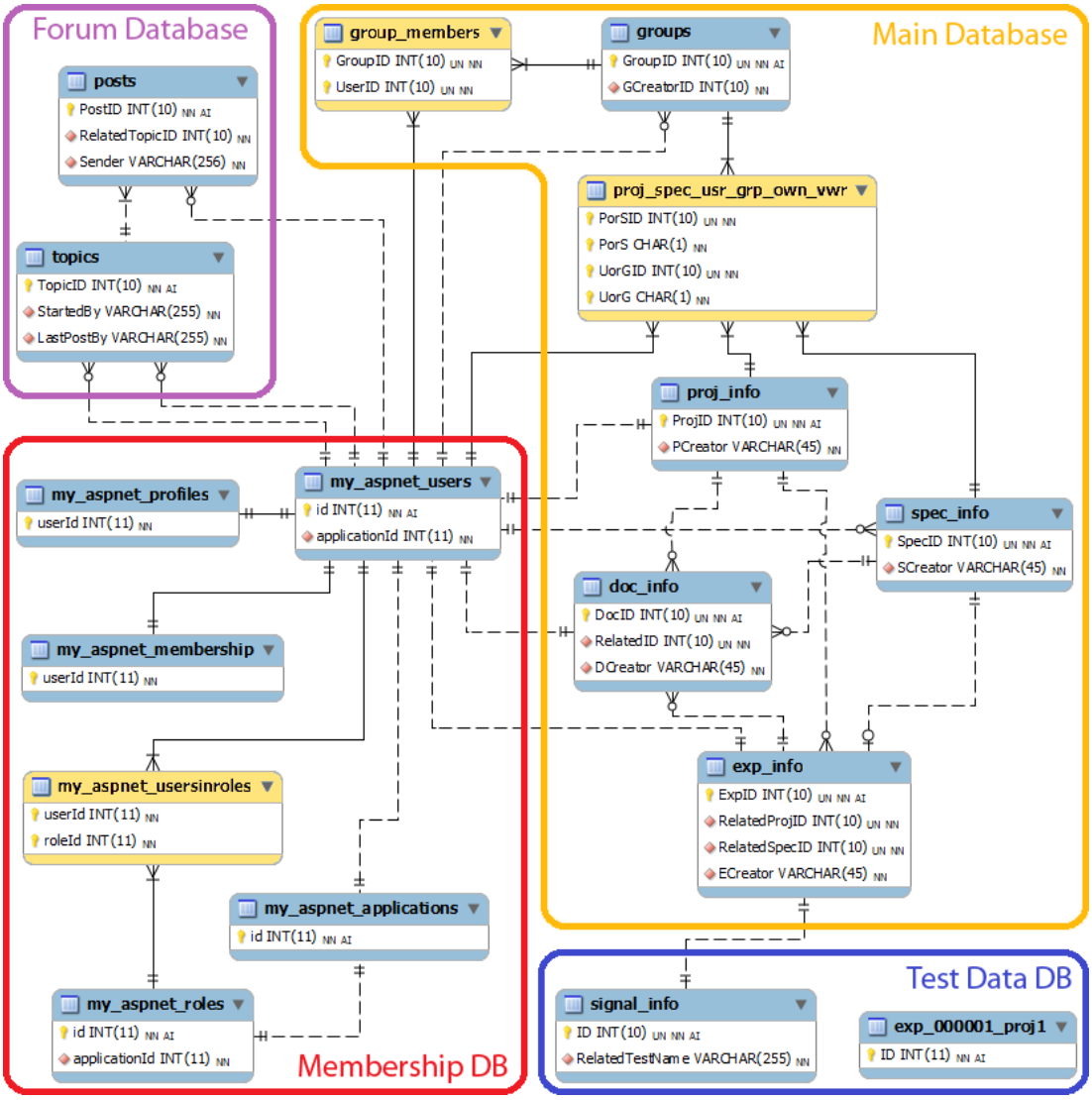


Figure 3.3, Simplified ER diagram of the database system

As it can be seen from Figure 3.3, database system of the project is composed of 4 individual databases: one for handling membership related issues (Membership DB), one for storing information about integrated forum functionality (Forum DB), one for storing experimental data and signal information for each experiment (Test Data DB), and lastly the main database to store information about projects, experiments, specimens, documentation, member groups and ownership/visibility settings (Main DB). The reason behind using multiple databases is mostly to provide mobility for the future needs. When required, all those databases can be distributed to several servers to increase the application performance. Also, baking up and maintaining separate databases of different functionalities is easier.

In the following sub-sections, all four databases (membership DB, main DB, test data DB and forum DB) existing in the project database system will be explained in detail.

3.3.1 Membership Database

Among the four databases, membership database is the only database which is created and managed automatically by MySQL's membership and role provider. The database consists of five entities and one connection table as shown in Figure 3.4. The entities in the database are 'my_aspnet_users', 'my_aspnet_profiles', 'my_aspnet_membership', 'my_aspnet_roles' and 'my_aspnet_applications' where the name of the connection table is 'my_aspnet_usersinroles'. The name of entities, connection table and attributes are automatically given by the MySQL.

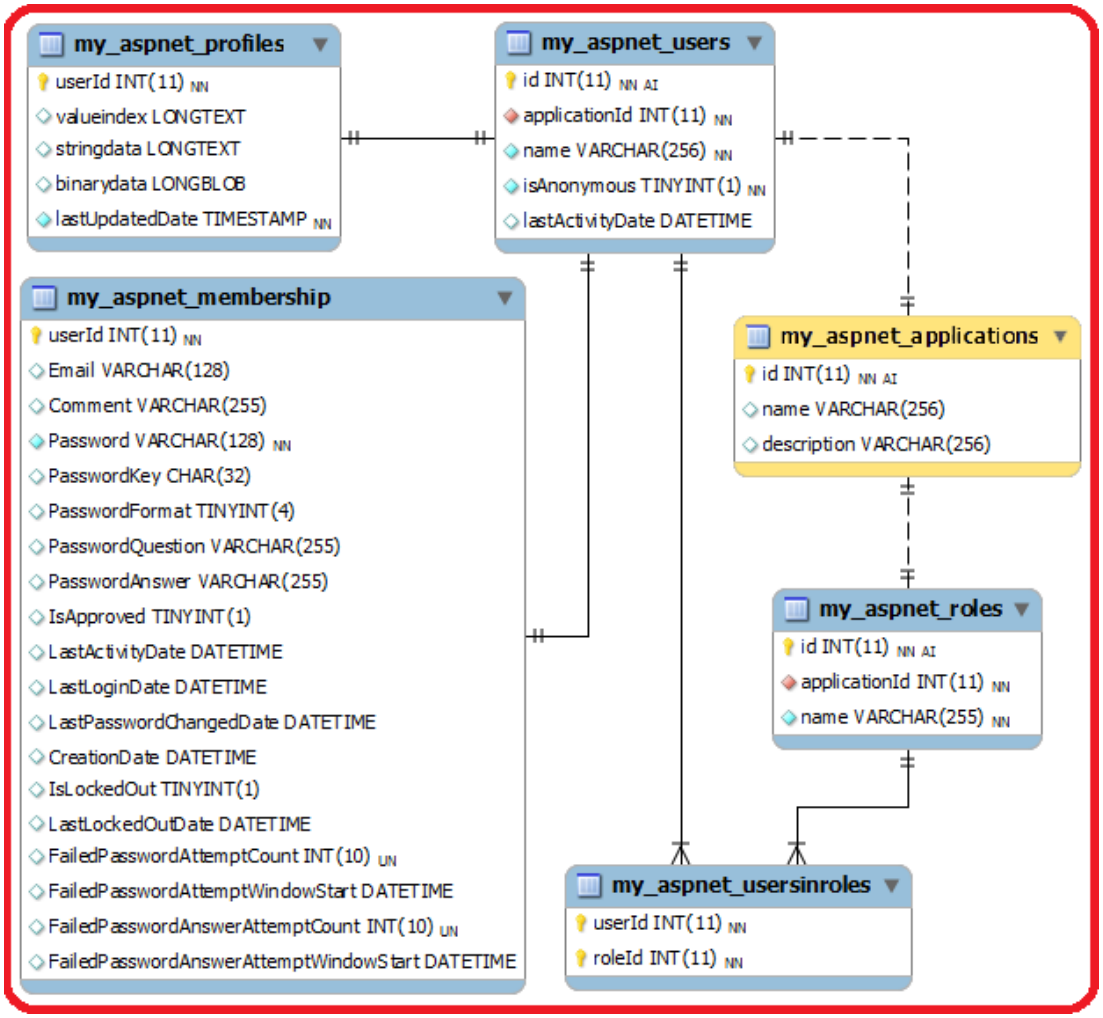


Figure 3.4, ER Diagram of Membership Database

The membership database is the place where all the information about website members is stored such as user profiles and roles, thus provides security against unauthorized users.

Functions of each database table and their attributes are explained below:

my_aspnet_users: This entity gives and stores the unique identifier for each newly created user with its auto incrementing *id* attribute. This *id* attribute is

also the primary key attribute of the entity. 'my_aspnet_users' entity also stores the application id in which the member is registered in its *applicationId* attribute; the user name of the member (not the real name) in its *name* attribute; the anonymity of the user (whether the user is an unknown guest or a registered one) in its *isAnonymous* attribute, and the last activity date of the user in its *lastActivityDate* attribute. This entity is the entity with the most number of relationships.

my_aspnet_profiles: This is the entity where the profile information about users are stored. Currently 6 profile information is collected from users. These are name, surname, gender and birth year of the user, institute that the user is related and the position of user in that institute. Names of the profile properties are stored in the *valueindex* attribute of the entity and the entries are stored in the *stringdata* attribute. The *binarydata* attribute is for storing member pictures but this field is not being used currently. The lastUpdateDate attribute stores the time when the user updated his profile information. The primary key attribute of this entity is *userId* and it is referencing the *id* attribute of the 'my_aspnet_users' entity.

my_aspnet_membership: Security of the website against unauthorized users is ensured by this entity. It stores the user's password in its *password* attribute for confirming the log in information of a user and authorizes him. Moreover, this entity counts the wrong password entry attempts in its *FailedPasswordAttemptCount* attribute for each user and locks the user for logging in if that count reaches to a predefined amount by changing user's *IsLockedOut* attribute to true. By doing so, account theft is being prevented. User e-mail address is also stored in this entity. Each time a new user is created, an e-mail confirmation is sent to the user. After confirming his e-mail, user is considered as approved and from that moment on, user can

successfully log in to the website. This approval status of the user is stored in the *IsApproved* attribute. Also, if a user forgets his password, user is asked to answer his predefined security question and a new password is e-mailed to the user. The security question and its answer is stored in the *PasswordQuestion* and *PasswordAnswer* attributes respectively. Another user property stored in this entity is the registration date of user and his last login date. These dates are stored in the *CreationDate* and *LastLoginDate* attributes of the entity. The primary key of this entity is the *userId* attribute which is referencing to the *id* attribute of my_aspnet_users entity.

my_aspnet_roles: The role names for each application is stored in this entity. The current web application uses 4 roles for users. These are *User*, *HigherRankedUser*, *Moderator* and *Admin*. *User* role has the least amount of privileges and every new user is instantly given a *User* role type. Most privileged role is the *Admin* role and only users having *Admin* role can change role types of website members. The differences between these roles is explained in Section 2.3. Each time a new role is created, a unique identifying integer value is assigned to that role and this value is stored in the automatically incrementing *id* attribute of entity. This attribute is the primary key attribute of 'my_aspnet_roles'. Name of the role is stored in the *name* attribute and the id number of the application related with the role is stored in the *applicationId* attribute of the entity.

my_aspnet_usersinroles: This is not an entity rather a connection table used for building links between users and their roles. This kind of table is created in order to reduce the data repetition in many to many relationships. For retrieving roles of a user or retrieving the users assigned to a specific role, this table is used. Both of its attributes, *userId* and *roleId* together, form the primary key of the table.

my aspnet applications: This entity is used for storing information about applications in the server. In the server of presented web application, the only application running is this thesis project and it is called as *METUce_ExpDB*. This entity has an auto incrementing *id* attribute which gives and stores the identifier integer value for each application created in the server. This *id* attribute is therefore the primary key attribute. There are two other attributes for this entity called *name* and *description* which store the name and the description of an attribute respectively.

3.3.2 Main Database

Main Database is the heart of the application. All the information about projects, experiments (excluding the test data), specimens, documents and the relations between them are stored in main database. Also, the information about groups which is used to classify users according to their common properties and their members is stored in this database. The database consists of five entities and two connection tables as shown in Figure 3.5. The entities contained in the database are 'proj_info', 'exp_info', 'spec_info', 'doc_info' and 'groups' where the names of the two connection tables are 'group_membes' and 'proj_spec_usr_grp_own_vwr'.

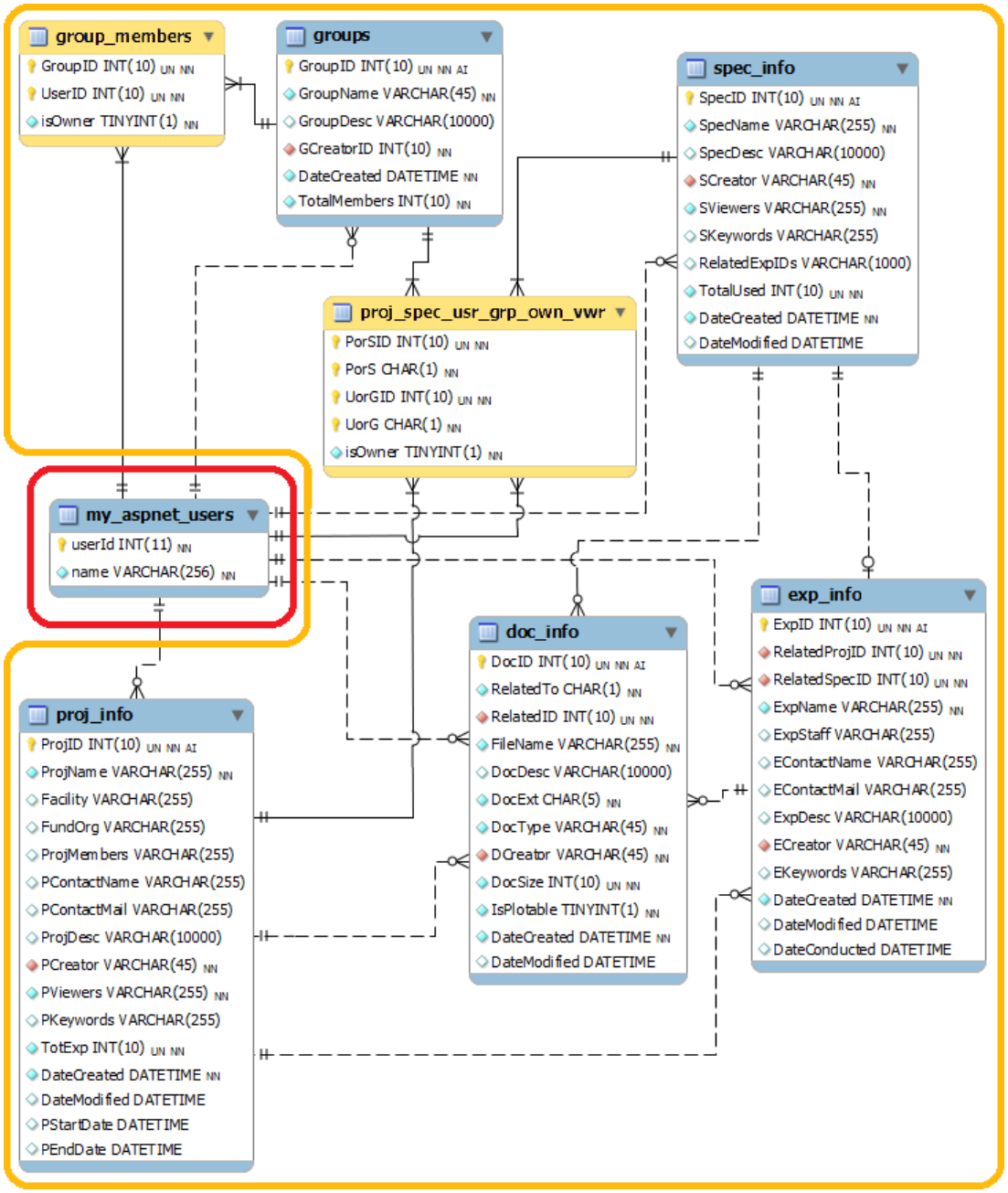


Figure 3.5, ER Diagram of Main Database

Functions of each table and their attributes are explained below:

proj_info: This is the entity where all the information about a created project is stored. This information contains project identification number (*ProjID*),

name of the project (*ProjName*), facility that the project belongs (*Facility*), funding organization of the project (*FundOrg*), project members (*ProjMembers*), name and e-mail address of the person who will be get in touch when necessary (*PContactName* and *PContactMail*), project description (*ProjDesc*), user name of the project creator (*PCreator*), visibility restriction level of the project (*PViewers*), keywords (*PKeywords*), total experiments belongs to the project (*TotExp*), the dates that the project is created and modified (*DateCreated* and *DateModified*), project start date and the deadline of the project (*PStartDate* and *PEndDate*). *ProjID* is the auto incrementing primary key attribute for this entity and the *PCreator* attribute is a foreign key which is referencing to the *name* attribute of 'my_aspnet_users' entity. Project creator and the project owners, which are indicated in the 'proj_spec_usr_grp_own_vwr' table, has privileges to edit the project information, to add new documents to the project or delete existing documents contained in the project. They even can delete the project from the database with all of its experiments and documentation. On the other hand, project viewers, specified by the *PViewers* attribute, has ability to see the project information, its experiments and the documents contained by both project and related experiments. But they cannot edit, insert or delete any of the information. The project visibility has a four level restriction and the level is determined by the project creator or owners. The base level restriction achieved by setting *PViewers* attribute as "Everyone". By doing so, it is permitted for everyone, guests and all site members, to see the project information, its experiments and documents. If *PViewers* attribute is set to "Site Members", then guest users, who are not a member of the website, are restricted to access project and related experiments. In order to achieve third level restriction, *PViewers* must be set as "Project Owners". This permits only the project owners and creator to see the project information and its experiments. The last level of restriction is the "Custom Selection".

Project creator and owners can one by one select the users or user groups which are intended to see the content. Every other user or user group is restricted to see the project information, its documents and its experiments.

spec info: This entity is used for storing all the information about test specimens. Each experiment must have its specimen information entered into the database. However, different experiments can use the same specimen for the test. Each new specimen is given a new identification number and it is stored in the *SpecID* attribute. This attribute is the primary key of the entity. 'spec_info' entity can store various information about a specimen such as name of the specimen (*SpecName*), description about the specimen (*SpecDesc*), user name of the specimen creator (*SCreator*), visibility restriction level of the specimen (*SViewers*), keywords (*SKeywords*), id numbers of the related experiments (*RelatedExpIDs*), how many times the specimen used in experiments (*TotalUsed*), the dates that the specimen information is created and modified (*DateCreated* and *DateModified*). As it is in the 'proj_info' entity, specimen creator is referencing the *name* attribute of the 'my_aspnet_users' entity and with specimen owners, they have the ability to edit the specimen information. They can add new documents or delete existing documents which are related to the specimen. Contrary to project owners and creator, specimen owners and creator have no power over related experiments. Ownership information of a specimen is also stored in the 'proj_spec_usr_grp_own_vwr' table. Similar to the 'proj_info', specimen visibility has four level of restriction: "Everyone", "Site Members", "Specimen Owners" and "Custom Selection". But, unlike 'proj_info', restricting visibility does not affect visibility of the related experiments. All the id numbers of related experiments stored in the *RelatedExpIDs* attribute as a text and each id is separated from one another with a comma.

exp info: This entity stores general information about experiments. However, no test data is stored in this entity. Test data for each experiment is stored in individual entities in Test Data DB. The information stored in 'exp_info' contains id number of the experiment (*ExpID*), the id of the project in which the experiment is defined (*RelatedProjID*), the id of the specimen which is used in the experiment (*RelatedSpecID*), name of the experiment (*ExpName*), the experimenting staff (ExpStaff), name and e-mail address of the person who is responsible for the experiment (*EContactName* and *EContactMail*), description (*ExpDesc*), username of the experiment creator (*ECreator*), keywords (*EKeywords*), the dates that the experiment is created, modified and conducted (*DateCreated*, *DateModified* and *DateConducted*). *ExpID* is the auto incrementing primary key attribute whose value is determined by the system each time an experiment is created. In order to create and experiment, user must be one of the owners or creator of the related project. Related project id of experiment is stored in the *RelatedProjID* attribute. The ownership and visibility settings of an experiment is taken directly from that of the related project. *RelatedProjID* and *RelatedSpecID* attributes are foreign keys that references to *ProjID* attribute of 'proj_info' and *SpecID* attribute of 'spec_info' respectively. Another foreign key of this entity is the *ECreator* attribute which references to the *name* attribute of the 'my_aspnet_users' entity.

doc info: The information about documents related to projects, experiments and specimens is stored in this entity. However, this entity does not store the document itself, but stores the file name of the document under the *FileName* attribute. Documents are stored in the file system under pre-defined folder structure as shown in the Figure 3.6. The reason behind not storing the files in the database is to provide accessibility to those files for other programs without DBMS interaction. Moreover, storing files outside the

database reduces the time required to retrieve data from the database. The files are stored in a folder named "DocumentationFolder" which resides under the application path. This folder contains three sub-folders which represent the related entity that the document belongs and they are named as "Experiments", "Projects" and "Specimens". Each created project, specimen, and experiment creates a folder starting with "ProjID", "SpecID" and "ExpID" respectively and followed with the id of the entity under the related sub-folder. Then the files are stored under these identified folders where the information about file is kept in the database.

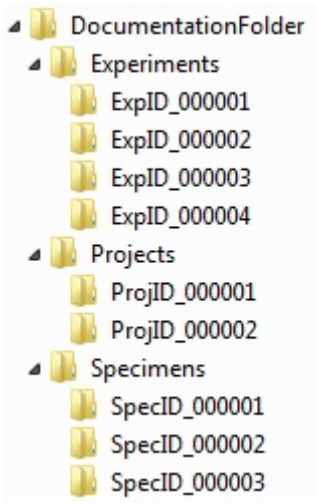


Figure 3.6, Predefined folder structure for storing documentation

The information about where the documentation belongs is decided by two attributes. First attribute is named *RelatedTo* which stores either 'P' for project, 'S' for specimen or 'E' for experiment. The second attribute is the *RelatedID* attribute which is a foreign key attribute that references to the primary key attributes of 'proj_info', 'spec_info' and 'exp_info' entities. *RelatedTo* and *RelatedID* attributes together is enough to determine where

the document belongs. Other information stored in the 'doc_info' entity is file name of the document (*FileName*), description of the document (*DocDesc*), the system extension of the document (*DocExt*), MIME type of the document (*DocType*), user name of the document uploader (*DCreator*), size of the document in bytes (*DocSize*), plotability of the document (*IsPlotable*), and the dates when the document is created and modified (*DateCreated* and *DateModified*). The plotable documents are the ones which contains the test data inside them. Currently, only MS Excel documents are accepted as plotable documents. When a plotable document is uploaded, a new entity in the Test Data DB is created whose name begins with the experiment id and continued with the file name. All the data in the excel table is transferred to that entity and the signal information is stored in the 'signal_info' entity of Test Data Database.

groups: This entity is used to store the information about user groups. User groups are used in order to classify website members according to their common properties such as an institute that the members bound or a project that the members working on. Moreover, ownership and visibility levels of projects and specimens can be set on a group by group basis; thus, the information can be secured against unwanted users and groups. When a group is created, system automatically assigns a new identification number to the group and stores it in the auto incrementing *GroupID* attribute. This attribute is the primary key of the 'groups' entity. User ID of the group creator is stored in the *GCreatorID* attribute which is a foreign key and references to the *id* attribute of 'my_aspnet_users' entity. Other information stored in the 'groups' entity contains name of the group (*GroupName*), description about the group (*GroupDesc*), creation date of the group (*DateCreated*) and total number of members resides in the group (*ToatalMembers*). Although the total number of members is stored in 'group'

entity, IDs of group members and their ownership statuses are stored in 'group_members' table. Only group owners can add or remove group members and only they can change the group name and description.

group members: This is a connection table which links the groups and their members by pairing group id and the user id in its *GroupID* and *UserID* attributes. Both attributes together form the primary key of the table. *GroupID* attribute is referencing to the *GroupID* attribute of 'groups' entity and *UserID* attribute is referencing to the id attribute of 'my_aspnet_users' entity. Also, by its *isOwner* attribute, it stores the ownership status of a member. If a member is group owner, he can add or remove group members and can edit the group information.

proj spec usr grp own vwr: This is the most complex connection table in the database system. The main purpose of this table is storing the ID numbers of owners and custom viewers of both groups and users for each project and specimen. Thus, it decides whether a user/group has the privilege to modify or view the content of a project or a specimen. Therefore, this table provides privacy and security against unwanted users and groups. This table does not store information about experiments because ownership and visibility settings of experiments are inherited from those of the related project. The table has five attributes and four of them form the primary key. *PorS* attribute stores a single character which is either 'P' for project or 'S' for specimen. Together with *PorSID* attribute, they identify the related project or specimen that the rest of the attributes contains information about. The *UorG* attribute can store either 'U' or 'G' which implies that the owner or viewer is a user or a group. *UorG* and *UorGID* attributes together is enough to identify the specific user or group and the *isOwner* attribute decides whether the user/group is owner of the project/specimen or the user/group

does only have the privilege to see the information about specified project/specimen.

3.3.3 Test Data Database

Test Data Database is the database in which the test data and the information about the signal readings are stored. Unlike the other three databases, this database does not contain a pre-determined amount of entities. The only pre-defined entity is the 'signal_info' entity. Other than that, with each new experiment a new table is created in the database for storing the test data. All newly created test data tables will have a similar form. Therefore, in the presented ERD, only one example for experiment entity is given and that entity is called 'exp_000001_proj1'. The ERD of these two entities can be seen in Figure 3.7.

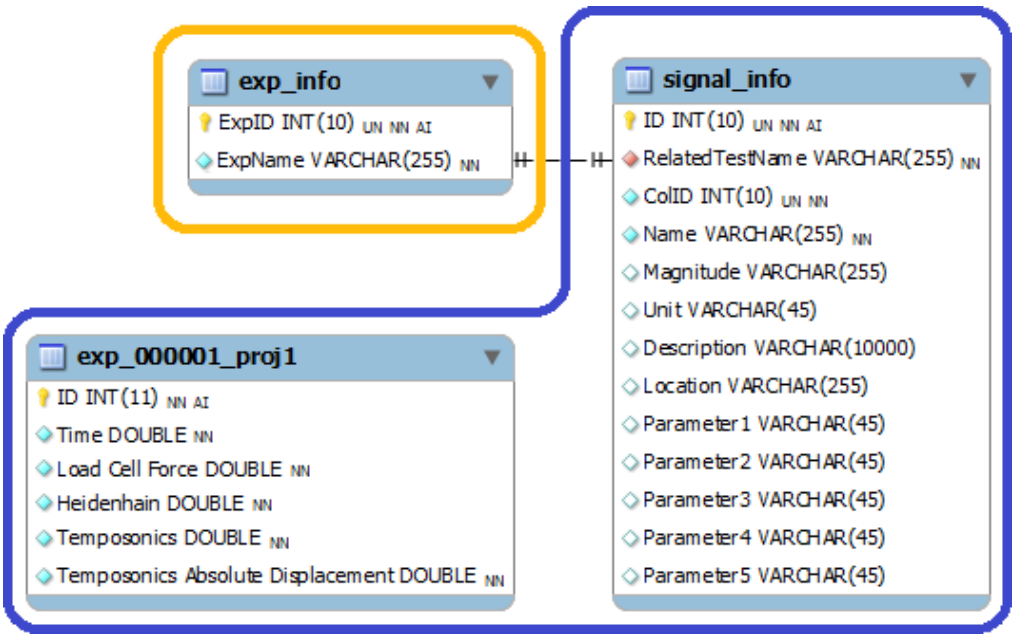


Figure 3.7, ER Diagram of Test Data Database

Functions of each entity and their attributes are explained below:

signal_info: Information about sensor readings which are collected during an experiment is stored in this entity. Each created 'signal_info' entity stores the name of the related experiment in its *RelatedTestName* attribute which is a foreign key referencing to *ExpName* attribute of 'exp_info' entity. *ColID* attribute stores the column number of the reading in Excel spreadsheet if the experiment data is inserted into the system as an Excel document. Looking at the values of these two attributes is enough to find the signal information about a specific reading. However, rather than making these two attributes as primary keys, a new auto incrementing attribute called *ID* is defined and selected as primary key. The information collected about readings contains the name of the sensor or reading which shows what the collected data is about (*Name*); unit of the data readings (*Unit*), detailed information about the sensor or the reading (*Description*), location of the sensor (*Location*) and five non-specified parameter which can be necessary depending on the test setup (*Parameter1* to *Parameter5*). Also, magnitude of the reading can be selected from a pre-defined list, such as acceleration, coefficient, current, displacement, energy, frequency, force, inertia, length, mass, moment, ratio, resistance, rotation, stiffness, strain, stress, time, velocity, voltage and wave length.

exp_000001_proj1: 'exp_000001_proj1' is an example entity which stores test data of an experiment. Whenever an Excel file is uploaded as a test data, a new similar entity is created in the Test Data database. The name of each entity is composed of three parts. The first part is "exp_" part and it is fixed for all experiment entities. The second part is a six digit right padded integer number which shows the ID number of the related experiment. This ID information is gathered from 'exp_info' entity. The third part contains the

filename of the uploaded MS Excel file. Therefore, by looking at the name of the entity, related MS Excel document can easily be found in the file system. After the creation of a new entity, attributes of it are created depending on the columns of the spreadsheet. The number of attributes is variable and equal to the number of columns in experiment's MS Excel spreadsheet plus one for the auto incrementing identity column which is also a primary key. Even though there is no need for a primary key for this entity, a primary key attribute is provided just for indexing issues. The name of the attributes are also taken from MS Excel spreadsheet. However, users may change them before the upload process of MS Excel file if desired. The value type of all the attributes are "double". Therefore, any numerical tabular data can be inserted into these entities. After all the attributes are created, the test data from MS Excel spreadsheet is transferred to the database for later use such as when plotting the test data.

3.3.4 Forum Database

As the website users communicate with each by using the integrated forum functionality, the topics created by the users and the messages posted in created topics are stored in "Forum Database". The information about created topics is stored in the 'topics' entity where messages posted under these topics are stored in the 'posts' entity. The entities, their attributes and the relations between those entities can be seen in the Entity Relationship Diagram of the database (Figure 3.8).

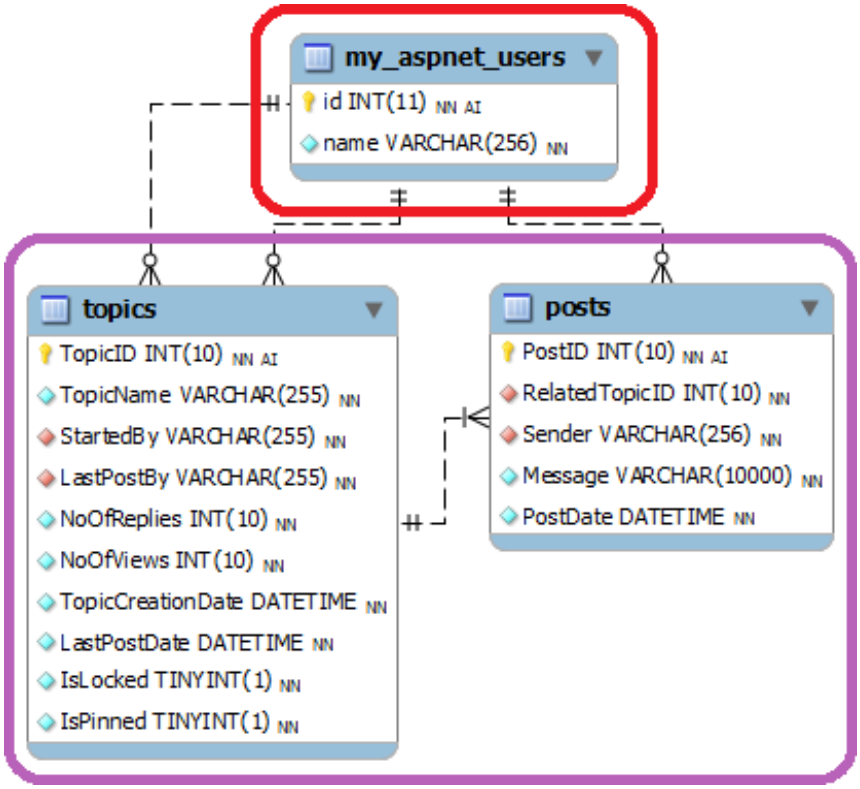


Figure 3.8, ER Diagram of the Forum Database

Functions of each database entity and their attributes are explained below:

topics: Information about created forum topics is stored in this entity. The stored information contains ID number of the topic (*TopicID*), title of the topic (*TopicName*), the user names of topic creator and the last message poster (*StartedBy* and *LastPostBy*), total number of replies under that topic (*NoOfReplies*), the number that shows how many times the topic is viewed (*NoOfViews*), the dates that the topic is created and the last post is made (*TopicCreationDate* and *LastPostDate*), whether the topic is locked for replies or not (*IsLocked*), and priority of the topic (*IsPinned*). Each time a new message is posted under a topic, the value of *NoOfReplies* attribute is increased by one. Similarly, each time a user looks at the contained

messages of a topic, the value of *NoOfViews* attribute is increased by one. If the *IsLocked* attribute of a topic is set to true, none of the users will be able to post a message under that topic. On the other hand, if the *IsPinned* attribute of a topic is set to true, that topic gains a higher priority against unpinned topics, and will be listed at the top of unpinned topics together with other pinned topics. The auto incrementing TopicID attribute is the primary key attribute for 'topics' entity. There are also two foreign key attributes which are StartedBy and LastPostBy attributes. Both of the attributes are referencing to the *name* attribute of 'my_aspnet_users' entity.

posts: This entity is used for storing information about messages posted under various topics. For each posted message, 'posts' entity contains information about ID number of the post (*PostID*), ID number of the related topic (*RelatedTopicID*), user name of the message sender (*Sender*), content of the message (*Message*) and the date of post (*PostDate*). The *PostID* attribute is auto incrementing primary key attribute. The *RelatedTopicID* is a foreign key attribute which references to the *TopicID* attribute of 'topics' entity. The Sender attribute is also a foreign key and is referencing to *name* attribute of 'my_aspnet_users' entity.

CHAPTER 4

THE WEBSITE

4.1 Overview

In this chapter, the platforms and programming languages used for application development are described together with the development stages of the website. Also, the tasks that the website users can perform are listed as a Use Case Diagram, and the execution processes of those tasks are explained briefly with given Sequence Diagrams.

4.2 Development of The Website

Website is the presentation layer of a web application and serves as a user interface of a web application. Users interact with the database system by using the website functionalities. Currently, all the data entry to the database, modifying or deleting it is handled by the web application.

Similar to desktop applications, various programming languages are being used for developing web applications. For the presented application, Microsoft's .NET Framework v2.0 was used as software developing framework. ASP.NET which is the web application framework of Microsoft, was used for designing the web interface, where C# programming language is used to bring all the functionality to the website. In order to provide communication between website and DBMS, MySQL's ADO.NET driver was integrated to the application. For the development environment, Microsoft's

Visual Studio 2008, which is the native Integrated Development Environment (IDE) for .NET Framework, was used.

In the design process, first login controls and main navigation menu which enables users to navigate through web pages were designed together with their page layout and visual properties. Since both the login controls and main navigation bar are needed to be accessed from any web page on the website, these web controls were put inside the main master page of the website called "Site.master". A *master page* is a feature which comes with ASP.NET framework. The content of a master page can be shared with other master pages and with any web page in the website. The main purpose of a master page is to remove the need to duplicate code for shared elements within a website. The pages which adopt the content of a master page is called a "content page". Website contains four more master pages other than "Site.master". These are "Browse.master" which contains searching controls, "Experiments.master", "Projects.master" and "Specimens.master" that contains special navigation menus related to experiments, projects and specimens respectively. After designing navigation menus and common controls in master pages, contents of all other pages are developed.

As a next step, button functionalities were granted and data binding operations for necessary controls are handled. For plotting the test data as a scatter graph, an external open-source .NET class library called "ZedGraph" was integrated into the application.

Finally, after completing the development of the METUce Experimental Database web application, the website was published and copied in a server computer which also contains the database system of the application. The server resides in the Civil Engineering Department of METU. Current

operating system of server is Windows Server 2003 and it uses Microsoft's Internet Information Services (IIS) 6.0 for hosting the website.

In order to explain the capabilities of the web application, a "Use Case Diagram" is given in Figure 4.1 where all the tasks that the application can perform is presented. In order to reduce crowdedness of relational lines between actors and use cases, some tasks are grouped together according to the entities that they are related to. Also, the DBMS, which enables addition, modification and deletion of database entries, is shown in the right side of the system rectangle of Figure 4.1. During the course of their processes, all the tasks interact with the DBMS to manage the data in the databases. Execution processes of each task is explained with sequence diagrams further in this chapter.

On the left side of the system rectangle, five different user types are listed as actors. Four of these user types are predefined user roles which are *User*, *HigherRankedUser*, *Moderator* and *Administrator.* Users that are not the member of the website (guests) forms the fifth user type.

Each user type can perform specific tasks and the relations between user types and the use cases that they can perform are linked together by different colored lines in Figure 4.1.

The tasks that a *Guest* can perform are logging in the website; creating a new membership account; browsing and searching publicly accessible projects, specimens and experiments; downloading the documentation related with publicly accessible projects, specimens and experiments; plotting experimental data of publicly accessible projects and browsing the list of topics created in the forum.

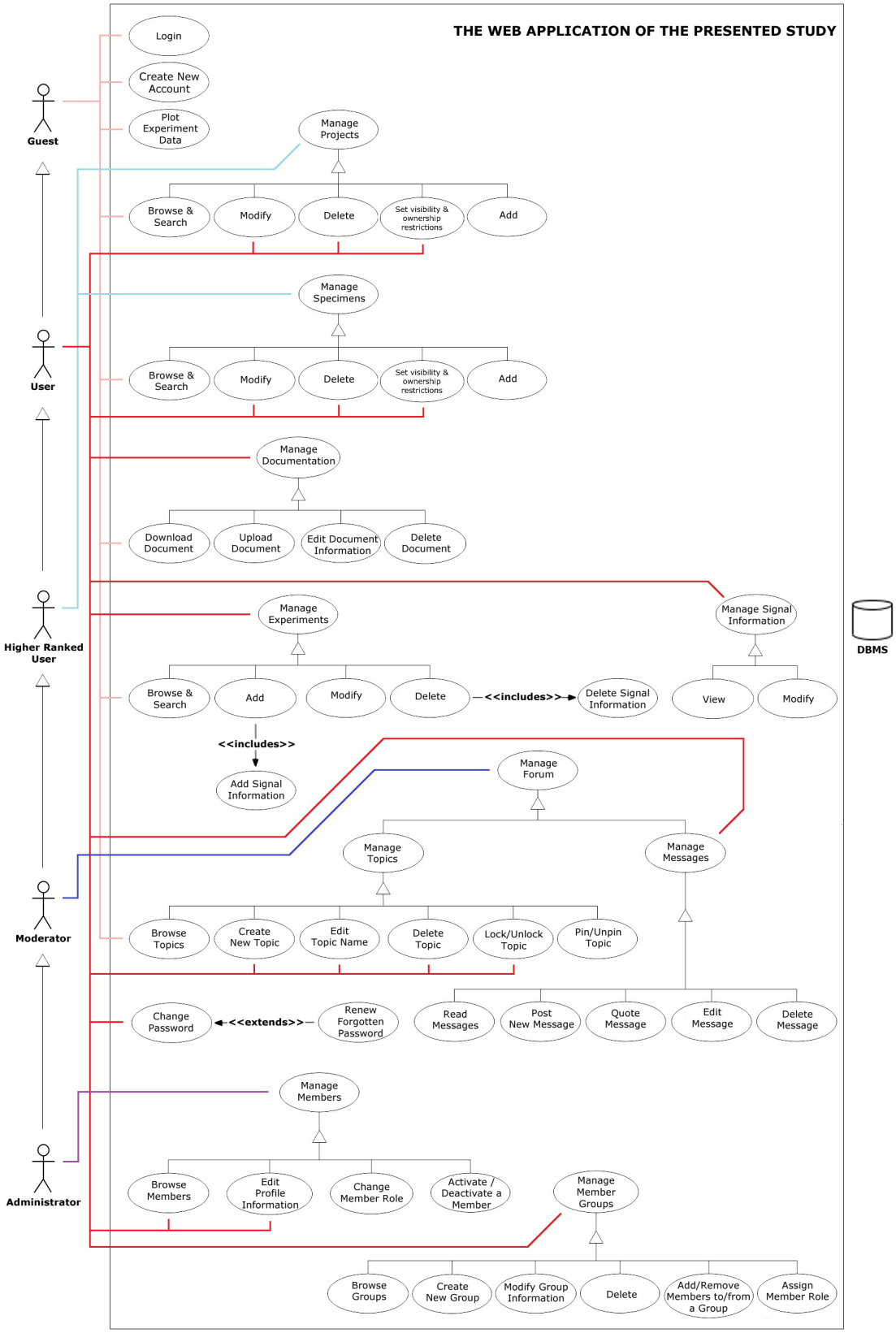


Figure 4.1, Use Case Diagram of the web application

Website members having the *User* role can do everything that a *Guest* can do and moreover, they can modify the information of owned projects and specimens. They can set visibility and ownership restrictions for them. They can also delete owned projects and specimens. However, in order to add a new project or specimen, members need to have at least *HigherRankedUser* role. Members with *User* role can create and delete experiments in owned projects. They can also view and edit information about experiments and their signal readings. *Users* can also upload any type of document under owned projects, specimens or experiments; edit document information or delete uploaded documents; edit their profile information; change their password or request new password if they forgotten the previous one; view information about other site members; create/delete member groups and add/remove members to/from groups which they own; set roles of members added under those groups as *Group Owner* or *Group member*; create forum topics and reply under those topics. They can also lock owned topics in order to prevent further message posts under those topics.

Higher Ranked Users can do anything that a *User* can, moreover they can create new projects and specimens. Implementation of this user role was crucial for information reliability and prevents irrelevant data uploads to the database by unwanted users.

Moderators can perform every task that a *Higher Ranked User* can but they are not affected by ownership and visibility settings of projects and specimens. They can also edit other user's profile information. Also, they have full power in forum tasks. They can edit or delete any topic or message they want; lock any topic or pin them to the top in order to attract more attention on those topics.

Final and the most powerful role is the *Administrator* role. Site administrators can do everything that other roles can. Moreover, they can change user roles or they can lock/activate other user's membership statuses.

4.3 Website Functionalities

In this section, the tasks and how they perform is explained briefly with presented sequence diagrams.

4.3.1 Creating A New Member Account

In order to use most of the website functionalities, users must register to the application. New account creation is handled in CreateAccount web page. The sequence diagram for creating a new member account is presented in Figure 4.2. Here, user selects a username and a password for his account and sends the information to the application together with his profile information. The application validates the information by checking the provided username and e-mail address. If both the username and e-mail address is unique, application creates the account by entering the information into the "Membership Database" and sends an account activation mail to the user's provided e-mail address. However, until the user follows the link provided in the e-mail message, the account stays unapproved. After confirming his e-mail address, user can login to the website. Newly created users are given least privileged "User" role type initially. Later, site administrators can change his role type.

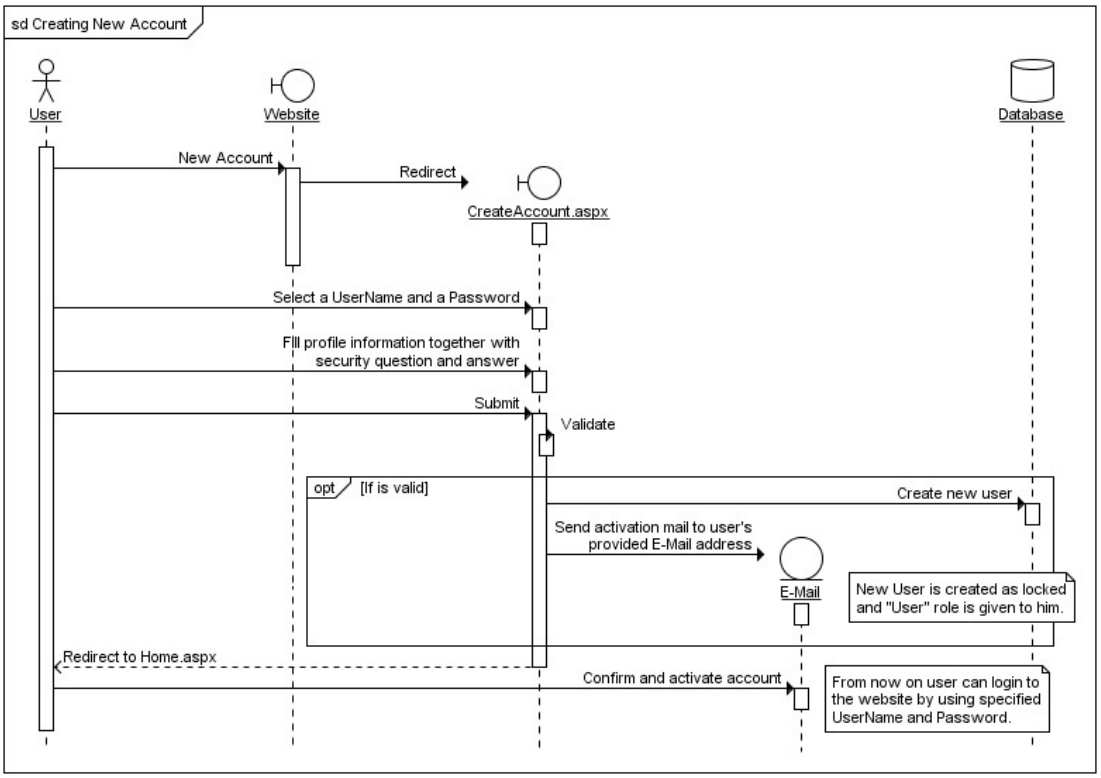


Figure 4.2, Sequence Diagram for creating a new membership account

4.3.2 Creating A New Member Group and Adding Members

Creating new member group is handled in GroupList page. The sequence diagram of creating new member group can be seen on Figure 4.3. On user arrival, page lists all previously created member groups. When user commands for new group creation, group name is asked to the user. After entering the required information, a new member group is created and the information about the group is entered to the "groups" table of the "Main Database".

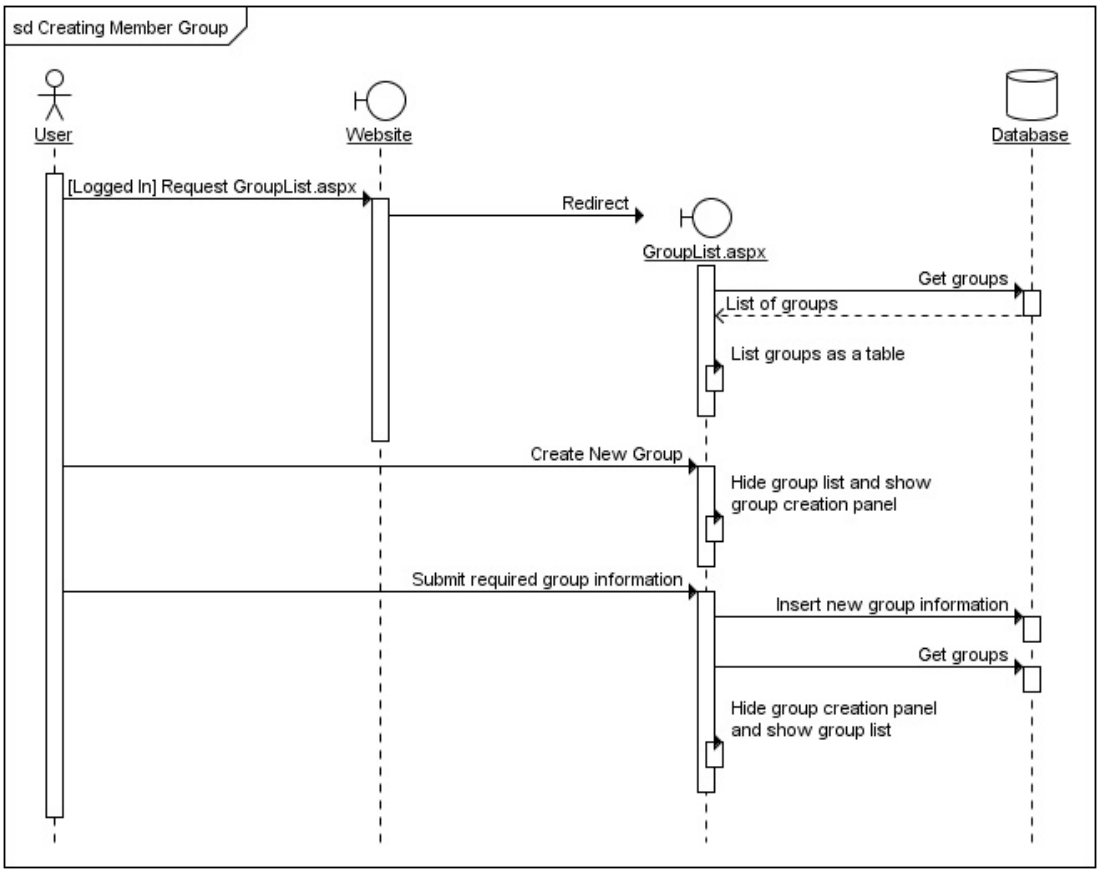


Figure 4.3, Sequence Diagram for creating a member group

After creation of a member group, desired members can be added to that group. The sequence diagram that shows the process of managing group members is presented in Figure 4.4. Managing group members is handled in GroupInfo page and only group owners can add or remove group members . They can also edit role types of group members. There are only two type of roles for group members. Users can either be a member of the group or they can be an owner. New members added to a group by selecting desired website members from the provided dropdown list in GroupInfo page. After adding a member to a group, his group role can be assigned by the group owners. Each time a user is included in a group or removed from it, or after

his role type is changed, the changes are reflected to the "group_members" table of the "Main Database".

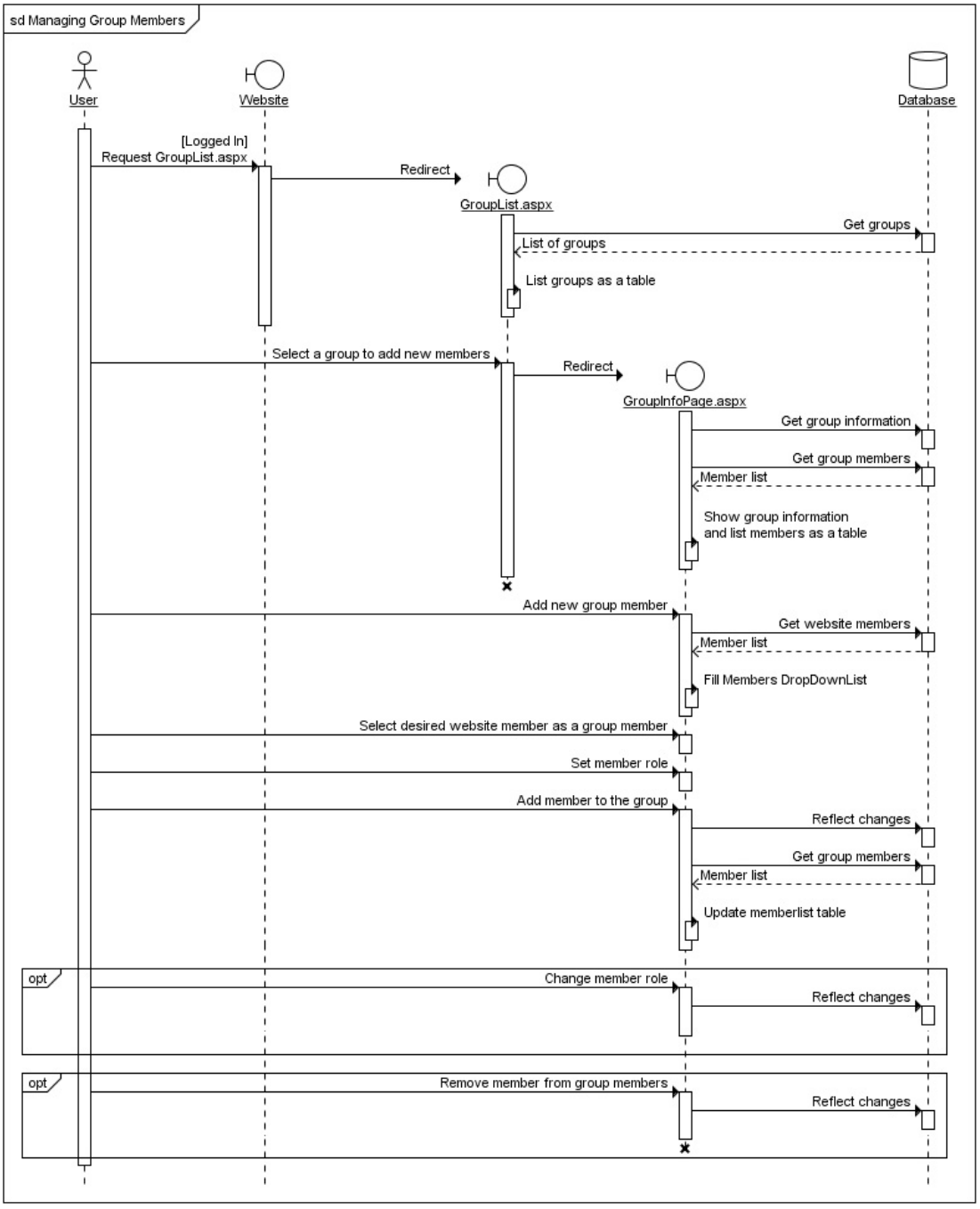


Figure 4.4, Sequence Diagram for managing group members

4.3.3 Creating New Project

To create a new experiment and upload test data to the website, first a user must create a project. Creating new projects is handled in CreateProject page. However, in order to create a project a website user must at least have *HigherRankedUser* role. The process of creating a new project is given as a sequence diagram in Figure 4.5.

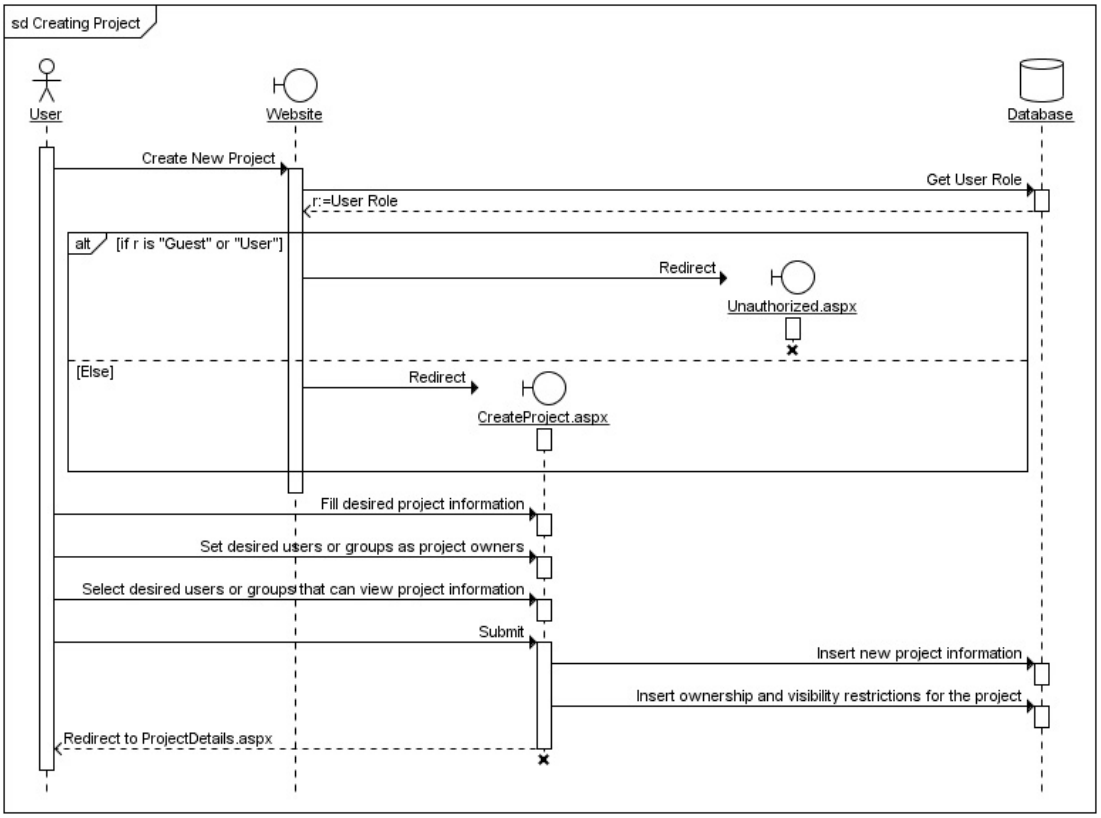


Figure 4.5, Sequence Diagram for creating a new project

During project creation process, user is asked to fill various information about the project such as name of the project, the facility of the project, starting and ending dates of the project etc. However, specifying the

attributes other than the project name is optional. After filling the necessary information, project creator can select the desired website members or member groups as project owners. Also, during project creation or after the creation process, project creator or owners can select the project visibility level. Submitting the entered information completes the creation process of the project. The project information is then entered in "proj_info" table of "Main Database" where the ownership and visibility settings of the created project is entered in "proj_spec_usr_grp_own_vwr" table of the "Main Database".

4.3.4 Creating A New Specimen

Creating specimen is handled by CreateSpecimen web page. Similar to project creation process, in order to create a specimen a user must at least have HigherRankedUser role. Figure 4.6 shows the sequence diagram of specimen creation process. In CreateSpecimen page, user is asked to fill some information related with the specimen. After filling the desired information, specimen creator can select desired members or member groups as specimen owners and can set visibility restriction level for that specimen. Submitting the information completes the creation process of the specimen. General information about the specimen is entered in "spec_info" table of "Main Database" and the information about groups or member that own or can view the specimen is entered in "proj_spec_usr_grp_own_vwr" table of "Main Database".

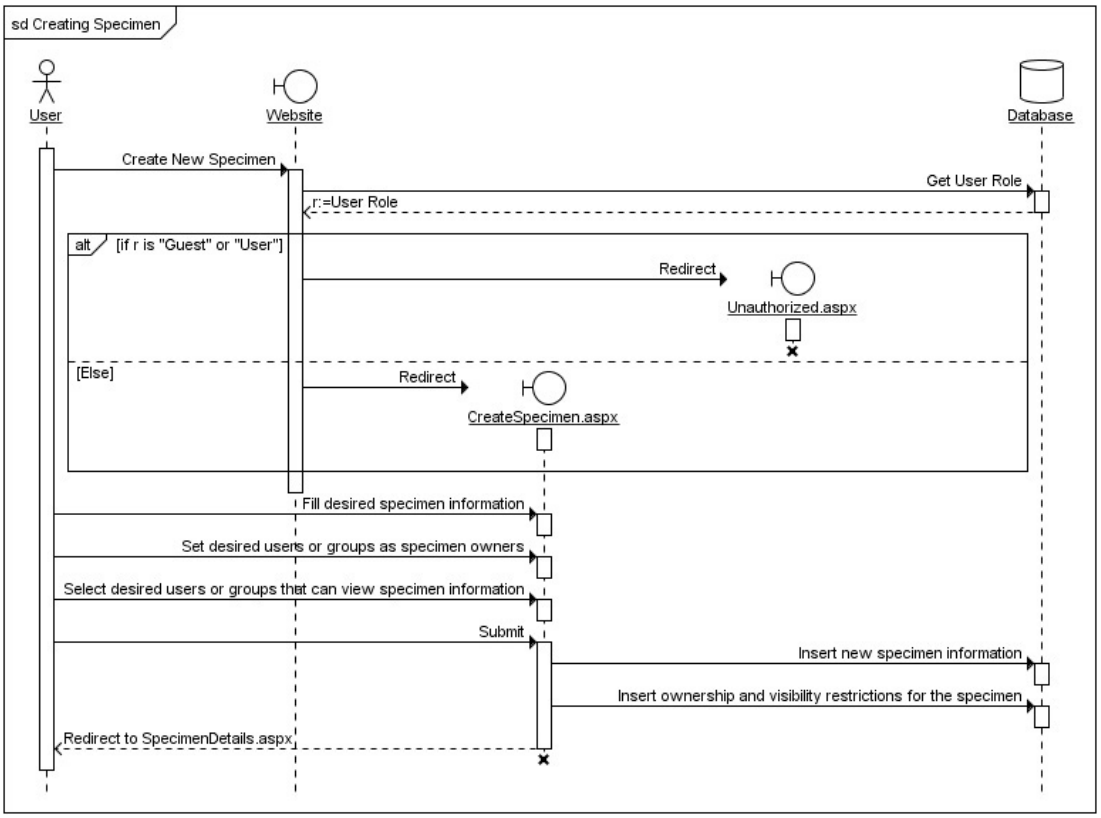


Figure 4.6, Sequence Diagram for creating specimens

4.3.5 Creating A New Experiment

In order to create an experiment, a user must first create the project that contains the experiment and the specimen on which the experiment is conducted. As diagrammatized in Figure 4.7, experiment creation is handled in AddNewExp page. The experiment creation process starts with uploading the Excel file which contains the test data. This process may take several minutes according to the file size and currently the maximum permitted file size is 100 MB. Upon completion of file upload, the spreadsheet names which contains a valid data set are listed to the user.

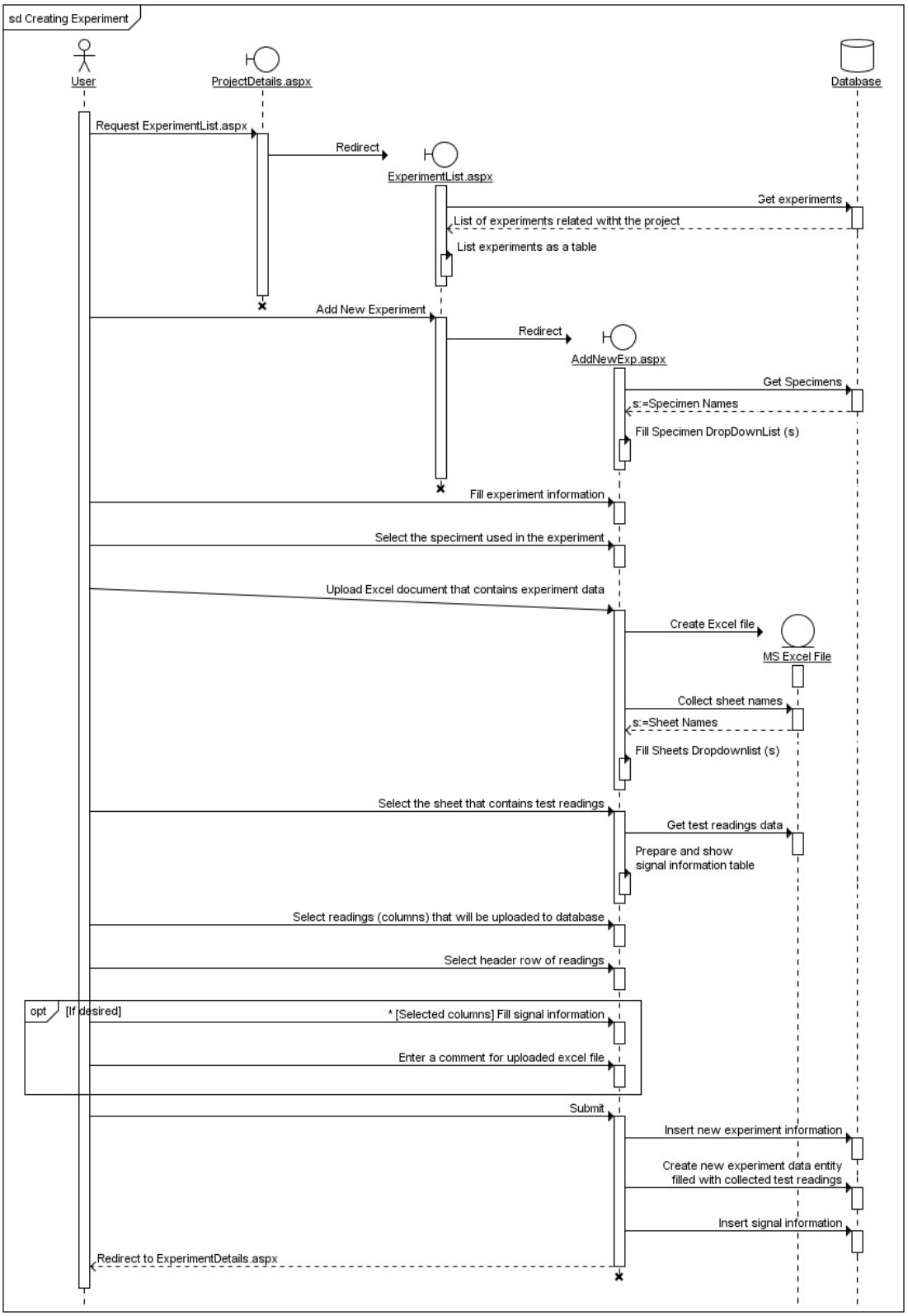


Figure 4.7, Sequence Diagram for creating a new experiment

After selecting the desired spreadsheet name the data is presented to the user as a table and user is asked to select the signal readings that will be inserted into the database. Also, the user is asked to fill the information about selected signals readings. Final step is submitting the experiment information which completes the experiment creation process and inserts the collected data into the database. General information about the experiment is entered in "exp_info" table of "Main Database" and the information about signal readings is entered in "signal_info" table of the "Test Data Database". Also, for storing test data, a new table is created in "Test Data Database" whose columns are represents the signal readings that are selected in experiment creation phase, and the test data collected from the Excel spreadsheet is inserted into this table.

4.3.6 Viewing and Editing Signal Information

Signal information, which makes experiment readings more meaningful, is collected during the experiment creation process and can later be viewed by website users. Also, signal information of an experiment can be modified by experiment owners. Viewing and modifying signal information is handled in ExperimentDetails page of the desired experiment and the process is shown in Figure 4.8. In ExperimentDetails page, all the Excel files which contains test data about selected experiment are listed. When a user asks for signal information details of an experiment file, the details are presented to the user as a table. If the user owns the experiment, he can edit desired information on this table.

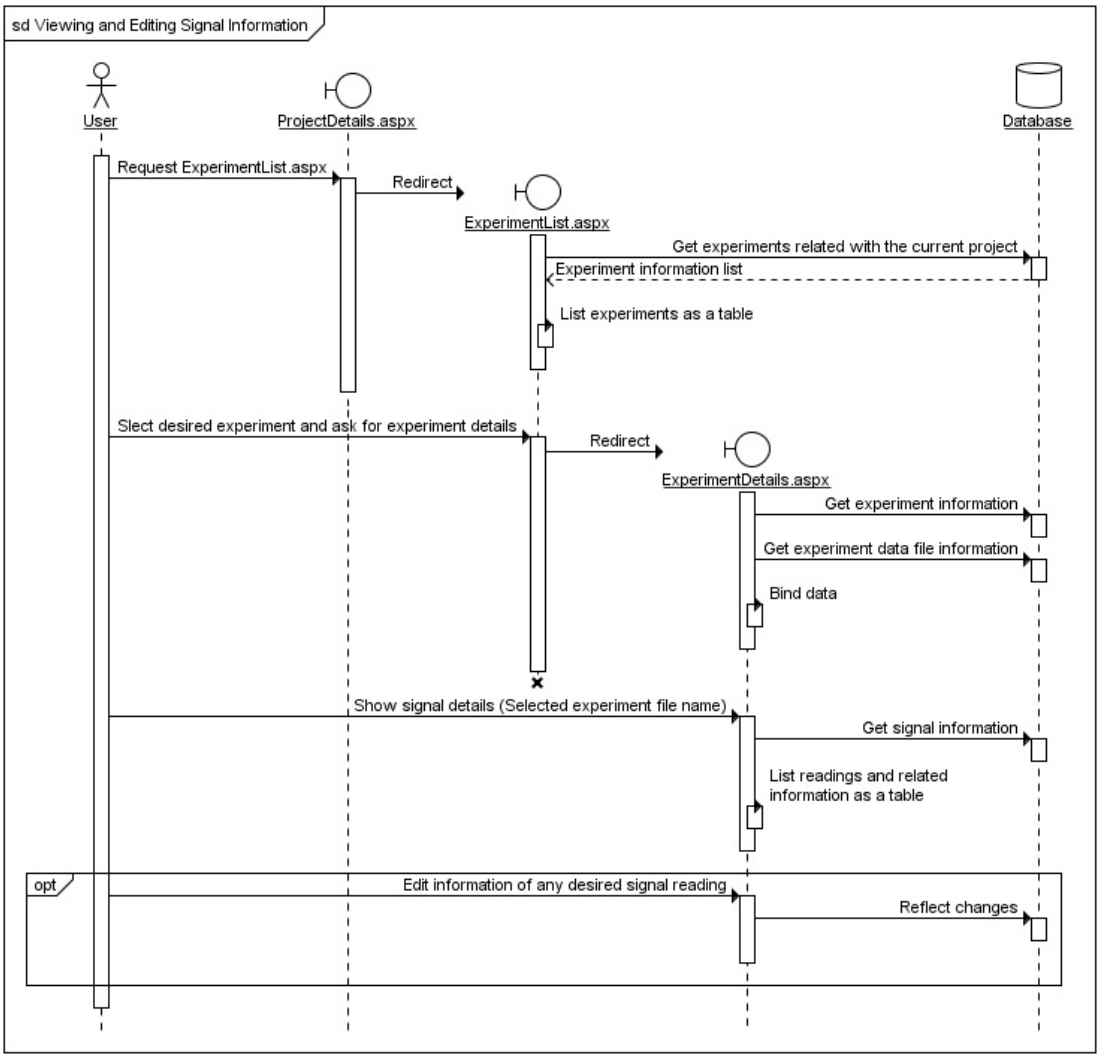


Figure 4.8, Sequence Diagram for viewing and editing signal information

4.3.7 Plotting Experiment Data

Plotting experiment data is a very useful functionality and greatly increases the understandability of the data. The developed web application can plot the experiment results as a scatter graph. The sequence diagram which shows the process of plotting an experiment is given in Figure 4.9. Plotting experiment data is handled in PlotGraph page. In order to reach this page,

user must select desired Excel file which contains the test data in ExperimentDetails page and click on the "Plot Experiment" button.

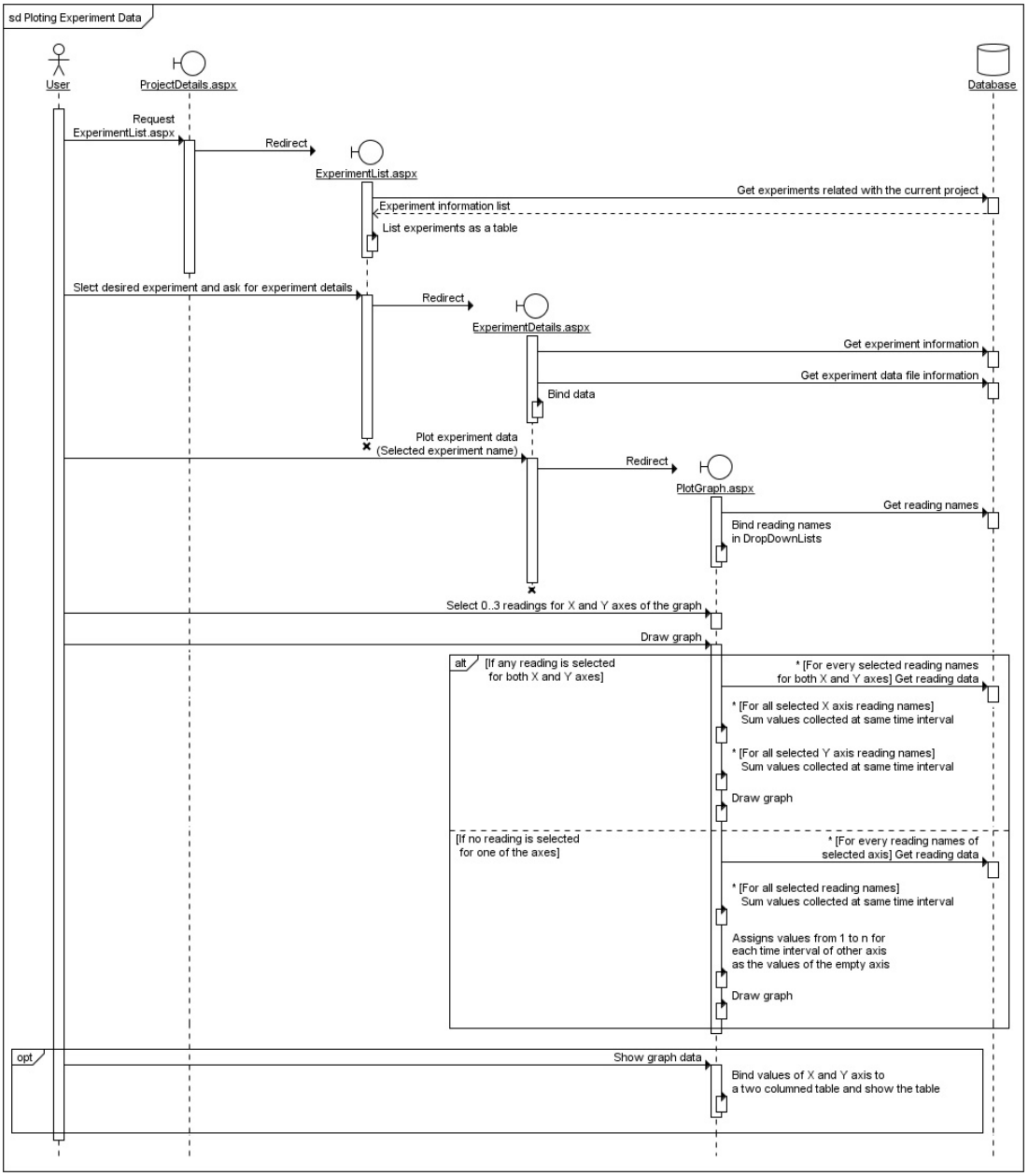


Figure 4.9, Sequence Diagram for plotting an experiment

When user arrives PlotGraph page, application collects the signal information related to the experiment and lists the names of signal readings. Here, user can select one to three signal readings for both of X and Y axes. Upon selection, detailed information about the selected signal is retrieved from the database and presented to the user. The collected data of selected signals is used as the values of scatter graph. If multiple signals are selected for an axis, the readings are summed and the summation is used as the values of scatter graph. Upon user request, website presents the data which is used as the values of X and Y axes in a two columned table.

4.3.8 Managing Documentation

Collecting documentation is very important for experiment databases because the selected entity attributes may not suit well for every situation. Even if the attributes are sufficient to describe the experiment in detail, filling the necessary information may take too long. However, uploading files related to a project, a specimen or an experiment is both quick and increase understandability dramatically.

The process of managing documentation for projects, specimens and experiments is given as a Sequence Diagram in Figure 4.10. In the sequence diagram "P/E/S Details Page" boundary type object refers to one of the ProjectDetails.aspx, ExperimentDetails.aspx and SpecimenDetails.aspx according to where the user want to add documentation.

Uploading, downloading and deleting documentation is handled in DocumentList page which can be accessed from details page of the related project, specimen or experiment. Upon user arrival on DocumentList page, website collects the information about documents that are related with

selected project, specimen or experiment, and lists them to the user as a table.

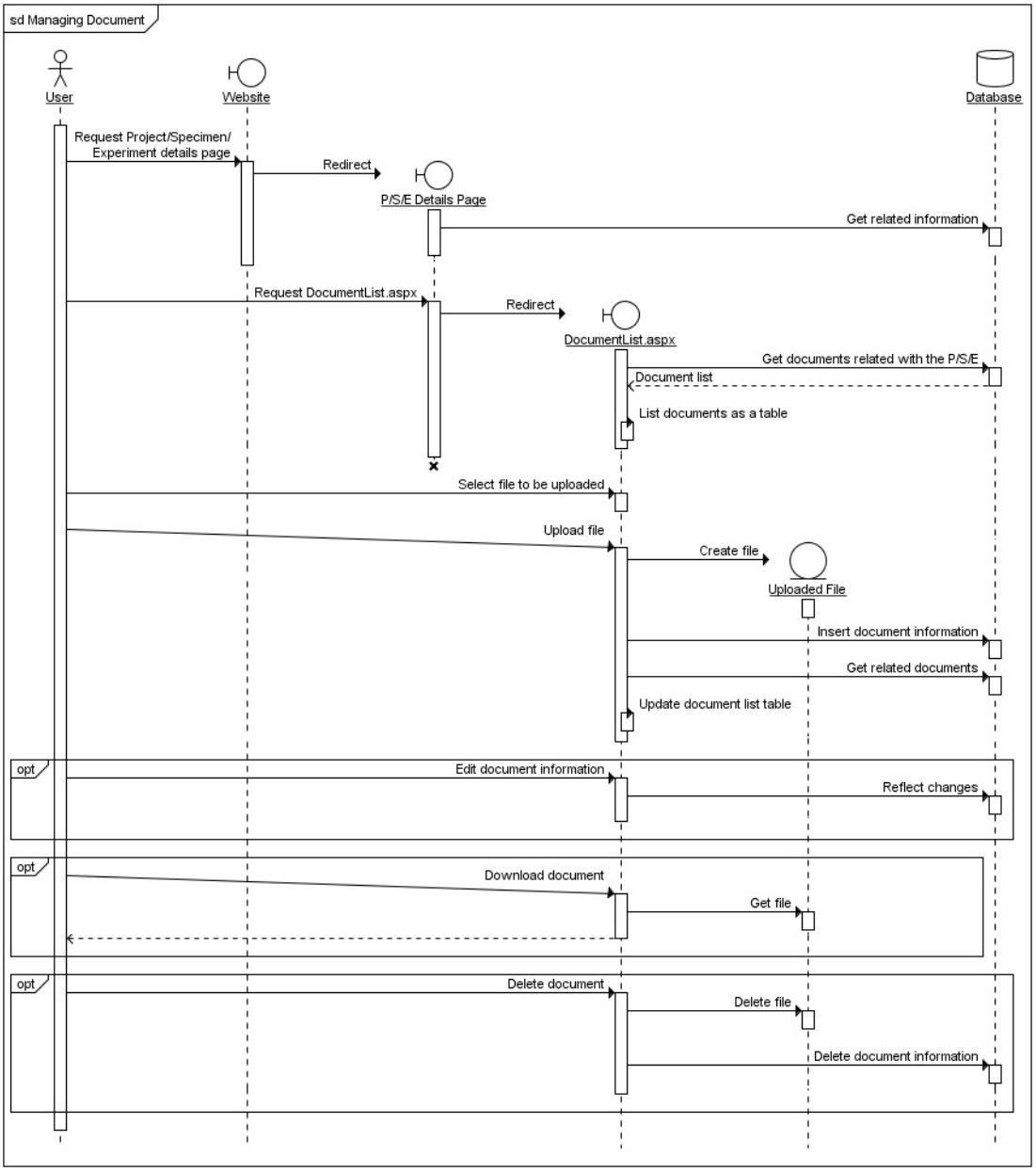


Figure 4.10, Sequence Diagram for managing documents

User can download files by clicking on their names, or if the user is an owner, he can upload a new one to the application. Any type of file can be uploaded to the website unless its size is greater than 100 MB. When a new document is uploaded, the information is inserted into the "doc_info" table of "Main Database" and the file is saved under a specific folder in the application path. Before file uploading, users can comment on the file and briefly describe the purpose of the file. This comment can later be edited, however.

4.3.9 Using Search Functionality

Searching is a very important feature and greatly helps users to find desired projects, specimens and experiments. Searching can be performed in Search page of the website. The process of searching is diagrammatized in Figure 4.11. There are two searching options which are "Simple Search" and "Advanced Search". The default search option is set to "Simple Search" which scans the database for the entered word or phrase in the name, keyword and descriptions fields of the selected entity which is either project or specimen or experiment. On the other hand, in "Advanced Search", user can specify separate word/phrase for various attributes and the application returns the results which contains all of the specified word/phrase. The projects, specimens or experiments that satisfy the search conditions are listed to the user and user can directly go to the details page of desired project, specimen or experiment by selecting it.

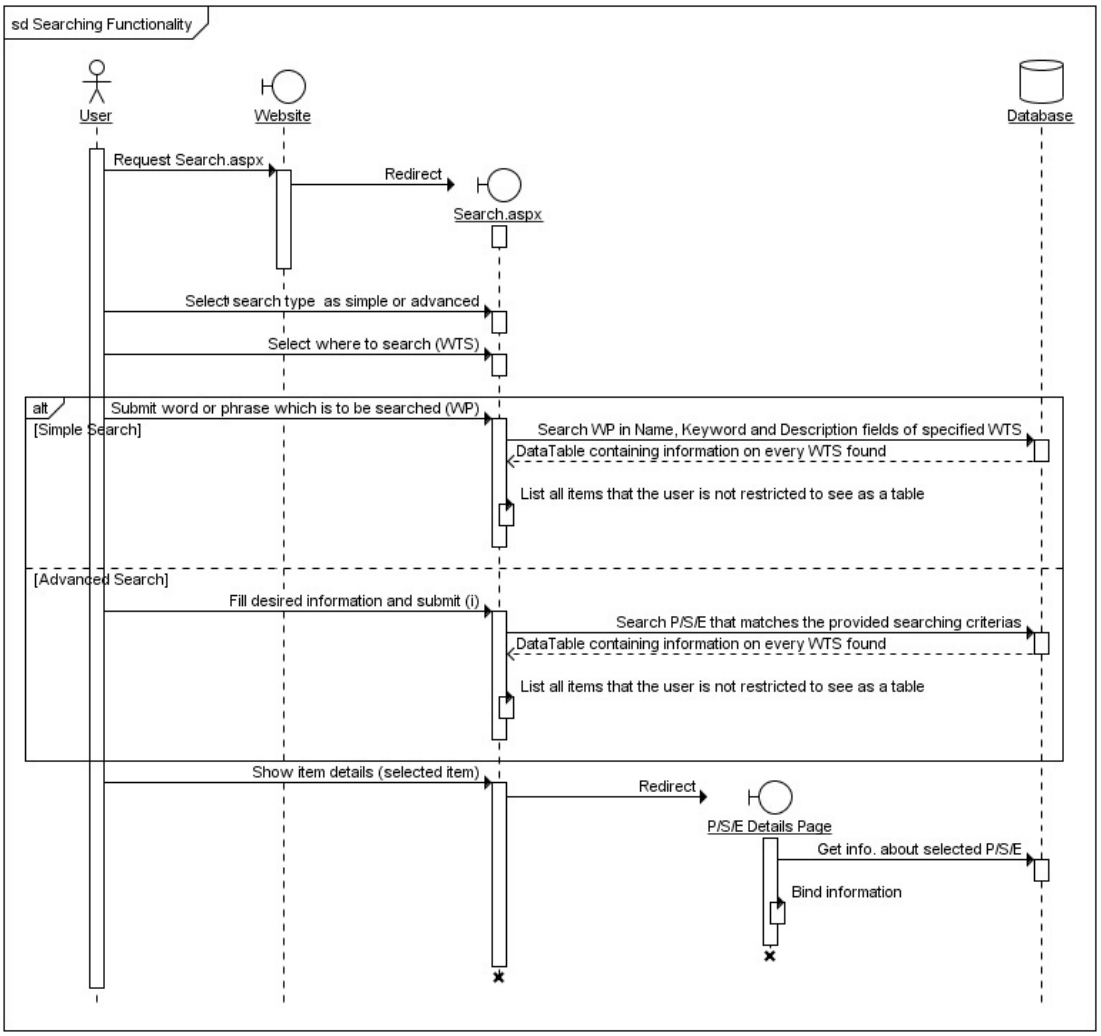


Figure 4.11, Sequence Diagram for using search functionality

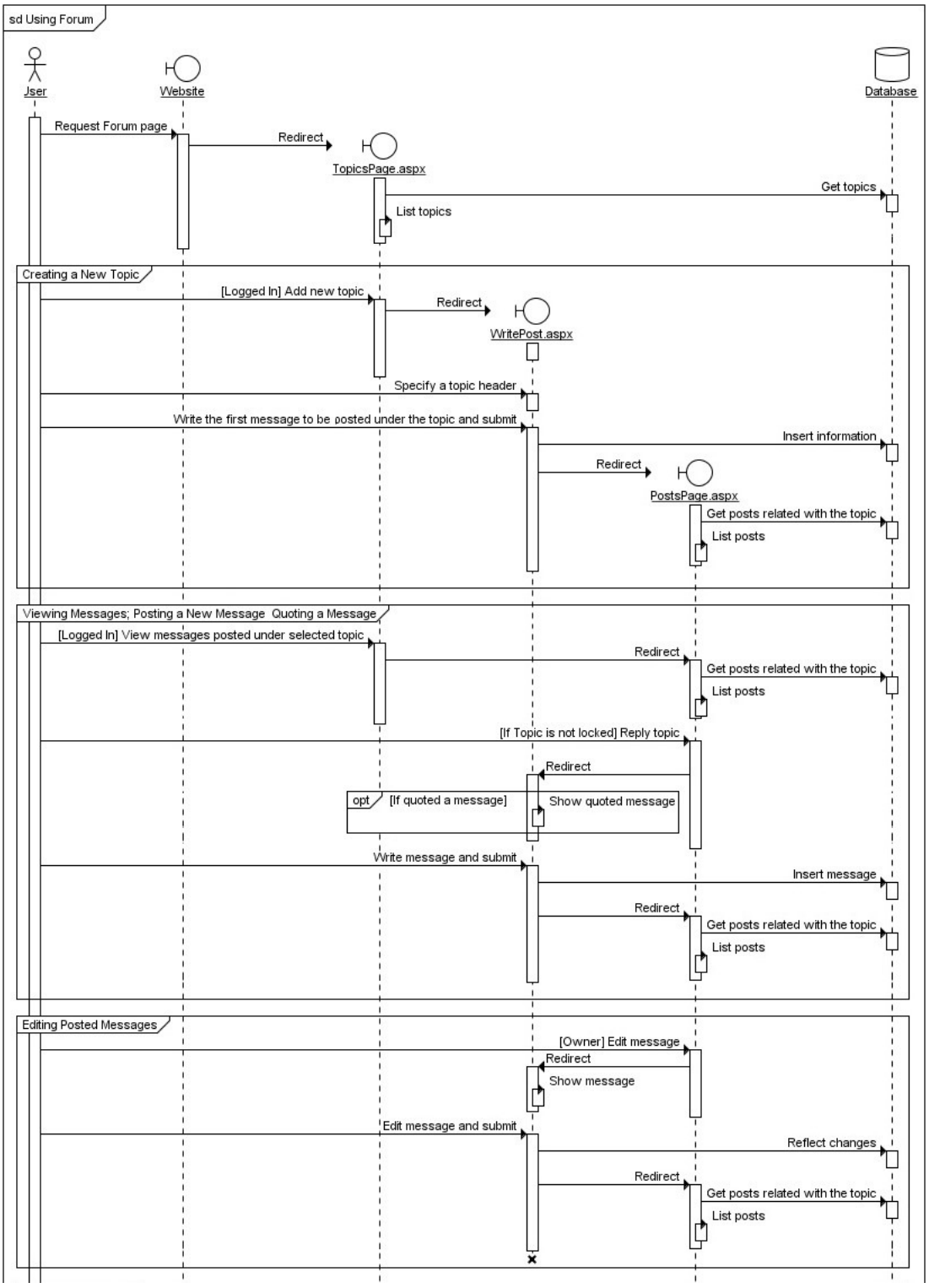
4.3.10 Using Integrated Forum

The forum is the only place that provides communication between website members. It consist of 3 pages: TopicsPage, WritePost and PostsPage. The sequence diagram of managing forum functionalities is presented in Figure 4.12. TopicsPage lists the topics of the forum to the user where PostsPage lists the messages posted under a selected topic. WritePost page allows users to specify the name of a topic on creation step and provides a textbox

for users where they can use when entering the first message of a topic or replying to a previously created topic.

A user can edit the name of a topic which is created by the him or he can delete those topics. Users can also edit or delete the messages that they have posted earlier. Moreover, users can quote previously posted messages when replying under a topic.

Other than creating, editing and deleting topics and messages, a user can lock or unlock a topic which is created by him. No message can be posted under locked topics. Another modifier of a topic is its pinned/unpinned status which can only be set by website moderators or administrators. Pinned topics are always listed on top in TopicsPage.


sd Using Forum
Jser
Website
Database
Request Forum page
Redirect
TopicsPage.aspx
Get topics
List topics
Creating a New Topic
[Logged In] Add new topic
Redirect
WritePost.aspx
Specify a topic header
Write the first message to be posted under the topic and submit
Insert information
Redirect
PostsPage.aspx
Get posts related with the topic
List posts
Viewing Messages; Posting a New Message Quoting a Message
[Logged In] View messages posted under selected topic
Redirect
Get posts related with the topic
List posts
[If Topic is not locked] Reply topic
Redirect
opt
[If quoted a message]
Show quoted message
Write message and submit
Insert message
Redirect
Get posts related with the topic
List posts
Editing Posted Messages
[Owner] Edit message
Redirect
Show message
Edit message and submit
Reflect changes
Redirect
Get posts related with the topic
List posts

(continues..)

(..started in previous page)

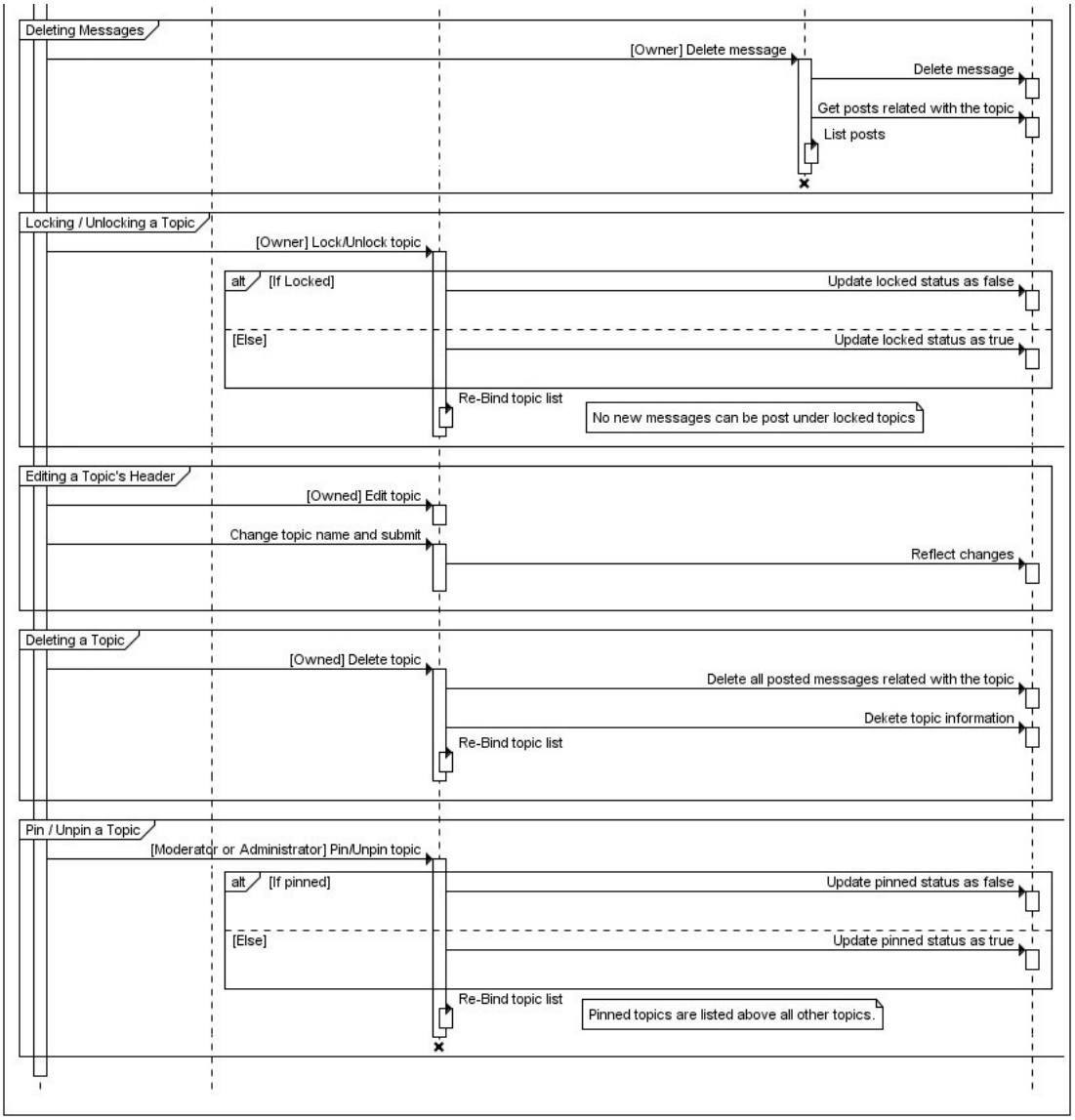


Figure 4.12, Sequence Diagram showing how to use integrated forum

4.3.11 Password Management

There is always a possibility that a user may forget his password or may want to change it with a new one. The process of retrieving a password or changing it is diagrammatized in Figure 4.13.

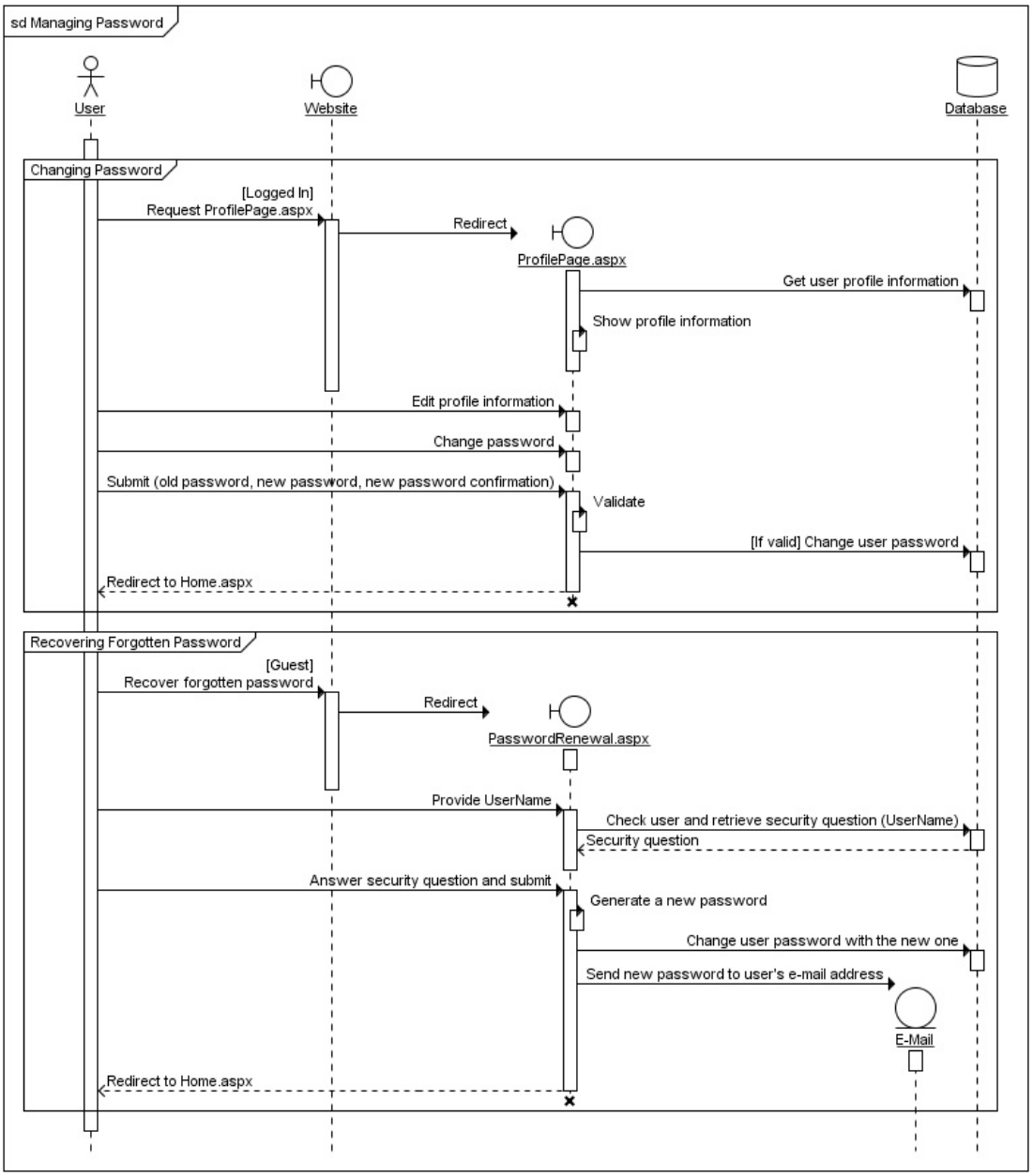


Figure 4.13, Sequence Diagram for managing password

Changing a password is handled in the profile page of a user. The user can change his password by entering his old password and specifying the new one. If the new password and its confirmation match, website approves the

new password and completes the process by reflecting changes to the "Membership Database".

On the other hand, renewing a forgotten password is handled in PasswordRenewal page. In this page, user is asked to answer his security question. If the answer is correct, then a new password is generated by the application and sent to user's e-mail address. User may later change this randomly generated password by following the changing password steps.

In the following chapter, a test scenario is followed step by step together with website screenshots in order to explain how to perform most of the tasks that are shown in the Use Case Diagram in Figure 4.1.

CHAPTER 5

TEST SCENARIO

5.1 Overview

This chapter introduces the user interface of the application and tries to explain how to use the main functionalities of the website with step by step screenshots.

Instruction starts with creating a new user account. Then an appropriate role is given to the user to enable project and specimen creation for the newly created user account. Next, a new member group is created and some of the previously created user accounts are added to this member group. After that, by using the new member account, a new project is created. Also, some documents related to the project is uploaded to the website. Later, a new specimen is created. Then, a new experiment which uses this specimen is added under previously created project by uploading the test data as an Excel document. After successfully creating the experiment, signal information of the experiment is checked and test data is plotted on a scatter graph. Then, two other experiments are created in order to show that various types of experiments can be uploaded to the application. Next, search functionality and forum is introduced. Also, how to change a user's password and how to retrieve a forgotten password is explained.

5.2 Preparations for Project Creation

This section demonstrates the tasks that are needed before starting project creation. In order to use most of the website functionalities a user must first register to the website. Moreoever, to create a project, a specimen or an experiment, he must have given the "HigherRankedRole" role type or any other more privileged role type. Also, in order to provide data security and reliability, a new member group can be created and some members may included in it.

5.2.1 Creating A New User Account

When a user opens the website for the first time, website cannot identify the user therefore the login control is initially in its anonymous state as shown in the Figure 5.1. In order to create a new account, user must click on the "New Account" button which is also highlighted in the Figure 5.1.

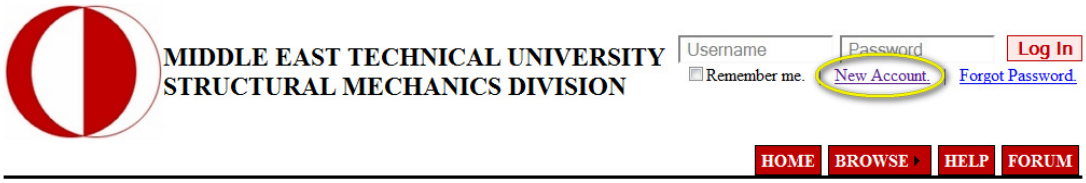


Figure 5.1, Anonymous state of the website

After clicking this button, user is redirected to CreateAccount.aspx page which is shown in Figure 5.2. In this page, user specifies his username and password and he is asked to fill his profile information.

As shown in Figure 5.2, a new test user account is created by filling the necessary information and clicking the "Create User" button. At this point application validates the page if the required information is filled correctly. Entering a valid e-mail address is crucial because a verification mail is sent to the provided e-mail before activating the user's membership. After the user verifies his e-mail address by clicking the link provided in the e-mail message, membership status of the user becomes approved and the *User* role is given to him. Alternatively, website administrators can also activate a user's membership.

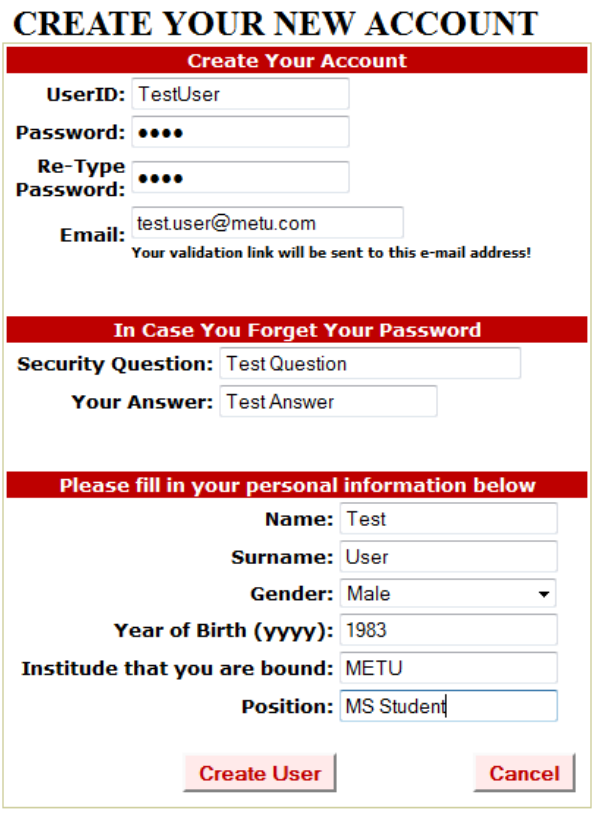


Figure 5.2, CreateAccount.aspx

From now on, user can login to the website by entering his username and password and clicking the "Log In" button as shown in Figure 5.3.

Figure 5.3, Logging In with specified username and password

After successful logging in with the newly created user, the state of login control is changed to logged in state and a welcome message is shown to the user in order to tell the user that he is authenticated. Also, a new navigation bar appears on the left side as shown in the Figure 5.4, which enables user to see or edit his profile information, to list all the members and groups of the website. If the user role is *Admin*, an "Administrative Controls" tab appears next to the "Group List" tab.

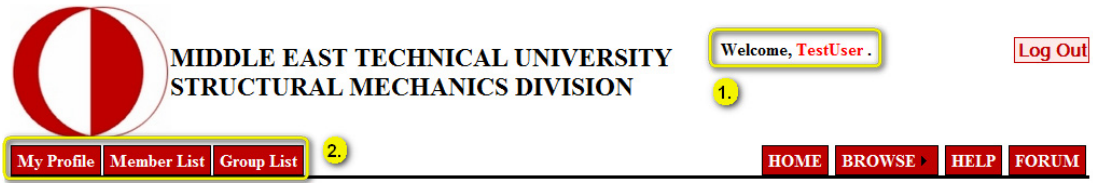


Figure 5.4, Logged In state of the web site

5.2.2 Using Administrative Controls To Change User Role

In order to create a project or specimen, the newly created user must have given a higher role than *User* which is the default role type given to a newly

created user as it is explained in Chapter 4. The Figure 5.5 shows assigning of a *Moderator* role to the newly created TestUser account.

Only users with *Admin* role can change other account's roles. Therefore, in order to change TestUser's role to *HigherRankedUser,* one of the website administrators logs in to system and changes the role of the TestUser account as can be seen in Figure 5.5.

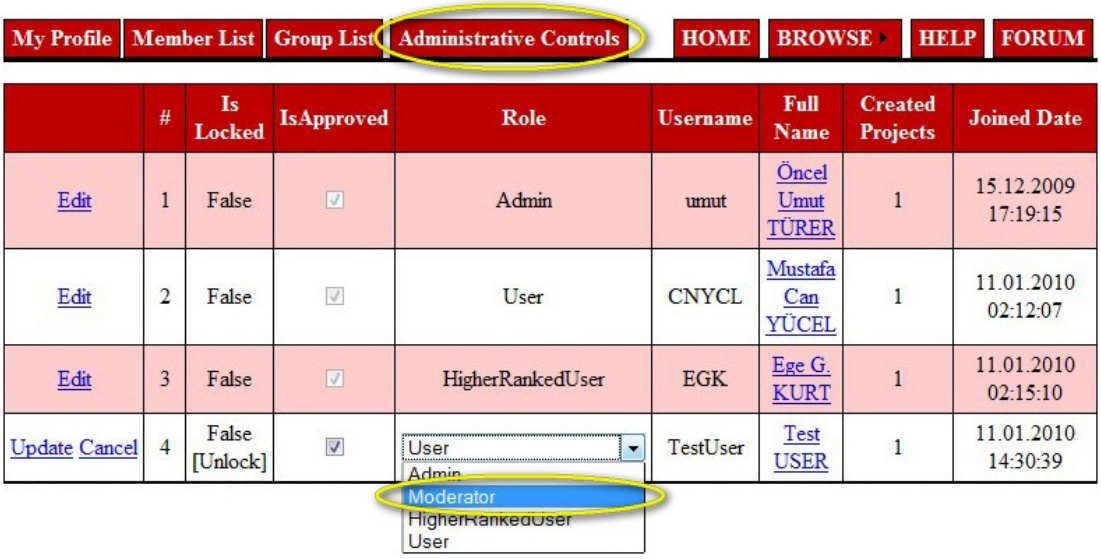


Figure 5.5, AdminControlsPage.aspx

5.2.3 Creating A New Member Group and Adding Members

Since the TestUser account is created and a new role which enables him to create projects and specimens is given to that account, how to to create a new member group can be explained as a next step. Member groups are useful when assigning project and specimen ownership and visibility privileges to more than one member. Also, newly created members can

easily gain same privileges of a group by joining it. This prevents modification of ownership and visibility settings of each project and specimen for every new user.

In order to create a new group, user must click the "Group List" navigation tab which redirects the user to GroupList.aspx page. In this page previously created member groups and a brief information about them is listed as shown in Figure 5.6.

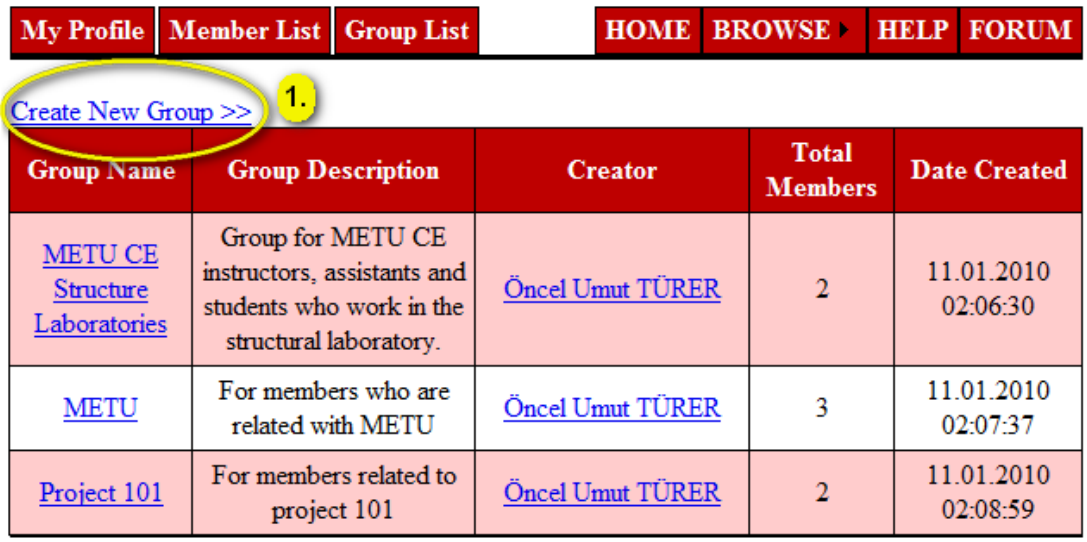


Figure 5.6, GroupList.aspx

Clicking the "Create New Group" button changes the content of the page and enables user to specify the group name and its description as shown in Figure 5.7.

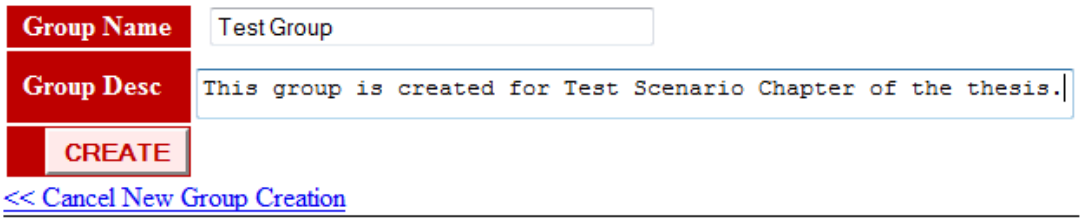


Figure 5.7, Creating new member group

Filling the necessary information and sending the information by clicking the "Create" button, inserts the member group information into the database. After the creation of the new member group completed, the website again lists the member groups in the system. Figure 5.8 shows the group list including the newly created member group.

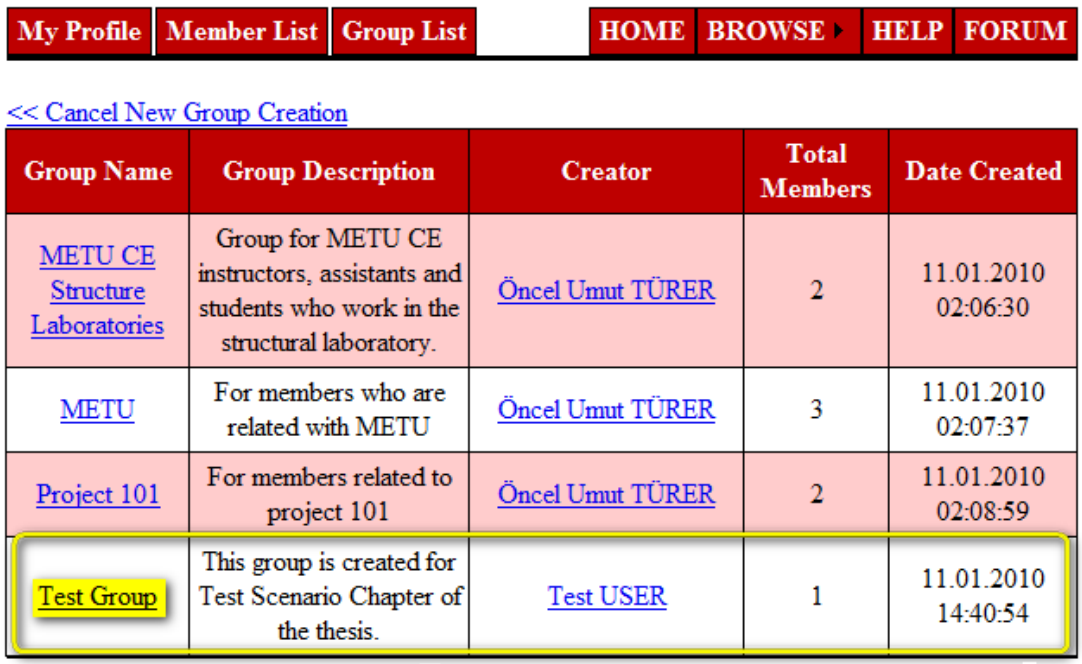


Figure 5.8, Group list showing the newly created member group

Clicking the name of the newly created member group redirects the user to GroupInfoPage.aspx where he can see or edit the information about that group. Also, a list of group members is displayed at the bottom of the page. Here group owners can add new members to the group by clicking the "Add New Group Member" button or they can remove existing members from the group. The detailed information about newly created "Test Group" is presented in Figure 5.9.

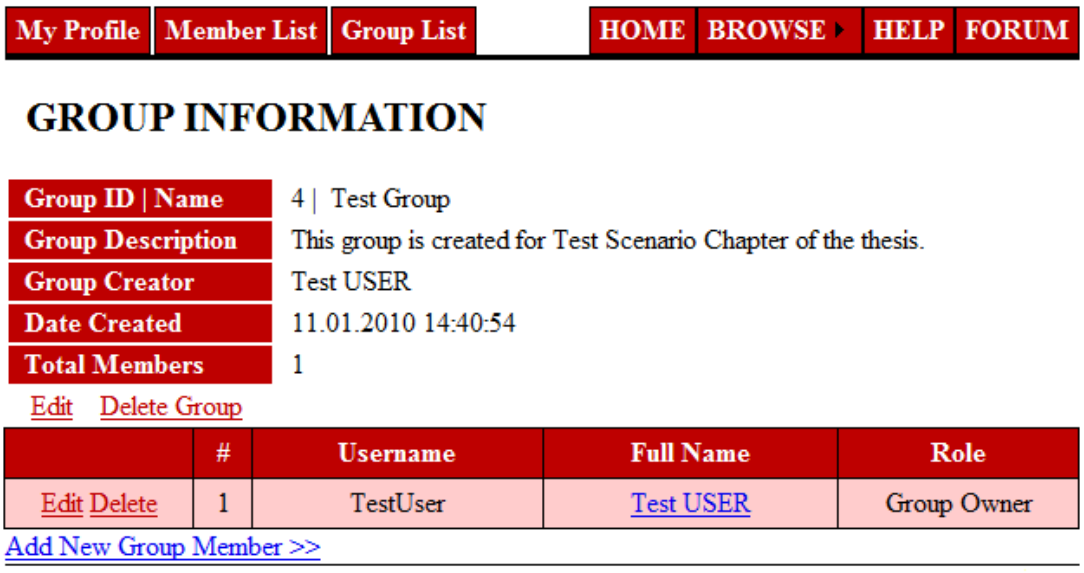


Figure 5.9, GroupInfoPage.aspx

After clicking the "Add New Group Member" button, new controls appear on the page where the user can select desired website members by using the dropdown control as shown in Figure 5.10. Also, user can specify the group role of the selected member before adding the member to the group. There are only two group roles. A group member can either be a "Group Member"

or he can be a "Group Owner" and only group owners can edit group information or add new members to a group.

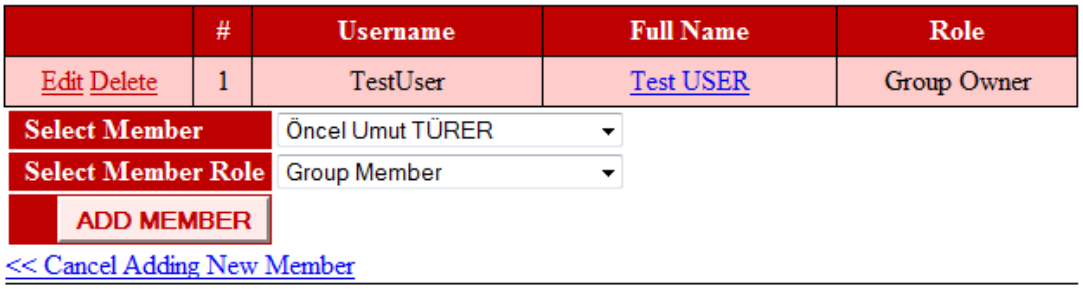


Figure 5.10, Adding new members to the group

Two new members were added to the test group as can be seen in Figure 5.11.

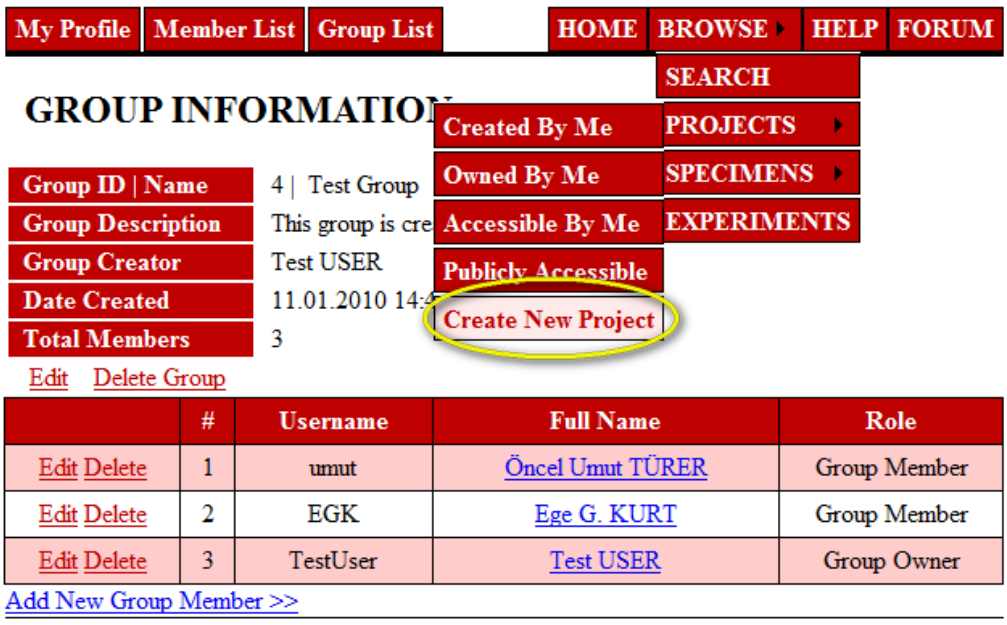


Figure 5.11, New group members and "Create New Project" menu item

Since all the preparations for project creation has completed, the test user can create a new project by clicking the "Create New Project" navigation menu item from Browse tab of right navigation bar. The "Create New Project" menu item is highlighted in Figure 5.11, and clicking this menu item redirects user to the CreateProject.aspx page of the website.

5.3 Pseudo-Dynamic Test Data Scenario

In this scenario, how to upload the test results of multiple PsD experiments which were conducted on the same specimen is demonstrated. The experiments were conducted in the Structural and Earthquake Engineering Laboratories of METU Civil Engineering Department. In order to upload the test data, a project was defined first. Next, the specimen on which the experiments conducted was created. Then, two different experiments are created and the test data related to those experiments were uploaded to the application database. Finally, how to view/modify the signal information and how to plot an experiment data as a scatter graph is demonstrated.

5.3.1 Creating The Project

After clicking on the "Create New Project" menu item which is highlighted in Figure 5.11, website redirects the user to CreateProject.aspx which enables project creation. The CreateProject.aspx page is shown in Figure 5.12.

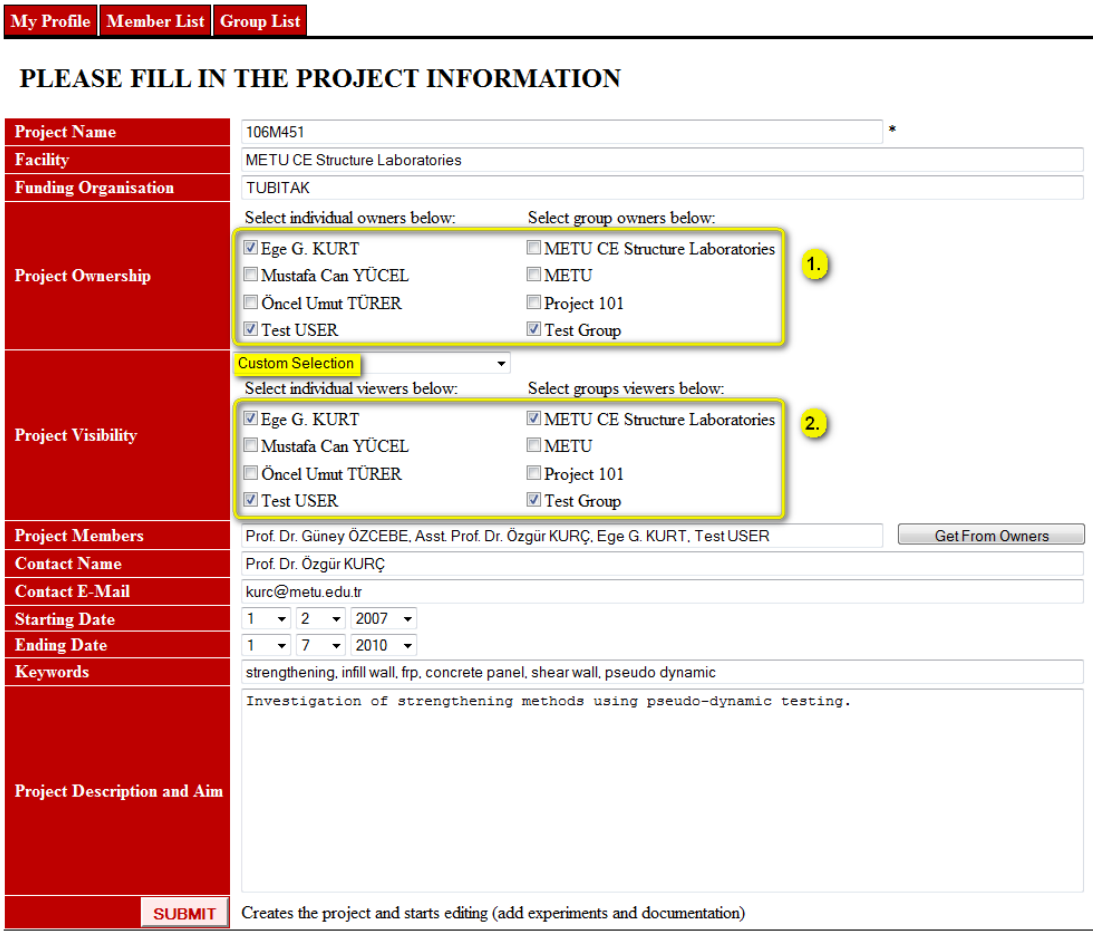


Figure 5.12, CreateProject.aspx

After reaching CreateProject.aspx, user is asked to fill some information related to the project. Also, user can select members and member groups as the project owners, and he can specify the visibility restriction level of the project. In Figure 5.12, visibility restriction is set to "Custom Selection" in order to show the extended controls of for the visibility adjustment of the project. When setting ownership and visibility of the project, it is not necessary to select a member if his group is already selected. However, selecting both the member and his group guarantees that the member can see or modify the project even though he is removed from the group.

Clicking "Submit" button completes the project creation process and redirects the user to ProjectDetails.aspx where the user can view, edit or delete the information about the newly created project. As shown in Figure 5.13, there is a tree-view navigation menu on the left of the page which lists the projects in the system that are visible to the user. By using this menu, user can navigate through projects easily. In this page, there is also another navigation bar just over the project details. This navigation bar has three menu items: Information, Experiment and Documents. The "Information" shows project details to the user where "Experiments" and "Documents" are used for listing experiments and documents related to the project respectively.

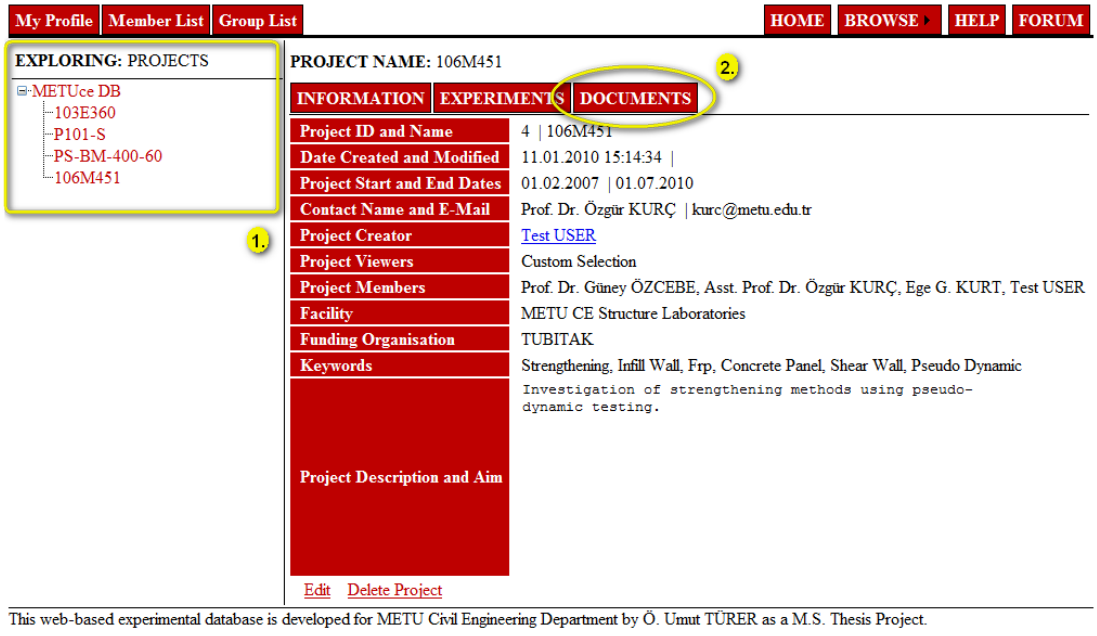


Figure 5.13, PojectDetails.aspx

5.3.2 Adding Documents Related To The Project

In order to add documentation to a project, user must click the "Documents" menu item in ProjectDetails page as highlighted in Figure 5.13. By doing so,

user is redirected to DocumentList.aspx and the documents related with the project are listed just below the navigation menu as shown in Figure 5.14.

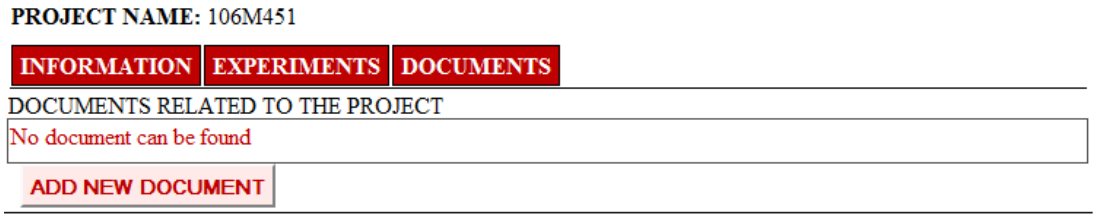


Figure 5.14, DocumentList.aspx

Since the project is just created, no related document can be seen in Figure 5.14 at this moment. By clicking the "Add New Document" button, user can specify a file to be uploaded to the website as shown in Figure 5.15. Also, user can comment on the document explaining the purpose of the file. Currently, the maximum permitted file size is limited to 100 MB. But, any file whose size is smaller than this limit can be uploaded to the website.

Figure 5.15, Uploading project document

Two files, the project contract and the project budget are uploaded to the website as example project documentations. After completing the upload process, new documents are now listed in the page as shown in Figure 5.16.

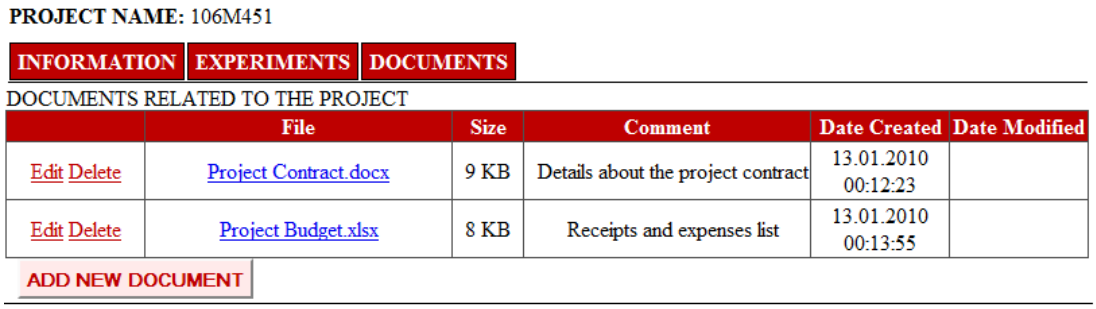


Figure 5.16, List of documents for created project

From now on, users can download these documents to their computers by clicking the name of the listed files. Other than downloading these files, project owners can also delete them or they can edit the documentation comments.

5.3.3 Creating The Specimen

During the experiment creation process, users are asked to select the specimen used in the experiment among the specimens found in the database. However, if the specimen used in an experiment haven't been created in the database, user needs to create the specimen information first.

In order to create a specimen, user needs to reach CreateSpecimen.aspx page. This process is similar to reaching the project creation page and the

quickest way to reach CreateSpecimen.aspx is by using the "Browse" tab of top-right navigation bar of the website as shown in Figure 5.17.

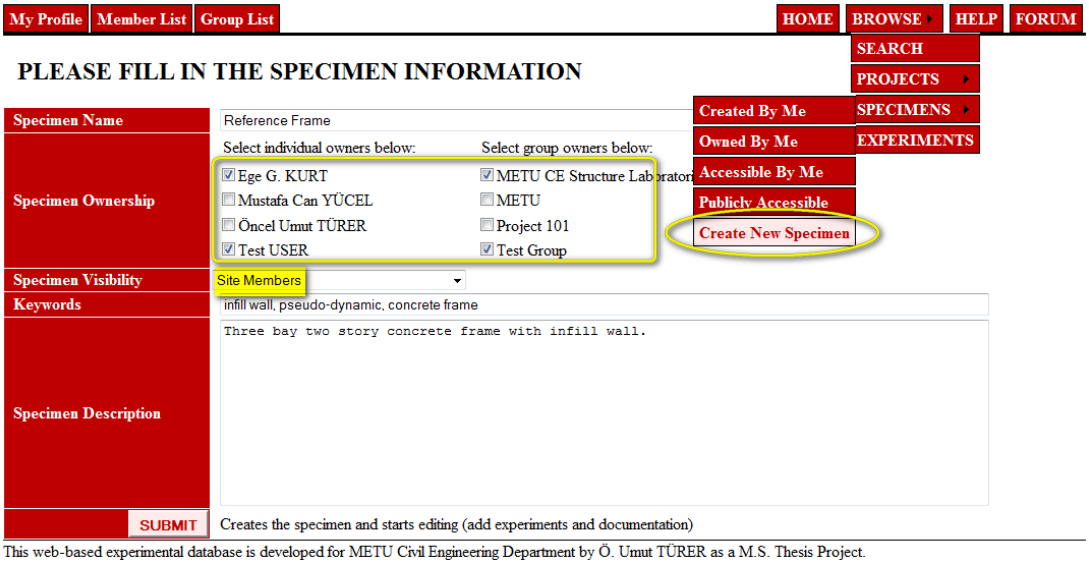


Figure 5.17, CreateSpecimen.aspx

In CreateSpecimen.aspx user is asked to fill some information about the specimen. Also, he needs to select specimen owners and visibility restriction level just like the project creation phase. As it can be seen from Figure 5.17, the specimen which is a three bay, two storey, R/C frame with infill walls was created. Two users and two member groups are selected as specimen owners and specimen visibility restriction is set to "Site Members" which means every member of the website can view the information of this specimen.

Clicking the "Submit" button completes the creation process of the specimen and the user is redirected to SpecimenDetails.aspx as shown in Figure 5.18. Similar to ProjectDetails.aspx page, this page also contains a tree-view navigation menu which lists all of the specimens that are visible to the user

for easy navigation. There is also a navigation bar which lets users to quickly access projects, experiments and documents related to a selected specimen.

By following the "Documents" tab of this navigation menu which is highlighted in Figure 5.18, a user can upload documentation related to the specimen. Any file whose size is smaller than 100 MB can be uploaded as a documentation such as the pictures of the test setup or the CAD drawings related to it. However, in this section how to upload specimen documentation is not explained because this process is exactly same as the document uploading process of project creation which is explained in Section 5.3.2.

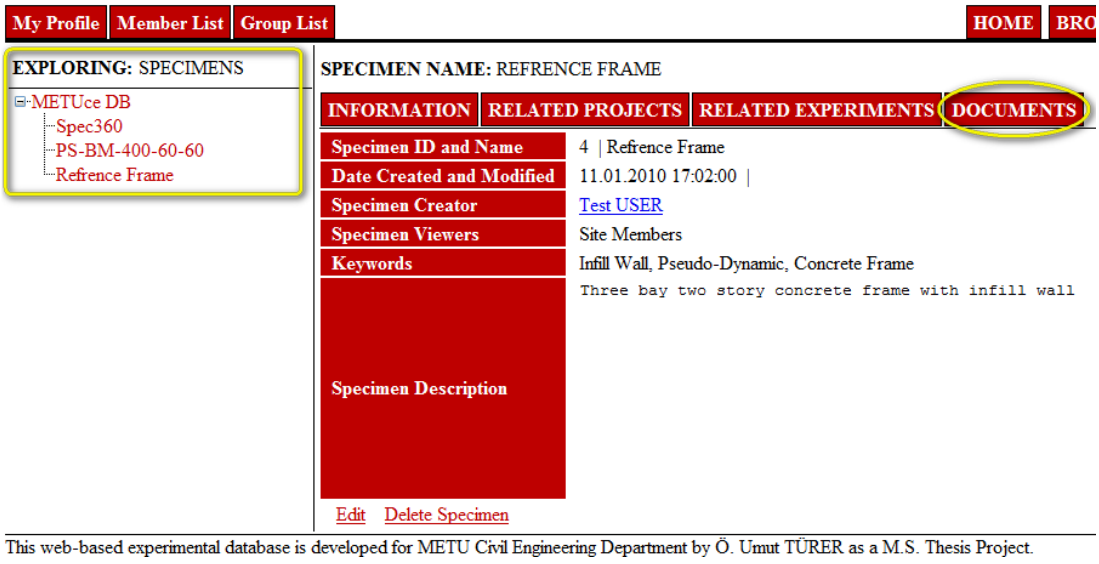


Figure 5.18, SpecimenDetails.aspx

5.3.4 Creating The First Experiment

In the first experiment, the three bay, two storey R/C frame with infill walls was subjected 50% of 1999 Duzce earthquake loads by using pseudo-dynamic testing techniques.

Since the project and the specimen is created, user only needs to create the experiment and upload the test data. Each experiment must be created under a project. Therefore, user needs to return the project details page of previously created project by using either "Projects" sub menu of "Browse" tab of top-right navigation bar or using the search functionality of the application. Search functionality of the website is explained in Section 5.5.

After reaching the ProjectDetails.aspx, user must click on the "Experiments" tab of navigation bar as highlighted in Figure 5.19 in order to reach ExperimentList.aspx which enables users to create a new experiment. Since the project has just been created, no experiment is listed in the presented ExperimentList.aspx. To add a new experiment, user needs to click on the "Add New Experiment" button which is also highlighted in Figure 5.19.

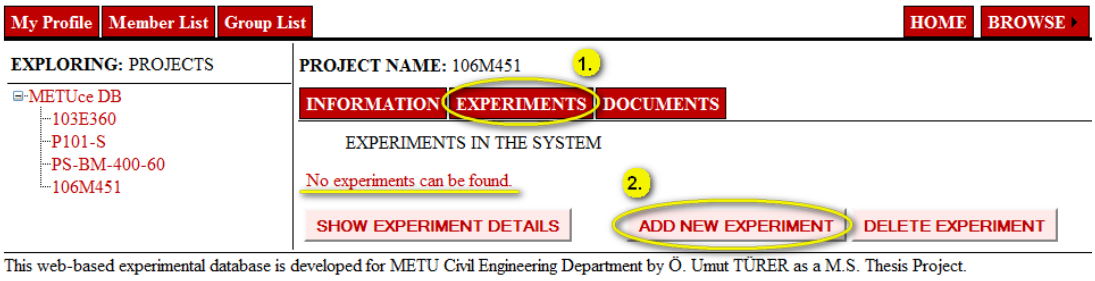


Figure 5.19, ExperimentList.aspx

Clicking the highlighted "Add New Experiment" button redirects users to CreateExperiment.aspx which can be seen in Figure 5.20. In this page, user is asked to fill some information about the experiment. Also, he is asked to select the specimen used in the experiment by using the dropdown control which is highlighted in Figure 5.20 and contains all the previously created specimen names. If the specimen used in the experiment haven't created

yet, user can create the experiment by clicking the "Create New Specimen" button near the drop down list.

After filling the general information about the experiment and selecting the specimen used in the experiment, user specifies the Excel file containing the experiment data by clicking the "Browse" button. 100 MB file size limit is also valid for Excel files. In order to upload specified file to the application server, the user must click on the "Analyse File" button.

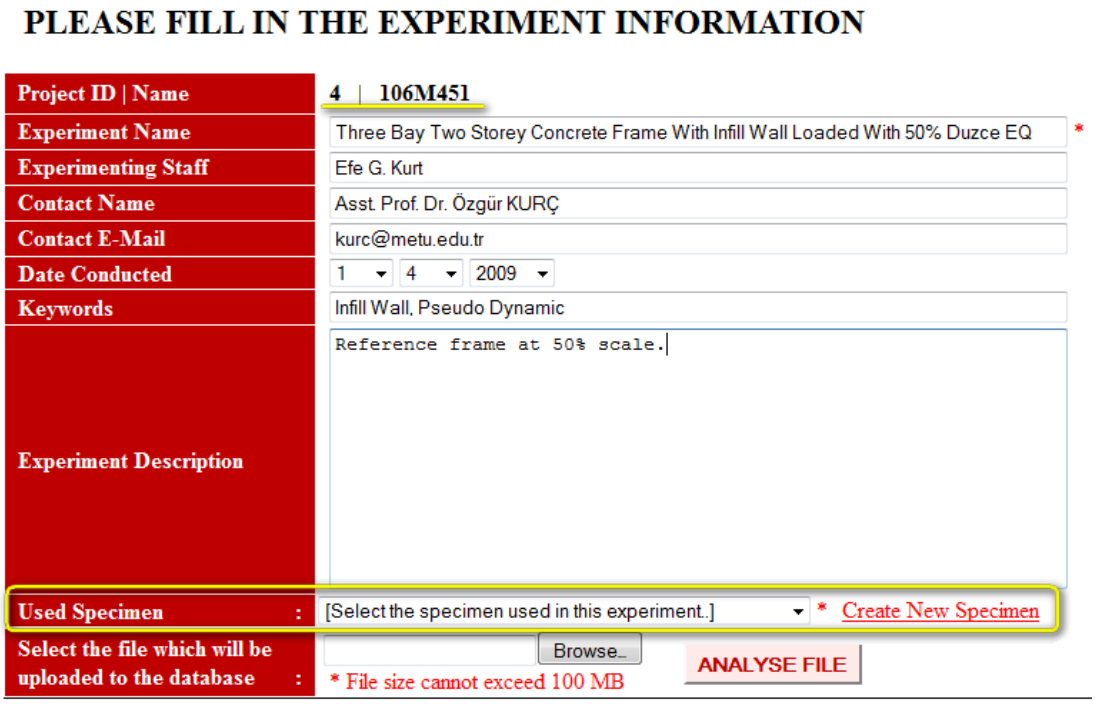


Figure 5.20, CreateExperiment.aspx

Upon completing the upload, application analyses the spreadsheets of Excel file and collects the names of the spreadsheets which contain data. Then the

names of those spreadsheets are bind in a dropdown control where user can specify the name of the spreadsheet which contains the desired test data.

As it can be seen from Figure 5.21, previously created specimen is selected as the experiment specimen, "Reference_All.xls" is selected as the Excel file and the spreadsheet named "50" is selected as the Excel spreadsheet which contains the desired test data. If user wants to change the uploaded file, he can click on the "Change File" button and specify the new Excel file.

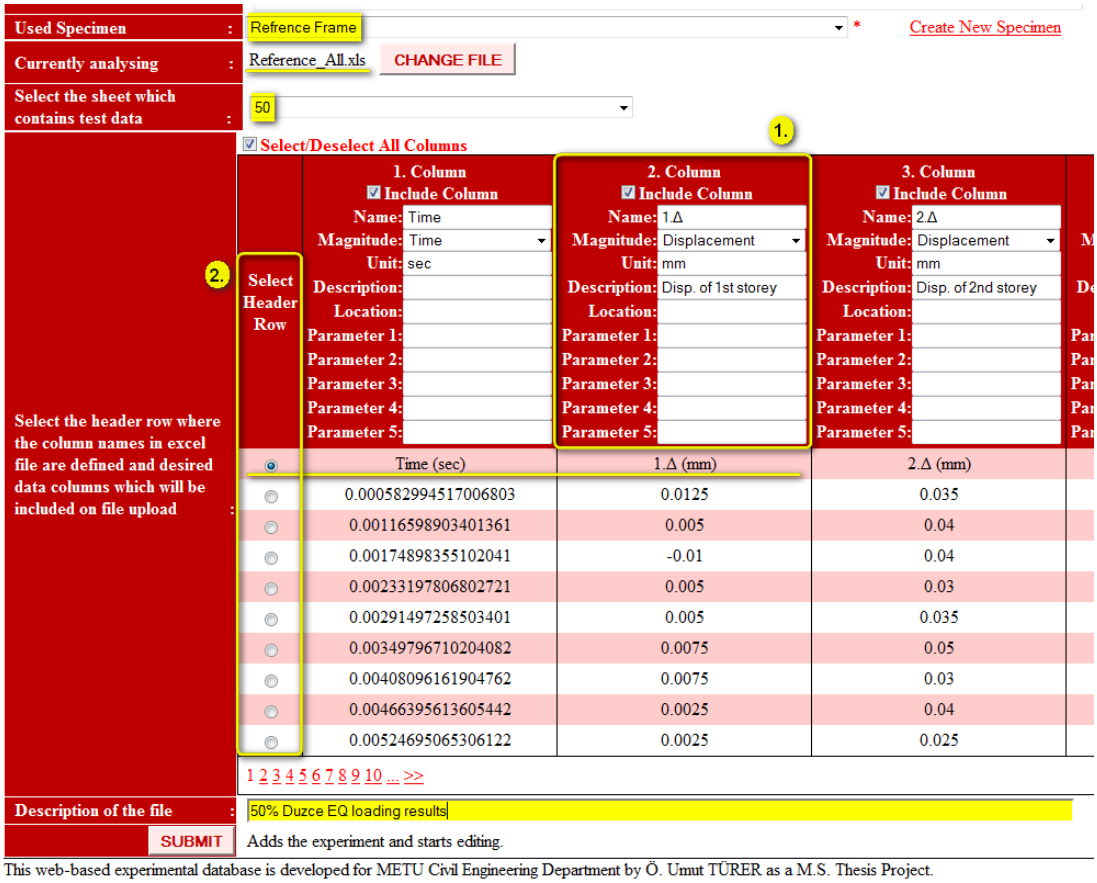


Figure 5.21, Uploading test data Excel file

After user selects the name of the spreadsheet, application analyses the content of the spreadsheet and presents the data readings to the user as a table. Also, a field which is used to collect the signal information for each reading is added over each table column as shown with a yellow rectangle labeled as "1." in Figure 5.21. By using these fields, user can select which columns to be inserted into database, also, they can specify the name, unit, magnitude, description and sensor location of signal together with five other custom parameter.

Before uploading the test data, user must select the heading row of the signal readings by using the radio buttons as highlighted with a yellow rectangle which is labeled as "2." in Figure 5.21. A heading row is an excel spreadsheet row which is found just above the first row where the signal readings starts. The heading row commonly contains information about the signal readings but it is not necessary. Also, by using the yellow textbox control, users can comment on the selected Excel file.

Clicking the "Submit" button, completes the experiment creation process and the user is redirected to ExperimentDetails.aspx which is presented in Figure 5.22. ExperimentDetails.aspx presents information about the experiment and lists the plotable Excel files related to the experiment. By using the "Add Another Plotable Data File" button, user can add additional Excel files related to the experiment such as modified or calibrated versions of the previously uploaded test data. Users can download these files by clicking on the file name as highlighted in Figure 5.22.

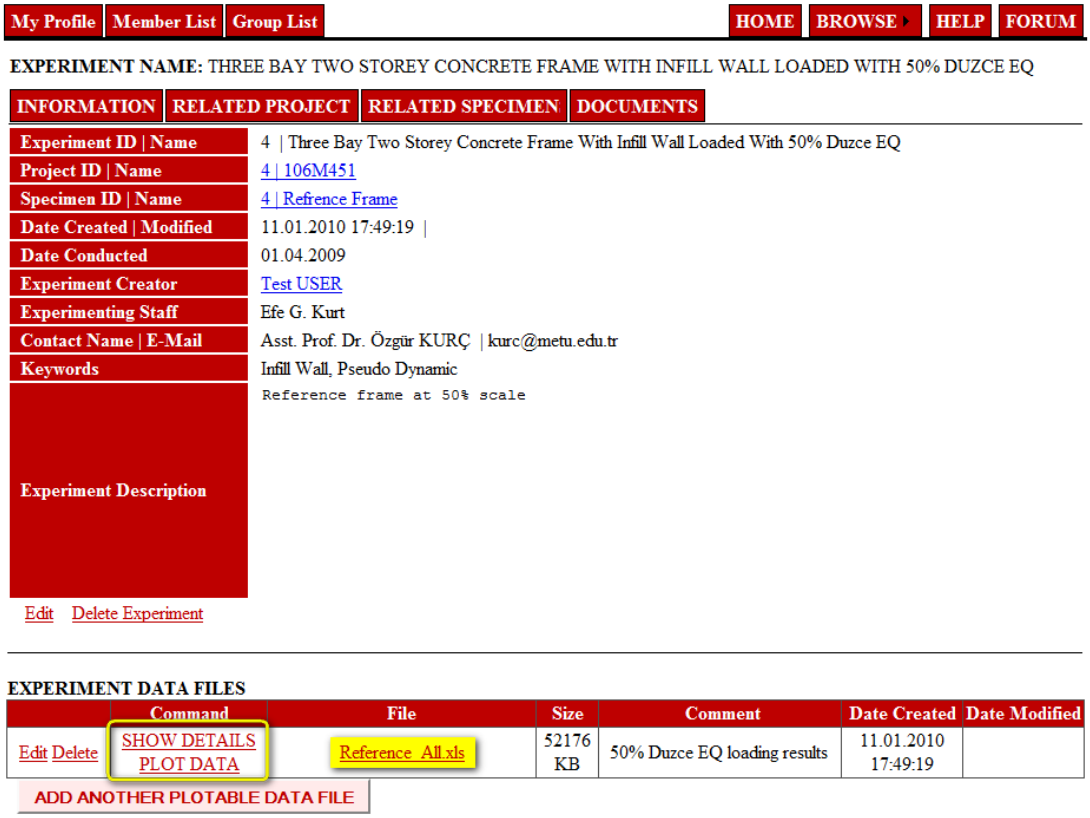


Figure 5.22, ExperimentDetails.aspx

5.3.5 Viewing Signal Information and Plotting Experiment Data

As the experiment created successfully, users are now able to view the signal information related to the test data and they can plot the experiment results by using the "Show Details" and "Plot Data" buttons which are framed with a yellow rectangle in the Figure 5.22.

When clicked on "Show Details" button in ExperimentDetails.aspx, a signal information table related with the selected experiment is presented to the user as shown in Figure 5.23.

EXPERIMENT DATA FILES

	Command	File	Size	Comme
Edit Delete	SHOW DETAILS PLOT DATA	Reference_All.xls	52176 KB	50% Duzce EQ loading

ADD ANOTHER PLOTABLE DATA FILE

EXPERIMENT SIGNAL INFORMATION

	Signal ID	Name	Magnitude	Unit	Description	Location	Parameter1
Edit	1	Time	Time	sec			
Edit	2	1.Δ	Displacement	mm	Disp. of 1st storey		
Edit	3	2.Δ	Displacement	mm	Disp. of 2nd storey		
Edit	4	1F	Force	kN	Force effecting on 1st floor		
Edit	5	2F	Force	kN	Force effecting on 2nd floor		
Edit	6	Baseshear	Force	kN	Baseshear force		
Edit	7	Drift 1					
Edit	8	Drift 2					
Edit	9	M1	Moment	kN.m	Moment at node 1		
Edit	10	M4	Moment	kN.m	Moment at node 4		
Edit	11	V1	Force	kN	Shear at node 1		
Edit	12	V4	Force	kN	Shear at node 4		
Edit	13	N1	Force	kN	Axial force at node 1		
Edit	14	N4	Force	kN	Axial force at node 4		
Edit	15	Φ1	Rotation	rad/km	Rotation at node 1		
Edit	16	Φ4	Rotation	rad/km	Rotation at node 4		
Edit	17	Φ2	Rotation	rad/km	Rotation at node 2		
Edit	18	Φ3	Rotation	rad/km	Rotation at node 3		
Edit	19	γ					

Figure 5.23, Signal information related with an experiment data

This table has a 12 columns and varying amount of rows changing according to the number of Excel spreadsheet columns that are inserted into the database. First column is for edit button which enables project owners to edit desired signal information at any time. Second one is for "Signal ID" which represents the column number in the Excel spreadsheet of related data readings. Other columns of signal information table represents the name

given to the signal reading, its magnitude, its unit, its description, and the location of the sensor respectively. There are also 5 more columns named from "Parameter1" to "Parameter5" which can be used when an extra information is needed. The information presented in this signal information table is exactly same as the information filled in Figure 5.21.

If a user wants to plot the experiment results in a scatter graph he should click on the "Plot data" button which is shown in Figure 5.22. By doing so, user is redirected to Plotgraph.aspx as shown in Figure 5.24.

When the page loads, it contains only the "Back" and "Draw Graph" buttons together with two dropdown controls. If a user clicks on the "Enable Summation of Multiple Readings" button, four new dropdown controls, which framed with the yellow rectangle labeled as "1." in Figure 5.24, appears on the web page. These dropdown controls list all the names of data readings that are inserted into the database. These names are same as the names of signal information framed with a yellow rectangle in Figure 5.23. From these dropdown controls, user selects desired data readings to be used as X and Y values of the scatter graph. If user does not select any of the listed signal names for X or Y axes, then integers from 0 to infinity is assigned one by one to the values of related axis for every value in other axis. This helps user to visualize the readings of the selected data. If a user selects multiple X axis or Y axis readings, those readings are summed and the graph is drawn by using the summed values. The signal information table framed with a yellow rectangle labeled as "2." can only be seen if a single couple of X and Y readings is selected, and it shows the information about selected readings.

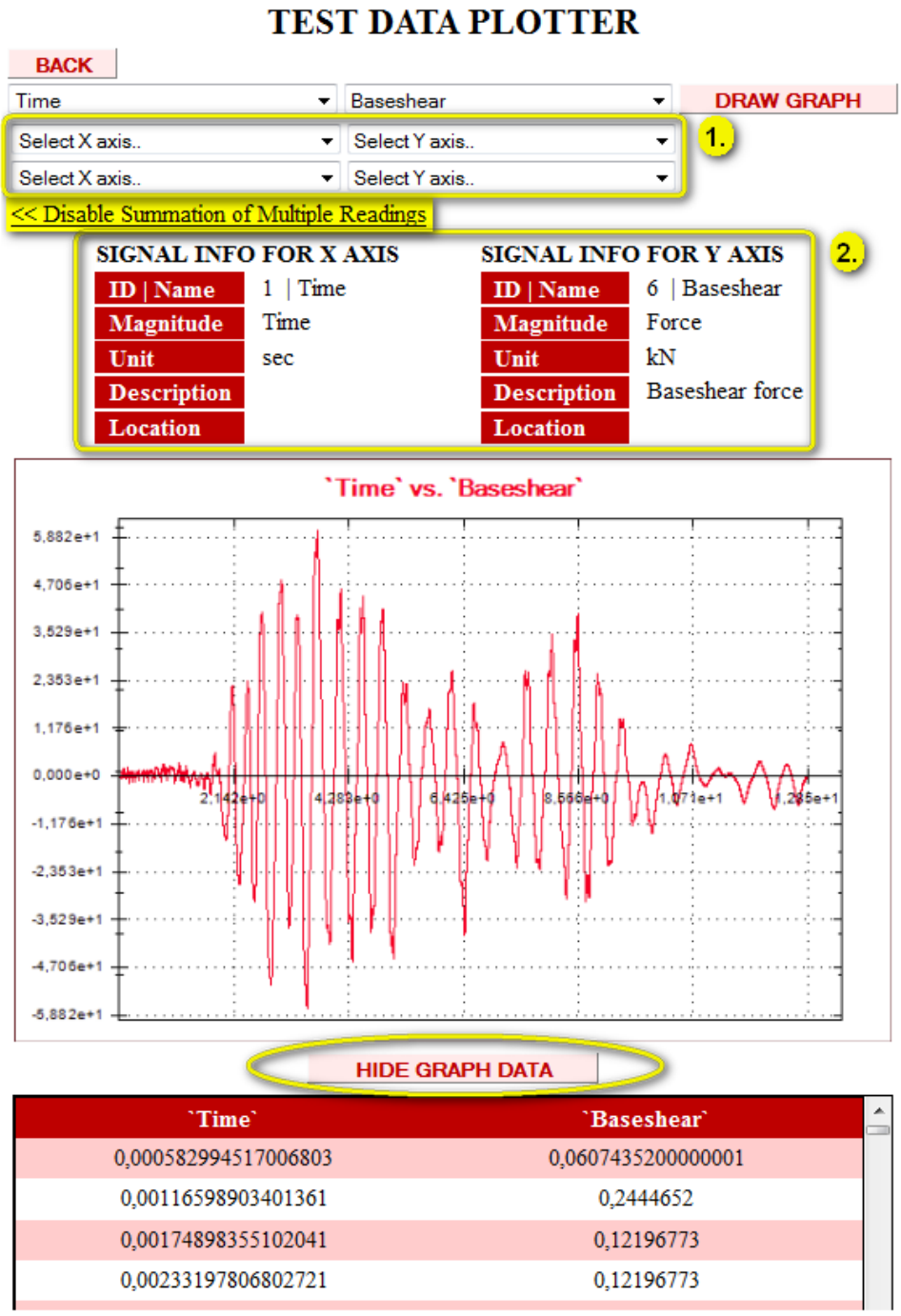


Figure 5.24, Plotgraph.aspx showing the test result of first experiment

On clicking "Draw Graph" button, a scatter graph is drawn by using selected signal readings as the values of X and Y axes and the graph is displayed as a PNG image on the page as shown in Figure 5.24. In the presented graph, elapsed time is selected as the values of X axis and the baseshear force affecting on the specimen is selected as the values of Y axis. Drawing the graph and converting it into a PNG image is handled by an open-source external class library called "ZedGraph". Upon clicking the "Show Graph Data" button in this page, a table which presents the values used for graph axes is presented to the user and the name of the button changes to "Hide Graph Data".

5.3.6 Creating The Second Experiment

The second experiment also performs a pseudo-dynamic testing on the same specimen which is used in the first experiment. In this experiment, the specimen was also subjected 50% of 1999 Duzce earthquake loads. However, this time, the infill walls were strengthened by using fiber reinforced polymers (FRP). The main purpose of uploading this second example is to show how the experiments are grouped according to the projects they belong and according to their specimens.

The process of creating the experiment and uploading the test data for this second experiment is exactly same as the first experiment. Even the project and specimen names for both experiments are same. Therefore, application automatically groups both experiments together under the name of same project and same specimen. In Figure 5.25, both first and second experiments are listed as the experiments of the project created in Section 5.3.1. Also, from Figure 5.26, it can be seen that both experiments are listed under the name of the same specimen. Finally, Figure 5.27 shows the graph of second experiment's test results. In the presented graph, elapsed time is

selected as the values of X axis and the tip deflection of the structure which is obtained by summing the deflections of first and second floors is selected as the values of Y axis.

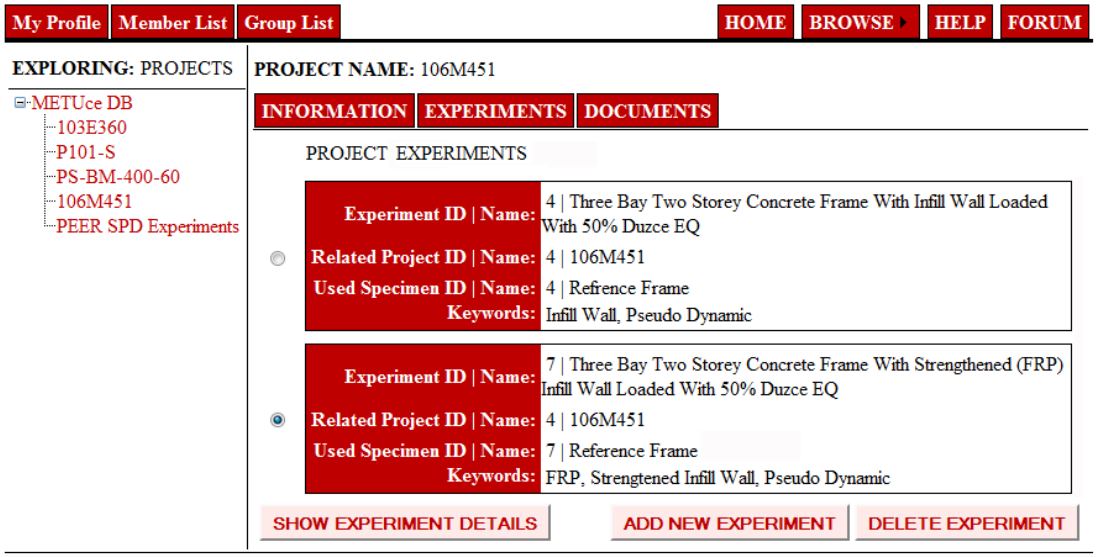


Figure 5.25, List of project experiments

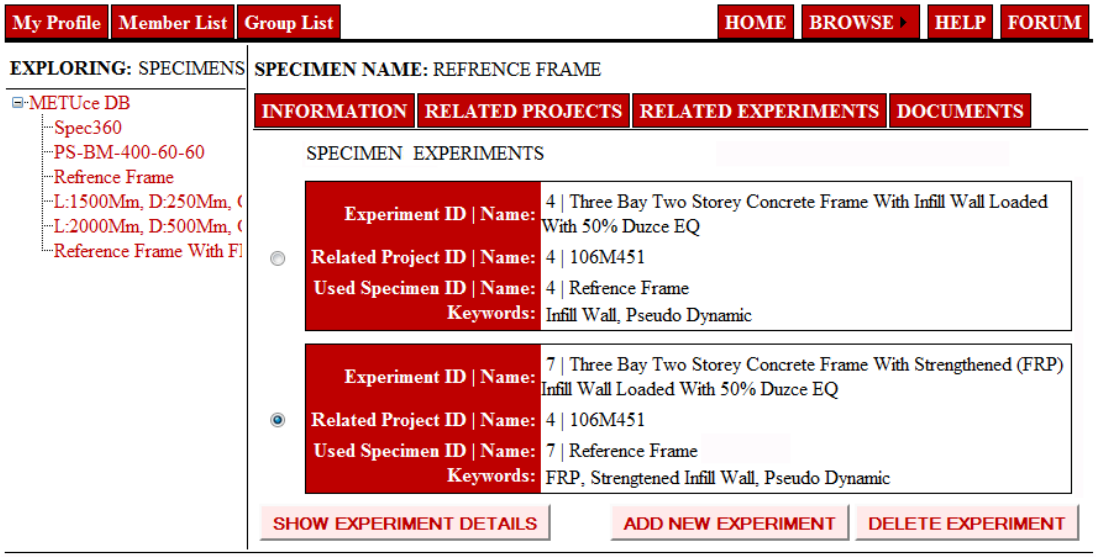


Figure 5.26, List of experiments that are conducted on a specific specimen

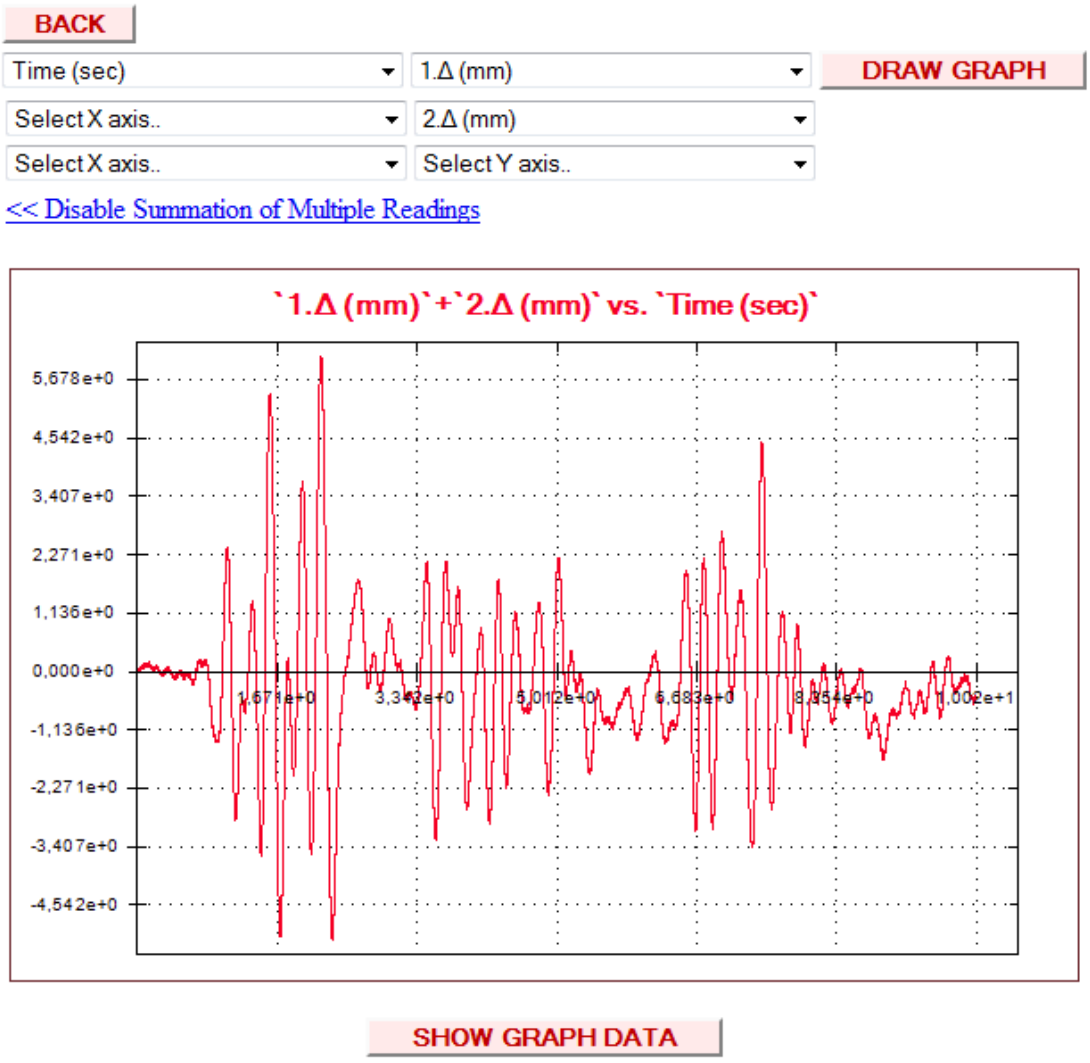


Figure 5.27, Graph of test result of the second experiment

5.4 Quasi-Static Test Data Scenario

As it is introduced in Chapter 1, there are several web-based experiment databases used in the world today, and one of them is PEER Structural Performance Database. In this section some quasi-static test results acquired from PEER SPD is uploaded to the application database in order to show that the application accepts test data from various types of experiments. Also,

being able to collect data from other databases can be accepted as the transition step of becoming a part of a distributed database system.

The collected data was stored as formatted text documents in PEER SPD. Therefore initially, the data downloaded and transferred into an Excel file format. Then a new project was created to upload PEER experiments. Next, specimens used in these experiments were created with the information taken from the PEER SPD website. Finally, the experiments were created by uploading the Excel files which contains the test results of PEER SPD.

Figure 5.28 show the details page of "Davey 1975, No.2" experiment which is one of the experiments taken from the PEER SPD, and Figure 5.29 presents the hysteresis graph drawn by using the test data of "Davey 1975, No.2" experiment.

EXPERIMENT NAME: DAVEY 1975, NO. 2

INFORMATION | RELATED PROJECTS | RELATED SPECIMENS | DOCUMENTS

Experiment ID | Name	6 | Davey 1975, No. 2
Project ID | Name	5 | PEER SPD Experiments
Specimen ID | Name	5 | L:2000Mm, D:500Mm, Octagonal Cantilever With Hammer Head
Date Created | Modified	19.03.2010 11:51:36 |
Date Conducted	
Experiment Creator	Öncel Umut TÜRER
Experimenting Staff	
Contact Name | E-Mail	|
Keywords	
Experiment Description	Material Properties --- Concrete Strength: 34.8 (MPa) Transverse Steel: Yield Stress: 312 (MPa) Longitudinal Steel: Yield Stress: 371 (MPa) Strength: 562 (MPa) Geometry --- Diameter: 500 (mm) Cross-Section: Octagonal Length: L-Inflection: 1,750 (mm) L-Measured: 2,000 (mm)

Edit Delete Experiment

EXPERIMENT DATA FILES

	Command	File	Size	
Edit Delete	SHOW DETAILS PLOT DATA	PEERspd - Davey 1975, No. 2.xlsx	29 KB	

ADD ANOTHER PLOTABLE DATA FILE

Figure 5.28, Information about the PEER SPD experiment named "Davey 1975, No.2"

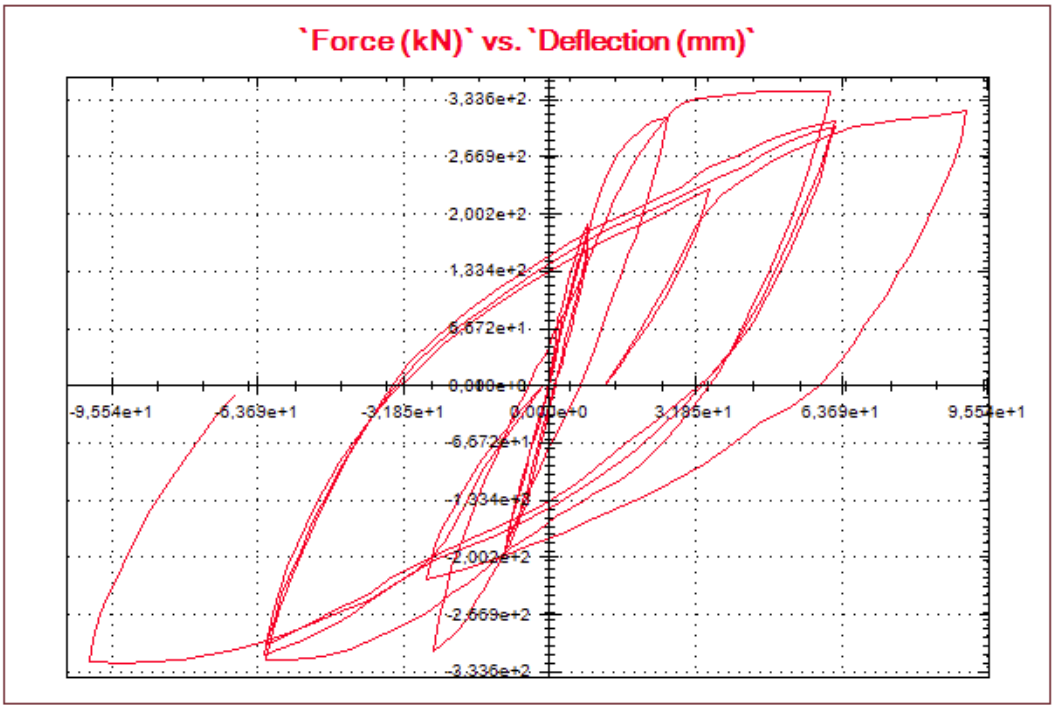
TEST DATA PLOTTER

BACK

Deflection (mm) | Force (kN) | DRAW GRAPH

Enable Summation of Multiple Readings >>

SIGNAL INFO FOR X AXIS		SIGNAL INFO FOR Y AXIS	
ID | Name	1 | Deflection (mm)	ID | Name	2 | Force (kN)
Magnitude	Displacement	Magnitude	Force
Unit	mm	Unit	kN
Description		Description	
Location		Location	



HIDE GRAPH DATA

`Deflection (mm)`	`Force (kN)`
0	0
0,33581	15,36
1,0075	38,4
1,3433	54,722

Figure 5.29, Plotting the hysteresis graph of PEER SPD experiment named "Davey 1975, No.2"

5.5 Using Search Functionality

A search functionality was integrated to the application to enable accessing the a specific project, specimen or experiment easily.

Searching functionality is handled in Search.aspx page that can be reached by clicking "Search" menu item under "Browse" tab of right navigation bar. At the top of this page there are some tips telling how to use wildcards or how to search for phrases as can be seen in Figure 5.30. Just below the tips, there is a textbox control and a dropdown control. Dropdown control has three items: Projects, specimens and experiments. These items determine where to search the word or phrase that is entered in the textbox. Initially search option is set to "Simple" which mean the specified word or phrase is searched in name, keyword and description fields of a project, specimen or experiment.

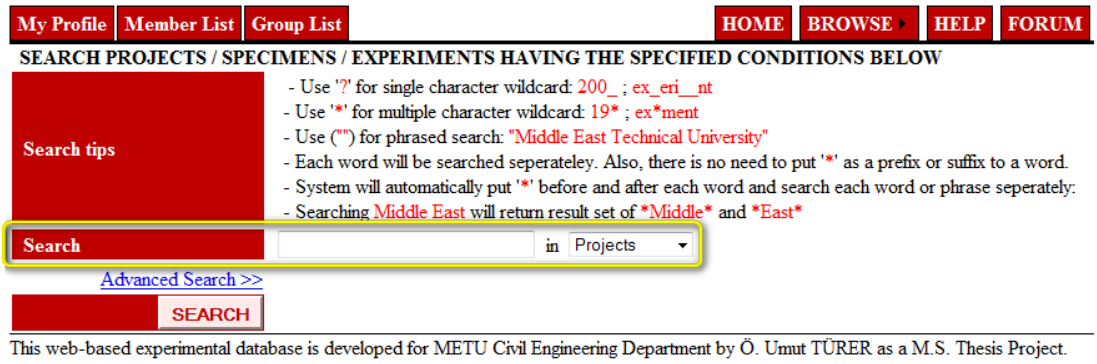


Figure 5.30, Search.aspx with simple search active

If a user wants to search different words or phrases for different fields, he should activate the "Advanced" search option by clicking on the "Advanced Search" button. "Advanced" search options can be seen in Figure 5.31.

In "Advanced" search, users can individually specify a word or phrase for keyword, name and description fields. Also, they can specify the creator and creation date range for a project, specimen or experiment. The yellow rectangle in Figure 5.31 highlights the extended search options for projects. These options appears only when the "Where to search?" drop down list is set to "Projects". With these extended options, users can specify a word or phrase for name of the project facility and funding organization. Also, they can specify a range for starting and ending date of the project. In order to activate a search option, users must check the related boxes named "Include in the search".

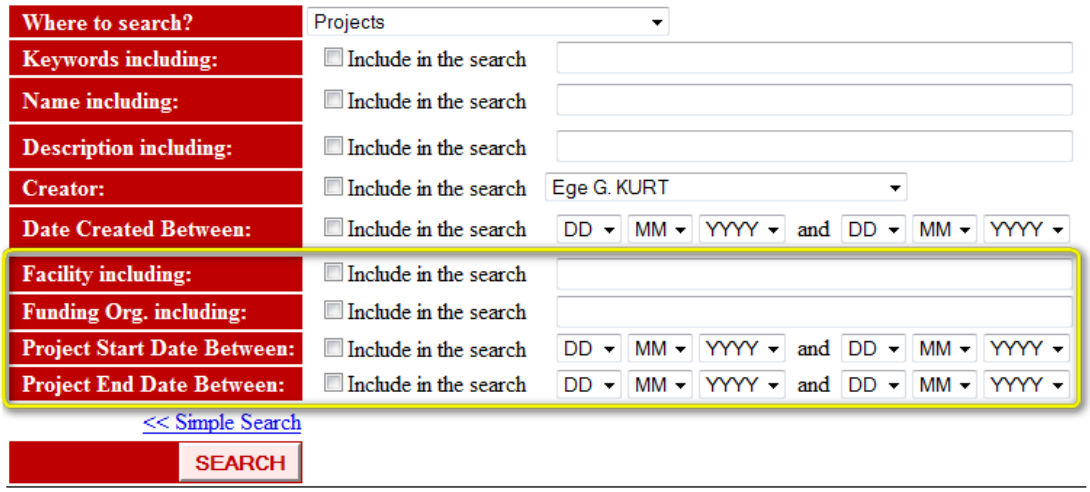


Figure 5.31, Advanced search option

In Figure 5.32, an "Advanced Search" example is given. In this example, experiments having "Pseudo Dynamic" in its keywords and created after 1 January 2008 is searched and two results including the one which is created as a part of test scenario are returned as a result. If the desired item is listed in the search results, user can go to the details page of it by selecting the

radio button related to the item and then clicking on the "Show Experiment Details" button.

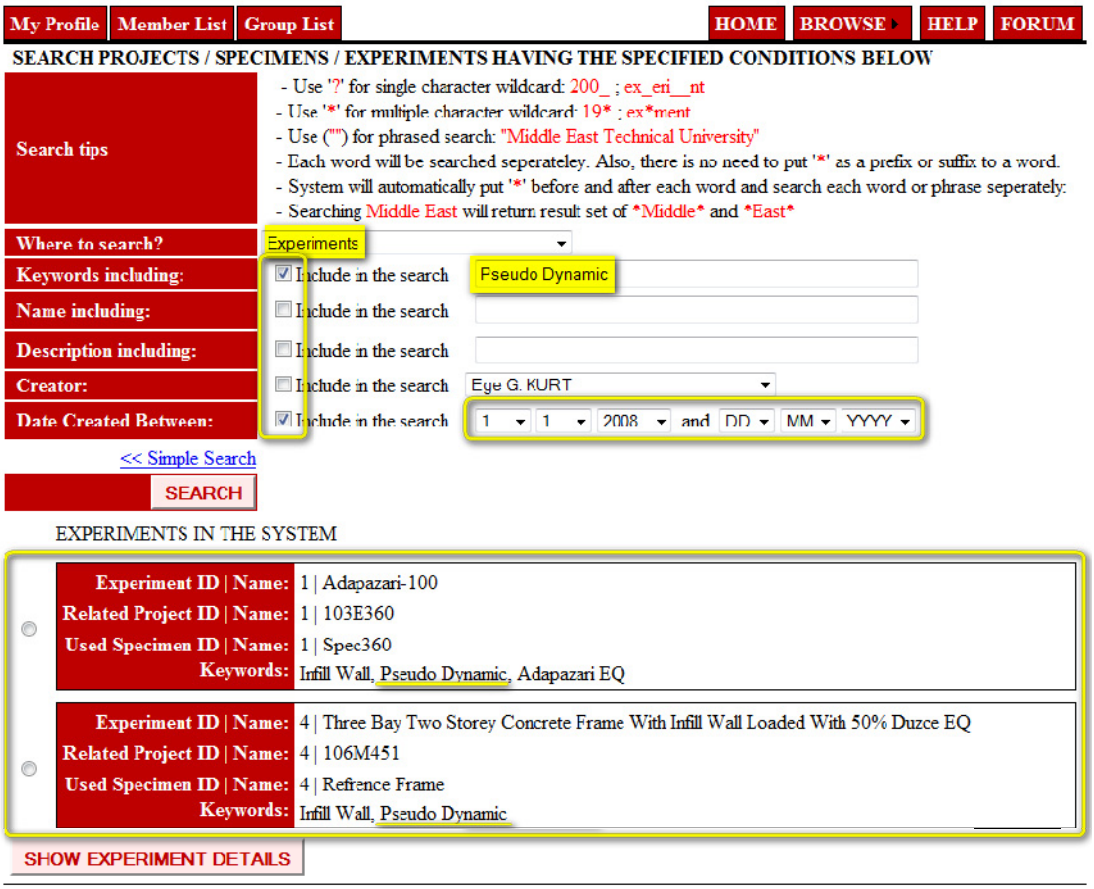


Figure 5.32, Search results for specified searching conditions

5.6 Using Integrated Forum

Currently, the only functionality of the website that provide communication between website members is the integrated forum. By using the integrated forum, members can share their opinions with each other or they can discuss with each other.

In order to reach the forum, users must click on the "Forum" item on the right navigation bar. This loads the TopicsPage.aspx as shown in Figure 5.33. This page lists the topics created by users starting with the newest topic. However, website administrators or moderators can pin an important topic such as announcements and then these topics is shown on the top of other topics. Pinned topics are displayed with a blue pin icon.

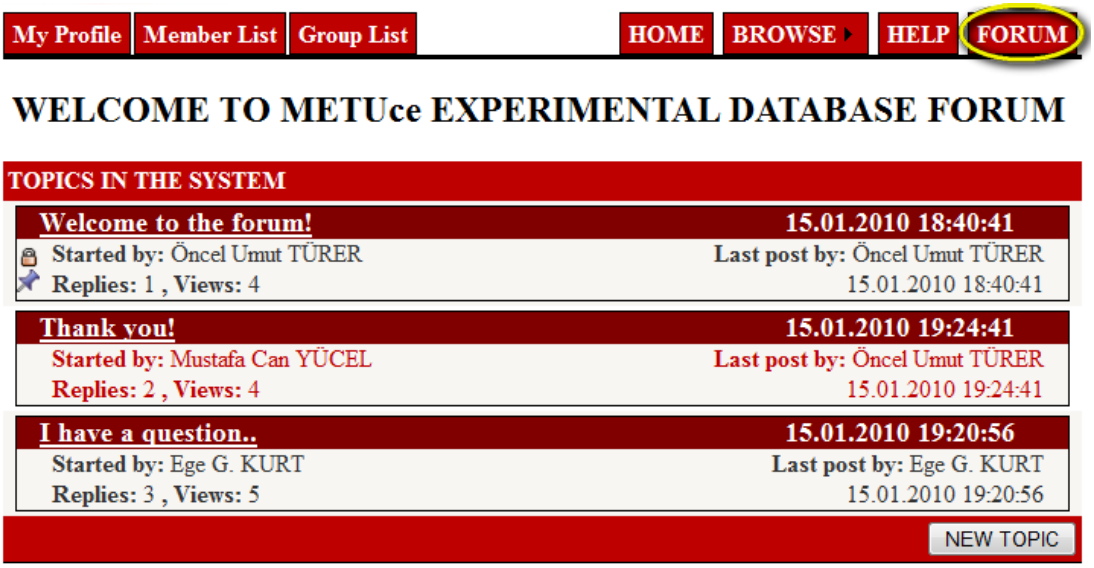


Figure 5.33, TopicsPage.aspx

To create a new topic, user must click on the "New Topic" button below the topic list. This button brings the WritePost.aspx to the screen as shown in Figure 5.34. In this page, user specifies the topic name and writes the first message which will be posted under the specified topic name.

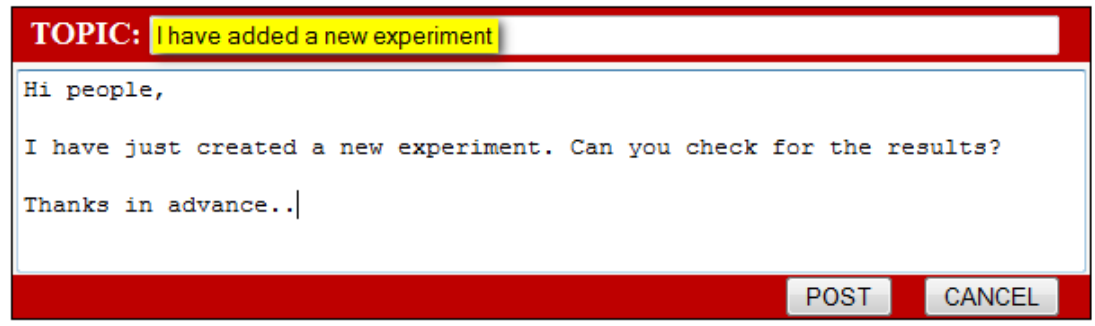


Figure 5.34, WritePost.aspx

After clicking the "Post" button, PostsPage.aspx is loaded and the messages posted under a specified topic is listed to the user as shown in Figure 5.35.

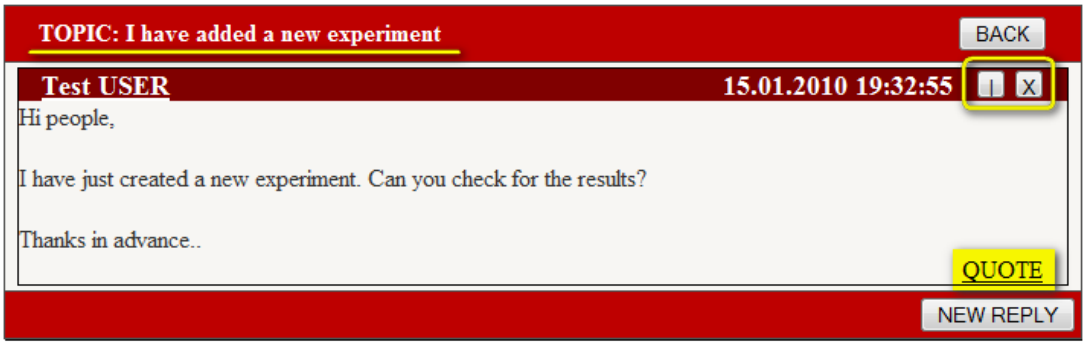


Figure 5.35, PostsPage.aspx

At the top of the page the name of the topic which contains the listed posts and a "Back" button which returns to TopicsPage.aspx is found. Below that, the messages posted under that topic is listed. However, since the topic is just created, only one message item is listed currently. Upper-left corner of each message, profile name of the sender is shown. Users can edit or delete

the messages which are posted by themselves by using the "|" and "X" buttons on the upper-right corner of the post as shown in Figure 5.35.

Users can also quote previously posted messages by clicking on the "Quote" button at the lower-right corner of a message. Clicking this button brings WritePost.aspx with the quoted message included in the textbox as shown in Figure 5.36. Everything else is same as writing a new post under a topic.

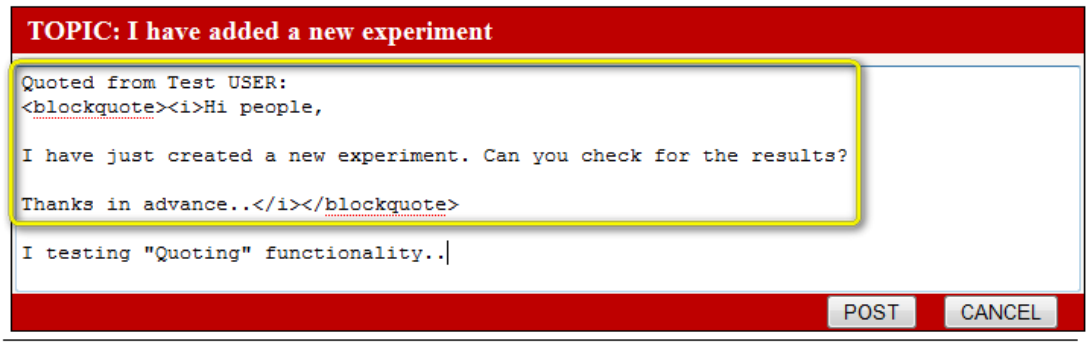


Figure 5.36, WritePost.aspx with a quoted message

After completing the message and pressing "Post" button, user is redirected to PostsPage.aspx again. Now, the newly added post with the quoted text is also listed in the page as can be seen from Figure 5.37. In order to differentiate a normal message from a quoted one, quoted text is written in italic format.

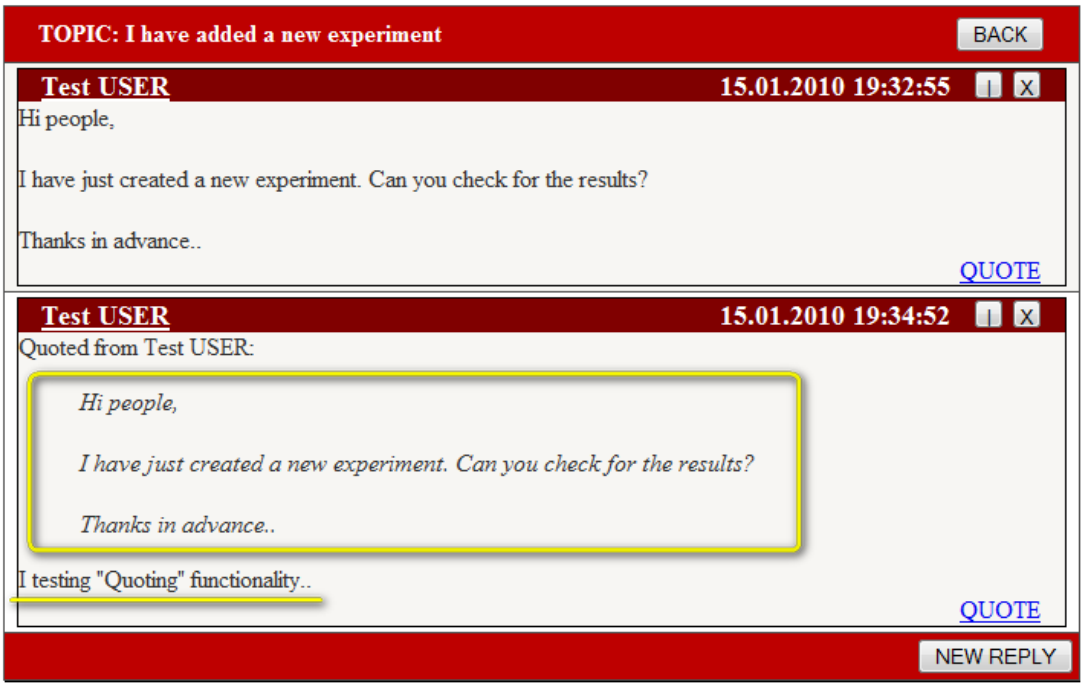


Figure 5.37, PostsPage.aspx showing quoted text in italic format

User can return back to TopicsPage.aspx by clicking the "Back" button when he finishes posting messages. As it can be seen from Figure 5.38, the newly created topic is also listed in the page. Since the user is the creator of this new topic, three buttons labeled as "L/U", "|" and "X" became visible to the user for the topic.

As it is in PostsPage.aspx, the "|" button is used for editing the topic name where "X" deletes the topic and all the messages related to that topic. The "L/U" button is used for locking and unlocking the topic for message entries.

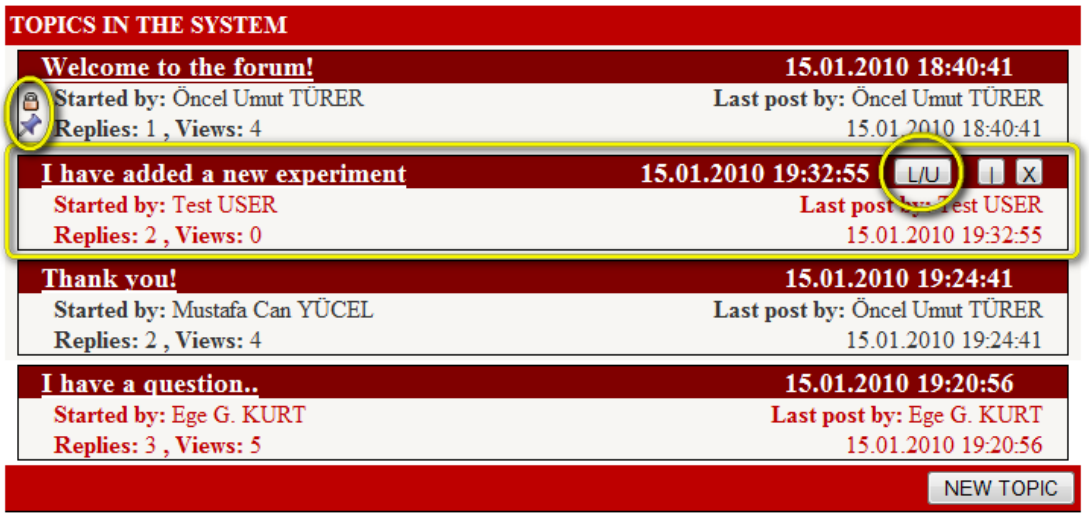


Figure 5.38, TopicsPage.aspx showing the newly created topic

If a topic is locked no one can post a message under that topic nor they can quote a message related to that topic. Locked topics are marked with a yellow lock icon as it can be seen from Figure 5.38.

5.7 Changing Password and Retrieving Forgotten Password

In time, a user may want to change his password or he may want to retrieve a new password since he forgotten the old one. In order change his password, a user must go to his profile page by clicking the "My Profile" navigation menu item. After reaching this page user must activate editing mode by clicking the "Edit" button. Then clicking "Change Password" button renders required web controls as shown in Figure 5.39. From here, user must provide his old password and enter a new password. After clicking bottom "Change Password" button, system changes user's password.

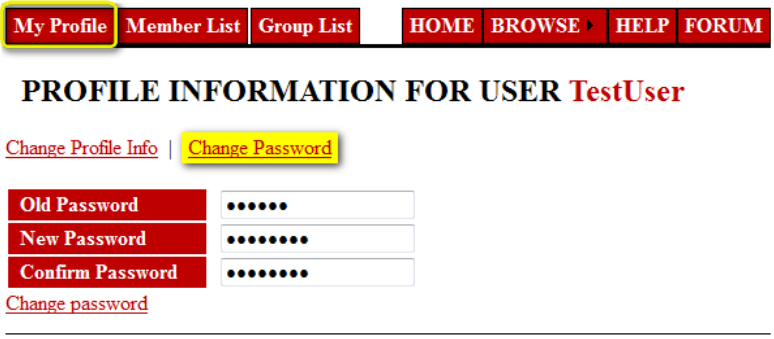


Figure 5.39, Changing password

In case a user forgets his password, he must reach the PasswordRenewal.aspx by clicking the "Forgot Password" button below login controls. At this page user must provide his UserName together with his security answer as shown in Figure 5.40. After clicking the "Finish" button, a new password is mailed to the user's e-mail address.

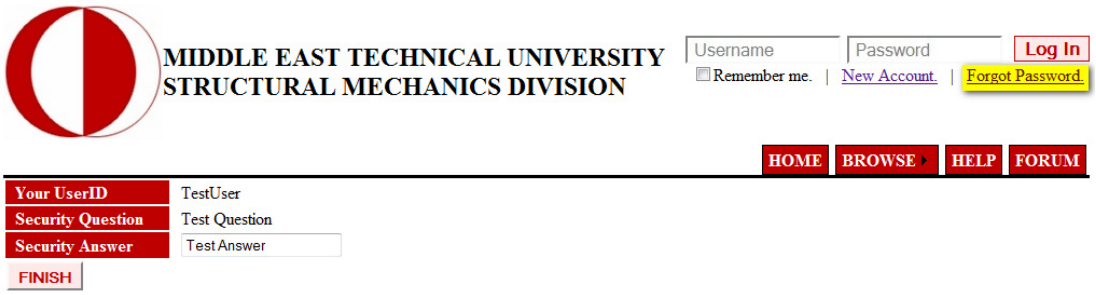


Figure 5.40, PasswordRenewal.aspx

CHAPTER 6

CONCLUSION AND FURTHER RECOMMENDATIONS

6.1 Overview

In this study, a web-based database application for storing experimental data and related documents at a single location is developed and the stored information is successfully shared among researchers and engineers. This chapter discusses conclusions obtained from the study and problems encountered during the development procedure. Also, restrictions and limitations that users may encounter is presented. Finally, recommendations for further improvement of the application is given.

6.2 Conclusions

At the end of the study, it is seen that database systems and the web technologies are perfectly applicable in structural and earthquake engineering. Valuable experimental data can successfully be stored in databases and shared via the Internet.

In development of a web-based experiment database, data reliability and security is very important. Therefore, in the presented application, data security and reliability is achieved by giving various role types to website users and by providing ownership privileges and visibility restrictions to projects, specimens, and experiments.

Also, maintainability of a web-based database is very crucial. In order to minimize further database changes, the initial data abstraction model and the following database architecture was designed carefully. Clear and understandable names were tried to given to entities and their attributes. Furthermore, defining entities with more specific attributes rather than broad ones enabled detailed categorization of the data. When developing a multiplatform application such as the presented web-based experiment database application, building the application on a good object oriented programming design pattern foundation significantly reduces the time required for the necessary application changes.

Aside from maintainability, in the design process of presented database system, the future performance of the application was also considered. The database system was developed as four separate databases. Therefore, in the future when the developed application slows down due to excessive enlargement of the application database, the workload of the server can be reduced by distributing these individual databases among other servers.

Lastly, the application's web interface was developed as user friendly as possible and user restrictions were minimized. Visuals were designed simple and instructive, numerous error handlings were made in order to cover user mistakes, users were permitted to upload their experiment data as widely used Microsoft Excel documents which also can import data from various other sources, and they were given the ability plot the experiment results as scatter graphs.

6.3 Restrictions and Limitations

Even though the application serve perfectly under most circumstances, in some cases users may encounter some restrictions. Necessity of using Microsoft Excel documents for uploading experiment data may be unsuitable in some cases where the test output comes as a formatted text document or as a CSV file. However, today Excel documents are widely used for storing experiment readings, and even if the test output is not in Excel format, Excel can easily import the content of a formatted text data or a CSV document.

Another restrictions that users may encounter is the formatting of Excel documents that are used as the data source of experiment data. Although the application automatically finds the spreadsheets which contains data inside, and detects the length and width of the data, it needs the test readings start from the first column of the selected Excel spreadsheet. But, data not necessarily start from the first row of excel spreadsheet because during upload, user selects the first row where the data starts. Even though this formatting restriction is a minor problem for users, in the future an algorithm can be written to detect the data anywhere inside the spreadsheet.

There is also a restriction on uploadable file size. In order to prevent uploading failures and depletion of empty disk space on the application test server, a 100 MB file size limit is set for the documents that will be uploaded. However, in the future this limit can easily be increased or even it can totally be removed.

6.4 Recommendations and Further Improvements

As it is pointed in the introduction part, experimental data is very important in structural and earthquake engineering and several web-based experiment

databases from all around the world are used for storing this important information. Therefore, upgrading the currently developed application and allowing it to import and export experimental data from/to other databases would be a great improvement and a crucial extension for turning the application into a distributed database system. In order to transfer experimental data among various databases, XML technology can be used because of its platform independence.

One of the most important features of the developed application is its experiment data plotting capabilities. It dramatically improves understandability and visualization of experiment data. Also, graphs play an important role in data comparison. Therefore it would be a great improvement to extend data plotting capability by adding simultaneous multiple data drawing and zooming capabilities.

Another improvement for the application would be adding a real-time wireless sensor listening capability. By doing so, application can be used as a remote health monitoring system. The readings coming from a sensor can be shown in the website. Also, thresholds can be set for some important readings and when those thresholds are passed, application can send an e-mail or an SMS message to a specified person.

Moreover, cameras can be installed around the structure which is being monitored and a live video streaming capability can be integrated into website in order to enable users to observe the structure remotely by using the web application. Also, a picture of the structure can be taken by those cameras and inserted into the application database whenever a threshold value is passed. This provides users to investigate the environmental conditions at the time when the threshold is passed.

REFERENCES

[1] Elmasri, R., & Navathe, S. B. (2004). *Fundamentals of Database Systems, 4th Edition.* Boston, MA: Addison Wesley.

[2] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2001). *Database System Concepts, 4th Edition.* McGraw-Hill.

[3] Goyal, A. (2008, August 28). *Database Management System Basic Concepts*. Retrieved January 19, 2010, from DBMS Basic Concepts: http://dbmsbasics.blogspot.com/

[4] *Database Model*. Retrieved January 19, 2010, from Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Database_model/

[5] *Database and Database Management System (DBMS)*. Retrieved January 19, 2010, from Network Dictionary: http://www.networkdictionary.com/software/dbms.php

[6] Shanker, U., Misra, M., & Sarje, A. K. (2008). Distributed real time database systems: background and literature review. *Distributed and Parallel Databases* , 127-149.

[7] McGregor, D. R., Cockshott, W. P., & Wilson, J. N. (1997). *Patent No. WO 97/32263.* Scotland/Glasgow.

[8] Hong, L. T., Givens, C., Lui, C.-C., Wright, M. J., & Fatehi, F. (1997). *Patent No. EP 0 350 208 B1.* DE, FR, GB.

[9] Ulin, T. (2003). *REAL-TIME DATABASE DESIGN -- Databases Form Core of Reliable Network Design -- Though often given short shrift, sound real-time databases are the foundation of high-availability networks, but which architecture?* Communication Systems Design.

[10] Stankovic, J. A., Son, S. H., & Hansson, J. (Jun, 1999). Misconceptions about real-time databases. *Computer* , 29-36.

[11] Casteleyn, S., Daniel, F., Dolog, P., & Matera, M. (2009). *Engineering Web Applications.* Berlin: Springer.

[12] *NEEScentral*. (n.d.). Retrieved January 19, 2010, from Network for Earthquake Engineering Simulation (NEES): https://central.nees.org/

[13] Berry, M., Parrish, M., & Eberhard, M. (2004). *PEER Structural Performance Database User's Manual v1.0.* Berkeley, CA: Pacific Earthquake Engineering Center.

[14] *PEER Structural Performance Database*. Retrieved January 19, 2010, from Pacific Earthquake Engineering Research Center: http://www.ce.washington.edu/~peera1/

[15] Chiou, B., Power, M., Abrahamson, N., & Roblee, C. (2006). An Owerview of the Project Next Generation of Ground Motion Attenuation Models for Shallow Crustal Earthquake in Active Tectonic Regions. *Fifth National Seismic Conference on Bridges & Highways.* San Francisco, CA.

[16] *PEER Next Generation Of Ground Motion Attenuation Database*. (n.d.). Retrieved January 19, 2010, from Pacific Earthquake Engineering Research Center: http://peer.berkeley.edu/nga/

[17] *Kawashima Laboratory*. Retrieved January 19, 2010, from Department of Civil Engineering, Tokyo Institute of Technology: http://seismic.cv.titech.ac.jp/en/

[18] Itoh, Y., Ishiyama, T., Liu, C., & Fukumoto, Y. (2005). Database for structural steel experiments under a distributed collaboration environment. *International Conference on Advances in Experimental Structural Engineering* (pp. 773-780). Nagoya, Japan: Ichiryusha.

[19] Zhou, Y., Kijewski, T., & Kareem, A. (2003). Aerodynamic Loads on Tall Buildings: Interactive Database. *Journal of Structural Engineering* , 394-404.

[20] *ResOlution of Site Response Issues from the Northridge Earthquake*. (n.d.). Retrieved December 10, 2009, from University of Southern California: http://gees.usc.edu/ROSRINE/

[21] Markovic, Z., & Lekic, D. (2006). Benefits of On-Line Geotechnical Database. *GeoCongress 2006: Geotechnical Engineering in the Information Technology Age* (pp. 1-4). ASCE.

[22] Archuleta, R., Steidl, J., & Squibb, M. (2005). The Cosmos Virtual Data Center: Open dissemination of worldwide strong motion data. In P. Gülkan, & J. G. Anderson, *Directions in Strong Motion Instrumentation* (pp. 209-222). Springer Netherlands.

[23] *The COSMOS VDC User Manual.* Retrieved January 19, 2010, from The COSMOS Virtual Data Center: http://db.cosmos-eq.org/CosmosVDCUserManual.pdf

[24] Swift, J. N., Stepp, J. C., Vaughan, D., Grimes, P., & Turner, L. (2002). Colloborative Research on Web Dissemination of Geotechnical Data. *Proceedings of the 2002 annual national conference on Digital government research* (pp. 1-4). Los Angeles, California: Digital Government Society of North America.

[25] Stepp, J. C. (2001). Consortium of Organizations for Strong Motion Observation Systems. *SMIP01 Seminar on Utilization of Strong-Motion Data* (pp. 123-130). Los Angeles, California: CSMIP.

[26] (n.d.). Retrieved January 19, 2010, from The COSMOS Virtual Data Center: http://db.cosmos-eq.org/

[27] Clary, J., O'Brien, J., & Urbonas, B. (1999). Development of National Stormwater BMP Database. *WRPMD'99 - Preparing for the 21st Century.* Tempe, Arizona: ASCE.

[28] Clary, J., Jones, J., Strecker, E., & Quigley, M. (2008). International Stormwater BMP Database: What's in It for You? *World Environmental and Water Resources Congress 2008 - Ahupua'A* (pp. 1-10). Honolulu, Hawaii: ASCE.

[29] Coutinho, R. Q., Oliveira, J. T., & Santos, L. M. (2000). Database of In Situ Test Results from Recife Soft Clays. *Innovations and Applications in Geotechnical Site Characterization (GSP 97)* (pp. 142-154). Denver, Colorado: ASCE.

[30] *Wireless Sensor Research*. Retrieved January 19, 2010, from Building and Fire Research Laboratory, NIST: http://www.bfrl.nist.gov/WirelessSensor

[31] Jang, W.-S., Healy, W. M., & Skibniewski, M. J. (2008). Wireless sensor networks as part of a web-based building environmental monitoring system. *Automation Construction* , 729-736.

[32] Stewart, J. (2008, June 1). *Crow's Feet Are Best*. Retrieved January 19, 2010, from The Data Administration Newsletter: http://www.tdan.com/view-articles/7474

[33] *NatHaz Modeling Laboratory*. Retrieved January 19, 2010, from University of Notre Dame: http://www.nd.edu/~nathaz/

[34] *NatHax On-line Wind Simulator*. Retrieved January 19, 2010, from University of Notre Dame: http://windsim.ce.nd.edu/

[35] *NatHaz Aerodynamic Loads Database*. Retrieved January 19, 2010, from University of Notre Dame: http://aerodata.ce.nd.edu/

[36] Itoh, Y., Liu, C., & Ishiyama, T. (2006). Development and applications of an experimental database for structural engineering education. *World Transactions on Engineering and Technology Education* , 87-93.

APPENDIX A

WEB PAGE STRUCTURE AND THEIR FUNCTIONALITIES

Figure A.1 lists the web pages of the application website. It also shows the structural relations between master and content pages. As it is expressed before, all the pages share the content of "Site.master". The ".aspx" extension is an indicator that tells the web page is created by using the Microsoft's ASP.NET framework.

Site.master:

As it is expressed before, all the pages share the content of Site.master. Therefore, everything on Site.master page is always rendered for every web page in the website. This master page contains the logo of METU for the top-left corner of the website. It also contains the login controls which are displayed at the top-right corner of the website. By using the login controls, users can create new member accounts in order to fully benefit from the websites functionalities. When a user wants to create a new account he is directed to CreateAccount.aspx web page to fill necessary information about his user name, password, e-mail address and profile information. After account creation, site members can directly log in to web site with their user names and passwords in order to fully benefit from the website functionalities.

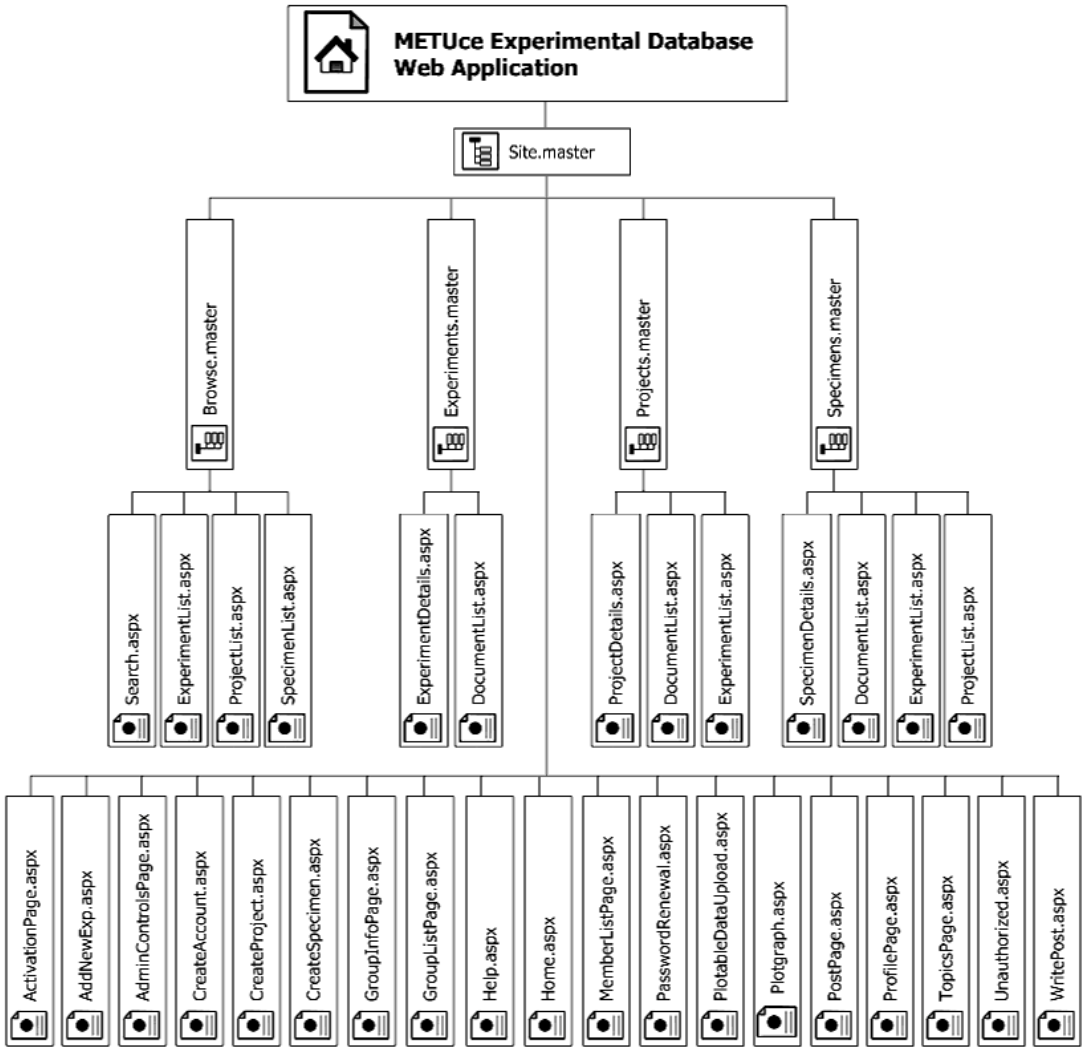


Figure A.1, Structural relation between master and content pages

Site.master has also two navigation menu for website navigation. By using the navigation bar on the right, users can navigate to home page of the application or then can visit the web pages where they can search or browse for specific projects, specimens and experiments or the pages that enables them to create new projects or specimens; they can navigate to help page in order to learn how to use the website; and they can log in to the application forum to discuss or share their knowledge with other site members. On the

other hand, the left navigation bar enables users to change their profile information; to see all the web site members or groups. Also, administrators can navigate to "AdminControlsPage.aspx" by using this navigation bar.

ActivationPage.aspx:

When a user creates a new account, an e-mail confirmation is sent to him in order to prevent fake account creation. After confirming the e-mail address, user is redirected to this page and asked to fill his user name. If the user name and e-mail address match, the user account is activated and user becomes a member of the website.

AddNewExp.aspx:

This page is one of the most important pages of the website because experiments are created by using this page. Also, their test data is uploaded in this page. AddNewExp.aspx can only be visited by clicking the "Add New Experiment" button on the ExperimentList.aspx when it is used for listing related experiments of the selected project. Before creating an experiment, the test specimen used in the experiment must have been created in the system. After filling the necessary information about experiment and selecting the used specimen, user is asked to upload the MS Excel file which contains the experimental data. Then, the data in the Excel file is presented to the user and he will be asked to select the spreadsheet columns to be inserted into the database. At this point user can enter information about each column. This information is stored as a signal information in the database.

AdminControlsPage.aspx:

This page can only seen by users having the *Admin* role. By using this page, administrators can set one of the *User*, *HigherRankedUser*, *Moderator* and

Admin roles for users, they can block a user for logging in the website by locking the membership account or they can activate membership status of a user without the email confirmation.

CreateAccount.aspx:

Users are directed to this page when they click the "New Account" button in login controls of Site.master. Before creating a new account, users need to specify a distinct user name, a password and a unique e-mail address. The email address must be valid because a confirmation e-mail message is sent to the user in order to prevent creation of fake membership accounts. Users can also specify a security question and an answer which is used when retrieving a forgotten password. Moreover, users asked to fill their profile information which includes user's real name, surname, gender, year of birth, institute that he is bounded and his position in that institute; however, filling this information is not obligatory.

CreateProject.aspx:

This page is for creating a new project which is can only be done by members having *HigherRankedUser* role or a higher one. When creating a project, user is asked to fill some information about the project such as project name, its facility, if exists the name of the funding organization, members of the project, contact name and e-mail address of the person who is responsible for the project, the starting and ending dates of the project, keywords related with the project and project description. Also, when creating a project, creator should select the project owners and set visibility restriction level of the project. Two types of owner can be set for a project: individual members and member groups. All of the website members and member groups are listed as checkboxes in the page, and project creator can scroll through these checkboxes and he assigns users or groups as owners

by putting a check in the boxes near their names. After project creation, all the owners will have the same privileges of the project creator, i.e. they can create experiments in that project, they can upload or delete documentation to the project or they can edit the information related with the project. On the other hand, the visibility restriction enables project creator to set who or what group members are able to see the project when they browse or search for projects. There are four restriction levels: "Everyone", "Site Members", "Project Owners" and "Custom Selection". The "Everyone" option is the least secure option which enables even the unregistered members to see the content and experiments of the project. This kind of project are called as "public projects". The "Site Members" option restricts unregistered website users to view the content of the project. The "Project Owners" option extends the visibility restriction and prevents all site members which are not owners of the project or a member of an owner group to view the content of the project. The last visibility restriction level option allows the project creator to decide the members or groups that are able to see the content of the project and its experiments.

CreateSpecimen.aspx:

This page works similar to the CreateProject.aspx and instead of creating a project, it helps the user to create a new test specimen. As it is in CreateProject.aspx, only users having HigherRankedUser or a higher role can process this task. When creating a new specimen, web page asks the name, keywords and description of the specimen and enables the user to set ownership and visibility restrictions for the specimen. These restrictions work with the same principle as it is in CreateProject.aspx.

GroupListPage.aspx:

This page lists the member groups in the application and enables website members to create new groups. Groups are created in order to put website members which have common properties together. When creating a new group, user is asked to fill the group name and description.

GroupInfoPage.aspx:

This page information about a selected group and its members. Also, it enables group owners to edit the group information, to delete it, to add new users to the group or remove them from the group. Also, group owners can assign ownership to other group members by using this page.

Help.aspx:

Frequently asked questions and a user manual is presented in this page.

Home.aspx:

This is the default opening page of the application. It gives a brief information about the application.

MemberListPage.aspx:

This is the page where all the website members are listed. In this page, the full name of the member, his role type, the number of created projects and the date of registration are also presented.

PasswordRenewal.aspx:

When a user forgets his password, this page enables him to be e-mailed a new password. In order to this, user have to give a correct answer to his previously specified security question.

Plotgraph.aspx:

This is the page where the test data of experiments is plotted as a scatter graph. This page contains six dropdown menus, 4 of which is hidden initially and becomes visible when the user clicks on "Enable Summation of Multiple Readings". All the dropdown menus contain the names of signal information that is collected from the MS Excel spreadsheets. The data related with reading names of the left three dropdown menus are used as X values of the graph where the data related with reading names of the right three menus are used as Y values. After user selects the readings which represents X and Y values of the graph, a small table appears showing the signal information related with selected reading names. After clicking the "Draw Graph" button, the application collects the data related with selected reading names from the database and renders the desired graph to the user by using the collected data. If a user selects multiple reading names for X and/or Y values, data related with the same axis of the graph is summed before rendering the graph. The drawing process of this graph is handled by a third party open source .NET class library called "ZedGraph". If desired, users can see the X and Y values of the drawn graph in a tabular form.

PlotableDataUpload.aspx:

This page is used for uploading a new plotable test data to a previously created experiment. This page follows the same procedure of the AddNewExp.aspx except it does not create a new password.

ProfilePage.aspx:

This page is used for displaying the profile information of a site member. Also, a user can update his profile information by using this page.

Unauthorized.aspx:

This page shows a warning message to unregistered users when they want to see a restricted web page and invites them to sign in the website.

TopicsPage.aspx:

This is the main page of the integrated forum. It lists the topics created in the forum. In this page, website members can create new topics or see the messages posted in that topic by clicking the topic name. Also, topic owners can edit the topic name or they can delete it from the database together with all the contained messages. Each topic can be locked to message entry or they can be pinned to the top of other pages; however, locking and pinning can only done by website administrators.

PostPage.aspx:

When clicked on a topic name, website members are redirected to this page which lists the messages posted under the selected topic. Users can create new posts or edit their own posts. Also, users can quote other messages when they want to post a new message by clicking the "Quote" button of a desired message.

WritePost.aspx:

This is the page that the website members use when want to post a new message under a topic. Also, when creating a new topic, users specify the topic name from this page.

Browse.master:

This master page has the searching controls for projects, specimens and experiments. It is rendered when a user wants to search or browse for projects, specimens or experiments. However, its searching controls are only

visible when a user wants to make a search; else all of its content becomes hidden except the contained content page. Browse.master provides two searching options: simplified searching and advance searching. Simplified search option scans the entered word or phrase within the name, keyword or description of projects, specimens or experiments. On the other hand, in advance search option, users can specify individual words or phrases for keyword, name and description fields. Also, they can search projects, specimens or experiments according to the creator or the creation date. Project searching has some extra advanced options such as searching according to specified facility name, funding organization name or project start and ending dates. When the necessary fields are filled and the "search" button is clicked, application dynamically creates an SQL query and sends it to the DBMS. After executing the query, DBMS returns the results back to application. Then the results are directed to the appropriate content page where the result set is listed. The content pages of Browse.master are Search.aspx, ExperimentList.aspx, ProjectList.aspx and SpecimenList.aspx.

Search.aspx:

Search.aspx is the default content page of Browse.master. Actually this page is an empty page which helps users to focus on the master page's searching controls. After users fill the necessary search fields and click the "Search" button, Search.aspx page is redirected to one of the ExperimentList.aspx, ProjectList.aspx or SpecimenList.aspx pages according to the area of search and the search results are listed in the appropriate page.

ExperimentList.aspx:

This page is a content page for Browse.master, Projects.master and Specimens.master. When a user wants to search for specific experiments by using Search.aspx or he want to browse all visible experiments in the

database by selecting "Experiments" navigation item from "Browse" tab of right navigation menu of Site.master, ExperimentList.aspx is loaded as a content page of Browse.master. When a user want to list experiments related with a specific project or specimen, it is loaded as a content page of Projects.master or Specimens.master respectively. The page has a "gridview" control and three buttons called "Show Experiment Details", "Add New Experiment" and "Delete Experiment". "Gridview" is an ASP.NET specific tabular control which enables data binding. Therefore, the result set coming from the DBMS is bound to this control for listing. The control shows the name, id number and keywords of the listed experiments together with the id numbers and names of the related project and used specimen. A radio button is located in the left side of each listed experiment. If a user wants to see detailed information about an experiment, he should select the related radio button and click the "Show Experiment Details" button. By doing so, he is redirected to the ExperimentDetails.aspx where the detailed information about selected experiment is displayed. "Add New Experiment" button redirects users to AddNewExp.aspx for creating a new experiment under a selected project. Therefore, it is available only called from Projects.master page. On the other hand, "Delete Experiment" button is for deleting selected experiment and visible under both Projects.master and Specimens.master. However, when called from Browse.master, this button becomes hidden.

ProjectList.aspx:

ProjectList.aspx is a content page for Browse.master, Experiments.master and Specimens.master. Similar to the ExperimentList.aspx, this page has a "gridview" control for project listing. This page uses Browse.master as a master page when called for showing search results or when browsing created, owned, visible or public projects. For listing projects related with a specific experiment or specimen, it uses Experiments.master or

Specimens.master as master pages, respectively. Each list item shows project id number, name, its keywords, and the number of experiments contained in the project. Also, an image shows whether the project is publicly visible or is restricted to be viewed only by its owners or a specified group of members. The visibility restriction can be set from ProjectDetails.aspx by the project owners. Different from ExperimentList.aspx, this page has three buttons. One for navigating to ProjectDetails.aspx page for detailed information, the other two is for creating a new project or deleting the selected project. The "delete" button is visible to project owners only for security reasons.

SpecimenList.aspx:

SpecimenList.aspx is used as a content page for Browse.master, Experiments.master. It uses Browse.master when searching for a specimen, or when browsing a created, owned, viewable or a public specimen. When listing specimens used in a specific experiment, this page uses Experiments.master. SpecimenList.aspx has exactly the same controls that the ProjectList.aspx has and their functionalities are similar also. Rather than listing projects, contained "gridview" control lists the specimens together with the specimen's id number, name, keywords and the number of related experiments. Every list item contains a picture showing the visibility restriction of the specimen just like as it is in project listing. After clicking the "Show Specimen Details" button, user is directed to SpecimenDetails.aspx page for details.

Projects.master:

This master page is loaded when a user want to see the detailed information about a project. It loads ProjectDetails.aspx, DocumentList.aspx and ExperimentList.aspx as content pages. It contains two navigation menu

controls: a tree view menu which lists all the projects in the system that are visible to the user and a navigation bar which loads DocumentList.aspx or ExperimentList.aspx when a user want to see documents or experiment related to a specific project.

ProjectDetails.aspx:

This page is a content page of Projects.master and shows detailed information about a specific project and enables the project owners to edit contained information. In this page, project id number, its name, the creation and modification dates of the project as well as the starting and ending dates, name and e-mail address of the contact person, name of the project creator, visibility restriction level of the project, project members, facility that owns the project and name of the funding organization, specified keywords related to the project and description about the project can be found.

DocumentList.aspx:

This page is used as a content page for Experiments.master, Projects.master and Specimens.master. It lists uploaded files related to experiments, projects or specimens depending on the master page that calls DocumentList.aspx. Also, file uploading is handled using this page. Currently any file whose size is smaller than 100 MB can be uploaded to the website, and a description for each file can be specified before or after the upload. For each uploaded file, the page shows file name, its size, its creation and modification dates as well as the description about the file.

ExperimentList.aspx:

This page lists experiments of the selected project for easy accessing. Detailed information about this page is given under Browse.master.

Specimens.master:

Similar to Projects.master, this page also had a treeview menu which shows visible specimens in the entire database. Also, by using its navigation bar, users can view a list of related projects, experiments or documents. Its content pages are SpecimenDetails.aspx, DocumentList.aspx, ExperimentList.aspx and ProjectList.aspx.

SpecimenDetails.aspx:

This page is the first opening content page of Specimens.master. It gives detailed information about a specific specimen. This information contains the id number and name of specimen together with related keywords, its description, name of its creator, the creation date and its last edit date. The modification of this information is also handled in this page by specimen owners.

DocumentList.aspx:

This page lists related documents for a selected specimen. Detailed information about this page is given under Projects.master.

ExperimentList.aspx:

This page lists experiments where the selected specimen is used. This page is explained in detail under Browse.master.

ProjectList.aspx:

This page shows a list of projects whose experiments are tested the selected specimen. Detailed information is given under Browse.master.

Experiments.master:

This master page contains only one control which is a navigation bar. This navigation bar enables user to navigate between its two content pages. ExperimentDetails.aspx gives detailed information about selected experiment and DocumentList.aspx is the content page where the documents that are related with the selected experiment is listed.

ExperimentDetails.aspx:

This page mainly contains two data bound controls. The one on the top gives detailed information about the selected experiment such as name and id number of the experiment; the names and id numbers of related project and used specimen, the dates of creation and modification, the date when the experiment is conducted, name of the creator of the experiment, names of experimenting staff, name and e-mail address of contact person responsible for the experiment, related keywords and the experiment description. The second data bound control is the one at the bottom which lists plotable MS Excel test data files related with the experiment. Experiment owners can add new plotable MS Excel documents or they can delete them. When a new test data file is wanted to be uploaded, users should click the "Add Another Plotable Data File" button and this button redirects the user to the PlotableDataUpload.aspx page for data uploading. Every listed plotable file in this control has two link buttons left to their file names. "Show Details" button presents detailed signal information as a tabular list for the selected test. The information contains the signal ID, name of the signal, its magnitude, unit, description and location. All the presented information can be edited by the experiment owners. The "Plot Data" button redirects users to PlotGraph.aspx where any specified two signal information is plotted as a scatter graph.

DocumentList.aspx:

This page lists non-plotable files of a selected experiment. Detailed information about this page is given under Projects.master.

APPENDIX B

COMPLETE ERD OF THE DATABASE SYSTEM

The Entity Relationship Diagram given in Figure 3.3, shows only the primary key and the foreign key attributes in order to reduce crowdedness and to focus on the relations between databases and their entities. On the other hand, the ER diagram given in Figure B.1 shows all the attributes of entities; thus it is a complete ER Diagram of the application database system.

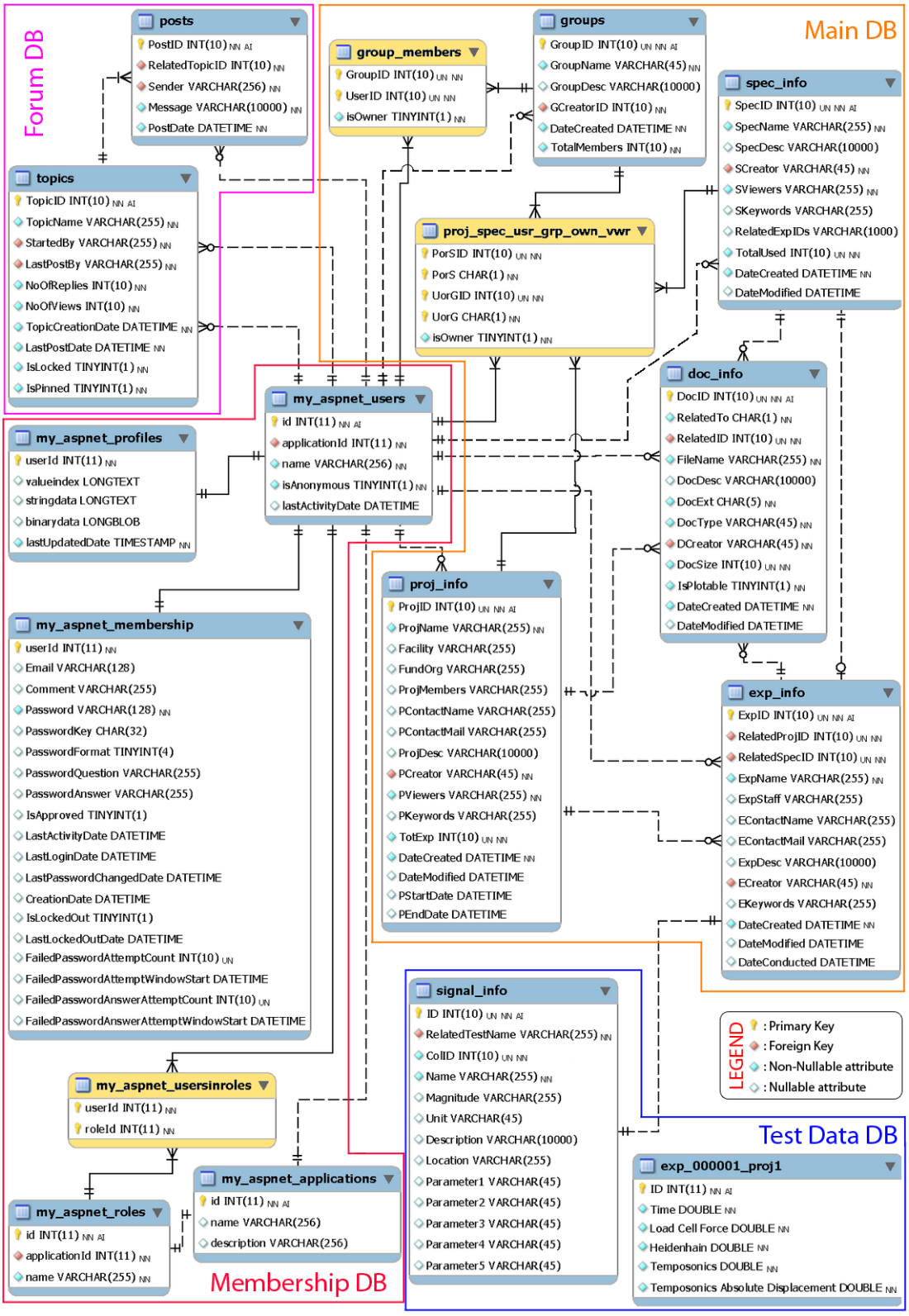


Figure B.1, Complete ER Diagram of the application database system