İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

GPS / GNSS CONTROL STATIONS' DATA MANAGEMENT SYSTEM DESIGN AND APPLICATIONS

M.Sc. Thesis by
Engin TUNALI

Department : Geomatics Engineering

Programme : Geomatics Engineering

Thesis Supervisor: Assoc. Prof. Dr. R. Nurhan ÇELİK

JANUARY 2011

İSTANBUL TECHNICAL UNIVERSITY ★ INSTITUTE OF SCIENCE AND TECHNOLOGY

GPS / GNSS CONTROL STATIONS' DATA MANAGEMENT SYSTEM DESIGN AND APPLICATIONS

M.Sc. Thesis by
Engin TUNALI
(501081606)

Date of submission : **21 December 2010**
Date of defence examination: **24 January 2011**

Supervisor (Chairman) : **Assoc. Prof. Dr. R. Nurhan ÇELİK (ITU)**
Members of the Examining Committee : **Assoc. Prof. Dr. Orhan AKYILMAZ(ITU)**
Assoc. Prof. Dr. D.Uğur ŞANLI (BU)

JANUARY 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GPS / GNSS KONTROL İSTASYONLARI VERİ YÖNETİM SİSTEMİ TASARIMI VE UYGULAMALARI

YÜKSEK LİSANS TEZİ

Engin TUNALI
(501081606)

Tezin Enstitüye Verildiği Tarih : 21 Aralık 2010
Tezin Savunulduğu Tarih : 24 Ocak 2011

Tez Danışmanı : Doç. Dr. R. Nurhan ÇELİK (İTÜ)
Diğer Jüri Üyeleri : Doç. Dr. Orhan AKYILMAZ (İTÜ)
Doç. Dr. D. Uğur ŞANLI (BÜ)

OCAK 2011

FOREWORD

This master thesis was written during the time-period from autumn until end of 2010, under the teaching supervision of Assoc. Prof. Dr. Rahmi Nurhan Çelik, Istanbul Technical University.

I would like to thank my supervisor Assoc. Prof. Dr. Rahmi Nurhan Çelik for his valuable advices and powerful support during searching and writing phases of this thesis work. Although, we worked and managed thesis work from different cities, I was aware of his expectations andit was important to me to gain his pleasure with outpust of this work. His vision forced me to take one step further and improve quality of work. I would like to present my special thanks to my family were always patient and sensitive during that time and to my friends who showed their tolerance on my nervous exhaustion and shared their best information to improve my thesis.

January 2011

Engin TUNALI

TABLE OF CONTENTS

Page

ABBREVIATIONS

ANSI	: American National Standards Institute
API	: Application Interface
ASP	: Active Server Pages
BLOB	: Binary Large Objects
CFS	: Cold Fusion
CS	: Control Station
CSS	: Cascaded Style Sheet
DBA	: Database Administrator
DBMS	: Database Management System
DCL	: Data Control Language
DDL	: Data Definition Language
DML	: Data Manipulation Language
DQL	: Data Query Language
ED 50	: European Datum 1950
ER	: Entity - Relationship
GIS	: Geospatial / Geographic Information System
GNSS	: Global Navigation Satellite Systems
GPS	: Global Positioning System
HTML	: Hypertext Markup Language
IBM	: International Business Machines
IMS	: Information Management System
INT	: Integer
ISO	: International Organization for Standardization
IT	: Information Technology
ITRF	: International Terrestrial Reference Frame
JSP	: Java Server Pages
NASA	: National American Space Agency
NAVSTAR	: Navigation Satellite with Timing and Ranging
OGC	: Open GIS Consortium
PHP	: Personnel Hypertext Pre-processor
RDBMS	: Relational Database Management System
SQL	: Structured Query Language
UTC	: Universal Time Coordinated
WEB	: World Wide Web
WGS 84	: World Geodetic System 84
W3C	: World Wide Web Consortium
XML	: Extensible Markup Language

LIST OF TABLES

Page

LIST OF FIGURES

Page

GPS / GNSS CONTROL STATIONS' DATA MANAGEMENT SYSTEM DESIGN AND APPLICATIONS

SUMMARY

In the world of information technology (IT), it is fair to state that the only constants are change and growth. Since the inception of the idea of using the computer for cartography and mapping in the 1950s, has never been a constant issue and renewal of IT with more advanced hardware and software enable users of spatial information to operate more easily applications and exceed frontiers of knowledge in time.

In the last 15-20 years, the world's largest change in IT has eased use and organization of Geographic information that this evolution pioneered the birth of a new concept. This concept is able to collect geographic data and IT under same context that is called World Wide Web (WEB) based Geospatial or Geographical Information System (GIS). In GIS, raw geographic data itself does not mean anything. To make it meaningful, data should be stored in appropriate databases. Data structure and relations should be established carefully for effective display to end users. For that reason, the database where data is stored and its design both are crucial for a sustainable and efficient GIS.

Many institutions that have common disciplines with Geomatics, consider database design so serious for ensuring healthy spatial data organization and creating reliable decision support. What other they care about is offer that database to users by the easiest and simplest way. Today, WEB is the most intense sharing platform that can handle this issue. Usage area and giving unlimited opportunity to share spatial information make the WEB platform first choice to share database among people. Nowadays, states, public and private institutions not only publish their data on database via web platform obtained from projects, but also provide users to add own data to database on the web environment after some data quality tests. A two-sided data stream (client - server) architecture has increased spatial data distribution over public this, has resulted with increasing awareness of the importance of spatial information by all areas of modern society.

In this thesis, the main goal is to design an online database with an advanced user interface on the WEB which storing GPS/GNSS (Global Positioning System / Global Navigation Satellite Systems) control station data. Furthermore, another goal of this thesis is to integrate online database interface with mapping service of Google to manage operations via online user created maps. For these purposes, the database including project information with GPS/GNSS control stations data was created according to relational database model principals and design steps from conceptual to physical.

Then, design was implemented in MySQL relational database management system (RDBMS) from where users can access to database and manage their data according to defined user privileges.

Control Stations and other related data tables were created convenient to regulation of Large Scale Map and Map Information Production in Turkey in order to have standardization in data management and sharing procedures.

Dynamic data management on the web such as: insert, delete and update control station data and navigate user requests to the database via forms are supported by dynamic WEB pages designed with Personal Hypertext Pro-processor (PHP) WEB scripting language. Furthermore, WEB map service of Google Maps were embedded to web interface. Therefore, internet users will obtain results of their requests on control station data visually, with the real world view of location. Google Maps Application Interface (API) enables users to create their own maps with optional visual features and also includes tools for transforming control station coordinates data into the map readable format.

The architecture of the system has been developed according to basic client - server environment on the WEB which enables data transitions between two sides. With help of this structure multiple users can access, query and manage control station data in the database. Results of queries and other requests are viewed on the dynamic WEB pages. Each user can populate his / her own data on free allocated space for use in the database without any conflicts by other users.

The system was designed not only for Geomatic engineers or related professions but also for random users around the world who interested in location of objects. This system contributes on data sharing and data standardization procedures like online GIS services on the internet. Moreover, data flow and distribution over the WEB environment reduce data collecting efforts and allow users to manage more economical projects. Therefore, it is suitable for private and public organizations who works with or use set of coordinates of objects and location infromation.

This architecture was named as “MySD” which is the abbreviation of “Map & Manage your Spatial Data” and published with that name on the internet. Being WEB based and open to public, offers many advantages to users such as: Creating reliable, fast and correct communication interface between user and database in order to manage and update GPS control station data, providing standard data distribution procedures between users instead of struggling with data transformation and collecting procedures and also visualizing user information and queries via WEB based Google Maps interfaces.

GPS / GNSS KONTROL İSTASYONLARI VERİ YÖNETİM SİSTEMİ TASARIMI VE UYGULAMALARI

ÖZET

Günümüzde bilgi teknolojileri dünyası ele alındığınıda değişmeyen tek şeyin değişim ve ilerleme olduğu açıktır. 1950'li yılların başında ortaya çıkan bilgisayar aracılığıyla harita üretimi fikri de aynı etki ile yerinde saymamış ve sürekli yenilenen yazılım ve donanım desteğiyle coğrafi veriyi esas alan biz kullanıcılara birçok aşılmaz sınırı aşma olanağı tanımıştır.

Son 15-20 yıl içinde dünyanın bilgi teknolojileri alanında gösterdiği bu büyük değişim coğrafi verilerin kullanımını ve organizasyonunu kolaylaştırarak yeni bir kavramın doğuşuna önayak olmuştur. Bu kavram coğrafi veri ile bilgi teknolojilerini aynı şemsiye altında toplayan Coğrafi Bilgi Sistemi'dir (CBS). CBS'nin temelinde yer alan coğrafi veri kendi başına bir anlam ifade etmez. Verinin bir anlam ifade etmesi için uygun veritabanında depolanması diğer veriler ile arasındaki ilişkilerin tanımlanması ve son kullanıcıya görsel olarak sunulabilecek yapıya getirilmesi gerekir. Bu sebep ile verinin saklandığı veritabanı ve bu veritabanının tasarımı CBS'nin sürdürülebilirliği ve verimli kullanılması bakımından büyük önem taşımaktadır.

Mesleğimizle ortak disiplinlerde çalışan birçok kurum ve kuruluş sağlıklı veri organizasyonunu sağlamak ve güvenilir karar destek mekanizmaları oluşturmak için veritabanı tasarımına çok önem vermişlerdir. Diğer önem verdikleri şey de oluşturdukları veritabanını ilgili kullanıcılara en kolay ve basit şekilde sunmaktır. Günümüzde bu sunumun en yoğun yapıldığı platform WEB tabanlı paylaşım platformlarıdır. WEB platformlarının kullanılma yoğunluğu ve insanlara sınırsız paylaşma imkanı vermesi açısından veri tabanı paylaşımında ilk sırayı alması doğaldır. Artık devletler, kamu ve özel kuruluşları gerçekleştirdikleri projelerde CBS'ye altlık oluşturacak verileri veritabanı üzerinden WEB tabanlı platformlar vasıtasıyla kullanıcılara sağlayabildikleri gibi kullanıcıların sağladıkları verilerde gerekli veri kalitesi testinden geçtikten sonra veritabanına WEB platformu aracılığyla eklenebilmektedir. Burada çift taraflı (istemci – sunucu) bir veri akışı söz konusudur ki bu durum gerçekleştirilen tez çalışmasının çatısını oluşturmuştur.

Bu tez çalışması ile küresel konum belirleme sistemi (GPS /GNSS) kntrol istasyonu verilerininin WEB tabanlı ilişkisel veritabanları ile depolanma ve güncellenme süreçleri irdelenmiş ve WEB tabanlı harita servisleri ile bu konumsal verilerin online harita üzerinde görselleştirilmesi amaçlanmıştır.

Bu amaca yönelik olarak, GPS / GNSS kontrol istasyonu verisi içerebilecek kabileyette bir veritabanı tasarlanmıştır. Bu veritabanı daha sonra dinamik WEB arayüzleri ile internet ortamında kullanıcıya sunularak online veri yönetim süreçleri izlenmiştir.

Bu ilişkisel veritabanı içerisinde depolanan GPS / GNSS kontrol istasyon verileri diğer bir online servis olan Google harita üretim arayüzü ile görsel oalrak sunulmuştur. Böylece kullanıcıların kontrol istasyonu verilerini hem dinamik WEB arayüzleri hem de online harita servisi aracılığyla yönetmesi amaçlanmıştır. Çoklu kullanıcı girişine uygun olarak tasarlanan bu arayüz, hem veri yönetimi hem de veri paylaşım standartlarının WEB ortamı üzerinden geliştirlmesi söz konusu olduğunda Türkiye'deki öncü uygulamalar arasında sayılabilir.

MySD (Map and Manage your Spatial Data'nın kısaltması) adını alan bu arayüz, sırasıyla veritabanı tasarımı, WEB arayüzlerinin tasarımı ve online harita servisinin entegrasyonu aşamalarının sonrasında ortaya çıkmış bir sonuç ürünüdür.

Veritabanı tasarımı bu çalışmanın ilk ve en önemli işlem adımıdır. Organizasyonun veya bireyin saklamak istediği veri esas alınarak amaca yönelik en uygun veritabanı tasarımı yapılmıştır. Bunun için son 20 ve 30 yıl içinde geliştirilen bütün veritabanı sistemleri karşılaştırmalı olarak ele alınmıştır. WEB tabanlı çalışması tasarlanan bu arayüz için en uygun veritabanı modeli olarak ilişkisel veritabanı modeli kullanılması uygun görülmüştür. İlişkisel veritabanı hem dünya genelinde sahip olduğu kullanım yaygınlığı hem de WEB ortamında birlikte çalışılabilirlilik esasları ele alındığında, hiyerarşik, ağ ve nesnel – ilişkisel model gibi diğer veritabanı modellerine göre üstünlük sağlamıştır. Verilerin birden çok ve birbiri ile ilişkilendirilmiş tablolar içinde satır olarak saklanabilmesi, kullanım kolaylığı ve güvenlik açıklarına karşı geliştirilmiş arayüzleri ile ilişkisel veritabanı pek çok uygulamada olduğu gibi bu tezin uygulamasında da temel yapı taşı olarak seçilmiştir.

Veritabanı tasarlanırken veritabanı hayat döngüsü (database life cycle) aşamaları sırasıyla izlenmiştir. Bu adımlar sırasıyla vurgulanacak olursa: Planlama, ihtiyaçların belirlenmesi, kavramsal tasarım, mantıksal tasarım, fiziksel tasarım ve tasarım sonrası destek. Bu işlem adımlarının ilki olan planlama aşamasında organizasyonun veya bireyin ulaşmak istediği hedef doğrultusunda nasıl bir veritabanı kullanması gerektiği tespit edilir. İlişkisel veritabanı modeli, planlama aşamasında ve bu tezin hedeflediği amaçlar gözetilerek seçilmiştir. Seçilen veritabanında hangi verinin saklanacağı konusu ihtiyaçların belirlenmesi aşamasında tespit edilir. Bu amaçla veritabanının potansiyel kullanıcıları ile görüşmeler yapılmış, GPS / GNSS kontrol istasyonu verisi ile çalışan organizasyonların görüşleri dikkate alınmış ve konu ile ilgili çeşitli belge ve dökümanlar taranmıştır. Veritabanında saklanacak verilerin neler olacağı bu çalışmalar sonucunda belirlenmiştir. Kavramsal veritabanı tasarımı aşamasında ise veritabanının varlık – ilişkisel model yapısı içinde kaba ve kağıt üstünde bir tasarımı yapılmıştır. Bu tasarımda veritabanını oluşturacak 8 adet varlık tablosu ve onların içinde saklanacak nitelik değerleri belirlenmiştir. Proje, yönetici kurum, personel, fotoğraf, GPS / GNSS, datum & projeksiyon, rapor ve uydu alıcısı bu aşamada belirlenen ve veritabanında saklanması düşünülen varlık sınıflarıdır. Mantıksal veritabanı tasarımı ile kavramsal tasarım aşamasında belirlenen varlık sınıfları tablo yapısına aktarılmıştır. Bu aşamada MySQL veritabanı yönetim dili kullanılmış olup tablolar ve aralarındaki ilişkiler bu dilin komutları kullanılarak oluşturulmuştur. Veri yönetimi ve olası veri yapısı değişiklikleri için kullanıcı erişiminin bazı sınırlamalar ile kurala bağlanması için tam yetkili, yarı yetkili ve ziyaretçi adı altında üç adet veritabanı kullanıcı tipi oluşturulmuştur. Kullanıcı hesapları doğrultusunda kimin hangi veriye ve ne kadar veriye ulaşma imkanı olduğu bir takım sınırlar ve kurallar ile tanımlanmıştır.

Bundan sonraki aşamada mantıksal tasarımda oluşturulan veritabanı yapısı tüm tablo yapısı ve ilişkileri, güvenlik ve kullanıcı kısıtlamaları ile birlikte MySQL 5.0.5 fiziksel veritabanı yapısı içine aktarılmıştır. Bu fiziksel veritabanı yapısı PhpMyAdmin 3.0.2 ilişkisel veritabanı grafik arayüzüne aktarılarak daha kullanıcı dostu bir hale getirilmiştir. Bu fiziksel tasarım sadece veritabanı yöneticisi tarafından görülebilmekte olup normal kullanıcılar bu fiziksel katmandan soyutlanmıştır. Veritabanı dışındaki kullanıcılar için veri yönetim işlemleri, WEB ortamı üzerinden dinamik sayfalar aracılığıyla yapılmaktadır.

Oluşturulan veritabanına veri eklemek, çıkarmak, güncellemek ve diğer veritabanı yönetim işlemlerini WEB üzerinden sürdürülebilir kılmak için WEB programlama dili PHP (Personal home page) ile dinamik kullanıcı arayüzleri geliştirilmiştir. Bu arayüzler yardımıyla kullanıcılar taleplerini veritabanına iletirler ve sonuçları da aynı dinamik WEB sayfaları aracılığyla görüntüleyebilmektedir.

Veritabanına yeni veri gönderme, var olan veriyi çağırma ya da güncelleme işlemleri sunucu taraflı PHP tasarım dilinin basit bir HTML (Hyper Text Markup Language) sayfası içinde kullanılması ile mümkündür. Bu yapının çalışma prensibi basit bir istemci - kullanıcı mimarisi ile özetlenebilir. Kullanıcı taleplerini kullanıcı taraflı bir HTML sayfası ile sunucuya iletir. Sunucu bu HTML sayfasını alır ve içindeki PHP kodlarını çözümleyerek talep edilen veriyi MySQL veritabanından alır ve HTML sayfası olarak istemci tarafına cevap olarak gönderir.

Kullanıcılar, bu istemci – sunucu mimarisini kullanırken birbirleri ile bir çakışma yaşamadan projelerini yönetebilirler. Buradaki temel nokta sıradan kullanıcıların bu mimarinin fiziksel arayüzlerinden soyutlanması ve sadece WEB sayfaları aracılığıyla veritabanı yönetimini gerçekleştirmeleridir. Bu WEB sayfalarının tasarım aşamalarında PHP, HTML, CSS ve JavaScript kullanılmış olup veritabanı yönetim süreçlerinin tamamı sunucu tarafında gerçekleşmektedir.

Oluşturulan veritabanı yönetim sistemine WEB arayüzlerinin yanı sıra online bir harita üretim servisi olan Google Harita'da entegre edilmiştir. Google Map 2005 yılında doğmuş ve kullanıcılara kendi haritalarını üretme imkanı veren bir internet servisidir. Pek çok uydu ve hava görüntüsünün biraraya getirilip yer yüzeyine referanslanmasıyla oluşturulan Google Maps, kullanıcıya gerçek coğrafi koordinatlarda (WGS 84 datumunda enlem ve boylam) konum bilgisi sağlamaktadır. Google Harita API (JavaScript temelli uygulama geliştirme dili) ile kullanıcılar kendi spesifik uygulamalarını online haritalar üzerinde gerçekleştirme imkanı bulmuşlardır. MySD yazılımına Google Harita servisi eklenerek GPS / GNSS kontrol istasyonlarının konumları harita üzerinde temsil edilmiştir. Google Harita servisi kullanıcıya konum verisi sağlarken WGS 84 (Dünya Jeodezik Sistemi) referans sistemini baz alır. WGS 84 referans sisteminde coğrafi enlem ve coğrafi boylam online Google Haritasında bir objenin konumunu belirlemek için yeterlidir. WGS 84 aynı zamanda GPS ölçmelerinin de baz aldığı referans sistemi olduğu için kontrol istasyonu koordinat verilerini (enlem ve boylam) kullanarak noktanın haritadaki gösterimi yapılabilir. Bu çalışma ile de veritabanı içersinde saklanan kontrol istasyon verileri harita üzerinde görselleştirilmiş ve Google Harita servisinin geliştirme arayüzü vasıtasıyla pek çok nokta operasyonunu içeren modüller sisteme entegre edilmiştir.

Bu mimari sadece Geomatik mühendislerinin kullanması için değil ayrıca konum verisi ile ilgilenen dünyadaki tüm kullanıcıların kullanması için tasarlanmıştır. Veri toplama ve paylaşım süreçlerine katkıda bulunmak bu sistem ile bir işlem adımı uzaklıkta olup, tüm veriye WEB ortamı üzerinden belirli şartlar altında erişim sağlanabilecektir. Geniş alana dağılmış verilerin tek bir ortamda toplanması ve tek ortamdan paylaşım ile veri standardının sağlanması tezin amaçladığı diğer bir konudur..

Ayrıca, WEB üzerinden veri akışının sağlanması ve buna paralel geliştirilen veri paylaşım standartları veri toplamak için harcanan kaynakların korunmasına ve veri yönetiminin mali olarak daha ekonomik bir şekilde yürütülmesini sağlayacaktır. Bu sebep ile bu tasarım, koordinat ve konum verisi ile çalışan kurum ve kuruluşlar için uygun bir yazılımdır.

Bu tasarım ingilizce "Map and Manage your Spatial Data" olarak isimlendirilmiş olup ve internette kısaltma ismi olan MySD ile yayınlanmaktadır. İnternet tabanlı ve halka açık olması kullanıcılara pekçok avantajı beraberinde getirmektedir. Veritabanı ile kullanıcı arasında güvenilir, doğru ve hızlı iletişim arayüzü kurarak GPS / GNSS kontrol istasyonu verisinin doğru şekilde yönetilmesini ve güncellenmesini sağlamak, veri standardını koruyarak kişiler ve kurumlar arası veri dönüşüm süreçlerini elemine etmek, sahip olduğu Google Haritaları görsel arayüzü ile kullanıcıya verisini WEB tabanlı harita üzerinde görüntüleme ve sorgulama imkanı sağlamak bunlar arasında sayılabilir.

1. INTRODUCTION

Science, business, education, economy, law, culture, all indicators of human development are related with the constant aid of data. It is to be known that raw data itself may not include useful information. Considering information as a building, raw data would be basic building block of that building. In other explanation, data need to be interpreted for getting useful, timely and good information which can be the key to planning and decision making. Because making a good decision is a key factor for organisational survival. The important thing is here to create a relationship between data and information [1]. Information can be defined as all of our knowledge about one subject. To share this information with others we need an information system. Data defines itself as representation of information in that system. This organized logically related data defines a database [2].

The human brain is the most complex system in the world. When people encounter with data, they encode it and manage all related processes within a brain. Although our brain is an advanced system and is able to store, interpret, and retrieve, it has a limit. Picking specific information from multiple data set and manage related information is far beyond the scope of a brain's capacity. People have used lots of method to overcome these limits throughout history. Maps on animal hide, discovery of clay tablets and process world's data into papers are some examples of these methods. With the invention of computer in the middle of 20th century, large data sets on paper transferred to digital environment so data has become more manageable and processed in order to get a desired result [3].

Looking at the past 20 years within improvement of IT, reliable store and effective management of data in computer environment via database has become much more important issue. Successful organizations in today's world should adopt powerful database functions to their system to handle with almost infinite number of data.

As a result, people can book flights more comfortable, manage their investment from home and even order meal from closest restaurant's database via web. Therefore, database technology has a major impact on the growing use of computers.

1.1 Brief History of Information Technology (IT)

The history of database dates from the mid-1960s. Database has proved to be exceptionally productive and of great economic impact. In fact, today, the Database market exceeds $8 billion, with an 8% annual growth rate [4]. From the first days of computers, storing and manipulating data have been main application focus. The first general-purpose database management system (DBMS) was designed by Charles Bachman at General Electric in the early 1960s which is known as Integrated Data Store. It formed the basis for the network data model, which was standardized by the Conference on Data Systems Languages and strongly influenced database systems through the 1960s [5]. In the middle of 1960's, International Business Machines introduced a new system called Information Management System (IMS) that added data communication capabilities to large scale databases [5]. American Airlines who was first customer of that system made a revolution in transportation area. All of major computer manufacturers produced successful database systems in the early 1970's. In addition to that in the middle of 1970's E. F. Codd formed the foundation of today's known database theory by introducing the relational model. Many organizations around the world that have been influenced by Codd's design, changed cultural and social life irreversible with their products [5].

In the 1980s, parallel to the development of relational model, the SQL query language for relational databases, developed as part of IBM's project and users gained opportunity to manage data in a database with SQL that still standard query language for database [5].

Use of the Internet has raised and the importance of databases to this growth has become even more obvious. The public internet and private intranets can be thought of as vast client/server architectures with very thin clients (browsers) and fat servers.

The servers store information in databases to be sent to the browsers in order to response to user's demands [2]. Besides, with appearance of Hypertext Markup Language (HTML), PHP, JavaScript and Extensible Markup Language (XML) design languages, governments and large organizations such as Amazon, e-Bay and ESRI in GIS environment published their databases with dynamic interfaces to public via Internet.

1.2 Spatial Data and Web Services

A GIS which is widely known concept in Geomatics environment is not just a typical database model in which different types of spatial data structures incorporate. There are many similar literatures about the architecture of a GIS. Most of the authors state these main components: data input, storage, analysis, output and the user-interface. The ways these five components are integrated distinguish a GIS from other IS.

The fact that GIS deals with geospatial data, in primary and geographic research, separate them from other IS [6]. Advanced IT and services of data via WEB platforms has changed distribution of geospatial data dramatically. Today, most governments and private organizations share spatial data to public via spatial databases depends on basic client-server architecture.

GPS that determines object's position according to satellites in orbits is one of the space based technology and may be the most widely used and shared data sets in GIS community. Since without coordinate pair coming from GPS measurements, we are away from locate objects to their real places on earth surface.

This client-server approach has brought some remarkable advantages in community of Geomatics:

- Data standardization,
- Data quality improvement,
- Interoperability among heterogeneous data,
- Combine mass amounts of data to one place,
- Reduce expense of data distribution and manage applications,
- Fast response to geographic related demands [7].

1.3Purpose of Thesis

In this thesis the main concern was creating advanced WEB based software which reflects latest technology in IT and spatial data services. This software will incorporate spatial database, WEB environment and online mapping service in order to improve data storage and distribution operations with better spatial data management.

This software was developed by using open source tools and programming languages. Therefore, it will be open for additional extensions in the future. The whole structure depends on a basic client – server architecture that allows users to interact with server side MySQL relational database and extract meaningful spatial data. This client – server architecture use WEB pages and dynamic forms for communicating. These pages and forms are designed by PHP and JavaScript programming languages.

In this software user requests of GPS / GNSS control station data are sent and received via dynamic WEB pages in which results are usually represented in tabular types. Moreover to that, another goal of the thesis is to integrate online mapping service of Google Map and Google Maps API for visualising stored GPS / GNSS control stations on Google Maps. Thus, user will be able to display stations on online maps.

MySQL, PHP and Google Maps integration for spatial database management also aims to improve data sharing and data optimization procedures by using power of the database and the internet. Users on WEB platform can access to the database of the software to manage GNSS data simultaneously under some permissions. The PHP generated dynamic WEB pages are used for abstraction of users from physical layer of database. PHP generated dynamic web pages use embedded SQL codes inside for communicating and querying the database.

This independent, open source software is called MySD (Map & Manage your Spatial Data) and developed in order to prove importance of WEB environment and databases in Geomatics applications.

In part two, fundamentals of database and design steps of relational databases will be briefly explained.

Part three will describe the SQL and MySQL which are the standard language for DBMS and allow user to conduct all crucial DBMS operations during design and management.

In part four, WEB design standards and PHP as web programming language to produce dynamic pages are briefly discussed. Moreover to these, user requests and data distribution protocols via PHP are other topics that are covered here.

In part five, we will focus on spatial data and advantages of using WEB services and for spatial data sharing.

Furthermore, in part five, online mapping service of Google Maps will be discussed. Moreover, Google Maps API (application interface) for user specific map production will be covered in that part.

In part six, application of thesis which is named MySD (Map and Manage your Spatial Data) software and its features will be explained. The database design phases, web interfaces for user access, and other GNSS CS data management interfaces will be briefly described here. Google Maps API, JavaScript-XML-PHP integration during data processing, queries and data display on Google Maps will be describe in this chapter.

Conclusions and future plans with recommendations will be discussed in part seven.

2. DATABASES

2.1 Databases and Main Concepts

According to informal Google search results, word of database has more than 20 different definitions so much in common less unique but most cause to problems for new beginners. To make it brief, a database can be defined as collection of interrelated data items that are managed as a single unit [8]. In addition, according to large definition, a database is a single large repository of minimum duplicated data that many users can simultaneously manage their operations and it is independent resource, not belongs to any organization or users [9].People may not be aware of existence of database in real life but we've been using its benefits more than a decade. Purchasing goods from supermarket, using credit card, booking a holiday, using local library, renting video and visiting internet based stores. All of these daily routine things hide a huge database behind them. Since a database is basically a way to capture, store, organize, and present data as information, they are used in nearly all types of organization.

2.2 Data

Historically, the term data is collection of facts related with objects and events that stored on computer disk and other storage engines. For example, in a geodetic database, the data would include facts such as latitude, longitude, datum data and object type. This type data is known as structured data. Numeric, character and dates are the most used structured data and they are stored in tabular form [2]. But today with fast development of databases and multimedia environment this definition should be enlarged. Since files, maps, photos, videos and audio files can be stored in a database anymore, now we define data as anything that has meaning in user's environment doesn't matter if it is structured or not.

Data on itself does not contain any information. So we can't use data and information interchangeably. Information is obtained after data have been processed [2].

Table 2.1: Data table without any information

C4	39.96906948	32.70064473	1047.789
C4	41.08403243	32.54202962	1078.023
C4	40.04046912	32.18492147	999.465
C4	40.07692904	32.60863423	999.465

There is nothing to be said about data in a table above (Table 2.1). These numbers and strings are meaningless. But presentation of that data in right context and adding some another tables maybe we will get some useful information as below (Table 2.2) for users such as point coordinates.

Table 2.2: Data table including meaningful information

Point Type	Latitude (decimal degree)	Longitude (decimal degree)	Ellipsoidal height
C4	39.93656649	32.81016469	1023.563
C4	39.90022909	32.77789235	1069.789
C4	39.94788507	32.64777302	990.563
C4	39.97696261	32.65206456	986.235

2.3 Database Management System

DBMS is advanced software gives user opportunity to create, manage and sustain database with controlled access [9]. To understand necessity for DBMS think about random university. University has a really large collection of data including departments, students, lectures and grades.

- This data is accessible all the time.
- Students, lecturers can access it concurrently.
- Multiple demands should be responded quickly and changes that made by users must be recorded on a database.
- Besides, some paths of databases should be restricted to students.

Traditional file systems are insufficient tools to maintain these processes. Not only they don’t have advanced software to give response to queries but also having security leak, being insecure to inconsistent changes coming by users and insufficient capacity to handle with that kind large data [5]. DBMS is a systematic tool of creating, storing, updating and retrieving data from database.

2.3.1 Advantages of using dbms

Using DBMS provides some advantages to users such as:

- Data independence which separates application programmes from data descriptions (metadata),
- Manages concurrent data access by multiple users,
- Prevent conflict of data from different users updating process,
- Protect users from possible system failure and unauthorized malicious access.
- Supports query language that user can retrieve data from database,
- Check data integrity while increasing data sharing [5].

2.3.2 Layers of data abstraction

Database systems have ability to give each user a view of same data, but the views can be altered according to user requirements. Although views are not able to store any data, they present changes that made in underlying database. This independence is coming from abstraction between layers in DBMS. Most modern DBMS's follow this architecture (Figure 2.1) composed of three primary layers: physical layer, logical layer, and external layer [10].

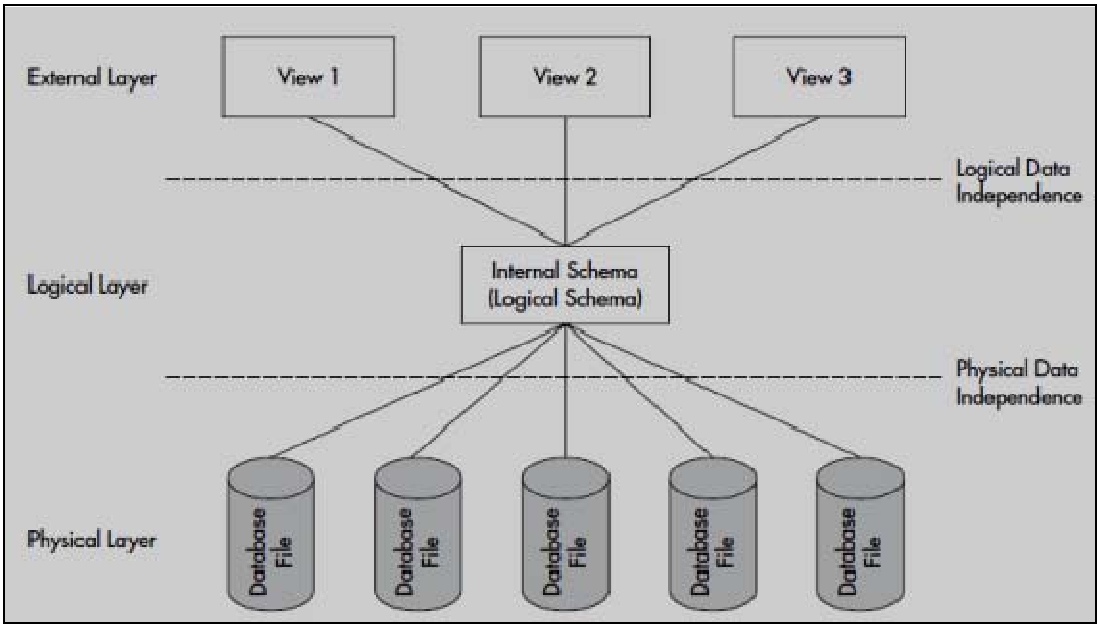


Figure 2.1: Layers of data abstraction [10].

The physical layer includes the data files that store all the data for the database. Almost all DBMSs allow the database to be stored in multiple data files, which are usually distributed over multiple physical disk drives. The random user does not need to know about storage options of data in data files. Physical data independence is provided by separation of physical layer and logical layer where database administrator (DBA) manages data operations such as configuring the database software and data files ability to alter the physical file structure of a database without disrupting existing users [10].

The physical layer has a concrete structure in the operating system files, whereas the logical layer is an abstract data structures gathered from the physical layer. The DBMS adjust the data in the data files into a common structure. This layer also known as schema, a name used for collected data items stored in a particular database. Logical level describes what data are stored in the database and what relationships exist among those data. Users can be protected from changes in the logical layer of the data. This is ensured by logical data independence [10].

The external level or view level is the highest level of abstraction. It is the view that the individual user encounter during database management process. In this layer, the database users, programmers or individuals query the database. In most modern databases, only the DBA deals with the physical and logical layers. The DBMS handles the transformation of selected items from one or more data structures in the logical layer to form each user view [10].

2.4 Data Models

There are some important terms that must be mentioned before describing data models.

Entity: An entity is a real world object that exists and is different from other objects. Particular person, department, place are examples of entities that are unique.

Entity Type: Collection of similar entities.

Figure 2.2:GPS control station - engineer entity realtionship model.

Relationship: A relationship is an association of entities. As depicted in previous page, measurement is a relationship between control station and engineer entities (Figure 2.2).

Attributes: Attributes are properties of entity types and entities are represented in a database by a set of attributes.

Null Value Attribute: Represents a value for a column that is currently unknown or is not applicable for a record.

Entity-Relationship (ER) Model: ER Model is based on a real world that comprises of collection of objects called entities and relationships among these objects. Moreover, ER model is an abstract and conceptual representation of data often for relational databases. With high level data independence, ER model is the tool of representation of database schema in DBMS that pictorially depicted by ER diagram. ER model can be converted to relational table structure for RDBMSs design in the following stages.

Relationship is association among entities that can be classified into one-to-one, one-to-many, many-to-many relationship and many to one that all depicted below (Table 2.3).

Table 2.3: Relationship types in RDBMS [1].

Relationship Type	Representation	Example
one-to-one	◄——►	President----Country
one-to-many	——<	Department----Employees
many-to-many	>+——+<	Employee----Project
many-to-one	——<	Employee----Department

One-to-many (1: m) associates one entity to more than one entity in other tables.

One-to-one (1 : 1) relation is rare one that one entity associates with only one entity just like in president-country example.

Many-to-many (m : n) relationship associates more than one entities to another a number of entities.

Many-to-one (m : 1) relation associates many entities to one entity in another table. Employees work for only one department.

A model is a representation of a real world objects, events and their associations. Data models help you to represent the data requirements of your organization and describe the structure of data with constrains and relationships in a logical manner. A model itself does not do anything and it is not a piece of software. The purpose of data model is representing data and makes data meaningful for user in a conceptual way [1].

2.4.1 Hierarchical model

The earliest databases used the hierarchical model, which originated from the file systems that the databases replaced, that records arranged in a hierarchy much like an organization chart. Hierarchical data follows the form of tree-like structures. Every object in the hierarchy has a ‘‘parent’’ object of the same type. Data are related in a nested one-to-many relationship [10].

2.4.2 Network model

In the network model, each file may be related with an absolute number of files. In spite of being flexible because any relationships can be implemented by using pointers, storage capacity of data is not enough and processes of data management are time consuming. It uses many-to-many relationship and is still popular on powerful mainframes and for high-volume transaction processing applications [2].

2.4.3 Object oriented model

In this model, attributes and methods that manage data in a database are enclosed in a structure called object classes. Encapsulation is the key word for this model because new classes are formed by more general classes which encapsulate them [2]. Moreover to these models, object relational and multidimensional data model can be added but in this work our main focus is relational data model and design of relational database.

2.4.4 Relational data model

The relational model is very simple and more flexible than other traditional models. In database design, the relational model is an "implementation" model because, difference from the ER model mentioned before that is DBMS independent, it is used to define how a database will be applied using a specific DBMS. In the relational model, data are physically represented within **tables** (Figure 2.3) that are also including **relations** with rows and columns [7].

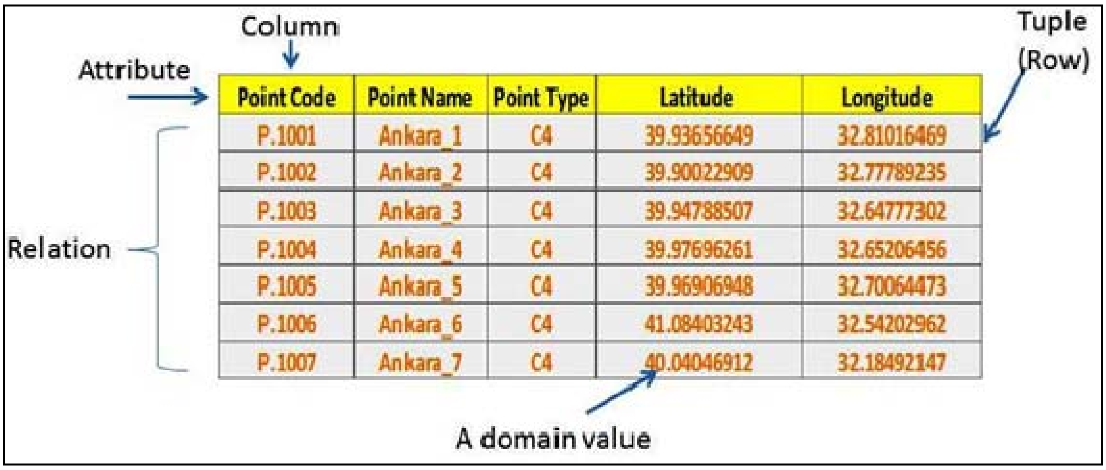


Figure 2.3: Tables with definitions to represent relational data model.

In the relational data model relations are used to hold information about objects represented in a database. Relations are represented in a table where columns correspond to attributes and rows correspond to **records or tuple**. In addition, every attribute in a relational database is associated with a **domain**. In relational table above (Figure 2.3) each record contains six values one for each attribute. It doesn't matter order of tuples because relations will stay remain [7].

Relational tables have some properties:

- The table has a unique name among all other tables in the database,
- Each cell of the table has only one value,
- Each column has a distinct name,
- The order of records and columns has no significance,
- The values of a column are coming from same domain [9].

2.4.5 Relatinal keys

Key is a group of attributes that is used to define a row in a relation. Since each value must be unique in a table, we define a unique column by using relational keys.

Superkey: A superkey is a subset of attributes of an entity-set that uniquely identifies the entities.

Candidate Key: A superkey that comprises of only the minimum number of columns necessary to provide uniqueness.

Primary Key: The key that is selected to identify records uniquely within the table. Table always has a primary key and column that selected as a primary key has a distinct attribute values.

Foreign Key: A column or columns within one table that matches the candidate key of other tables [1].

2.4.6 Relational intergrities

Data integrity constraints refer to the accuracy of data in the database. Data integrity is important to maintain data consistency for operations like INSERT, UPDATE, and DELETE which will be discussed in the following part.

Null: Null integrity means that the data value is not known or is not applicable for a record.

Entity Integrity: Entity integrity expresses that a primary key cannot accept null value.

Referential Integrity: A foreign key in a table, must match a candidate key value of some record in a other relational table or the foreign key column in a table must be completely null.

Domain Integrity: In the relational model domain integrity is used to delineate the properties of the columns of a table. Domain means of all possible values that attribute can take. Name, type, size can be referred by domain. For example age column is set to have integer value not string or character [9].

2.5 Relational Database Design

Database design process deals with various phases (Figure 2.4) that integrates all related data and includes operations of recording facts about organizations and projects. Furthermore, database design is a complex process. Because of the determination of relationships among individual parts and their representation for preserving correct functionality are highly involved, design processes are divided into some phases.

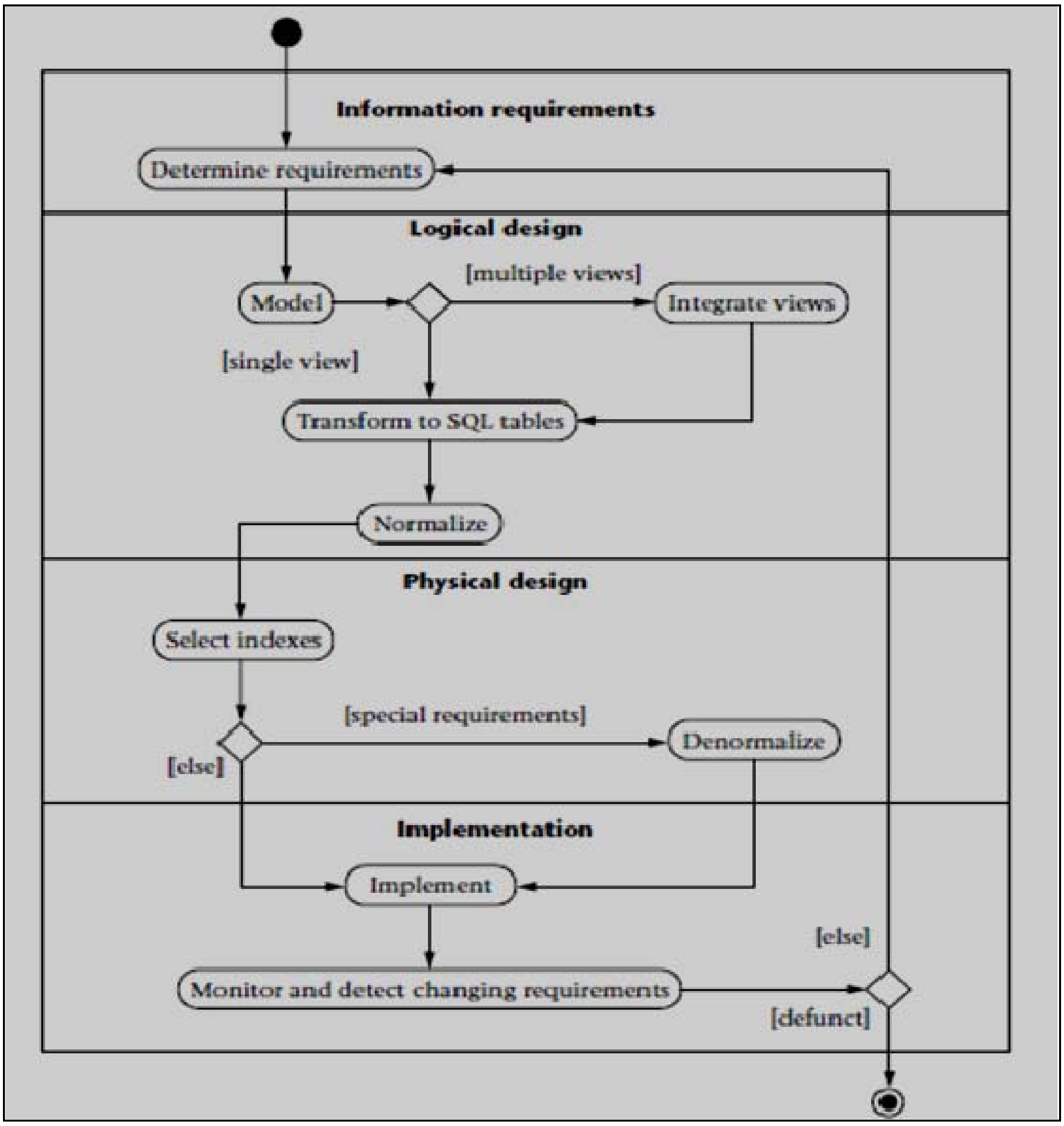


Figure 2.4: Relational database design phases [9].

- **Planning:** Planning is the phase that organizations evaluate their future expectations and estimate their management requirements with DBMS according to feasibility reports.

- **Requirements Gathering:** In this phase documents, descriptions, survey and interview results related with system demands are gathered and database requirements to fulfil project are evaluated [9].
- **Conceptual Design:** The conceptual design phase includes modelling the externals of the application and database in high level of abstraction without using any data model and DBMS. The layout of reports, screens, forms, web pages, and other data related presentation tools are designed in this phase [10].
- **Logical Design:** In this phase conceptual design is transformed into logical model that is ready to be implemented into RDMS. Also known as internal design as constructing a model of information used in enterprise based on a relational data model. ER model is the base of construction internal design [10].
- **Physical Design:** During the physical design phase, the logical design is converted to the hardware and software that is going to be used to implement the databases and user applications. Describes the base relations, file organisations, and indexes design used to achieve efficient access to the data, and any associated integrity constraints and security measures are designed in physical design phase [10].
- **Construction:** In the construction phase, the application developers query and test the programming units. If quality expectations, hardware and software requirements are received successful program units are moved to production unit to introduce random users [9].
- **Implementation:** Implementation in database design provides environment for installing a new application programs, forms, web pages, reports and database objects into the system and also provides tools for data conversion operations.
- **On-going Support:** This phase is related with quick response and maintaining life cycle of system for performance issues, unexpected results, system errors and failures or the impossible demands coming from users for improvements [10].

In the next part we will discuss the SQL, relational DBMS language.

3. STRUCTURED QUERY LANGUAGE (SQL)

3.1 Brief Introduction to SQL

"S-Q-L" or other pronunciation "See-Quel" is the most widely used commercial relational database language that used by both professionals and random users. SQL was a production of IBM's System-R projects that conducted in the middle of 70's [9]. Oracle was the first DBMS that supported SQL and then followed by IBM, Microsoft and others. According to ANSI (American National Standards Institute) and ISO (International Organization for Standardization) standards it has been accepted as a formal language for RDBMS management [2].

3.2 SQL Environment

The SQL environment (Figure 3.1) includes DBMS with two databases (one of them as test database) and programs to use DBMS to access these databases. Each database is part of catalogue that constitutes a description of a database. And schema is the structure in that catalogues that includes user created objects.

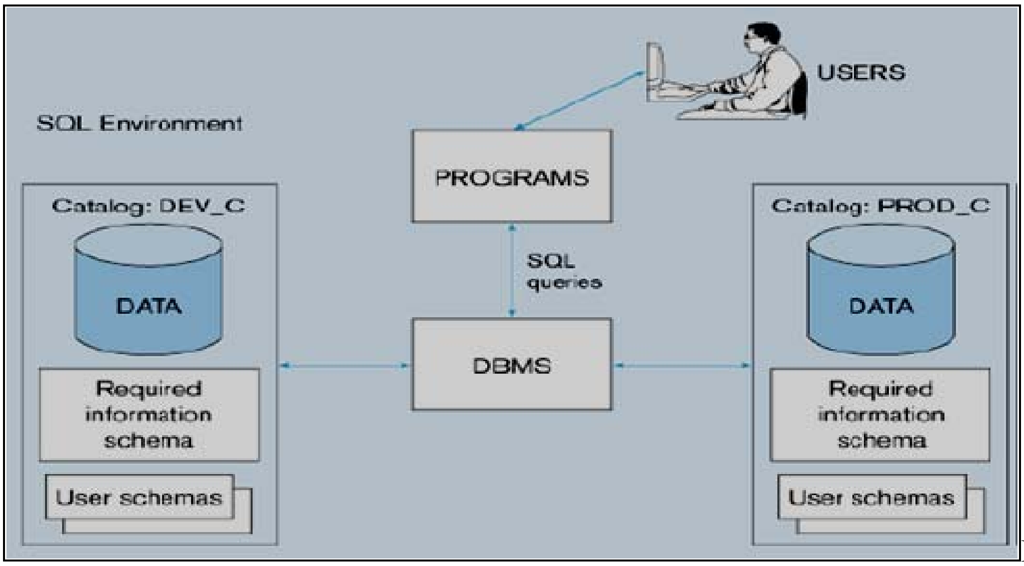


Figure 3.1: SQL environment in DBMS [2].

3.3 SQL Language

A DBMS language SQL has abilities such as: Create and edit database and table structures, manage a basic data tasks such as insert, delete and modify data in the table, perform both simple and complex queries, easy to learn and work with minimum user performance, control access to data and provide data manipulation processes [9].

All these operations are mentioned above are carried out by SQL commands that classified into four types.

3.3.1 Data query language (dql)

The "SELECT" statement (3.1) is used to query data from database. Select with clauses statement.

- SELECT: Lists the columns that are to be returned in the results.
- FROM: Lists the tables or views from which data is to be selected.
- WHERE: Provides conditions for the selection of rows in the results.
- ORDER BY: Specifies the order in which rows are to be returned.
- GROUP BY: Groups rows for various aggregate functions.

SELECT point_id, latitude, longitude, height **(3.1)**

FROM GPS

ORDER BY point_name;

3.3.2 Data definition language (ddl)

This type of language includes statements that create and modify objects in a database. DDL works with physical design of database objects not with data. It has CREATE, ALTER and DROP statements. "CREATE TABLE" statement (3.2) to create a new database objects.

CREATE TABLE Points **(3.2)**

(Point_ID DECIMAL (12.4) NOT NULL

CONSTRAINT pk1 PRIMARY KEY (Point_ID)

"ALTER TABLE" statement (3.3) for altering, removing, deleting columns and constrains such as foreign key, primary key, unique value.

ALTER TABLE `Points` **(3.3)**

ADD CONSTRAINT `gps_ibfk_1` FOREIGN KEY (`Point_code`)

“DROP TABLE” statement (3.4) to remove database objects (tables, views) from database when they are not needed. Adding “CASCADE CONSTRAIN clause causes to drop all child data related by key.

ALTER TABLE `points` **(3.4)**
ADD CONSTRAINT `gps_fk_1`
FOREIGN KEY (`point_code`)
REFERENCES `project` (`proint_code`)
ON DELETE CASCADE ON UPDATE CASCADE;

3.3.3 Data manipulation language (dml)

DML statements allow users to add, change and remove columns (implementation) from table in a database by using INSERT, UPDATE, DELETE commands.

INSERT” statement (3.5) add a new row to table. **(3.5)**
INSERT INTO `points` (`point_id`, `point_no`, `point_name`)
Values ((39, 'P.1001', 'Ankara_1');

“UPDATE” statement (3.6) updates the data values in table. Using WHERE clause helps to user define a specific data value in the table to update.

UPDATE points SET point_name = ANKR_1; **(3.6)**
WHERE point_id = 921;

“DELETE” statement (3.7) removes one or more rows from the table. With WHERE clause we can define specific data to remove.

DELETE FROM points **(3.7)**
WHERE point_id = 36

3.3.4 Data control language (dcl)

These commands are used by DBA for control and maintenance of database. According to privileges users can manage database from create a new user account to change column in the table. “GRANT” for give privileges and “REVOKE” to take it back.

3.4 MySQL

Since database software has become more public, open source database systems are freely available anymore. One of these is MySQL, a SQL client/server RDBMS coming from Scandinavia. MySQL contains SQL server, client programs for accessing the server, administrative tools, and a programming interface for operating your own applications [11].

3.4.1 Why mysql ?

MySQL is advanced software that differs from other free open sources such as PostgreSQL, SQlite in many respects like performance, support features, licensing service and price[11].

- Speed: MySQL is the fastest database system in the world.
- Easy to use: Simple and less complex database management.
- Query Language Support: MySQL is interoperable with SQL.
- Capability: MySQL has many interfaces that multiple users can access to multiple databases simultaneously.
- Connectivity and Security: With today's WEB technology MySQL is accessible from anywhere via internet.
- Portability: MySQL can run on Windows, Linux, Unix, NetWare from biggest servers to personal laptops.
- Small size, Availability, Free of Cost
- Open Source Codes: All codes of software can be obtained from web site of MySQL (www.mysql.com).
- Support: The MySQL Reference Manual is a huge tutorial that comes with MySQL distributions. Moreover to that MySQL user community, training classes, monitoring services makes it so widespread [11].

MySQL can manage database with more than 50.000 tables and 5 billion rows. Today, National American Space Agency (NASA) and United States Census Berau are the main customer of that product.

3.4.2 MySQL data types

In MySQL there are three main data types: Text (Table 3.1), Numeric (Table 3.2), and date / time (Table 3.3) data types.

Table 3.1: Text data types in MySQL [12].

Data Type	Description
CHAR(size)	Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters.
VARCHAR(size)	Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters.
TINYTEXT	Holds a string with a maximum length of 255 characters.
TEXT	Holds a string with a maximum length of 65,535 characters.
BLOB	For BLOBs (Binary Large Objects). Holds up to 65,535 bytes of data.
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters.
MEDIUMBLOB	For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data.
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters.
LONGBLOB	For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data.
ENUM(x,y,z,etc.)	Let you enter a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted. 1. **Note:** The values are sorted in the order you enter them. 2. You enter the possible values in this format: ENUM('X','Y','Z')
SET	Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice.

Table 3.2: Numeric data types in MySQL [12].

Data Type	Description
TINYINT(size)	-128 to 127 normal. 0 to 255 UNSIGNED*. The maximum number of digits may be specified in parenthesis.
SMALLINT(size)	-32768 to 32767 normal. 0 to 65535 UNSIGNED*. The maximum number of digits may be specified in parenthesis.
INT(size)	-2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED*. The maximum number of digits may be specified in parenthesis.
FLOAT(size,d)	A small number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter.

DOUBLE(size,d)	A large number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter.
DECIMAL(size,d)	A DOUBLE stored as a string, allowing for a fixed decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter.

Table 3.3: Date data types in MySQL [12].

Data Type	**Description**
DATE()	A date. Format: YYYY-MM-DD **Note:** The supported range is from '1000-01-01' to '9999-12-31
TIME()	A time. Format: HH:MM:SS **Note:** The supported range is from '-838:59:59' to '838:59:59'.
YEAR()	A year in two-digit or four-digit format. **Note:** Values allowed in four-digit format: 1901 to 2155. Values allowed in two-digit format: 70 to 69, representing years from 1970 to 2069.

4. DYNAMIC INTERFACE DESIGN FOR WEB DATABASE

Today many governments and organizations offer their data and products through web sites and users can get information or order products by visiting these sites. Since internet users want to communicate with advanced DBMS via WEB pages that updated frequently, the dynamic pages that maintain communication of user and database take an remarkable place in software's architecture [13]. Creating the dynamic interface for database management is code embraced process. Therefore, user who is responsible from design should have preliminary knowledge about design languages. Today, WEB design standards are defined by World Wide WEB Consortium (W3C).

4.1 HTML

HTML is the name of language used to define a document founded in 1994 by Berners Lee. It is also known as markup language because HTML works by increase effect of regular text with `marks' that carrying special meaning for a Web browser handling the document [5]. It was also the first language that used to distribute information over internet via Web browsers.

```
<html>
<head><title>Hello HTML</title></head>
<body><p>Hello World!</p></body></html>
```

(4.1)

HTML commands are known as tags, usually opening tag and closing tag. Information can be seen via web browsers if only be placed correctly between these tags. As depicted with simple "Hello World" program of HTML (4.1), the text between <html> and </html> describes the web page while the text between <body> and </body> is visible page content [14].

Although HTML is sufficient to represent the structure of documents for static display purposes, the language itself are far beyond the scope of representing the dynamic data within a document for more advanced applications.

4.2 Client Side Scripting Languages

Client-side languages are written programmes run on the user's computer. Whatever you write on a page, it will never change and directly sent to users browser (Figure 4.1). An example of client-side programming is JavaScript. JavaScript can be used to run checks on form values and send alerts to the user's browser. Menu that pop up, link that open new windows are examples of applications that created with this scripting language. The problem with client-side scripts is the limit of control and problems with different operating systems and web browsers [14].

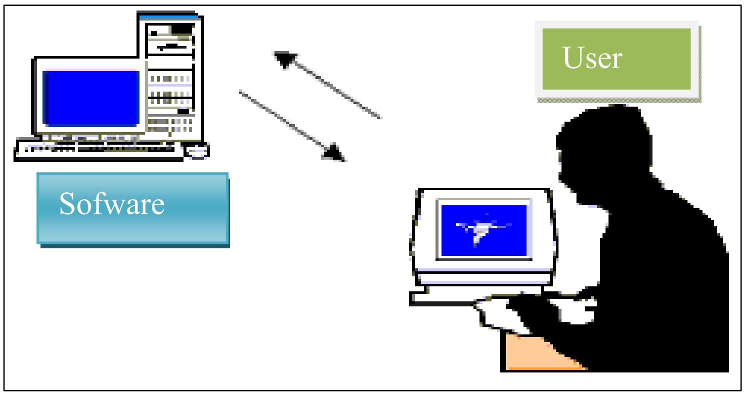


Figure 4.1: Sratic HTML transaction procedure.

4.3 Server Side Scripting Languages

These languages are designed to response requests that coming from users by accessing databases. All operations are managed in the server and results are sent with HTML page according to requests. Result HTML page coming from server includes PHP codes that provide dynamic structure (Figure 4.2). PHP, ASP (Active Server Pages), PERL-CGI, JSP (Java Server Pages) and CFS (ColdFusion) are well known server side languages [14].

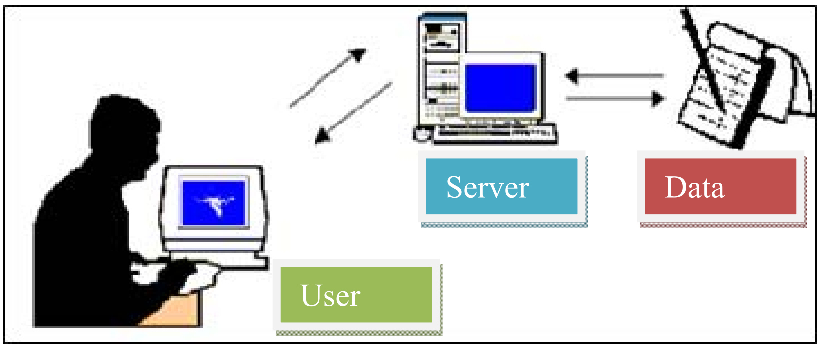


Figure 4.2: Dynamic HTML transaction procedure.

4.3.1 Personal hypertext preprocessor (php)

According to official web page www.php.net, PHP is a widely-used open source server side scripting language that is developed for database driven dynamic Web applications and can be embedded into HTML. Starting from the last that you can take an regular HTML page add some PHP code in it and get dynamic results.

(4.4)

```
<html>
 <head>
  <title>PHP Test</title>
 </head>
 <body>
 <?php echo '<p>Hello World</p>'; ?>
 </body>
</html>
```

The PHP code is put inside in special start and end processing instructions <?php and ?> as shown in “PHP Hello World” code (4.4) that allow you to get in and out of "PHP mode."

- PHP brings websites to dynamic in some ways:
- Sending feedback from your website directly to your mailbox,
- Sending email with attachments,
- Uploading files to a web page,
- Watermarking images,
- Generating thumbnails from larger images,
- Displaying and updating information dynamically,
- Using a database to display and store information,
- Making websites searchable and much more [15].

It was designed in 1994 by Rasmus Lerdorf to write WEB scripts not stand alone applications. At the time this thesis was written PHP was at version 5.3.

Working principle (Figure 4.3) of PHP pages are simple. When users visit one web page that written by PHP, the server checks the codes and process them in order to give response to user demands. Results are sent to the user with simple HTML page [15].

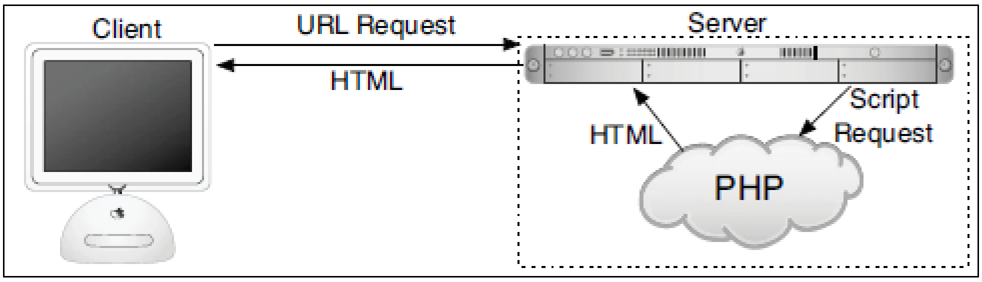


Figure 4.3: PHP in client – server model [13].

Furthermore, another remarkable feature of PHP is its database support. ODBC, MySQL, PosgreSQL, SQlite, Informix, InterBase, FrontBase, Solid, Empress are some number databases which work interoperable with PHP [15].

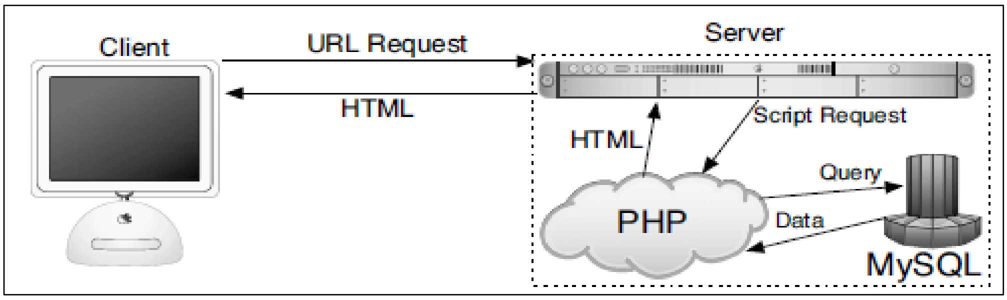


Figure 4.4: Working principal PHP-MySQL in WEB [13].

According to the recent tests and analyses, PHP gives best results with MySQL database. With PHP - MySQL integration user requests are analysed in the server. The simple HTML page coming from user is processed in PHP runner machine in order to find requests defined in PHP codes. Then, PHP interact with MySQL database for extracting wanted data or information (Figure 4.4). Responses of these requests are carried by PHP in the server side and forwarded to user (client side) in the type of HTML. During this operation PHP uses some SQL commands for interaction with MySQL database (Table 4.1).

Table 4.1:SQL commands in PHP programming language [16].

Command	**Description**	**Version**
mysql_select_db	Select a MySQL database.	**5.3**
mysql_db_query	Send a MySQL query.	**5.3**
mysql_result	Get result data.	**5.3**
mysql_create_db	Create a MySQL database.	**5.3**
mysql_drop_db	Drop (delete) a MySQL database.	**5.3**
mysql_error	Returns the text of the error message from previous MySQL operation.	**5.3**
mysql_fetch_array	Fetch a result row as an associative array, a numeric array, or both.	**5.3**
mysql_free_result	Frees the memory allocated for a result set.	**5.3**

5. GEOSPATIAL INFORMATION AND WEB SUPPORT

5.1 Geospatial Data

Geospatial data are set of data referenced to a place by number of geographic coordinates which can be collected, stored, updated, analysed and displayed graphically according to their attributes [17]. Geospatial data has two important properties:

- Data reference to a geographic space that means data with identical geographic coordinate system all around the world. So, data from different sources can be integrated and cross referenced spatially.
- Representation of geospatial data depends on geographical scale. For representing large areas of Earth's surface small scaled geospatial data should be generalised [7].

Geospatial data are collected based on two types of forms called vector and raster. The geographic object is the main spatial unit of vector form which is a recognizable discrete real world feature by a point, a line or a polygon [7].

In a spatial database, vector data is a topographic based data whose work is to supply the spatial referencing framework for data accumulation and analysis. Data in a topographic base contain geodetic and survey control stations and features that are found on a typical topographic map such as roads, rivers, urban areas and natural vegetation features [7]. In this thesis work we will focus on point type of geospatial data as control station data obtained by using GPS technology to store in the database and represented as a control station in the maps.

5.2 Global Positioning System (GPS) Data

GPS or Navigation Satellite with Timing and Ranging GPS (NAVSTAR GPS) is the name of technology that have been using by Geomatic engineers and random users for more than twenty years. By using the GPS technology we can determine two values anywhere on earth.

- Exact location of objects within accuracy range between meters to millimetres.
- The precise Universal Time Coordinated (UTC) time within accuracy of nanoseconds.

GPS using at least 24 satellites with advanced atomic clocks in 6 orbit, constellated to ensure at least 4 satellites are in radio communication with GPS receiver at any station, orbiting earth for approximately 12 hours with 55˚ inclination angle for obtaining most accurate object coordinates [18]. We will not focus on fundamentals and working principal of GPS and coordinate systems here such as: Satellites, signal types and structures, error modelling which can be found in the GPS / GNSS text books. Some of these topics can be studied from "GPS satellite Surveying (Leick, 1995)" and "GPS Theory and Practice (Hoffman and Wellenhof, 1994)".

But users have to know basic terms that the related with GPS measurements:But users have to know basic terms that related with GPS measurements:

- **Geographic latitude (λ):**The geographic latitude (abbreviation: Lat.) of a point on the Earth's surface is the angle between the equatorial plane and a line that passes through that point and is normal to the surface of a reference ellipsoid which approximates the shape of the Earth.
- **Geographic longitude (ϕ):** The Longitude (abbreviation: Long.) of a point on the Earth's surface is the angle east or west from a reference meridian to another meridian that passes through that point.
- **Ellipsoidal height (h)**: Distance of a point from the ellipsoid measured along the perpendicular from the ellipsoid to this point, positive if upwards or outside of the ellipsoid. Only used as part of a three-dimensional geodetic coordinate system and never on its own.

The latitude, thelongitude and the geodetic height which are obtained from GPS measurements, are enough to determine of object position on the earth. Furthermore these are also minimum required data for representing the GPS / GNSS CS on the online Google Maps. The coordinate system of measurements should coincide with the Google Maps reference system, WGS 84 in order to eleminate transformation and conversion processes [18].

World Geodetic System (WGS 84), reference system for GPS measurement which is a three-dimensional, right-handed coordinate system with its original coordinate point at the centre of mass and also has been using as the default reference system to represent spatial objects in Google Earth first commercial mapping software of Google, produced in the 2005.

In Turkey, the standards of GPS measurements and data are determined by Regulation of Large Scale Map and Map Information Production which also was used as core for MySD’s control station data storage policy.

5.3 Sharing Spatial Data

Geospatial data may be acquired by local governments, municipalities, private companies, academic institutions, and non-profit organizations. The collection and management of that geospatial data are the most time consuming and costliest part of the GIS. It also should also take into consideration increasing number of geospatial data brings the methods for acquiring data and the new ways for using it variously and of course the new data sharing opportunities [17].

Data sharing is a key concept of modern spatial database systems. Spatial data sharing is not a new act. Agencies at different levels of government and private organizations have shared data for mapping at different scales and for different purposes. While random users are able to request digital map layers from both government and private data providers. It is also widespread for engineers, surveyors and scientists to share spatial data and use spatial data gathered by others [7].

The need to organize collecting and sharing standards of geospatial data between government agencies, the private sector, and academia is not a new theme.

It recurs, in part, because it is widely recognized that collecting data multiple times and distribute it in different formats for the same purpose is wasteful and inefficient, unfortunately, it is what happening in Turkey for a long time.

There are many aid of sharing spatial data:

- Reducing the expenses of using spatial information by minimising the need to collect or convert data again and again when creating a new spatial database,
- Powerful applications that requires mass amount of data coming from different sources, collected in various times and stored in different formats,
- Quality increase of spatial data, and therefore, high degree of right decision making, by allowing different applications to share and implementation to a standardized reference data sets,
- Reducing costs of software and data maintenance [7].

Data non-interoperability is the main problem that preventing effective sharing processes of geospatial data. This non-interoperability can be classified as data and access non-interoperability. One is related with data incompatibility while another is issue of software related such as diversity in storage format.

5.4 WEB Services

To overcome these heterogeneities in geospatial data sharing, Open GIS Consortium (OGC) with various services has been introduced to GIS community. WEB and Map services on Internet by OGC and other organizations have improved data and access interoperability in GIS community. Moreover, WEB enabled geo-database number has increased dramatically according to last development of IT and data sharing procedures.

Architecturally, Web-enabled spatial database systems use a client/server configuration with four tiers (Figure 5.1).

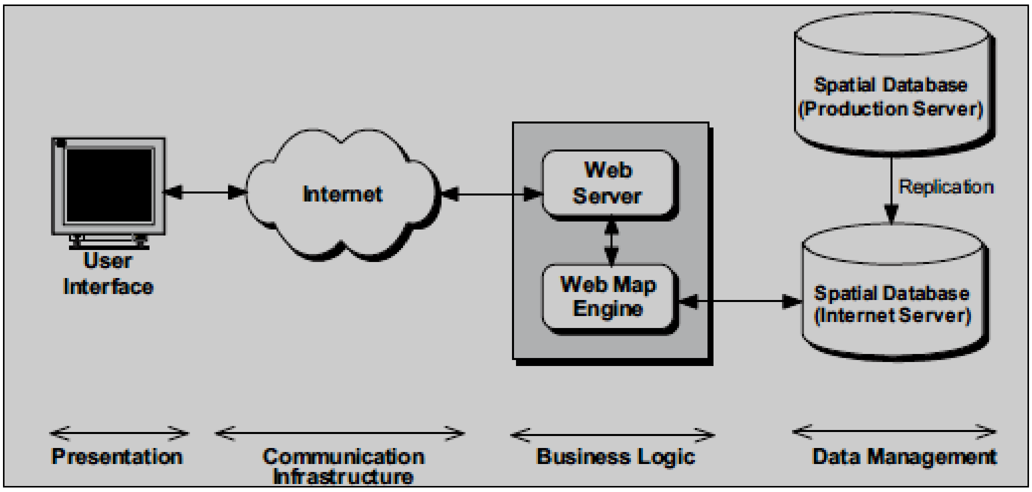


Figure 5.1:WEB map: Client – server four tiers architecture [7].

ArcGIS Server, Internet Map Server (IMS), MapInfo, MapXtreme and GeoMedia WebMap are some examples to corporations using this architecture. There are also open source products available free of charge, such as ALOV Map from the University of Sydney, Australia, MapServer from the University of Minnesota are some of spatial database examples designed according to this architecture.

Although WEB services give almost everything to geospatial data users, they are so expensive systems and requirement for big servers to maintain these services force random users to search another tools. Online, free, open source mapping tools such as Google Earth, Google Maps, Yahoo Maps and Bing Maps fastened spatial data distribution via Internet that user can also query addresses, locations and routes through these map interfaces.

5.5 Google Maps

Internet age has brought so much new applications but the year of 2005 will be remembered as the year for Geomatics community. That year Google released beta version of Google Map which was first free mapping service include satellite maps that user able to view any location (Figure 5.2) that represented by set of coordinates on the earth [19].

Figure 5.2: Location search of control stations according to coordinates.

Google Maps may look like exceptional tool but it is a result of a simple structure. It's just combination of HTML, Cascaded Style Sheet (CSS), and JavaScript. Satellite images loaded in the background with Ajax calls and then inserted into a <div> tags in the HTML page. As you visited different place in the map, the API sends data about the new coordinates and zoom properties of the map in Ajax calls that return new images [20].Google Maps use WGS 84, which is the same reference system of GPS measurements. The coordinates in the map are expressed by latitude and longitude in decimal degrees.

5.5.1 Google maps application interface

The Google Maps API is based on a very simple suite of classes, within a JavaScript container, directly within an HTML page. All of the functionality of a Google Map is based on this simple expound of an embedded JavaScript object into the HTML. JavaScript is able to build into most browsers with functionality for creating dynamic applications. At the time this thesis was written Google Maps API was in version 3.
All Google Maps applications begin with a simple map. To this map, you can add some of elements designed to gain the application functionality. All maps support four main elements:

- **Overlays:** Points of interest, lines define areas, routes and other information that being displayed on a map.
- **Events:** Operations, such as the user clicking on a point of interest, info window's appearances and etc.
- **Information:** Broaden data about the map, markers on the map, or map region as a part of the map.
- **Controls:** Main controls elements that enable the user to change zooming range and moving the map [21].

Markers as overlay are the main focus in our work since we want to represent GPS /GNSS control stations data in a database as a marker (Figure 5.3) in Google Maps. The markers identify object’s location and before adding markers, latitude and longitude information of the control station should be determined first. Then, JavaScript codes will locate the CS (5.1) to called map in previous action [22]. Google maps use a standard marker icon by default that users can change it with another symbol.

varmyLatlng = new google.maps.LatLng(41.363882, 27.044922); **(5.1)**
var marker = new google.maps.Marker({
position: myLatlng, map: map, title:"Hello World!" });

Figure 5.3: Adding control station to the map via Google Maps API.

Creating and working with dynamic interface through Google Maps requires preparing Maps API readable documents. Flat text files and XML files are widely used formats in Google Maps. The files including GPS control stations list should be converted to XML file format before embedding into the HTML.
XML file has an advantage of being interoperable with Google Maps to maintain data exchange and simplify operations for storing, updating and displaying CS data on Google Map. Although XML language is not so easy to write by hand, there are some bridge tools that converting source data of HTML and PHP into the XML type for making data readable by Google Maps. Generally, the key is to generate information that can easily be parsed by JavaScript to display information.
The basic component of XML language is the <tag> which is a single word enclosed in angle brackets (5.2).

<Row><Longitude>-97.772588</Longitude> **(5.2)**
<Latitude>30.249405</Latitude>
<Pt_Name>ANKR_345</Pt_Name></Row>

We had mentioned about MySQL database in part three. In this work XML was used as a bridging tool between MySQL database and HTML display page of Google Maps. The Maps API contains a method called GXmlHttp() which is used to create a browser neutral XmlHttpRequest() object and pull all coordinates of XML into HTTP to present them in the map(Figure 5.4).

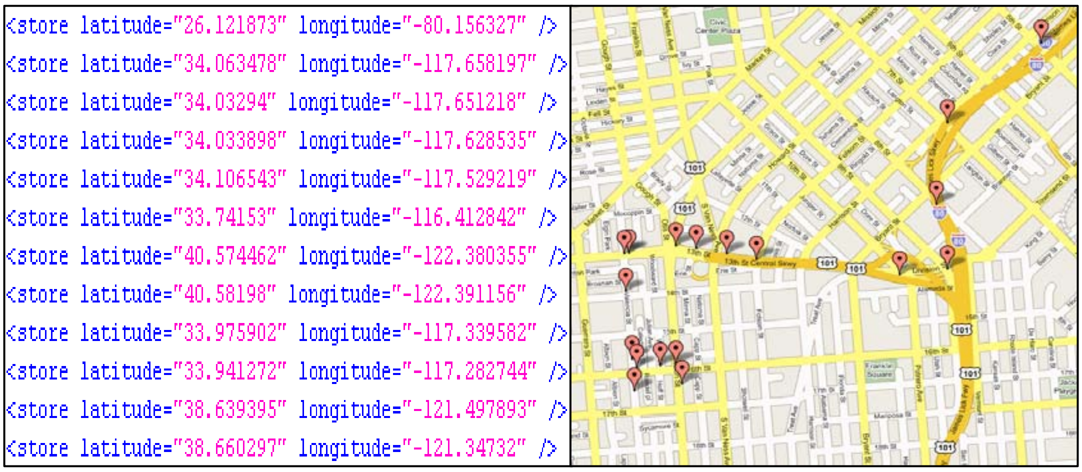


Figure 5.4: XML output of CSs on the map called by GXmlHttp() method.

6. WORK: MySD (MAP & MANAGE YOURSPATIALDATA)

6.1 Introduction to MySD

6.1.1 What is mysd?

- MySD is a WEB based relational database driven software which enables users store and manage their GPS / GNSS control stations data on the WEB environment. MySD is the abbreviation of Map &Manage your Spatial Data or with more technical definition it is a Web based dynamic MySQL spatial database with online Google Maps support. It embraces different and especially open source programming languages for one specific purpose: GPS / GNSS control station data management via web platform in which user is abstracted from the physical view of database.
- MySD comprises of MySQL database for storing data, dynamic WEB interfaces produced by PHP for interaction with the database and online mapping service of Google Maps for displaying location of GPS /GNSS control stations.
- Users of MySD access to the database after some basic registration process. MySD users are classified into three different account types including different level of privileges of data management. MySD users manage their data without any conflicts coming from other users.. Moreover, MySD serves as a guide during data storage and management in case of any wrong attempt.
- Google Maps API integrated MySD also has the ability of representing GPS / GNSS control station data on the Google Maps according to their geographic coordinates latitude and longitude defined in WGS 84 reference system. Therefore, users will receive control station information not only in tabular forms but also on the map with real world location satellites view.
- MySD is free source software which is open to additional improvements.

- MySD primarily designed for storing GPS / GNSS data. Additional geospatial tools of wide range of Geomatics engineering applications will be implemented in the future.
- MySD is working well with most internet browsers such as Google's Internet Explorer, Mozilla's Firefox, Safari and Opera. The software was not tested in Linux environment yet but it is expected to work without any problem.
- MySD users are not affected from any changes in the database structure and upgrades in software.

6.1.2 What is mysd not?

- MySD is not just a tool for storing GPS / GNSS control stations and related data. Users of MySD query the data with specific requests and each request bring new application interface with different results on browser's view.
- MySD is not static software. Since it is WEB based and open to public, data transitions, uploading, updating and removing attempts happen all the time, simultaneously.
- MySD is not a complete GIS tool in which data is analysed and results are represented graphically. These advanced tools are in development stage and will be available in the future.
- MySD is not a transformation or mathematical toolbox for GPS data. It is user responsibility to overcome mathematical requirements of data before implementing it into the database.
- MySD does not claim unlimited access and countless free operations during data management. Since the information are valuable today, accessing to the data sets and data management interfaces are restricted according to software's data sharing policy.
- MySD is not stand alone software program that requires any install wizard or setup exe file. It is WEB based software which means without internet connection it is not accessible unless you are working on your local host or local intranet.

6.1.3 Necessity of mysd in Geomatics

Most of the problems in Geomatics, are data related. Difficulties during data collection, inappropriate data storage techniques and gathering unrelated spatial data for a particular purpose causes waste of time and money. In addition, lack of regulations about data storing and sharing procedures is another issue in Turkey.These problems often results with inefficient outputs andproducts, unsustainable projects and failed investments.

The main concern of MySD is not only storing and publishing data but also helping to determine standards of data management and increase interoperability between private and public organizations for improving decision making mechanism.

Since the internet is the most powerful tool for sharing information, MySD was designed by considering all IT and WEB tools that have been developed recently. Being independent from any hardware, software and complicated system requirements, users can operate their data any time, from anywhere, simultaneously, without disturbing others. So instead of creating stand-alone software, MySD is designed as web tool that freely available to everyone with some restrictions.

6.1.4 GPS data storage

In Geodesy, GNSS is the widespread and the most modern measurement technique for getting accurate coordinate data of objects. Coordinates as mentioned before are considered as the most important units of spatial objects. Today, many disciplines need to use location information in their projects. Geodesy, remote sensing, photogrammetry, surveying, geology, geophysics, environmental planning, irrigation, construction, archaeology and disaster management use GNSS coordinates data directly or as an auxiliary data sets in their databases.

Because of majority of numeric values as geographic coordinates coming from GPS / GNSS measurements which are easy to store in the database, MySD was designed for being able to work with GPS / GNSS control stations data. In addition, geographic coordinate pair latitude and longitude are needed any way during visualization process via Google Maps.

Despite the fact that most of today's modern GPS receivers include their own recording units for storing coordinates or computer connection tools for increasing recording capacity during measurements, they are not designed as a basic database. Queries, gathering specific GPS data from different measurements in one particular environment and multiple user connections are beyond the scope of these storing units.

6.2 Architecture of MySD

Working principle of MySD depends on a basic client – server architecture (Figure 6.1) on the WEB environment. The clients send their requests of map or data to the server. Then the server processes these requests and sends results to the client with Google Map.

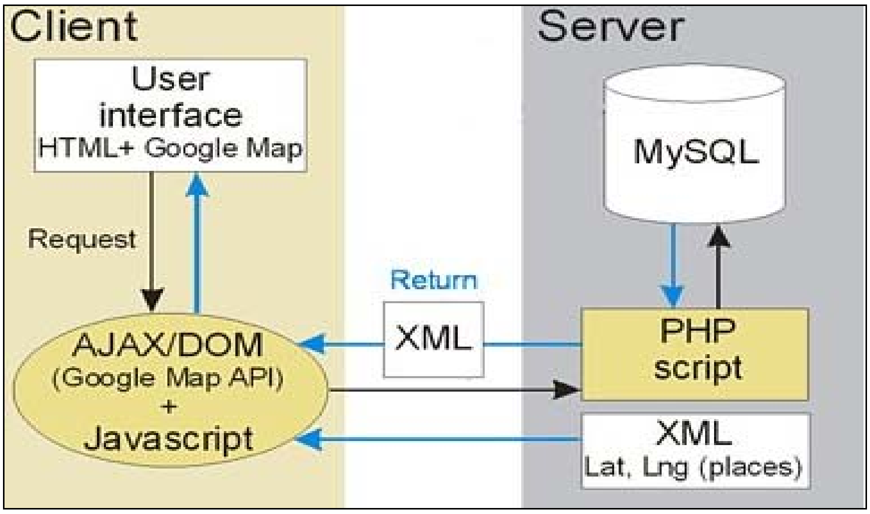


Figure 6.1: Client- Server architecture depicts working principle of the work.

The request would be packaged according protocols supported and delivered by the operating system. The server application would then verify the request and send back either the GPS project data on the WEB page (if the request meets the appropriate criteria) or else generate an error message. The operating system ensures that the messages are delivered successfully and the applications concerned are responsible for dispatching and handling requests for services in the appropriate manner.

6.2.1 Database design of mysd

Relational database design for MySD can be divided into some traditional steps as which are parts database development life cycle (Figure 6.2).

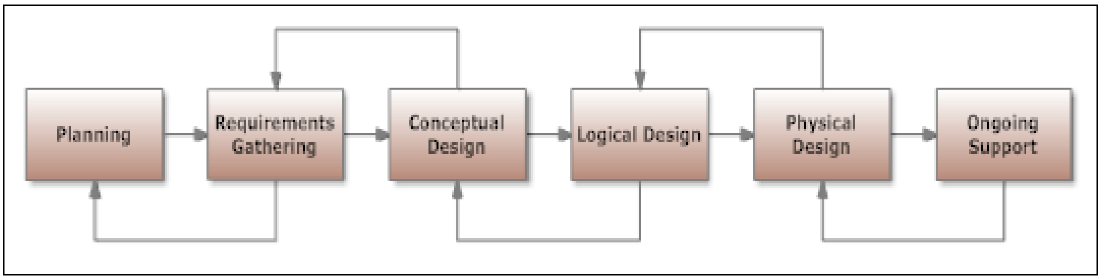


Figure 6.2: Relational database design phases of MySD.

Planning: In this phase, the goals of the GPS projects compared with the latest database technology to ensure that the GPS projects can reasonably be expected to be successful using that database. If more recent version of the DBMS is needed, or if different DBMS is required, the planning phase would give right answers for that.

Requirement Analyses: In this phase of design, the data to be stored in a GPS database is determined. The main focus in this phase is determining of what instead of how. What kind of GPS data and database is needed to fulfil user's requirements? This phase was an informal process that involves:

- Interviews with Geomatics user groups and Geomatics engineers,
- Surveys that conducted on potential users of GPS / GNSS database,
- Document reviews that gathered from private and public constitutions.

According to researches and surveys one schema was created in mind and visualized in SmartDraw 2010 to create the primary view of design (Figure 6.3).

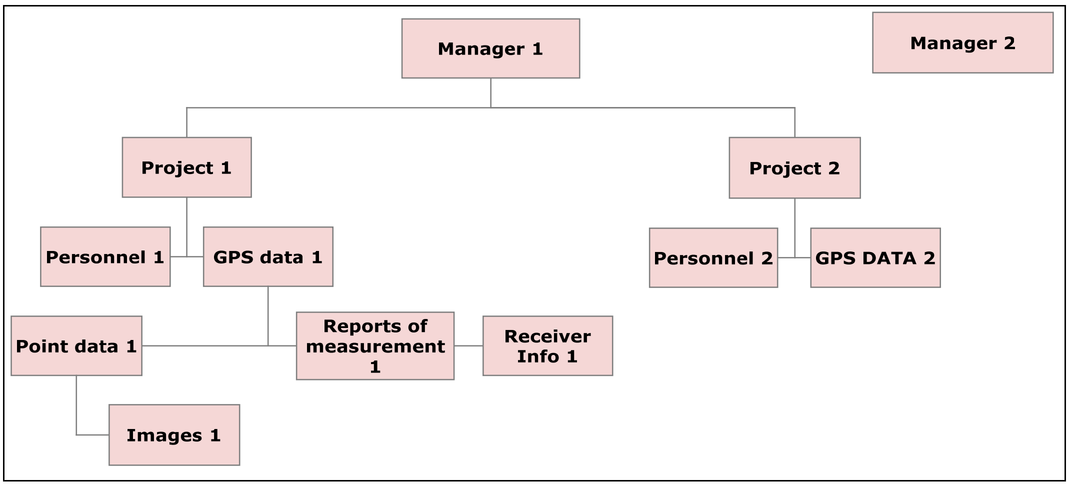


Figure 6.3: First visual design of plan after requirement analyses.

According to requirement analyses, basic data needs for MySD database were determined:

- Data of manager or corporation which conducts GPS / GNSS project. Manager’s area of study, contact information and some descriptive data.
- Project data which includes project description, project location, contact information and other definitions to represent project.
- GPS / GNSS control station data which vary from station name, latitude, longitude, ellipsoidal height to station protection type, date of measurement, velocity of station, reference epoch, standard deviation, orthometric height, address of station and receiver used in station measurements.
- Personnel data to give necessary information about project’s personnel who work as employee in projects.
- Image data of control stations and also report files to explain project results.

6.2.2 Conceptual database design for mysd

In the conceptual design, the high level description of data in MySD database was developed in terms of entity – relationship (ER) model. Manager (Figure 6.4), project (Figure 6.5), datum &projection (Figure 6.6), personnel (Figure 6.7), receiver (Figure 6.8), images and report (Figure 6.9), and GPS (Figure 6.10) are tables in the database and they are first designed as ER model before transformed into relational table structures in logical design phase.

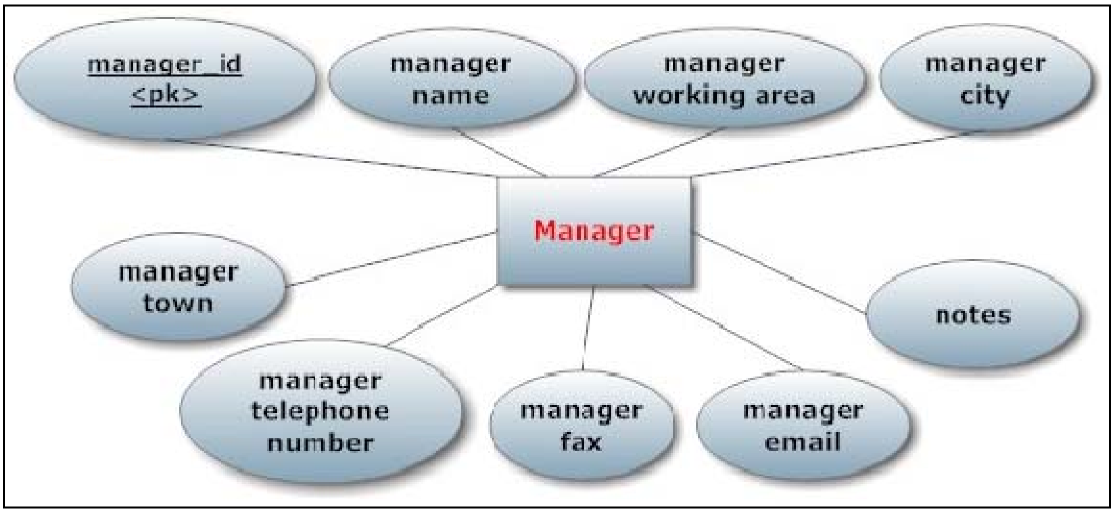


Figure 6.4: Manager entity and its attributes.

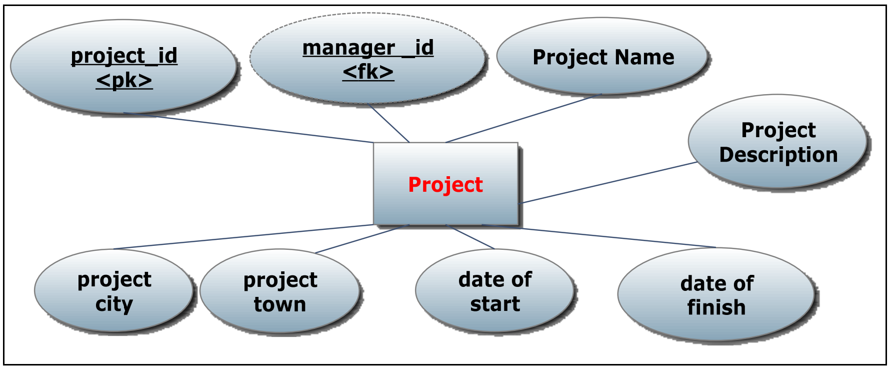


Figure 6.5: Project entity and its attributes.

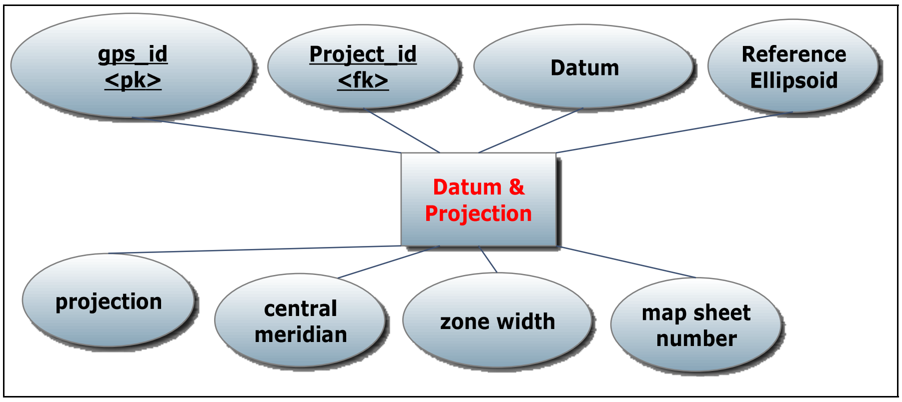


Figure 6.6: Datum-Projection entity with attributes.

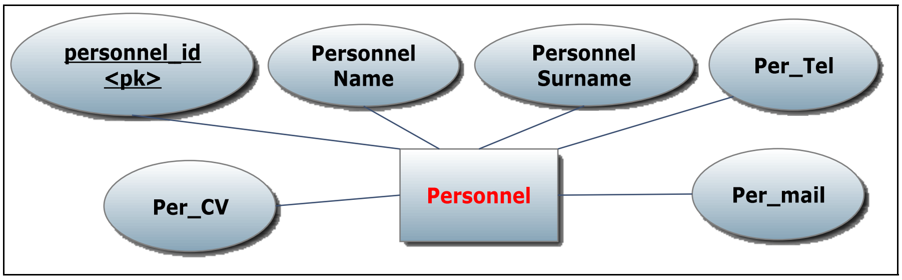


Figure 6.7: Personnel entity and its attributes.

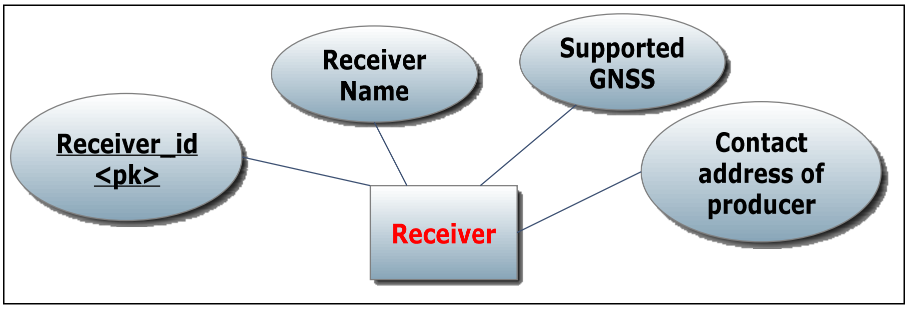


Figure 6.8: Receiver entity and its attributes.

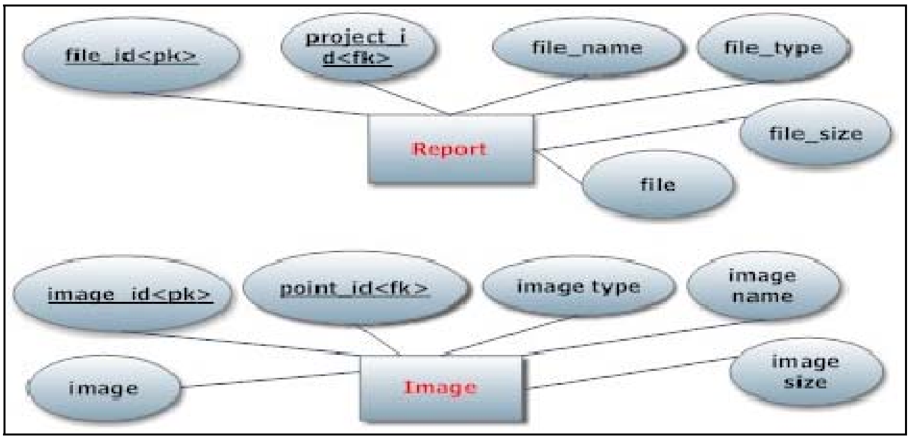


Figure 6.9: Report and image entities and their attributes.

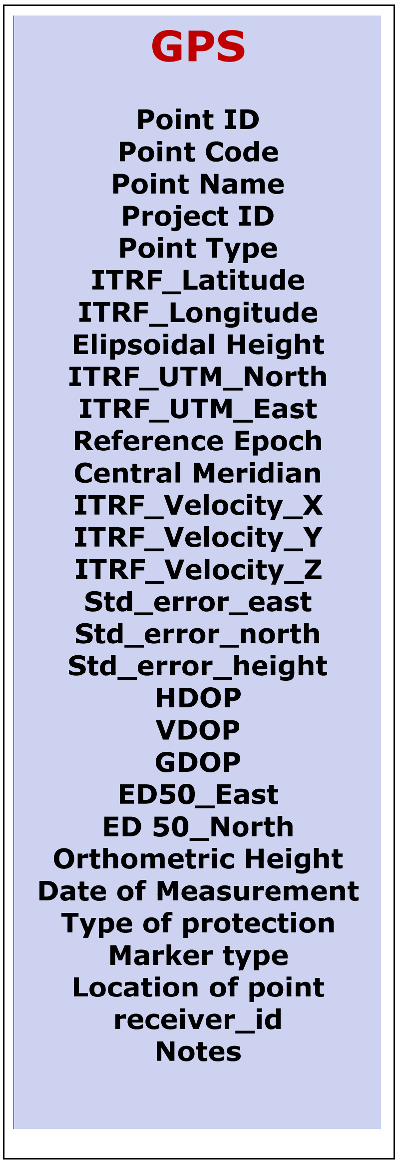


Figure 6.10: GPS entity and attributes in the conceptual design phase.

6.2.3 Logical database design of mysd

Logical design phase includes transformation of conceptual design into logical model. Thus, ER model (Figure 6.11) which was established in conceptual design phase transformed into the relational table structure. In this phase, relational database model is created which is independent from any DBMS and physical environment. Each simple attribute in ER model becomes a column header in relational table.

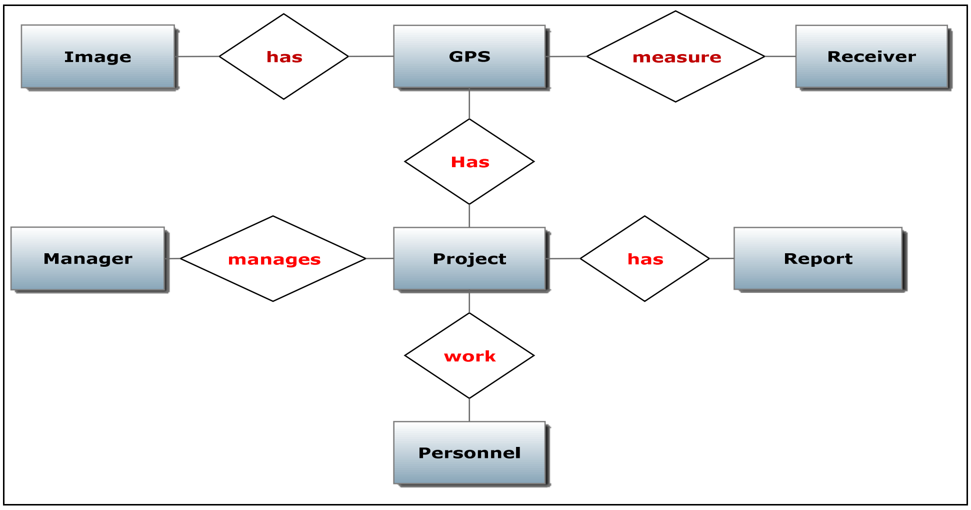


Figure 6.11: ER model of the software database in the conceptual design.

For the purpose of creating internal part of MySD database eight tables were created according to previous ER model.

Manager table (Table 6.1) includes project manager attributes.

Table 6.1: The manager table in logical design phase of the database.

Field	Data Type	Description
manager_id<pk>	int(11)	Unique manager identification number.
manager_name	varchar(255)	Manager or corporation name.
manager_area_of_work	varchar(500)	Describing working areas of this corporation or manager.
manager_city	varchar(255)	Manager city.
manager_town	varchar(255)	Town of the manager.
telephone_number	int(20	Telephone number: numeric value
fax_number	int(20)	Fax number: numeric value
Email	varchar(255)	Email address of corporation
PRIMARY KEY:		**manager_id**

Project table (Table 6.2) that includes random attributes necessary to define a project,

Table 6.2: The project table in logical design phase of the database.

Field	Data Type	Links to	Description
project_id<pk>	int(11)		Unique project identification number.
manager_id<fk>	int(11)	Manager table >corporation_id	Unique identifier of project manager.
project_name	varchar(255)		Project name.
project_description	varchar(500)		Project description.
project_city	varchar(255)		City of the project.
project_town	varchar(255)		Town of the project.
date_of_start	date(d:m:y)		Start date of project.
date_of_finish	date(d:m:y)		Finish date of project.
PRIMARY KEY:		***project_id***	
FOREIGN KEY:		***manager_id***	

Personnel table (Table 6.3) includes main data about project personnel.

Table 6.3: The personnel table in the logical design phase of the database.

Field	Data Type	Links to	Description
personnel_id	int(11)		Unique id number for personnel of project.
project_id	int(11)	Project table ->project_id	Project id number to define a particular project.
Personnel_name	varchar(255)		Name of personnel.
Personnel_surname	varchar(255)		Surname ofpersonnel.
Personnel_tel	int(255)		Telephone number of personnel.
personnel_mail	varchar(255)		Mail information of personnel.
personnel_cv_pdf	longblob		CV file of personnel (word, pdf, txt, jpeg).
PRIMARY KEY:		***personnel_id***	
FOREIGN KEY:		***project_id***	

The GPS table (Table 6.4) includes basic values of GPS measurements and some data obtained after post-processing.

Table 6.4: The GPS table in logical design phase of database.

Field	Data Type	Links to	Description
point_id	int(11)		Unique number to define a particular control station (CS).
point_no	varchar(255)		Control station number: be determined according to regulations in country.
point_name	varchar(255)		Name of the CS defined by project team.
project_id	int(11)	Project table -> with project_id	Define unique id of project that CS are measured for.
Itrf_latitude	decimal(12,8)		ITRF latitude of CS in decimal degree format.
Itrf_longitude	decimal(12,8)		ITRF longitude of CS in decimal degree.
Ellipsoidal_height	decimal(8,3)		Ellipsoidal height of CS.
Itrf_Utm_east	decimal(11,4)		Universal Transvers Mercator easting coordinate of CS.
Itrf_Utm_north	decimal(12,8)		Universal Transvers Mercator northing coordinate of point.
Reference_epoch	float(6,2)		Reference epoch: an instant in time chosen as the origin of a particular period for survey.
Central Meridian	int(3)		A **meridian** (or **line of longitude**) is an imaginary arc on the Earth's surface from the North Pole to the South Pole that connects all locations running along it with a given longitude.
Itrf_velocity_X (m/year)	float(6,4)		Velocity (movement per year) of CS in Cartesian X direction.
Itrf_velocity_Y (m/year)	float(6,4)		Velocity (movement per year) of point in Cartesian Y direction.
Itrf_velocity_Z (m/year)	float(6,4)		Velocity (movement per year) of CS in Cartesian Z direction.
Std_deviation_East	decimal (5,4)		Standard deviation of easting value with the measurements.
St_deviation_North	decimal(5,4)		Standard deviation of the northing value associated with the measurements.
St_deviation_Height	decimal(5,4)		Standard deviation of the Z associated with the measurements.

HDOP	float(8,3)		Horizontal dilution of precision (Error causes of satellites geometry).
VDOP	float(8,3)		Vertical dilution of precision.
GDOP	float(8,3)		Geometric dilution of precision.
ED50_east	float(12,4)		East coordinate of CS in ED50 datum.
ED50_north	float(12,4)		North Coordinate of CS in ED50 datum.
Orthometric_height	float(8,4)		Orthometric height of a CS that used for engineering application.
date_of_masurement	date (d:m:y)		Measurement date of point.
type_of_protection	varchar(255)		Materials that used to protect CS (barrier, fence)
Marker_type	Varchar(255)		Defines material used as a CS on the surface.
Location_of_point	varchar(255)		Address of measured control station (zip-code, street, town and city.)
receiver_id	int(11)		Used to define one particular receiver used in measurement.
notes	varchar(255)		Other notes to describe the control station.
PRIMARY KEY:		***point_id***	
FOREIGN KEY:		***project_id***	

The image table (Table 6.5) was designed to store image file of each point that defined in previous GPS table.

Table 6.5: The image table in logical design phase of the database.

Field	**Type**	**Links to**	**Comments**
image_id	int(11)		Unique number for image file
point_id	int(11)	GPS table->point_id	Point id represents a particular measured CS in survey.
image_name	varchar(255)		Name of the image.
image_type	varchar(255)		Defines image's format jpeg or png.
Image	longblob		Photograph itself as BLOB file
width	int(11)		Width of photo in pixel size
height	int(11)		Height of photo in pixel size
caption	varchar(255)		Text that entered by user to define photo's explanation or
PRIMARY KEY:		***image_id***	
FOREIGN KEY:		***point_id***	

The report table (Table 6.6) to store results files (PDF or MS.Word) of projects and receiver table (Table 6.7) to store receiver sets used in projects.

Table 6.6: The report table in logical design phase of the database.

Field	Data Type	Links to	Comments
file_id	int(10)		Unique number to define project file.
file_name	varchar(255)		Name of the file.
file_type	varchar(255)		Type of the file:Word, PDF, TXT.
file_size	int(11)		Size of file in byte.
file	longblob		File itself as long-blob type.
project_id	int(255)	Project table ->project_id	Integer number to represent a particular project.
PRIMARY KEY:		***file_id***	
FOREIGN KEY:		***project_id***	

Table 6.7: The receiver table in logical design phase of the database.

Field	Data Type	Links to	Description
receiver_id	int(11)	GPS table> receiver_id	Unique id for receiver used in GPS survey.
receiver_name	varchar(255)		Receiver product name.
supported_gnss_systems			Supported GNSS system.
web_producer	varchar(255)		Web address of receiver producer.
PRIMARY KEY:	***receiver_id***		

Datum & projection (Table 6.8) table to store initial data before survey.

Table 6.8: Datum-projection table in logical design phase of the database.

Field	Data Type	Links to	Description
gps_id	int(11)		Unique number to identify a particular GPS survey id.
project_id	int(11)	Project table ->project_id	Project id of GPS survey.
datum	varchar(255)		Datum information of GPS survey.
ellipsoid	varchar(255)		Ellipsoid information of GPS survey.
projection	varchar(255)		Projection information of GPS survey.
central_meridian	int(2)		Central meridian of GPS survey area.
zone_width	int(1)		Selected zone width, optional for survey.
map_sheet_no	varchar(255)		Map sheet number of GPS survey area.
hemisphere	Varchar(20)		Define hemisphere for work: North or South.
PRIMARY KEY:		***gps_id***	
FOREIGN KEY:		***project_id***	

Logical design process is managed by database designer MySQL Workbench 5.2 (Figure 6.12) software which is free and available on the MySQL web site (www.mysql.com). All schemas and relations between tables are designed and depicted with diagram (Figure 6.13) by using its advanced tools.

Figure 6.12: MySQL Workbench 5.2 softwareinterface.

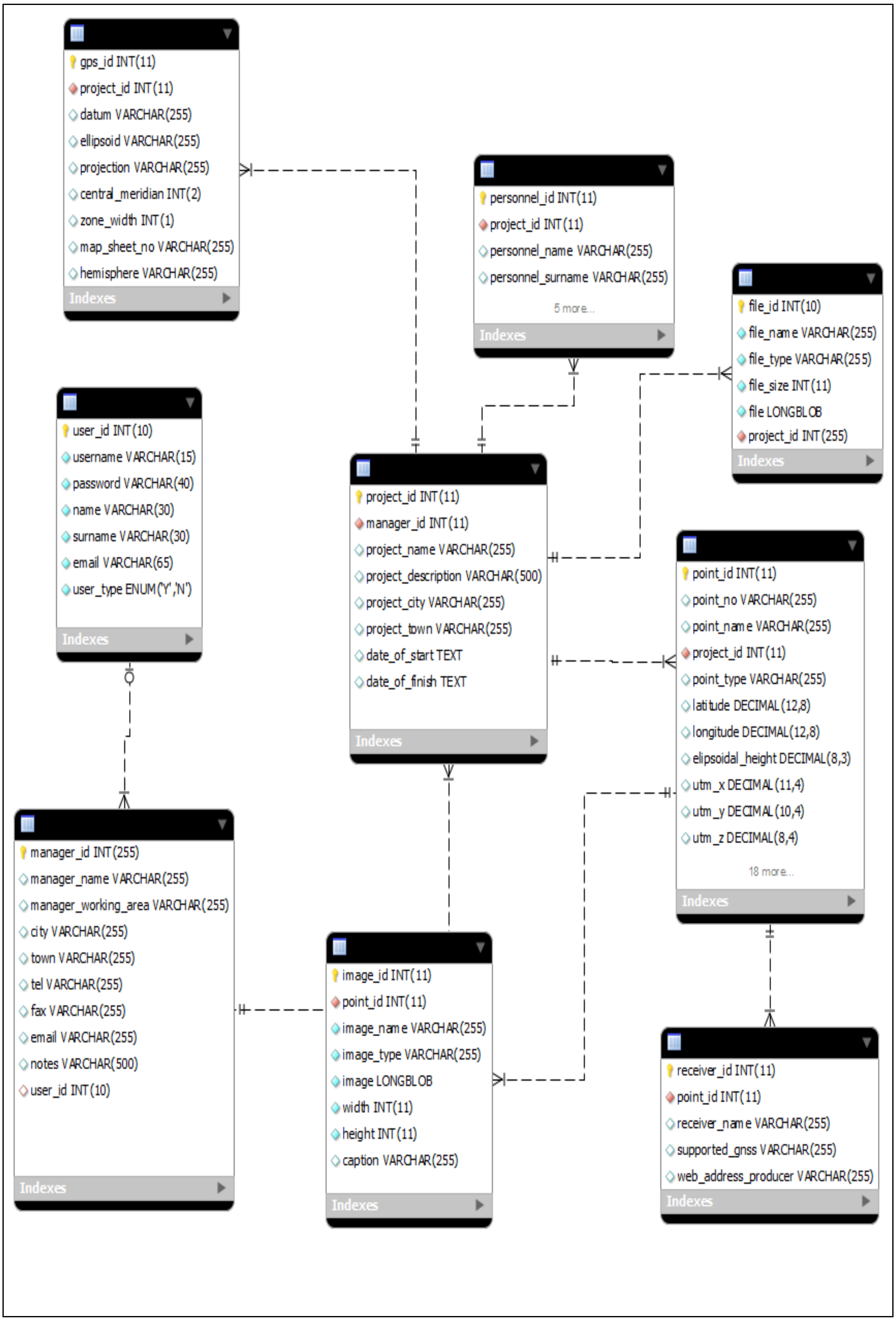


Figure 6.13: Diagram of the logical design to define relational tables.

Relations among tables are displayed (Table 6.9) with help of Smart-Draw 2010 engineering notations.

Table 6.9: Relation types between tables in logical design phase.

Table	Table	Relation	Description
Manager	Project	many-to-many	One manager can manage one or multiple projects and also one project can be managed by multiple managers.
Project	GPS	one-to-many	Each project can have multiple GPS surveys. But one GPS survey belongs to only one particular project.
Project	Projection-datum	one-to-many	Each project can have one and more datum-projection value. But one particular datum-projection value belongs to only one project.
Project	Personnel	one-to-many	There are many workers in one project but each particular worker belongs to one particular project.
Receiver	GPS	one-to-many	There are many CS points can be measured by one particular receiver while one particular point is measured by only one particular receiver.
Project	Report	one-to-many	Each project has many reports but no any report belongs to more than one project.
GPS	Images	one-to-many	Each CS has many images but each image can be used to define only one CS.

6.2.4 Physical database design of mysd

The structure including relational tables derived from logical design phase saved as independent SQL file to be ready for use in any MySQL DBMS. Then, SQL file was imported by PhpMyAdmin which was chosen to be the MySQL DBMS interface behind MySD. Transactions, security, recovery, and all other database integrity operations related with physical design are conducted by PhpMyAdmin's MySQL DBMS interface.

6.3 Database – WEB integration for MySD

6.3.1 PhpMyAdmin

Although MySD can be managed by simple MySQL command prompt, it is much easier to use user friendly graphic interface of PhpMyAdmin (Figure 6.14). PhpMyAdmin is not only the most popular free graphical interface for MySQL but also having PHP based management system by which users create, update and manage their data in the relational database by using dynamic WEB pages. PhpMyAdmin can be downloaded from http://www.phpmyadmin.net.

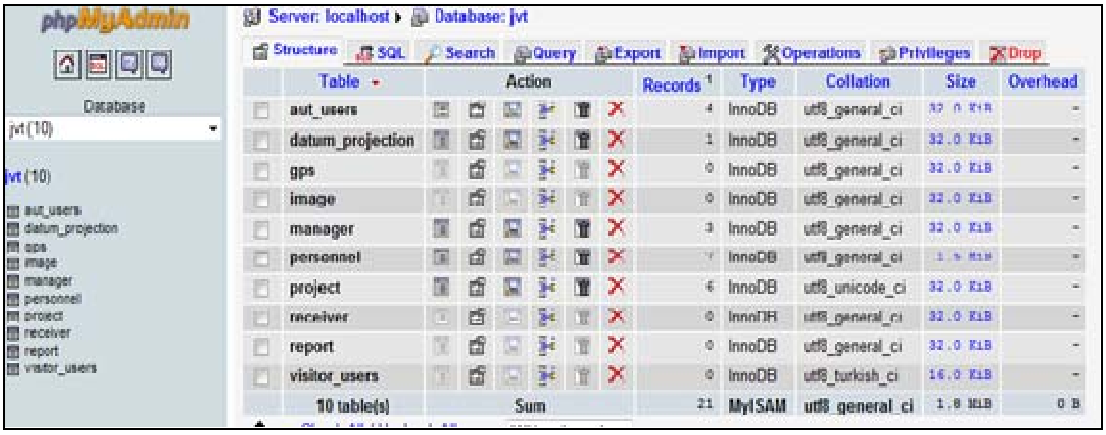


Figure 6.14: PhpMyAdmin: Graphical interface of MySD's database.

6.3.2 PhpMyAdmin – Dreamweaver integration for web design

Developed by Macromedia, Dreamweaver is the name of the software that used to create the dynamic web pages for MySD application interfaces. Interoperability with MySQL, PhpMyAdmin and all other Adobe products (Photoshop, Flash and Fireworks) is the main reason of Dreamweaver for being used as design programme of MySD WEB interfaces. During WEB design of software, both, code screens and Dreamweaver's graphic design interfaces were combined to achieve desired result.

Registration and login: MySD was designed for being accessible to everyone via internet. But, this access and login attempts were ruled with some privileges and restrictions. According to type of user account, users are classified in order to view part of database and manage database operations.

For this purpose three account types were determined in MySQL (Figure 6.15). First, “**gdb_authorize**” who has all privilege of data management also known as moderator or DBA. The second one, “**gdb_user**” account is able to access all data but authorized to manipulate only his records while the “**gdb_visitor**” account is allowed to view data but prevented from altering any of them.

	User	Host	Password	Global privileges [1]	Grant	
	gdb_authorize	localhost	No	ALL PRIVILEGES	Yes	
	gdb_user	localhost	No	SELECT, INSERT, UPDATE, DELETE, FILE	No	
	gdb_visitor	localhost	No	SELECT	No	

Figure 6.15: Defined user account types in MySQL.

User will define himself to the system via basic registration form including name, surname, contact information and moreover will define a username and password.

Figure 6.16: User registration and login form in MySD.

According to username and password filled in the login page, the system will assign user to one of the account types defined in MySQL. Therefore, user’s privileges and access restrictions will be identified. In MySD, sessions of the WEB site will be different for different user account.

Insert data: Data insertion and uploading via WEB form for populating database tables in MySQL, is managed by PHP form operation called “POST” method. This method provides data transfer from client to server. User sends data to database by filling fields in the WEB page. Manager, Project, Datum, Personnel tables were populated via this method.

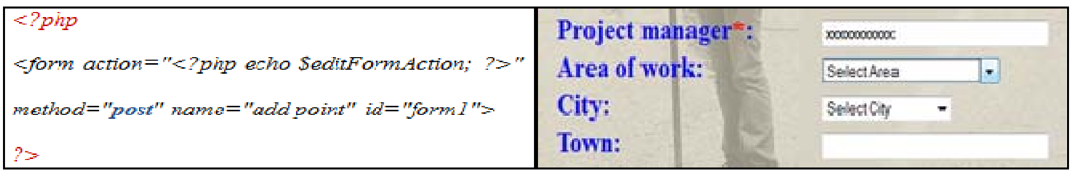


Figure 6.17: Code and visual “POST” method sending data via PHP forms.

Import Point Data: Instead of importing GPS data for each point each time, an application (Figure 6.18) was developed by using PHP which imports “csv” or “txt” file containing multiple GPS / GNSS control station data and assign their values to the MySQL database.

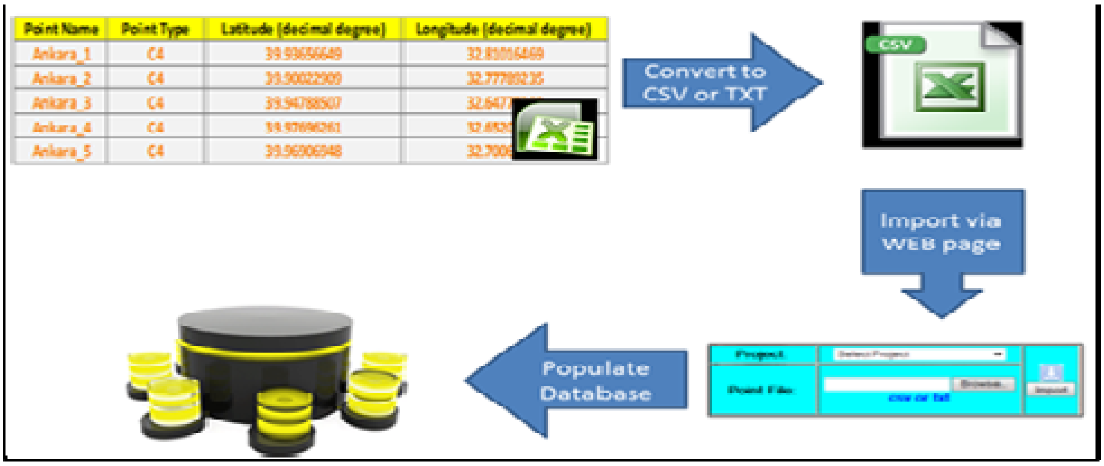


Figure 6.18: Process diagram for importing GPS data into database.

The **sample excel file** was linked in the web page for downloading. Thus, users can add their own GNSS stations values to suitable columns in excel file. The excel file should use point (.) separator instead of comma (,) for decimal part of data for successful importing process.

Loading image tool: Image loading tool was designed to load one image for each CS. The widely used **png, jpeg** and **gif** are some of the acceptable image formats for loading. Loaded images are stored as a BLOB data type in MySQL database.

Loading Personnel and report file: For the purpose of loading personnel files and brief project reports into the MySD database, another data loading tool was developed. With this tool users are able to send their Portable Document Format (PDF) and Microsoft Word (MS Office Word) files to the database without any version limitation.

The tool was not developed for simultaneous files upload to the database but this ability will be upgrade in the future.

Update and remove data: Data updating and removing processes and their WEB interface design are complicated procedures. Because, whenever rows in the master (referenced) table are deleted or updated, the respective rows of the child (referencing) table with a matching foreign key column will get deleted or updated as well. This is ensured by “CASCADE” actions which defined in MySQL.

For example, delete one manager record in the database (Figure 6.19), will cause of failure of all project, personnel, point and image and report file related with it in the database.

Figure 6.19: Delete action in MySD WEB interface.

Results of that action should take into consideration by user. Because, this move is irreversible and unless the database has the backup, all data will be gone.

6.3.3 Queries

PHP, server side scripting language can be embedded into HTML. This ability makes it prominent tool in dynamic web applications for sending user requests to and retrieving related responses from the database. These requests in MySD were defined by SQL queries in PHP.

For example, when user search for the CS and its project table information in the database, the request is conducted via SQL query (6.3) which including relations between tables defined by foreign keys and specific requests for extracting rows related to point with ID number 10.

```
SELECT  point, manager,  project_name, project_description, city, town            (6.3)
FROM manager AS a, project AS b,  gps AS c, imagesAS d
WHERE a.manager_id = b.manager_id
AND b.project_id = c.project_id
AND c.point_id = d.point_id
AND WHERE point_id = 10;
```

Furthermore, multiple searching forms were created in MySD. Therefore, users have opportunity to search specific piece of data. An example for SQL code (6.4) by which the system will search only the entry coming from users.

```
SELECT latitude, longitude,                                                        (6.4)
utm_x, utm_y, utm_z, point_type
FROM gpsWHERE point_code = 'user_entry';
```

6.3.4 Google maps api integration to database

The complex database structure with huge amounts of data is beyond the scope of ordinary user's perception. In addition, it is a well know reality that people can grasp and understand information which represented visually much more effortless comparing to the written ones. Therefore, open source online mapping service Google Map was integrated to the MySD to fill the gap of visual displaying tools of the database.

The Google Maps which is gathered by satellite images and aerial photographs including different sets of data have been using by most Geomatics engineers not only for searching an address or location of objects but also for implementing GIS applications, producing thematic maps and creating 3D terrain models.

In this work GPS / GNSS control stations data have to be processed in order to being readable by Google Maps. This is Google Maps API's responsibility to create user defined maps which displaying control stations coming from MySQL database. JavaScript, XML, AJAX and PHP were the languages used here which provide dynamic data flow between database and Google Map Interface.

To sum up the basic steps (Figure 6.1) for Google Map API integration:

- MySD users who want to submit or retrieve particular piece of control stations' data, use Google Map API which is embedded in client side HTML (Figure 6.20) pages. Usually, the whole process is triggered by clicking on a marker or on a submit button on the map.
- AJAX takes these requests and interacts with PHP for sending these requests to the MySQL database.

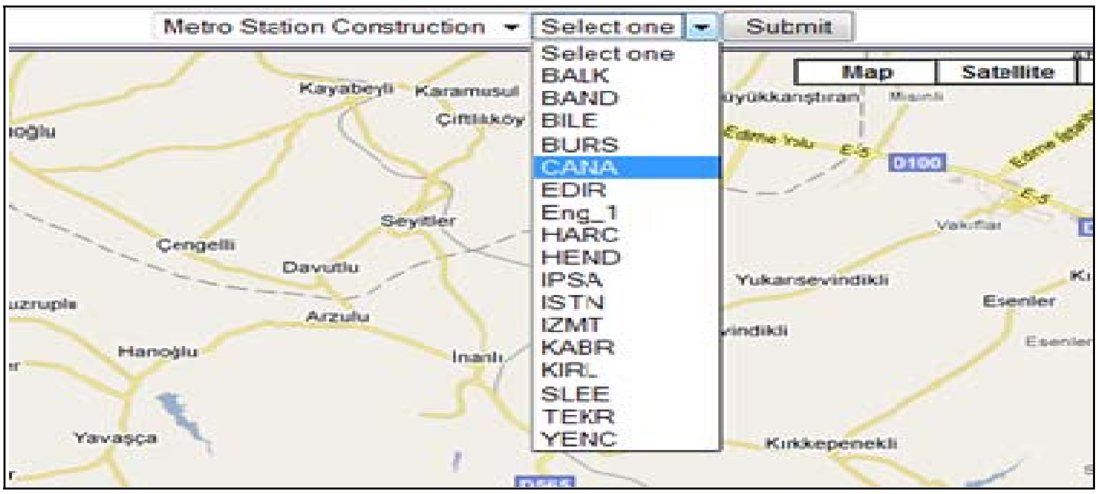
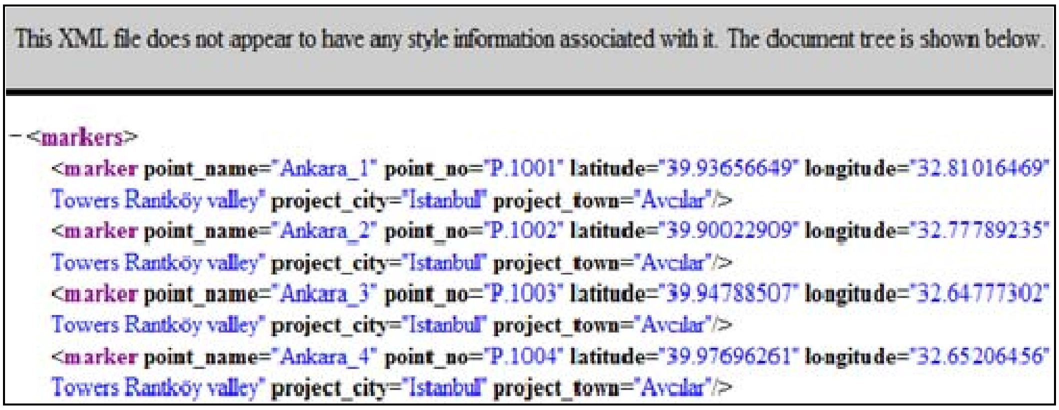


Figure 6.20: User request on the Google Map with submit button.

- In the server side DBMS receives these requests and extracts the particular data from tables after processing PHP codes in the HTML page.
- **DOM_XML:** PHP uses the DOM_XML or echo_xmlfunctions to take care of transforming data into the XML format (Figure 6.21). The XML can be regard asreconciledformat between our relational database and Google Maps API.


This XML file does not appear to have any style information associated with it. The document tree is shown below.


```
- <markers>
    <marker point_name="Ankara_1" point_no="P.1001" latitude="39.93656649" longitude="32.81016469"
    Towers Rantköy valley" project_city="Istanbul" project_town="Avcılar"/>
    <marker point_name="Ankara_2" point_no="P.1002" latitude="39.90022909" longitude="32.77789235"
    Towers Rantköy valley" project_city="Istanbul" project_town="Avcılar"/>
    <marker point_name="Ankara_3" point_no="P.1003" latitude="39.94788507" longitude="32.64777302"
    Towers Rantköy valley" project_city="Istanbul" project_town="Avcılar"/>
    <marker point_name="Ankara_4" point_no="P.1004" latitude="39.97696261" longitude="32.65206456"
    Towers Rantköy valley" project_city="Istanbul" project_town="Avcılar"/>
```


Figure 6.21: Requested data in XML that readed and parsed by PHP.

- Once the XML is working in the browser, it's time to move on to actually creating the map with JavaScript,
- Re-create the map upon loading by inserting new markers from a server-side list, each with an info window to display its information,
- User side HTML load the same page and dynamic map with the new data sets can be seen on the browser (Figure 6.22). The Internet Explorer, Google Chrome and the Mozilla Firefox preview the user requests without a problem.

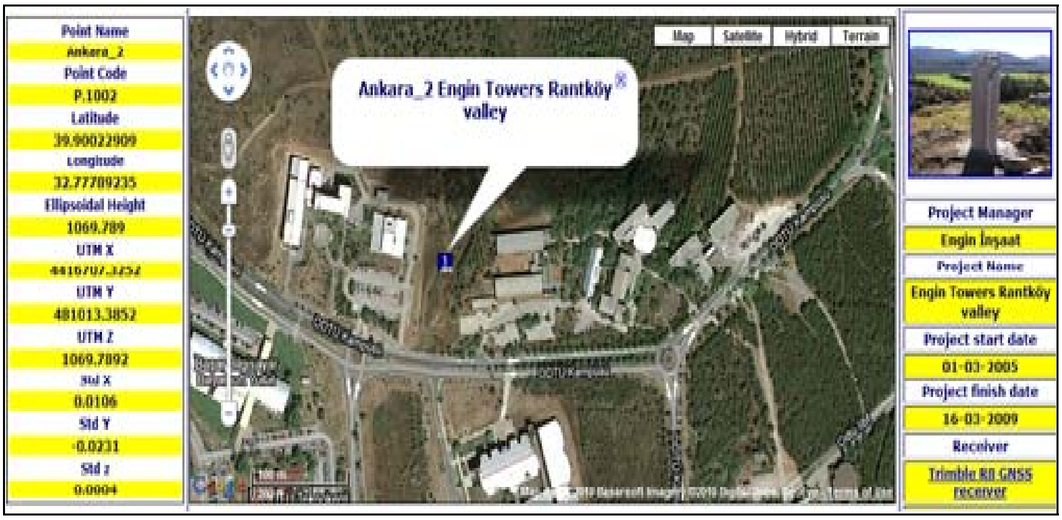


Figure 6.22: Preview of Google Maps with requested data.

The **Haversine formula** (6.5) is an equation significant in Geomatics and all navigation related professions, giving great-circle distances between two points on a sphere from their longitudes and latitudes [19].
That is, the shortest distance over the earth's surface – giving an 'as-the-crow-flies' distance between the points (ignoring any hills!).

R = earth's radius (mean radius = 6,371km) **(6.5)**

$\Delta lat = lat2 - lat1$

$\Delta long = long2 - long1$

$a = \sin^2(\Delta lat/2) + \cos(lat1).\cos(lat2).\sin^2(\Delta long/2)$

$c = 2.atan2(\sqrt{a}, \sqrt{(1-a)})$

$d = R.c$

This formula has been implemented to our system using SQL query to calculate distances between control stations according to their coordinates. Thus, user can see the closest GPS control station in the database according to defined circle radius in Google Maps.

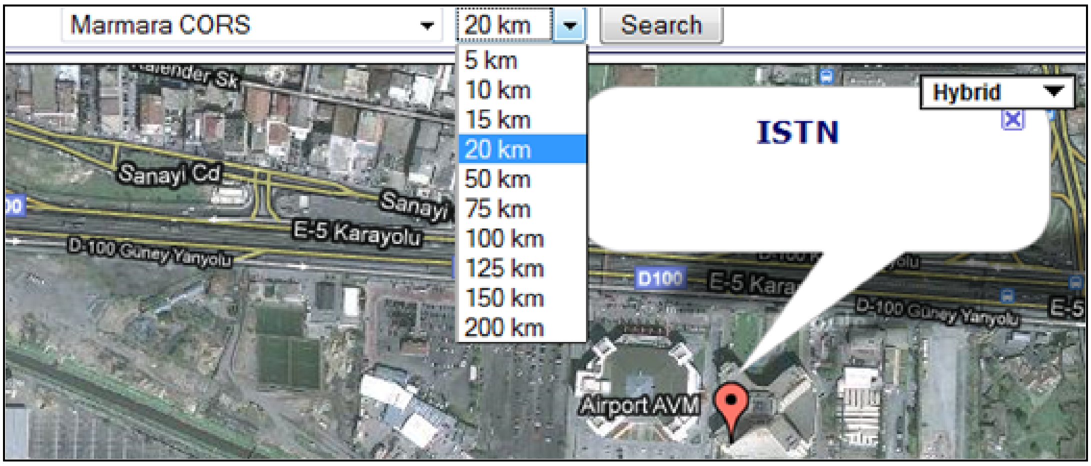


Figure 6.23: Control station request according to defined radius.

The other formulas related to the geographic coordinates such as bearing, midpoint and destination are still in development phase and as long as they finished, they will be implemented into the system. Therefore, with these functions users can use graphical map interface and more effectively for the purpose of analyses, future estimation and proper decision making.

7. CONCLUSIONS AND RECOMMENDATIONS

The universal increase of spatial information during the past decade contrasts sharply with the earlier era. This era was characterised by the umbrella term of GIS a highly specialised technology of interest professional users and researchers for specific applications. One of the major dynamic forces behind the recent spread of spatial information is the increasing availability of spatial data from government and commercial sources, distributed via the Internet through such mechanisms as spatial databases, digital geo-libraries and spatial data warehouses and clearinghouses.

Another major powerful force is the increasing awareness of the importance of spatial information by all areas of modern society. Policy makers in the public sector, for example, have embraced the notion that spatial information is an important requirement of good management, a fundamental piece of the economy, and access to this information is a civil right that allows citizens to participate in public affairs. Moreover, the changes in the collection, management and use of spatial information mentioned above would probably not have happened without the power of databases and WEB.

In this work, in order to prove importance of databases and WEB technology in Geomatics, one sample WEB based Geodetic DBMS (MySD) software was developed. The purposes of this web based DBMS software are storing and conducting GPS / GNSS control station data on the WEB. Thus, users can send requests and receive responses simultaneously from online database. Moreover to that, this data can be presented on online mapping service of Google Maps with real world location view by using geographic coordinates values of control stations stored in the database. MySD database was designed according to relational database model notations including parent and child table relations, referential integrities and attribute tables. Then, this structure was combined with WEB environment for being operable from anywhere with internet connection.

MySD is also suitable for use in public and private organizations for which fast response is necessary, such as: emergency, disaster management projects, hospitals and other land information related services

In addition, reducing expenses of data distribution and running applications makes it suitable tool not only for Geomatics engineer and related projects but also for city planners, civil engineers, municipality services, military services, geological researches, irrigation, forestry, land information system and archaeology applications.

By using MySD, user or corporation will help to improve standardization and quality of spatial data, will combine mass amounts of spatial data to one place which will reduce efforts on data collection and search processes, will contribute on spatial data sharing and distribution over internet and as a result of all decision making mechanism will improve according to reliable, multiple source gathered data.

This database was primarily designed to store only GPS project and control station data. Although it is not suitable for data coming from total station and other measurement methods in addition to gravimetric and meteorological data sets, open source structure, makes it possible to develop extension packages in the future which include all. Google Maps powered by this database served as WEB map service to represent data according to geographical coordinate latitude and longitude of control stations.

Google Maps API is a new born dynamic baby which has been growing so fast. Locate and display control station are quite simple procedures that can handle by map API but locating control station data from the database is time consuming process and user have to work with multiple programs at the same time to obtain desired results. The software in this work has succeeded to visualize control station data from the database in the map but advanced queries and special functions needed to be implemented after gaining adequate AJAX knowledge. The Google Maps API works fine with Internet Explorer and Mozilla Firefox but map loading time is changeable according to connection speed.

Dynamic WEB design language PHP proved its capability to work with relational databases and SQL queries. Different sessions for different user accounts, insert pages, update and delete functions, loading and importing tools are some examples for PHP's product in this software.

Despite the fact that MySQL is the most popular DBMS nowadays, with this work it has been realized that it is insufficient for generating advanced geometrical queries.
Tools for storing object shape-files, vector data except points and multimedia files have not developed yet in MySQL free version which means you have to combine auxiliary software to implement them in your database. Using Oracle or PostgreSQL DBMS may provide better solutions with spatial data.
The biggest lack of the software is inability to download data from the WEB interface by different file formats. But this will be solved soon after some experimental practice over the software. In addition, backup and security procedures of the database have not been resolved yet that according to unwanted remove of data then, backup data will act to fulfil missing data pieces.
To sum up, the main goal of this thesis is to create WEB based spatial database that allow users to store, manage data on the web and visualize results on the interface of the Google Maps. This goal was accomplished with a few leaks which will be resolved and improved in the future.
This software will be online soon at:
"http://yunus.hacettepe.edu.tr/~etunali/mysd/gds_index.php".
User guide of the software can be found in appendices and on the web site.

REFERENCES

[1] **Sumathi, S.,Esakkirajan, S.,** 2007: Fundamentals of relational database management, Springer-Verlag Berlin Heidelberg, New York.

[2] **Hoofer, A. J., Prescott, B. M., McFadden, R. F.,**2007: Modern Database Management (8.th edition), Pearson, New Jersey.

[3] **Riccardi, G.,**2003: Database management with web site development applications, Pearson, Boston.

[4] **Url-3, Anonymous,.** <*http://www.idc.com/*>,accessed at 04.11.2010.

[5] **Ramakrishnan, R.,Gehrke, J.,** 2002: Database Management Systems, McGraw-Hill, New York PP.

[6] **Imbroane, A. M., Bucur, L.,** 2007. Database Management Systems for spatial data structures, MPearson, New York.

[7] **Yeung, W. A.,Hall, B. G.,** 2007: Spatial Database Systems: Design, Implementation and Project Management, Springer.

[8] **Erickson, J.,** 2009: Database Technologies: Concepts, Methodologies, Tools and Applications, IGI Global, Hershey.

[9] **Connoly, T., Begg, C.,** 2004: Database Solution: A step by step guide to building databases (2.nd edition), Pearson, London.

[10] **Opple, A. J.,** 2009: Databases, A beginners guide, McGraw-Hill.

[11] **DuBois, P.,** 2009: MySQL(4th edition), Pearson, New York**.**

[12] **Url-2**<*http://www.w3schools.com/*>*,*accessed at10.11.2010.

[13] **Ullman, L.,** 2008: Visual Quick Pro GuidePHP 6and MYSQL 5for Dynamic Web sites, Peachpit Press.

[14] **Özaslan, Ö.,** 2003. Jeodezik Veritabanı Tasarımıve WEB Tabanlı Yönetimi, *MSc thesis,*İTÜ, İstanbul, MA.

[15] **Powers, D.,** 2006: PHP Solutions: Dynamic Web Design Made Easy, Apress, New York

[16] **Url-3**< *http://www.php.net/*>*,*accessed at10.11.2010.

[17] **Folger, P.,** 2009: Geospatial Information and Geographical Information System (GIS): Current Issues and Future Challenges.

[18] **R, N, Çeilk.,**2007**:** Istanbul Technical University, GPS technique lecture notes.

[19] **Michael Purvis., Jeffrey Sambells., Cameron Turner., 2006,** Beginning Google Maps Applications with PHP and Ajax, Apress.

[20] **Svennerberg, Gabriel.,** 2010: Beginning Google Maps API 3, Apress, NewYork

[21] **Brown, M. C.,** 2006: Hacking Google Maps and Google Earth, Wiley, Indiana.

[22] **Url-4 <** http://code.google.com/apis/maps/>, accessed at 25.09.2010.

APPENDICES

APPENDIX A.1 : User Guide of MySD

APPENDIX A.1: USER GUIDE of MySD

1. Login and Registration

- Use unique username and password to access to the software.
- If you are not registered user then define one account via registration form (Figure A1. a).
- Username and password should be unique and username should contain at least six characters. Otherwise program will display an error message.

2. Create Manager

- **User information:** After successful registration and login procedures, you will encounter with main screen of the MySD (Figure A1.b). You can check your user information from **User→User's info** and update your personal data.
- You have to follow some subsequent procedures to use programme correctly.
- Before importing control station file to MySD, you need to create "manager" record from **Manager→New manager** and populate fields with your data (Figure A1. c). Without defined manager, neither of project and point can be inserted to the MySD.
- You can edit manager information by selecting **manager→edit manager** and update necessary fields.

3. Define Project and Insert Project Personnel

- After creating manager, user can define project under this management by selecting **Project→New project**. Then, one manager must be selected from multiple select list. Without any manager selected, users can not add new project. In this phase project name will be checked by the system to prevent duplicate entry. The crucial part that users must fill **project city** and **town** fields while defining project. They will be used in geocoding process in the future by Google maps.
- The project data can be update by selecting **Project →edit project →edit** (pen image).

- Adding personnel data is managed by **Personnel→add personnel**, in the following screen select one particular project from multiple select list to insert the personnel to it. Microsoft Word 2003, 2007 and 2010 or PDF files are suitable file to be loaded as personnel CV (Figure A1. d).

4. Importing Point File

- You can import multiple control stations via one file to MySD database. Before importing control station file into MySD you have to download an excel file linked in **GPS Data→import point file→download sample excel**(Figure A1. e).Then, populate rows with your own GPS control station data. You must keep this table structure (**column headers**) unchanged otherwise, the table imported improperly to the database.
- After populate excel table select **File→Save as→CSV** (comma delimited) and save your file. Control station name, code, latitude, longitude, location (address) of point must be filled in excel table (Figure A1. f). Then, CSV file can be imported to database via importing tool in **GPS Data→import point file**. Do not change anything in CSV format. Do not use another table that you created.
- If you want to insert only one control station then select **GPS Data→Insert one point**(Figure A1. g)and fill the fields with convenient data of your control station.
- You can check control stations by selecting **GPS data→Project** points. Moreover, you can view all control stations belong to other projects by selecting GPS data→all points in DB.
- You can edit and update your control station data from **GPS Data→Project Points→Edit** (pen image).

5. Loading Image File

- Images of control stations can be loaded in MySD by selecting **Images→Upload Image.** In the following window you must select one control station and load the image of that control station. User can load multiple images for one particular control station. But image name must be unique and the system will not accept duplicate entry for it.

6. Delete Data from MySD

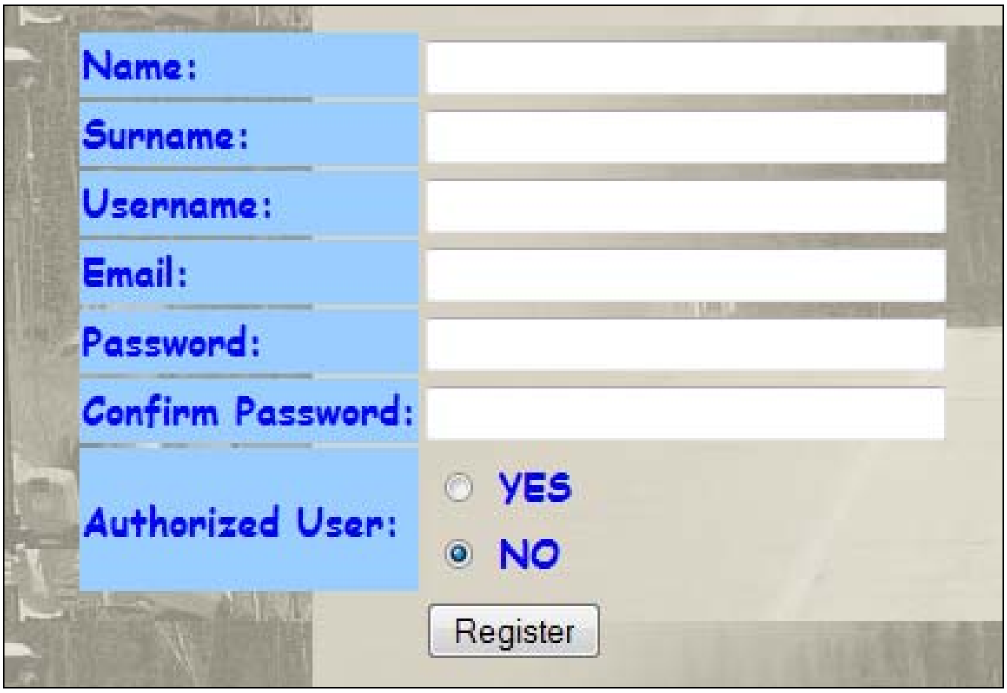
- You can delete data from data sets by selecting delete command (red **X** image) while editing your files. The delete command is irreversible and you have to think twice before attempt to delete one data records.
- In addition, deleting one record will cause to delete all child data of this data in other tables.

7. Mapping

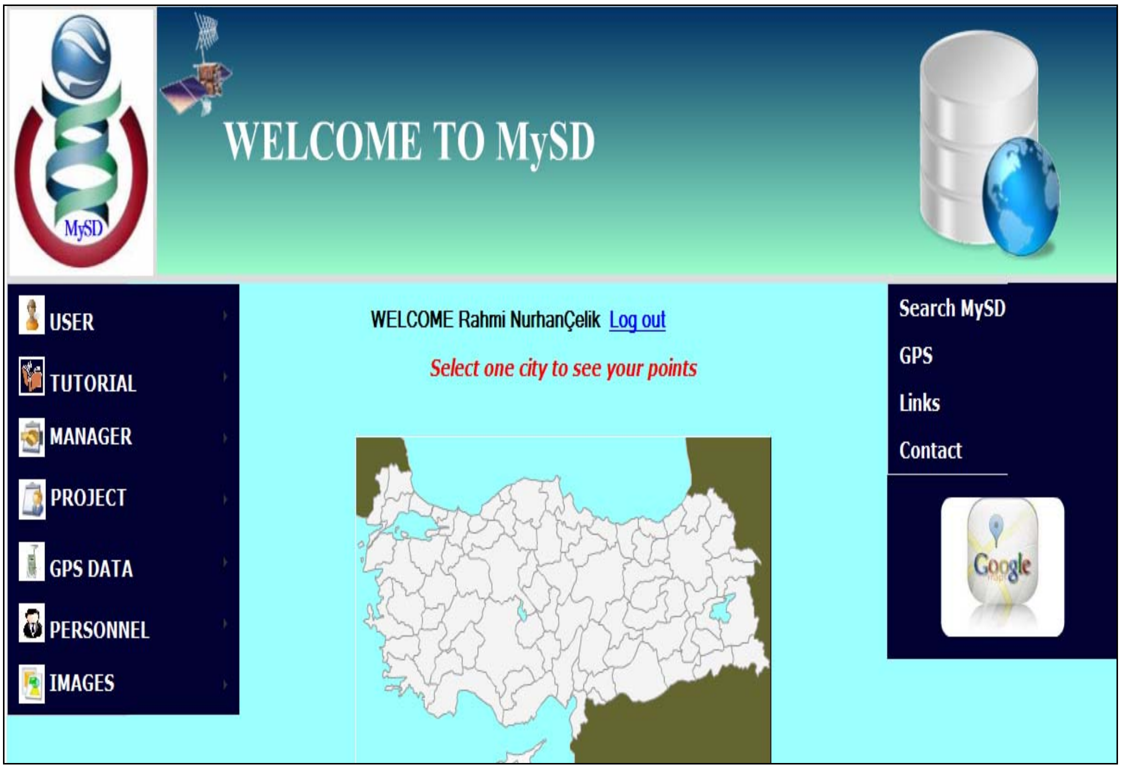
There are two ways to view your control station on the map.

- One by selecting **GPS Data→Project Points→gmaps** (Figure A1. h) and you'll obtain control station real location view on Google maps with information on the sidebars (Figure A1. i).
- The other one from homepage by using map and select one city to view control stations in that city.
- After selecting one city from the map you can reach to the points from multiple select list and also can search points according to distance defined by radius to another control station to find out closest or farther one to your selection (Figure A1. j).

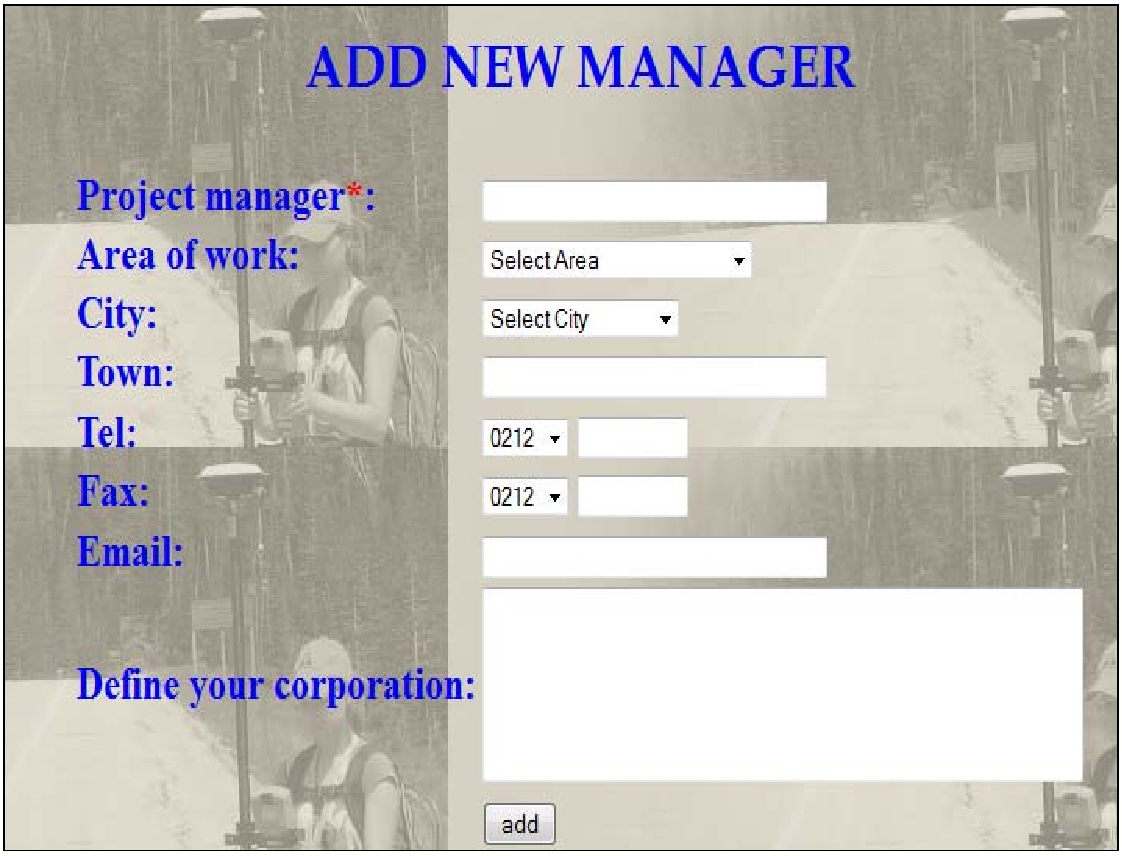
More detailed tutorial can be found on WEB site with videos and PDF files.



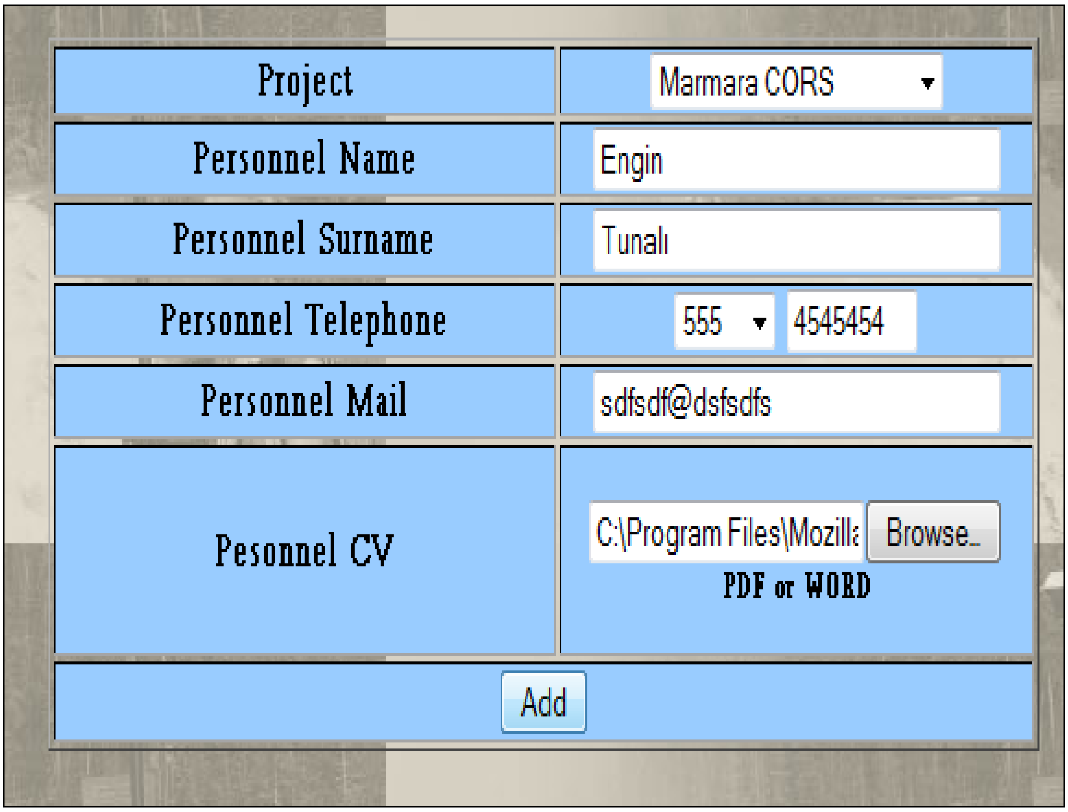
(a)


WELCOME TO MySD
MySD
USER
TUTORIAL
MANAGER
PROJECT
GPS DATA
PERSONNEL
IMAGES
WELCOME Rahmi NurhanÇelik Log out
Select one city to see your points
Search MySD
GPS
Links
Contact
Google

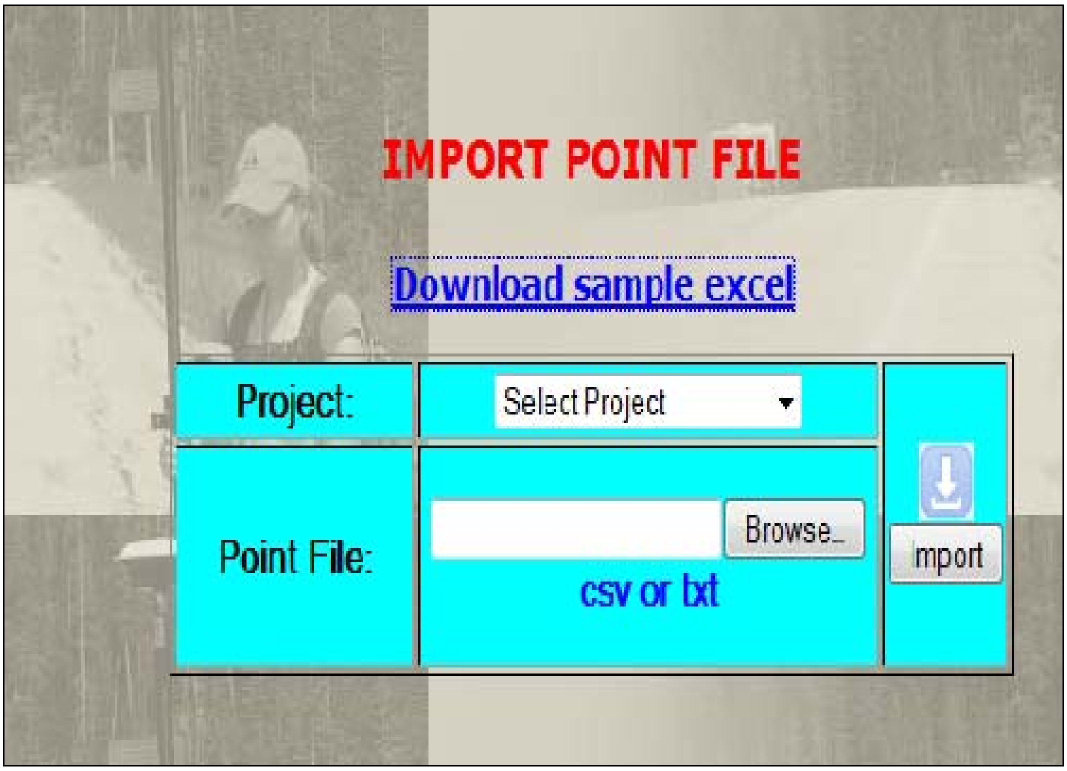
(b)


ADD NEW MANAGER
Project manager*:
Area of work:
Select Area
City:
Select City
Town:
Tel:
0212
Fax:
0212
Email:
Define your corporation:
add

(c)


Project
Marmara CORS
Personnel Name
Engin
Personnel Surname
Tunalı
Personnel Telephone
555
4545454
Personnel Mail
sdfsdf@dsfsdfs
Pesonnel CV
C:\Program Files\Mozill
Browse...
PDF or WORD
Add

(d)


IMPORT POINT FILE
Download sample excel
Project:
Select Project
Point File:
Browse...
csv or txt
import

(e)

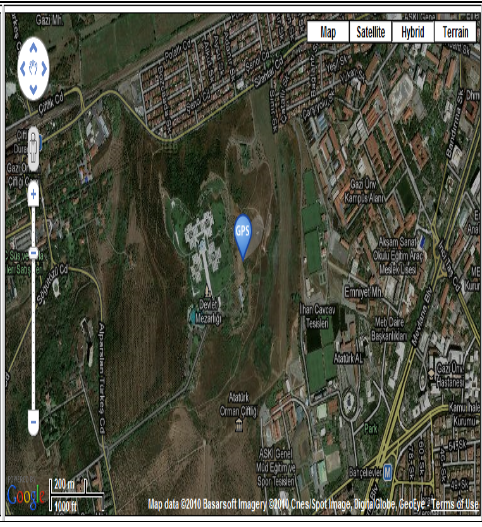
A	B	C	D	E
Point Code	Point Name	Point Type	Latitude	Longitude
P.1002	Ankara_1	C4	39.93656649	32.81016469
P.1002	Ankara_2	C4	39.90022909	32.77789235

(f)

Point Code	Point Name	Point type
P.1001	Ankara_1	CORS-TR
WGS84 geographic coordinates		
φ-Latitude (degrees-decimal)	λ-Longitude (degrees-decimal)	h-Ellipsoidal Height(m)
39.93656649	32.81016469	1023.563
UTM cartesian coordinates		geo-utm convertion tool
UTM X (m) :	UTM Y(m):	UTM Z(m):
4420733.9965	483780.7253	1023.5632
Reference Epoch:	2005.00	
Velocity X:	0.0123	
Velocity Y:	-0.0023	
Velocity Z:	0.0003	

(g)

Point Name
Ankara_1
Point Code
P.1001
Latitude
39.93656649
Longitude
32.81016469
Ellipsoidal Height
1023.563
UTM X
4420733.9965
UTM Y
483780.7253
UTM Z
1023.5632
Std X
0.0123
Std Y
-0.0023
Std Z
0.0003



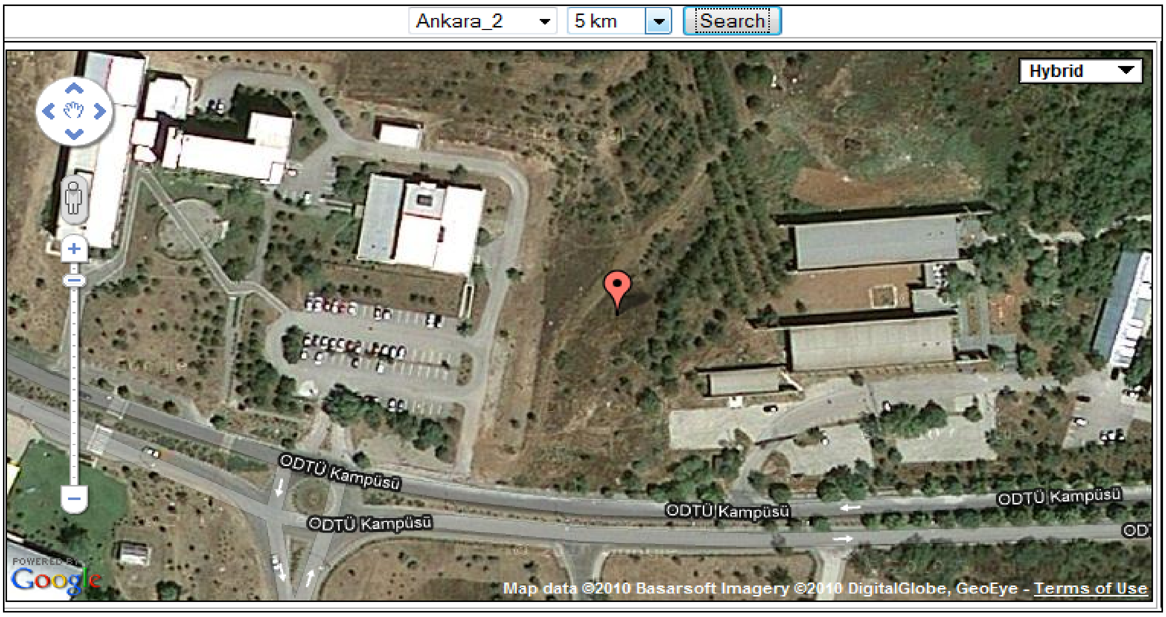
Project Manager
I.P. Melih
Project Name
I.P. Melih Rant Towers
Project start date
01-03-2005
Project finish date
16-03-2009
Receiver

(i)

GPS point list

Point Code	Point Name	Latitude	Longitude	Ellipsoidal Height	
P.1001	Ankara_1	39.93656649	32.81016469	1023.563	gmap
P.1002	Ankara_2	39.90022909	32.77789235	1069.789	gmap
P.1003	Ankara_3	39.94788507	32.64777302	990.563	gmap
P.1004	Ankara_4	39.97696261	32.65206456	986.235	gmap
P.1005	Ankara_5	39.96906948	32.70064473	1047.789	gmap
P.1006	Ankara_6	39.08403243	32.54202962	1078.023	gmap
P.1007	Ankara_7	40.04046912	32.18492147	999.465	gmap
P.1008	Ankara_8	40.07692904	32.60863423	999.465	gmap
P.1009	Ankara_9	40.02722704	32.38113999	1023.786	gmap
P.1010	Ankara_10	39.95739255	32.44877457	896.128	gmap

(h)



(j)

Figure A.1: MySD software WEB interfaces: (a) Registration. (b)Homepage. (c)Manager. (d) Personnel. (e) Importing point file. (f) Exceltable. (g) Insert CS. (h) Gmap. (i) GPS CS. (j) Search CS.

CURRICULUM VITAE

Candidate’s full name: Engin TUNALI

Place and date of birth: Bulgaristan, 30.01.1983

Permanent Address: Merkez district, Asya street, no:2/5,

Avcilar-Istanbul, TURKEY

Universities and
Colleges attended: Istanbul Technical University, Geomatics Engineering.

Contact: 555-4507605 / etunali@hacettepe.edu.tr