

**DEVELOPING A PARCEL-BASED INFORMATION SYSTEM BY OBJECT-  
ORIENTED APPROACH**

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

**BY**

**EMRAH TUFAN**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE  
OF MASTER OF SCIENCE  
IN  
THE DEPARTMENT OF GEODETIC AND GEOGRAPHIC INFORMATION  
TECHNOLOGIES**

**DECEMBER 2003**

Approval of the Graduate School of Natural and Applied Sciences.

---

Prof. Dr. Canan ÖZGEN

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

---

Assoc. Prof. Dr. Oğuz IŞIK

Head of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

---

Tuncay KÜÇÜKPEHLİVAN

Co-Supervisor

---

Assist. Prof. Dr. Zuhall AKYÜREK

Supervisor

Examining Committee Members

Prof. Dr. Adnan YAZICI

Assoc. Prof. Dr. Vedat TOPRAK

Assist. Prof. Dr. Zuhall AKYÜREK

Assist. Prof. Dr. Şebnem DÜZGÜN

Tuncay KÜÇÜKPEHLİVAN, MSc.

## **ABSTRACT**

### **DEVELOPING A PARCEL-BASED INFORMATION SYSTEM BY OBJECT-ORIENTED APPROACH**

TUFAN, Emrah

M. Sc., Department of Geodetic and Geographic Information Technologies

Supervisor: Assist. Prof. Dr. Zuhar AKYÜREK

Co-Supervisor: Tuncay KÜÇÜKPEHLİVAN, MSc.

December 2003, 112 pages

The cadastre contains parcel related data which must be up-to-date. The cadastral data in any country constitute a very big dataset. Therefore parcel related data should be carefully managed.

Today, using a database is an effective way of data management. The relational database management system can be a good one for parcel related data. However when the information system concept is considered, just relational database management system is not enough. Some tools are needed in order to manipulate the data in the relational database management system. Object oriented analysis and design is a good choice to develop these tools.

In this study, a parcel-based information system is developed and it is implemented for Çankaya Municipality. During the development relational database management system is used for attribute data management, and object-oriented analysis and design is used for development of application to manipulate the data in the relational database management system. The cadastral data are separated into two parts and each part is handled separately. The first part is the geographic or spatial data. These data are handled by the help of the MapInfo Professional Version 6.5. The other part is the attributes of these spatial data. For this part, relational database is designed and implemented on Microsoft SQL Server Version 2000. During the development of the relational database, conceptual database design is performed by enhanced entity-relationship (EER) model. Then in the logical design, the conceptual model is mapped into the relational model. After data storage area is created, the application is developed on that data by using principles of object-oriented design and analysis and unified modeling language. By the help of the software developed, the data management can be carried out easily. By this study, a solution is proposed for the cadastral data management problem of the municipalities, and this study is one of the first studies developed for parcel related data using object-oriented approach for municipalities.

**Keywords:** Parcel, parcel-based information system, cadastre, object-oriented modeling, unified modeling language, EER Model

## **ÖZ**

### **NESNE YÖNELİM YÖNTEMİ İLE PARSEL TABANLI BİR BİLGİ SİSTEMİ GELİŞTİRİLMESİ**

TUFAN, Emrah

Yüksek Lisans, Jeodezi ve Coğrafi Bilgi Teknolojileri Anabilim Dalı

Tez Yöneticisi: Yrd. Doç. Dr. Zuhal AKYÜREK

Yardımcı Tez Yöneticisi: Yük. Müh. Tuncay KÜÇÜKPEHLİVAN

Aralık 2003, 112 sayfa

Kadastro, parsel ile ilgili güncel bilgileri içerir. Bir ülkede kadastral veriler çok büyük bir hacim tutmaktadır. Dolayısıyla parsel ile ilişkili veriler dikkatli bir şekilde yönetilmelidir.

Günümüzde veritabanı kullanımı, veri yönetiminin etkili bir yoludur. İlişkisel veritabanı yönetim sistemleri ise parsel ile ilişkili veriler için iyi bir çözümdür. Fakat bilgi sistemleri mantığı altında düşünüldüğünde, sadece ilişkisel veritabanı yönetim sistemleri yeterli olmamaktadır. Veritabanı içerisinde tutulan verileri işleyecek bir takım araçlar da gerekmektedir. Nesne yönelimi analiz ve tasarım yöntemi, bu gibi araçları geliştirmek için iyi bir çözümdür.

Bu alıřmada, parsel tabanlı bir bilgi sistemi geliřtirildi ve ankaya Belediyesi iin uygulamaya konuldu. Sistemin geliřtirilmesi srecinde, iliřkisel veritabanı ynetim sistemleri coęrafi olmayan veriler iin kullanıldı ve veritabanı ierisindeki verilerin ynetimini saęlamak iin nesne ynelimli analiz ve tasarımı yntemleri kullanarak bir yazılım geliřtirildi. Bunlara ek olarak, kadastral veriler iki kısıma ayrıldı ve her bir kısım farklı bir řekilde ele alındı. Bunlardan birincisi coęrafi veya mekana baęlı verilerdi. Bu veriler MapInfo Professional yazılımının 6.5 versiyonu ile ynetildi. Dięer veriler ise mekana baęlı verilerin nitelik verileri olarak dřnld. Bunlar iin de iliřkisel bir veritabanı tasarlandı ve Microsoft SQL Server yazılımının 2000 versiyonunda uygulamaya konuldu. İliřkisel veritabanının tasarlanması ařamasında kavramsal veritabanı tasarımı iin EER model kullanıldı. Daha sonra mantıksal tasarımı ařamasında, kavramsal model iliřkisel modele aktarıldı. Verilerin depolanacaęı alan yaratıldıktan sonra, bu veriler zerine nesne ynelim analiz ve tasarımı yntemi ve birleřtirilmiř modelleme dili kullanarak bir yazılım geliřtirildi. Bu yazılım sayesinde, verilerin ynetimi veritabanına doęrudan bir eriřim olmadan kolay bir řekilde gerekleřtirildi. Bu alıřma ile, belediyelerdeki kadastral verilerin ynetimi ile ilgili problemlere bir zm nerildi ve bu alıřma belediyeler ierisindeki parsel ile ilgili veriler iin nesne ynelim yntemi ile yapılmıř olan ilk alıřmalardan bir tanesidir.

Anahtar Kelimeler: Parsel, parsel tabanlı bilgi sistemi, kadastro, nesne ynelim yntemi, birleřtirilmiř modelleme dili, EER model

To my family

## **ACKNOWLEDGMENTS**

I would like to express my sincere thanks to my supervisor Dr. Zuhâl Akyürek for her guidance and insight through out this study. Also thanks to Tuncay Küçükpehlivan for his valuable suggestions and encouragement on this study.

I would also thank to my colleague Umut Durak for his technical support about object-oriented methodology and Mehmet Servi for his conceptual support about cadastral system in Turkey.

I am grateful to all Çankaya Municipality personnel, who were ready whenever I need help.

Thanks also go to my friends in Ankara who kept me motivated.

Finally, my greatest thanks go to my family for their great patience and support during my life and education.



## TABLE OF CONTENTS

ABSTRACT .....	iii
ÖZ .....	v
ACKNOWLEDGMENTS .....	viii
TABLE OF CONTENTS .....	ix
LIST OF TABLES .....	xi
LIST OF FIGURES .....	xii
CHAPTER	
1. INTRODUCTION .....	1
1.1 What is an Information System? .....	1
1.2 Types of Information System .....	2
1.3 Parcel-Based Information System .....	4
1.4 Scope of the Thesis .....	6
1.5 Outline .....	8
2. CADASTRAL SYSTEM PROCESSES .....	9
2.1 City Planning and Cadastral System Process in Turkey .....	9
2.2 City Information System Studies in the World .....	11
2.3 City Information System Studies in Turkey .....	22
3. SYSTEM DEVELOPMENT AND IMPLEMENTATION .....	27
3.1 Development Methodology .....	27
3.3 Database Design and Implementation .....	29
3.3.1 Conceptual Database Design .....	29
3.3.2 Logical Database Design .....	32
3.3.3 Database Implementation .....	32
3.4 System Implementation .....	33
3.4.1 Application Architecture .....	33

3.4.2 Client Side Implementation .....	34
4. PARCEL-BASED INFORMATION SYSTEM FOR ÇANKAYA MUNICIPALITY .....	38
4.1 Software Design and Implementation .....	38
4.1.1 Requirement Analysis .....	38
4.1.2 System and Software Design .....	42
4.2 Overview of the System .....	43
4.2 Map Handling .....	44
4.3 Database Operations Manipulations .....	51
5. CONCLUSION AND RECOMMENDATION .....	76
REFERENCES .....	82
APPENDICES	
A. DATA TAKEN FROM MUNICIPALITY .....	86
B. DATA DEFINITION LANGUAGES .....	90

## LIST OF TABLES

A-1 Ecrimisil and Çiçekçiler Liste .....	86
A-2 Hilal.....	86
A-3 Dağıtım ve Tahsis.....	87
A-4 İpotek Formu: Tahsis Sahibi Bilgileri.....	87
A-5 İpotek Formu: Tahsis Edilen Taşınmazın Bilgileri .....	88
A-6 İpotek Formu: Ödeme Bilgileri .....	88
A-7 İpotek Formu: Taksit Bilgileri .....	88
A-8 Paraları Ödenenler .....	89

## LIST OF FIGURES

1-1 The Functions of Information System (Laudon and Laudon, 1998) .....	2
1-2 Types of Information System (Laudon and Laudon, 1998) .....	3
1-3 The role of the cadastre to facilitate an efficient land market and effective land use administration .....	5
1-4 Parcel, Parcel-Block and Building .....	6
2-1 Information System Architecture of City-Wide Information System of PRC (Zhi and Wang, 1995). .....	13
2-2 Information related to urban landscape elements and their management in Seoul (Oh, 2001). .....	15
2-3 Data integration of LIS .....	16
2-4 Main functions of LSIS .....	17
2-5 An integrated spatiotemporal system architecture (Lorentzos et al., 1999). .....	19
2-6 Class Symbols in GeoOOA (Kösters et al., 1996). .....	21
2-7 Temporal class symbol in GeoOOA (Kösters et al, 1996). .....	21
2-8 GeoOOA Model of the Land-Registry (Kösters et al, 1996). .....	22
2-9 Cadastral Parcel Query of Yomralıoğlu and Mataracı's study (Yomralıoğlu and Mataracı, 1999). .....	23
2-10 STER Model of Kocaman's study (Kocaman 2001). .....	24
3-1 The software life cycle (Somerville, 2001). .....	28
3-2 The EER model of parcel-based information system ..... <b>Error! Bookmark not defined.</b>	
3-3 The coding strategy .....	35
3-4 The implementation details of map functions. ....	37
4-1 Packages .....	39

4-2 Use case diagrams for package citizen.....	39
4-3 Use case diagrams for package planning office.....	40
4-4 Use case diagrams for package municipality .....	40
4-5 Use case diagrams for package municipality (continued) .....	41
4-6 Class diagram of a system.....	42
4-7 The logon mechanism.....	44
4-8 The map window of a system.....	44
4-9 The map information button and dialog box which retrieve information from map.....	46
4-10 The layer control dialog box .....	47
4-11 The geometric selection tool .....	47
4-12 The label addition properties of a system.....	48
4-13 Import button and import file dialog box .....	48
4-14 Export button and export table dialog box.....	49
4-15 The ruler button and ruler dialog box .....	49
4-16 Creating new parcel .....	50
4-17 Modifying an existing parcel .....	50
4-18 The info button, available layers, municipal parcel block dialog box and municipal parcel block update form.....	51
4-19 The New menu and submenus of New menu. ....	53
4-20 The Building and Expropriation information that is related to that building. .....	53
4-21 The Update menu and submenus of Update menu.....	54
4-22 The dialog box that is used to call parcel block data .....	54
4-23 The activity diagram of actor “municipal work department head” and use case “confirms the plan that was approved” from package municipality. .....	55
4-24 Sequence diagram of use case “confirm the plan that was approved” for an actor municipal work department head from package municipality. .	56
4-25 The confirmation procedure of a municipal plan .....	57
4-26 The activity diagram of actor “map department employee” and use case “ownership query” from package municipality. ....	58

4-27 Sequence diagram of use case “query an ownership” for an actor map department employee from package municipality. ....	58
4-28 The ownership query for map department .....	59
4-29 The activity diagram of actor “map department employee” and use case “specific area query” from package municipality. ....	59
4-30 Sequence diagram of use case “query a specific area” for an actor map department employee from package municipality. ....	60
4-31 The specific area query for map department.....	60
4-32 The activity diagram of actor “map department employee” and use case “parcel, parcel-block query” from package municipality. ....	61
4-33 Sequence diagram of use case “query a parcel and parcel-block” for an actor map department employee from package municipality. ....	61
4-34 Municipal parcel query .....	62
4-35 The activity diagram of actor “municipal work department head” and use case “confirms the plan that was approved” from package municipality. .....	62
4-36 Sequence diagram of use case “confirm the plan that was approved” for an actor map department employee from package municipality. ....	63
4-37 The municipal plan control approvement dialog box .....	63
4-38 The activity diagram of actor “hiring department employee” and use case “parcel-block/parcel query” from package municipality. ....	64
4-39 Sequence diagram of use case “query a parcel and parcel-block” for an actor hiring department employee from package municipality. ....	64
4-40 The tenant information query .....	65
4-41 The activity diagram of actor “hiring department employee” and use case “region query” from package municipality. ....	65
4-42 Sequence diagram of use case “query a region” for an actor hiring department employee from package municipality. ....	66
4-43 The region query demonstration for hiring department .....	66
4-44 The activity diagram of actor “hiring department employee” and use case “person query” from package municipality. ....	67

4-45 Sequence diagram of use case “query a person” for an actor hiring department employee from package municipality. ....	67
4-46 The person query for hiring department .....	68
4-47 The activity diagram of actor “hiring department employee” and use case “business query” from package municipality. ....	68
4-48 Sequence diagram of use case “query a business” for an actor hiring department employee from package municipality. ....	69
4-49 The business query dialog boxes for hiring department .....	69
4-50 The activity diagram of actor “expropriation department employee” and use case “name-surname query” from package municipality. ....	70
4-51 Sequence diagram of use case “query a name and surname” for an actor expropriation department employee from package municipality... ..	70
4-52 Person query dialog boxes for expropriation department .....	71
4-53 The activity diagram of actor “expropriation department employee” and use case “report date query” from package municipality. ....	71
4-54 Sequence diagram of use case “query a report date” for an actor expropriation department employee from package municipality. ....	72
4-55 The reporting date query dialog boxes .....	72
4-56 The activity diagram of actor “expropriation department employee” and use case “address query” from package municipality. ....	73
4-57 Sequence diagram of use case “query an address” for an actor expropriation department employee from package municipality. ....	73
4-58 The address query dialog boxes .....	74
4-59 The activity diagram of actor “building land assignment department employee” and use case “query the payment information” from package municipality. ....	74
4-60 Sequence diagram of use case “query the payment information” for an actor building land assignment department employee from package municipality. ....	75
4-61 Instalment plan query .....	75

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 What is an Information System?**

In the second half of the 20<sup>th</sup> century, the global enterprises had growth rapidly. As a result of this rapid development, the information system concept had become an important term for the management of these enterprises. At the beginning, it was believed that the management process and the information handling are the personal activities and they are handled by face-to-face communication. But today, it is accepted that, the information can be handled by only the well-organized systems.

According to Laudon and Laudon, (1998) the information system is explained as a set of interrelated components that collect (or retrieve), process, store and distribute information to support decision making, and control in an organization. By the help of the information system, data, which can be defined as the streams of raw facts representing events occurring in organizations or the physical environment, are translated into information (Laudon and Laudon, 1998). During translation three important steps or activities are carried out; input, processing and output. These steps are given in Figure 1.1.



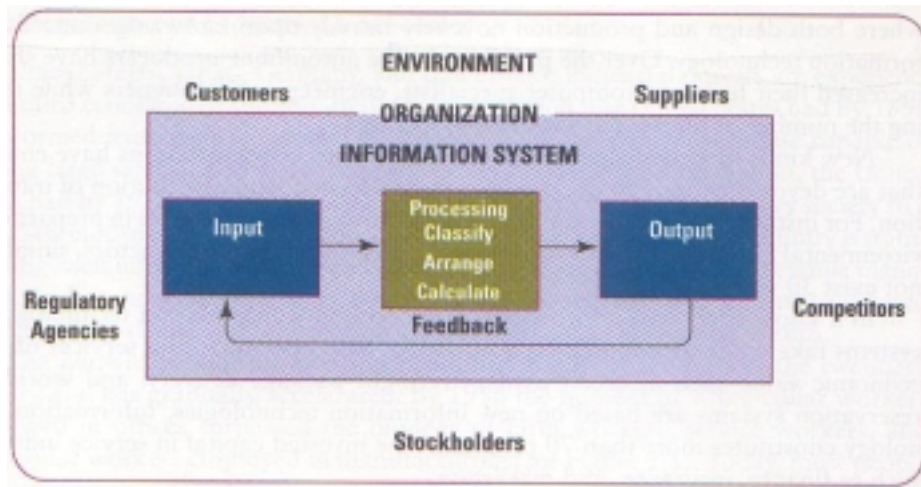


Figure 1-1 The Functions of Information System (Laudon and Laudon, 1998)

## 1.2 Types of Information System

In an organization there are four different organizational levels. These are the strategic level, management level, knowledge level and operational level. For each level or unit in organization, there are different kinds of information system. For strategic level, executive support systems (ESS), for management level management information system (MIS) and decision support system (DSS), for knowledge level, knowledge work system (KWS) and office automation system (OAS) and for operational level, transaction processing system (TPS) are placed or established in an organization. The types of information system can be seen in Figure 1.2.

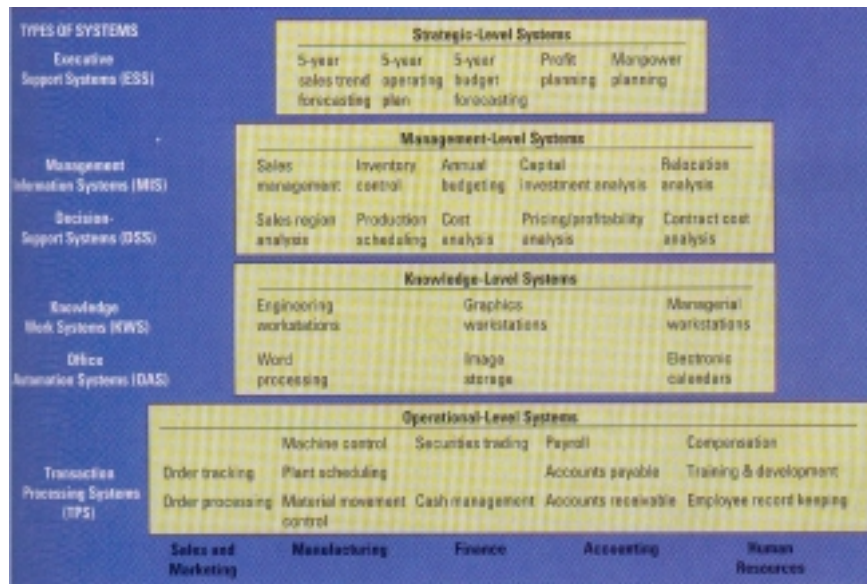


Figure 1-2 Types of Information System (Laudon and Laudon, 1998)

Laudon and Laudon (1998) define these systems as;

- **Transaction processing systems:** These are the computerized systems that perform and record daily routine transactions necessary to conduct the business.
- **Knowledge work system:** These are the systems that help knowledge workers for creating new knowledge.
- **Office automation systems:** These are the computer systems that increase the productivity of the data workers in an office. For instance; word processing and electronic mail system.
- **Management information systems:** These are the systems that provide reports and summaries to managers, for planning, controlling and decision making.
- **Decision support systems:** These are the systems that combine data and sophisticated analytical models to support decision making.

- **Executive support systems:** These are the systems which are designed to address unstructured decision making by means of advanced graphics and communications.

### **1.3 Parcel-Based Information System**

According to International Federation of Surveyors, the cadastre is a parcel-based and up-to-date information system which contains a record of interest in land. In the UN Bogor Declaration in 1996, the vision of the cadastre for the future is to “develop modern cadastral infrastructures that facilitate efficient land and property markets, protect the land rights of all, and support long term sustainable development and land management.” (The Bogor Declaration, 1996).

Cadastre or parcel-based information system can be seen as the basic infrastructure to support different systems in the area of land management which can be seen in Figure 1.3 (Enemark and Sevetdal, 1999).

- Land Tenure System, to secure legal rights in land such as mortgage
- Land Value System, to assess the value of land and properties and to levy land taxes
- Land Use Control System, to enable comprehensive and detailed land use planning
- Land Development System, to enable regulation and implementation in change of land use.

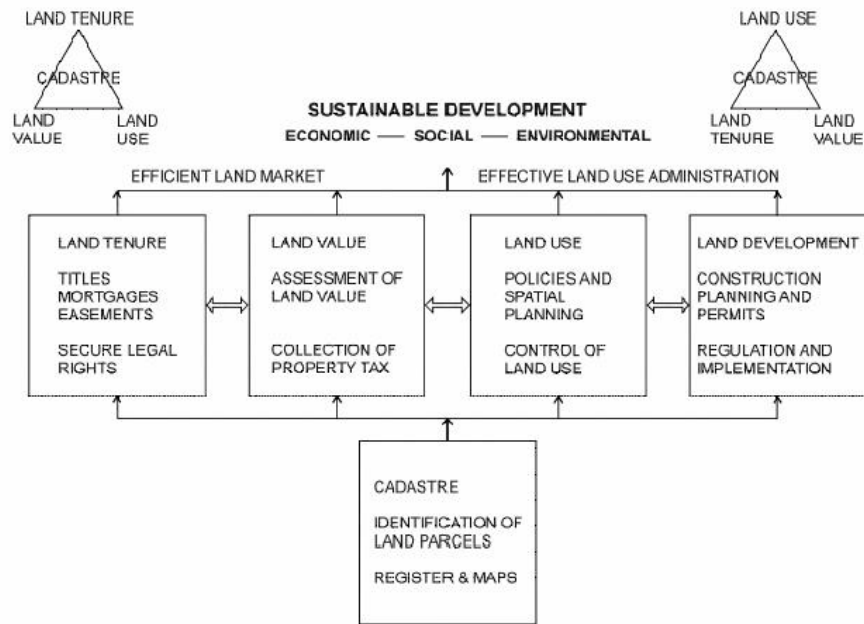


Figure 1-3 The role of the cadastre to facilitate an efficient land market and effective land use administration

The basic and the smallest geographic unit of the parcel-based information system or cadastre is the land parcel (Figure 1.4). All the land related information which is called as property or real estate are constructed on the land parcels. These information can be; land rights, land values, land use, housing and building, population and administration.

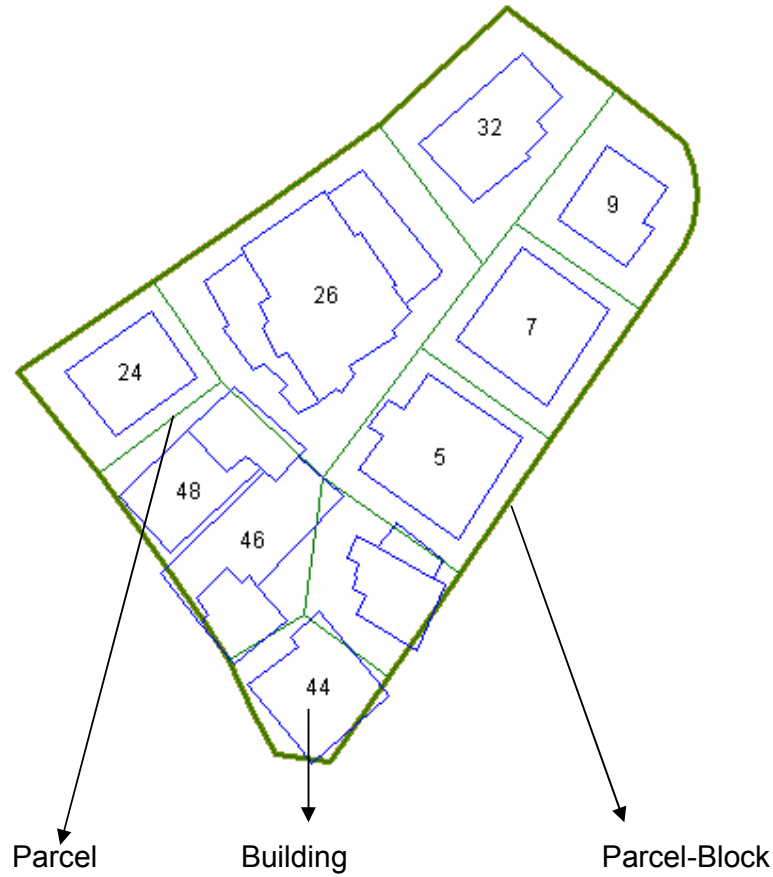


Figure 1-4 Parcel, Parcel-Block and Building

#### 1.4 Scope of the Thesis

The urban areas in Turkey have been exposed to migration especially for the last 15 years. As a result of this, the population of these urban areas has been growing and growing and number of urban areas or cities whose population is over one million has increased. Because of the population growing, the new problems in cities have appeared. One of these problems is increasing volume of cadastral data. Therefore the management of cadastral data in big cities or metropolises is getting difficult. Therefore in each metropolis, the public institutions whose job is managing cadastral data should find an efficient way of managing cadastral data.

Having almost three million population, Ankara which is the capital city of Turkey is a metropolis and has a problem of managing cadastral data as

well. The one of the biggest sub-province of Ankara is Çankaya and Çankaya Municipality is responsible with the management of the cadastral data which is inside the administrative border of the municipality.

When the interviews were performed with the personnel of Çankaya Municipality, it was seen that, there has not been made an investment about the cadastral data management so far. All the spatial data have been kept on the paper maps and all the non-spatial or attribute data have been kept on either excel, word sheets or on paper documentation. As the cadastral data have been stored on these media, the efficiency of the cadastral works in municipality has lost. Because, using excel or word sheets or non-existence of central data management results in redundant data. Besides, data retrieving cannot be performed quickly. Therefore, the municipality needs information system approach in order to solve the data management problem.

In information system approach, central data management plays an important role. There are several ways implementing central data storing area. One of them is using database management systems. Today, relational database management system is the most commonly used data handling methodology in many public and private enterprises.

The scope of this study is to develop a parcel-based information system in order to manage cadastral data in municipalities on which there has not been made an investment about managing parcel related data so far. After system is developed, it is aimed to implement the developed system to Çankaya Municipality.

Moreover while developing the parcel-based information system, there are several tools and methodologies are aimed to be used. These are;

- Analysing the requirements of a parcel-based database for Çankaya Municipality.
- Constructing a relational database management system. During construction, at the conceptual level, enhanced entity-relationship

(EER) model, and at the logical level, the relational model is aimed to be used.

- Normalizing the database in order to eliminate the data anomalies and implementing the normalized database on one of the database management system.
- Developing an application on the database to manipulate the data in the database. During development of an application, it is aimed to use the principles of object-oriented analysis and design.
- Using unified modelling language (UML) in order to model the application.
- Using waterfall methodology during development. In addition, it is aimed to combine the steps of the waterfall methodology and components of the UML. In other words, for each step of the waterfall methodology, one or more components of the UML are aimed to be applied, so that the software development can be controlled precisely.

## **1.5 Outline**

Literature survey containing cadastral system and city planning process in Turkey is given in Chapter 2. This chapter includes city information system studies in the world and in Turkey, as well.

In Chapter 3 system development and implementation are discussed. The relational database design is given. Moreover coding strategy is described.

In Chapter 4 the developed system is discussed. In this chapter usage of the UML is given and main functionalities of the system are given and sample queries which can be made by the system are demonstrated.

Finally Chapter 5 discusses the concluding remarks and future work that can be carried out on parcel-based information system.

## CHAPTER 2

### CADASTRAL SYSTEM PROCESSES

In this chapter parcel-based information system is discussed. The cadastral works and city planning process in Turkey are briefly explained and the similar studies performed in the world and in Turkey are given.

#### 2.1 City Planning and Cadastral System Process in Turkey

In our country, the city planning process is not going beyond the 19<sup>th</sup> century. In the Ottoman Empire, the primary concerns of the city planning were the strength of the buildings, suitability of the buildings to the health of the people, cleanliness of the environment and the prevention of the fires. After the reform activity in the Ottoman Empire, the primary concern of the city planning activities had become to give the proper municipality services to the citizen (Keleş, 2000).

In Turkey, city planning process is divided into four major parts as; a research, making a plan, approving a plan and implementation of a plan.

- **Research** In this phase of the city planning process, the data about the city are collected. There are three important advantageous of the research. The first one is the healthier predictions about the future of the city. The more complete and the accurate data result in the more accurate predictions of the future of the city. The second advantage of



the research is that they provide valid reasons to the people who approve the plan. The last advantage is that, they are the reference point for the people who want to evaluate the planning process and the implementation of the plan.

- **Making a plan** According to the current law (İmar Yasası 3194), the municipality whose population exceeds 10.000 must make a city plan (Kapan, 1998). In the other municipalities, the necessities of making a plan is decided by the municipality council.
- **Approving a plan** In our country the city plan approval is performed by considering the several laws. According to construction law (İmar Yasası 3194), the city plan is approved by the municipality council (Kapan, 1998). If the aim of the plan is related with the tourism or cultural and natural treasury of the country then the plan is approved by the Tourism or Public Works and Settlement Ministry.
- **Implementation of a plan** The implementation of the plan is performed by the municipalities.

Determining both legal and geometric position of all the real property can be described as the aim of the cadastral works (Web Reference 7). In Turkey, all the cadastral code is founded and improved to achieve this aim.

The cadastral work on land is started with the cadastral working plan and cadastral working period which are decided by the cadastral directorship. Having decided the plan and the period, the geometric position of the land have been decided which is called as cadastral parcel. Then, the numbering procedure has been applied and cadastral maps are created.

After cadastral parcels and maps are created, the municipal plan is made by municipalities. There are two types of municipal plan, in the latest municipal code (İmar Yasası 3194) which is founded in 1985.

- **Verse Municipal Plans:** They are generally 1:5000 or 1:10000 scales.

In this type of plan, the land use and region type which is residence, commercial, industrial areas, green lands, non-settlement areas and the transportation network of the settlement areas are decided (Keleş, 2000).

- Application Municipal Plans: They are the 1:1000 scales plans. In this type of plan, parcel-blocks of the several city regions, the roads of the city, building order, storey number and the storey height of the buildings are shown (Keleş, 2000).

After drawing the municipal plan, the ownership information is decided.

## **2.2 City Information System Studies in the World**

As the GIS technology is developing, the use of it in the administration field is also increasing. The need of storing administrative information in any city or center of population is increasing dramatically because of the population growing. GIS is the proper solution for this kind of information, and the number of cities that use GIS technology is increasing day-by-day.

Budapest is the one of these cities. In Budapest the number of registered flats had been increased from 300000 to 750000 during the 1990's. In addition the number of registered and received applications had been raised from 160100 to 252000 and while the numbers of unsettled applications were 40000 in 1990, this number became 368000 four years later (Farago, 1999). Therefore, in order to handle this situation BIIR (Budapesti Ingatlan-nyilvantartasi Információs Rendszer) system was developed. The aim of this system was to upgrade the paper-based land registration system to computer based information technology-based system (Farago, 1999). At the beginning, the system was considered to have two parts. One of them is land registration and the other is cadastral mapping system. The two systems had two separate databases. Later, the works that were for connection of these two databases were started. After the system is implemented, there was great saving in staff time.

Another study about parcel-based information system was performed in People's Republic of China. (PRC) Because of an increasing economic and investment activities, PRC cities needed to automate their information processing functions. Therefore the State Council started a city-wide information system project in 1991. According to Zhi and Wang, because of the size, computerization level and having sufficient number of people who are experts in information system, city X which had three million population and 100 square kilometers area were chosen as a pilot city (Zhi and Wang, 1995). The city-wide information system in PRC is consisting of several subsystems. These systems support economy, land, materials management, banking, foreign trade, city planning and development, residents, tourism, disaster prevention and news. While the system had been developing the three information service goals were aimed;

- The system will provide the timely, comprehensive and integrated information to the city administrator
- The system will provide the information to some city units to improve local economy, city development, services, education, technology, culture etc.
- The system will provide services for citizen of a city.

At the beginning a city-wide information system was divided into two levels; the first one was the management support system for top city administration and the other was the system that match the existing political and economic structure and functions. But later, this architecture was unable to meet the needs of a user from different hierarchical level. Moreover, there were data excessiveness, redundancy and inconsistency among the city levels. As a result of these problems the third level was added between the first two levels. At the new level, there are public information service system for public, integration and management system, and specialty information service system for industry. The architecture of the city information system of PRC is given in Figure 2.1.

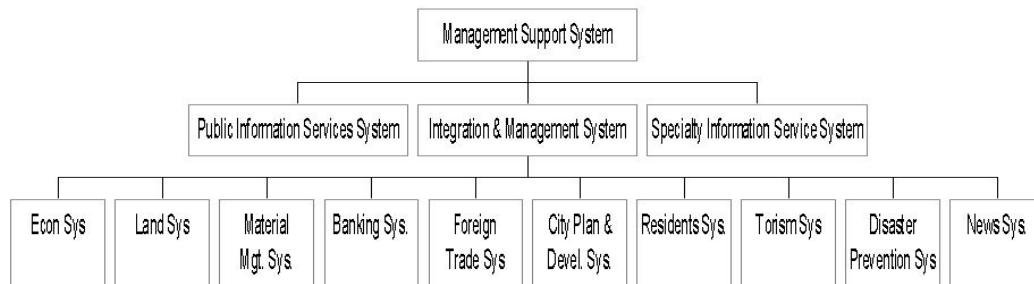


Figure 2-1 Information System Architecture of City-Wide Information System of PRC (Zhi and Wang, 1995).

The most important of these three systems is the integration and management and it is for coordinating, utilizing and managing the data recourses of various systems at different levels.

Trujillo which is one of the cities of Peru has also started city information system studies. Turkstra et al., state that, the development stage of city information system of Trujillo has been divided into three major parts (Turkstra et al, 2003). The first developing component of the system in Trujillo was the fiscal cadastre. Because by using fiscal cadastre, an increase in the tax income of the municipality was aimed. So that, the political support of the system was guaranteed. During the development of the fiscal cadastre, the first two steps being performed were that, base maps of the city were converted from computer aided design structure to geographic information system structure, cadastral coding system was established.

The second step of the city information system was an environmental atlas. It contains the information about the natural environment such as digital terrain model of Trujillo, built-up areas of city and cultural aspects.

The third and the last component is the municipality web site. At the time the paper was written, the web site of municipality was not operational but it is planned to increase the interactivity between citizen and municipality.

Seoul which is the capital of the South Korea has also started city information system studies. Oh says that, landscape information system

(LSIS) is developed in order to support decision making activities in the urban landscape management process (Oh, 2001).

During the development process of LSIS, the information related with the urban landscape elements are separated into three categories:

- Land base: The elements or data which determine the physical appearance of the landscape are included in this category. For example; Natural topography, roads, lot areas etc.
- Land cover: In this category, there is vegetation and artificial element on the land base category, such as buildings.
- Landscape management: Elements in the final group contains major viewpoints for landscape control, landscape resources, and development limits.

During the formation process of LSIS, landscape elements are further classified as

- Baseline landscape
- Skyline and streetscape
- Landscape (building) envelope

These are given in Figure 2.2

Landscape elements		Landscape types		
		Baseline landscape	Skyline and streetscape	Landscape (building) envelope
Land base/land form	Topography and geology	Topography and geology		
	Roads Lot areas Architectural and urban planning controls	Roads and sidewalks		Lot boundaries Zoning, lot coverage, FAR, building spacing and height, setback, other architectural and planning controls
Land cover/related developments	Trees and vegetation		Trees and vegetation	
	Road facilities		Overpasses, overhead bridges, streetlights, signal lights, electric poles	
	Buildings and structures		Single or detached houses (low-rise), apartments (high-rise), office buildings, factory buildings, temporary buildings, other structures	
	Proposed developments		Development proposals	
Landscape management	Viewpoints for landscape control	Viewpoints (3-D coordinates)		
	Landscape resources	Landscape resources		
	Development limits	Development limits/thresholds (VTCC)		

Figure 2-2 Information related to urban landscape elements and their management in Seoul (Oh, 2001).

During the database development of the LSIS, data are classified into two according to source from which data are acquired. The first one is the urban information system (UIS) of Seoul. In order to prevent data redundancy, some data are obtained from UIS. These are topography, roads, buildings, land ownership etc. Information not obtained from the UIS, such as development proposals and elements in the landscape management category are acquired from individual sources. The data integration of LSIS is shown in Figure 2.3

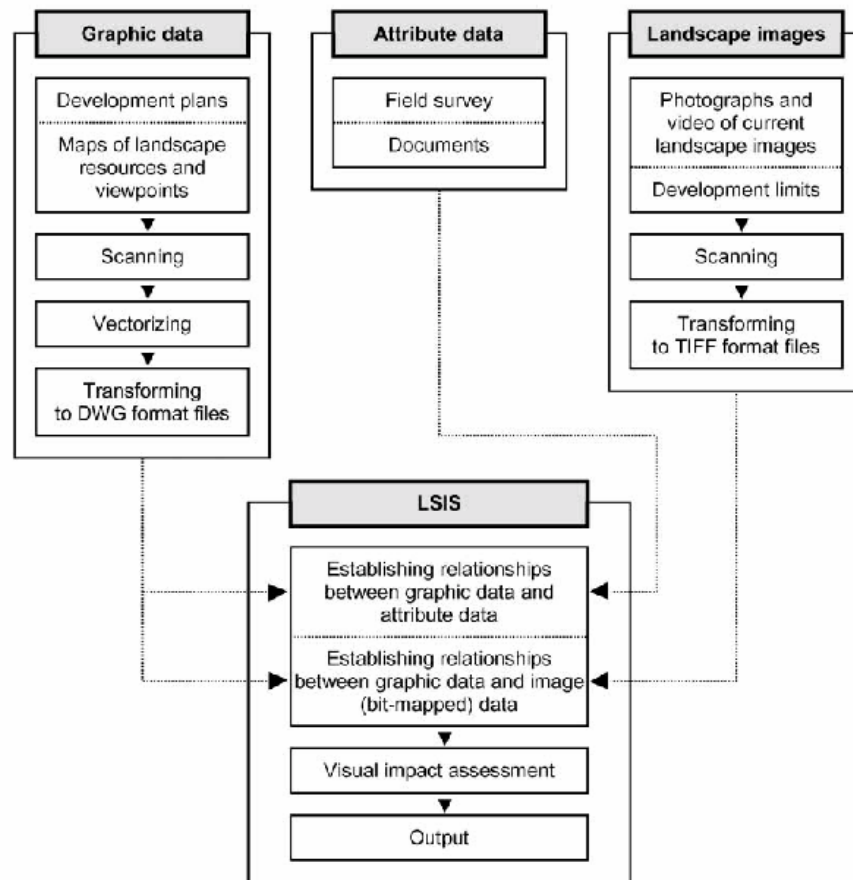


Figure 2-3 Data integration of LIS (Oh, 2001).

The LIS was developed in the AutoCAD Map environment. AutoLISP is used as a programming tool. Figure 2.4 demonstrates the functionality of LIS.

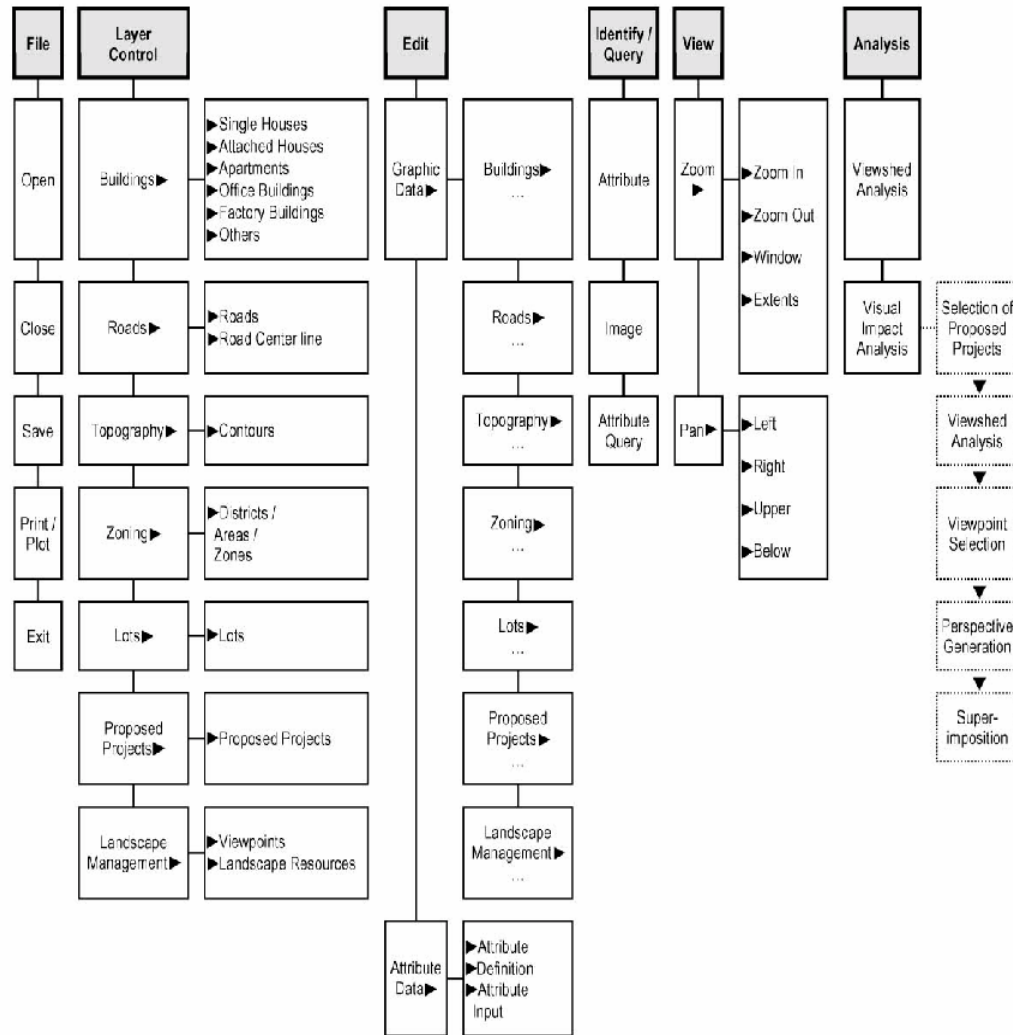


Figure 2-4 Main functions of LSIS (Oh, 2001).

After the Great Hanshin-Awaji Earthquake in 1995, the government made a plan to encourage the adoption of GIS in Japan. After this date, some advanced Japanese local governments such as Kanagawa Prefecture and Tokyo Metropolis started their GIS based urban information system (UIS).

Kugo et al. (2002) says that, the aim of the city planning in Japan is to paint a picture of how the city should look roughly 20 years hence (Web Reference 5). To achieve this goal city plans include the control of land use, creation and improvement of city facilities and city and land development projects.



In Japan, the city planning and the GIS relation is established by the help of two components; basic survey of city for city planning and the city planning maps. The Basic surveys are generally performed every five years and they focus on the land use and structures. City planning maps are divided into two. The first one is the administrative map, which is at a scale of 1:10000 and used to maintain consistency and the coordination at the city level. The other one is the basic city-planning map, which is also called as engineering maps. They show details of individual city plan.

Kanagawa Prefecture is located in southwest of the Japanese capital and it has the population of roughly 8.5 million. It contains 37 municipalities and the 2 government ordinance-designed cities whose population is over a million. The UIS of Kanagawa Prefecture was established to provide data for making city policy decisions in the prefecture and it is a link between basic city planning surveys and the GIS. The system is based on the regions called sub-zones and basic city planning surveys. Sub-zone is a region whose average area is 1.2 ha and has an average population of 500. In addition, land use and structures are stored at a scale of 1:2500.

The system is useful for city planning. Three useful applications of the system are the application for revising boundaries, grasping the condition of city facilities and the development and improvement of city land and city stability. In the first example; the areas in which the urbanization is banned or allowed can be determined. In the second one the park quality of the city can be determined by using the buffering function. Finally in the last one the city safety can be determined by several aspects such as; the threat of building collapse, the threat of fire.

In his PhD thesis Liou (1999) performed a conceptual study about cadastral information system (Liou, 1999). In this thesis, both the objects which have spatial and temporal properties which are called spatiotemporal were studied. In thesis the new modeling approach which was spatiotemporal object model (STOM) was used. In fact the basic of STOM is Coad and

Yourdon's object-oriented analysis (OOA) (Coad and Yourdon, 1991). The basic principles of the OOA were used in the study. These are the abstraction, encapsulation, inheritance and association. The problem of time dimension in cadastre was achieved by temporal object system. This system contains spans, interval and instants as well as the history of the object. In other words the alteration of an object was studied by mainly from two perspectives;

- Behavior of an object cannot be predicted and is instant
- Attributes of an object can be changed continuously and constitutes an interval in time.

In addition the object-oriented database management system was preferred as a tool to handle data management because of distributed characteristics of data.

Another study about spatiotemporal system was performed by Lorentzos et al (Lorentzos et al., 1999). Actually, this study was applied in the field of agriculture but because of the logic behind the system, it can be also applicable for city information system. In that study an integrated spatiotemporal system was developed. The system architecture in that study is given in Figure 2.5.

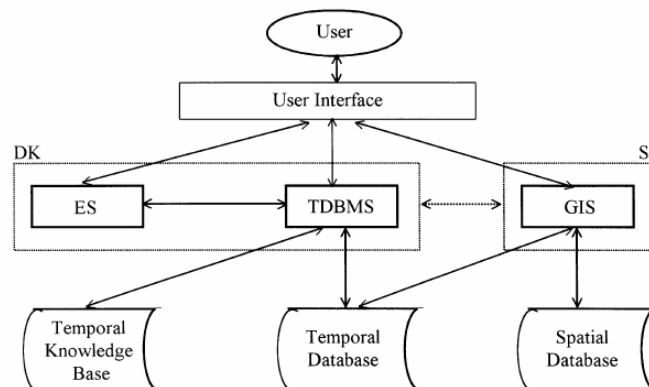


Figure 2-5 An integrated spatiotemporal system architecture (Lorentzos et al., 1999).

The system consists of two components;

- Data and knowledge (DK): It includes two sub-components which are expert systems (ES) and temporal database management systems (TDBMS)
- Spatial Data (S): This component only consists of geographic information system (GIS).

The system functionality is separated into four:

1. Temporal data management: The TDBMS and user interaction is provided and temporal data can be retrieved and maintained by users.
2. Temporal knowledge management: The user and ES interaction is provided so that temporal knowledge can be maintained and used in consultation session.
3. Spatiotemporal data management: The user can interact with the GIS so that maintenance of spatial data and retrieval of spatiotemporal data can be provided.
4. Spatiotemporal knowledge management: The user can interact with the GIS so that displaying the result of a consultation can be possible.

In recent year, the object-oriented technology is becoming more and more popular. As a result of this trend the researches have been performed on this field.

For example Kösters et al. (1996), used object-oriented methodology to model geographical data. However, they did not use the classical object-oriented approach because they believe that it is not capable to model the specific requirements of the geographic information system (GIS). Therefore they developed a modeling methodology called as GeoOOA which is based

on the classical object-oriented methodology but allows more adequate solution for the GIS. In GeoOOA, the class is separated into two types. One of them is the conventional class and the other is the geoclass which appears to be the point class, the line class, the region class and the raster class. (Figure 2.6) In conventional class, non-geographic objects are modeled and in geoclass, as the name implies geographic objects are modeled.

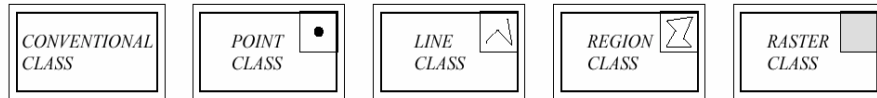


Figure 2-6 Class Symbols in GeoOOA (Kösters et al., 1996).

In GeoOOA, the temporal and non-temporal objects are also modeled as different class types whose name is temporal class. Temporal class symbol is given in Figure 2.7.

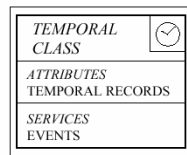


Figure 2-7 Temporal class symbol in GeoOOA (Kösters et al., 1996).

In the study, Kösters et al. (1996), can model the spatial and temporal information. As an example, land registry case is given in the paper which is given in Figure 2.8.

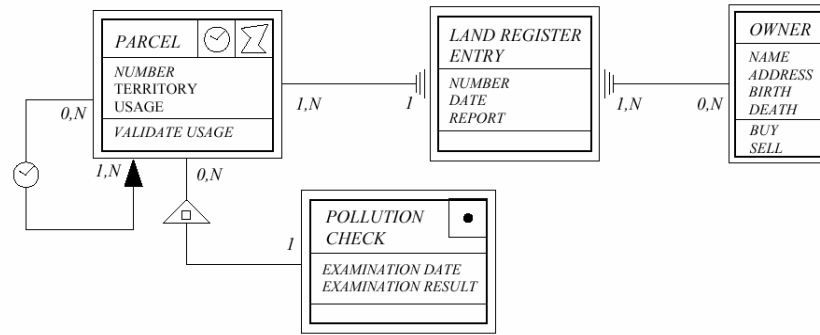


Figure 2-8 GeoOOA Model of the Land-Registry (Kösters et al., 1996).

### 2.3 City Information System Studies in Turkey

Because of a rapid population increase, the cadastral data in Turkey have also been increasing. As a result of this fact, the studies have been started in order to translate the cadastral information to the computer-based information system. One of the important studies was made by Yalın (1986). In his PhD thesis joining the three components to establish a multipurpose cadastral system were forced (Yalın, 1986). These components are;

- Cadastral map
- Multipurpose land information system
- Ownership registration system

Another study is performed by the Yomralıoğlu and Mataracı (1999). In their study, they integrated the title-deed and cadastral information to city information system. First of all, the computer-aided drawings (CAD) data were taken and put into GIS environment by using the ArcView software. At the end, the nine layers were obtained which were Parcel, parcel-block, building, point, triangulation, polygon, border, map border and usage border. Then the attribute information which is in DBF format was being imported into the system and integrated with the geographic information system to

establish the city information system. The Figure 2.9 is example of the system established.

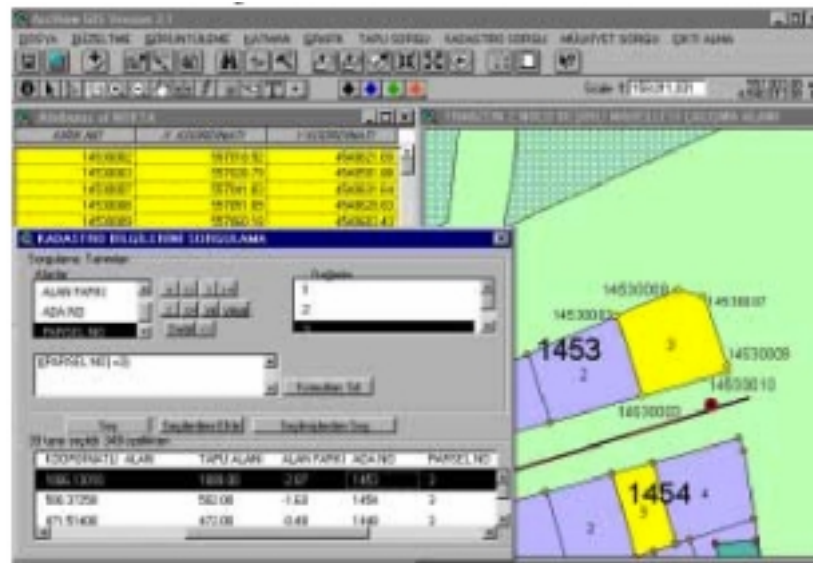


Figure 2-9 Cadastral Parcel Query of Yomralioğlu and Mataracı's study (Yomralioğlu and Mataracı, 1999).

One of the studies about city information system was performed by Kocaman in METU (Kocaman, 2001). In that study, conceptual modeling of a cadastral information system was developed by Spatiotemporal Entity-Relationship (STER) model.

The STER model is actually based on the entity-relationship model. It has two level of conceptual modeling. The first level is easier; less complicated and has less detail. The second level has a more detail and all information which is used to describe the application is given in that level (Figure 2.10). Besides, in the second level, STER model is extended into the entity-relationship diagram.

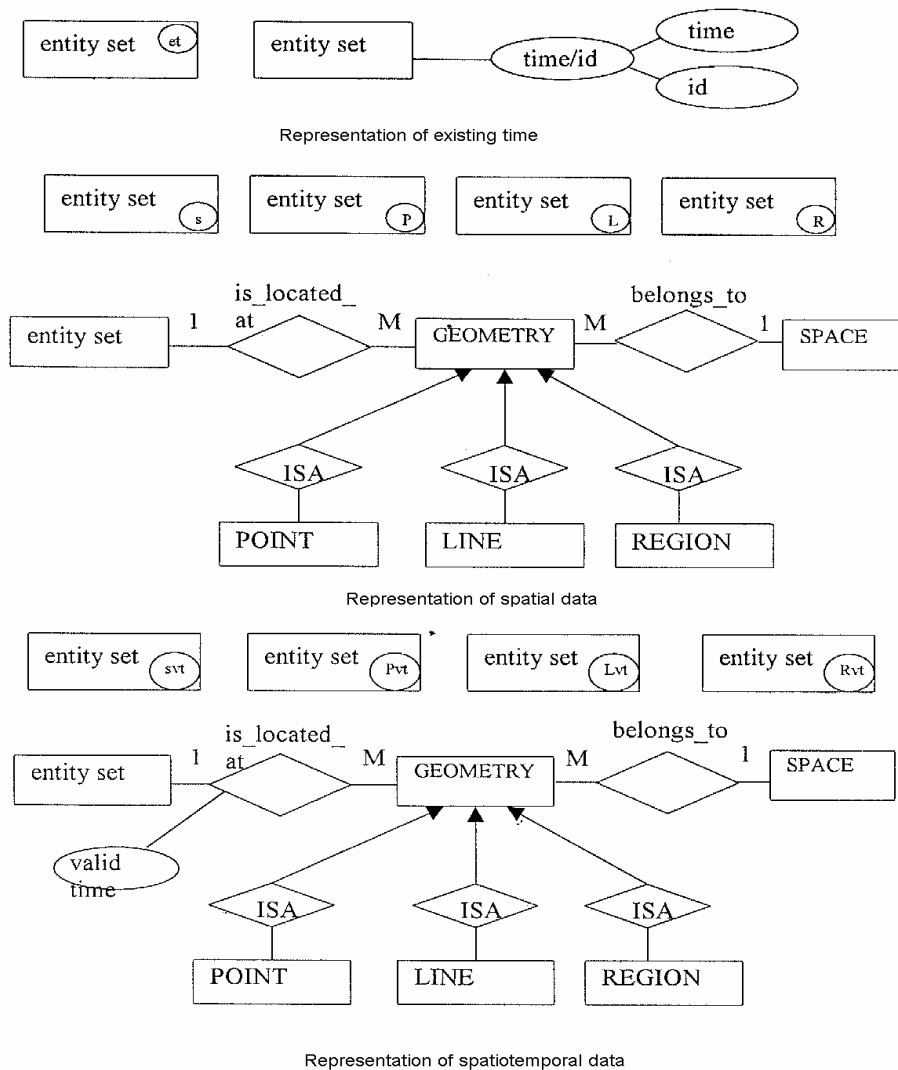


Figure 2-10 STER Model of Kocaman's study (Kocaman, 2001).

After conceptual design was completed, the logical design was performed. The conceptual schema was mapped into the relational model and normalization procedure was applied until all the relations became in boyce-codd normal form. After that the system was implemented on Oracle 8i DBMS software. In that study, MapInfo was used to show the geographic data to the user.

In Turkey, municipalities are also paying attention to cadastral information system and some of them have already transferred their systems to computer-based information system. One of them is Bursa Yıldırım Municipality. In this municipality, after making a feasibility analysis, the data were classified into three groups as; non-spatial attribute data such as, accounting records, attribute data of spatial data such as, address records and graphic information of spatial data. Then the information system establishment studies were started in October 1999. The studies were almost completed in August 2001. According to Kısa (2001), by the help of this system, the municipality has gained a capability to perform a lot of analysis, such as address query, tax loan query, population density analysis etc (Kısa, 2001).

Another municipality which has started city information system studies is Bursa Gürsu. The starting date of the studies is 12.02.2002 (Web Reference 6). At the beginning of the project, the four central districts are included. The studies are carried out by considering GIS and management information system.

The first thing during the project is to collect data by site survey. Data about buildings, people living in these buildings, real estate, water subscription, and environment cleanliness tax are collected in that survey. While site survey is continuing, the map related works are also carried out. In that part of the studies, the cadastral maps- municipal plans, satellite images and orthophotos are imported into the system. Then the maps and data that are collected by site survey are integrated in the GIS environment.

By the help of the system, address, buildings and person queries can be easily performed. In addition to these, real estate, water subscription, and environment cleanliness tax information are easily acquired (Web Reference 6).

In 2000, TKGM has started a new project called TAKBİS. (Land Registry and Cadastral Information System) The project is defined as an e-



government project (Web Reference 1). By the help of this project, TKGM services can be planned, managed and given more precisely and rapidly. In addition the data related with the ownership can be given to other association and enterprises easily and quickly so that they can be used in a broader area or fields. Moreover by the help of this project TKGM and other association and enterprises can be integrated easily.

The main objectives of TAKBİS project are:

- To keep an up-to-date data related with the land such as title deed records, cadastral maps in the computer environment by considering information system concept.
- To put the cadastral works into the multipurpose land information system.
- To give certain capabilities to TKGM so that it can provide better service.
- To provide easy-to-use computer aided environment so that, all the user of a system can use the system easily (Web Reference 8).

The system is planned to have four parts which are title deed registry application software, cadastral works application software, project observation software and source management (office automation) software (Web Reference 8).

In Turkey, the parcel related data in municipalities has not been a major subject in studies. All the studies so far have dealt with parcel data either geometrically or for registration purposes. In addition, the low level users have not been considered in existing studies. In other words, user should have some level of knowledge in order to use the system developed in the studies.

## **CHAPTER 3**

### **SYSTEM DEVELOPMENT AND IMPLEMENTATION**

#### **3.1 Development Methodology**

In this section a method that is used to develop the parcel-based information system is explained.

The method mainly consists of two parts:

1. Software design and implementation,
2. The database design and implementation.

In software part, there are five sub-steps

- Requirements analysis and definition
- System and software design
- Implementation and unit testing
- Integration and system testing
- Operation and maintenance

The method which is following these steps while developing a software is called as waterfall model or software life cycle because of cascading from one phase to another (Somerville, 2001). This is illustrated in Figure 3.1.

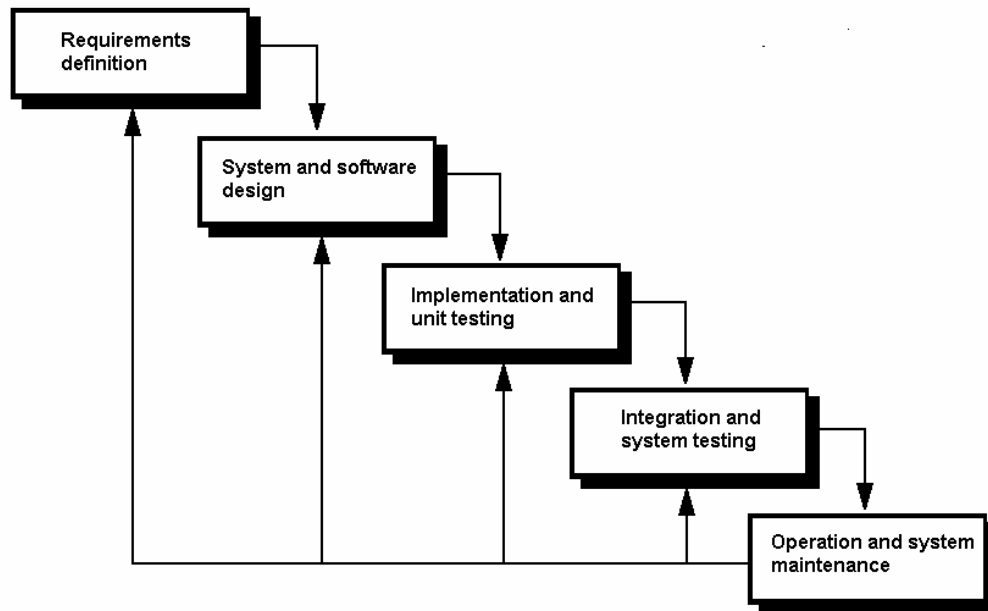


Figure 3-1 The software life cycle (Somerville, 2001)

1. *Requirement Analysis and Definition:* In this stage interviewing with system users, the system's services, constraints and goals are established.
2. *System and Software Design:* In this stage the overall system architecture is established. Basic software system abstractions are determined.
3. *Implementation and Unit Testing:* In this stage, software design is applied as a combination of small set of programs and each program units is verified against the specification.
4. *Integration and System Testing:* The small set of program units are integrated and whole system is verified to determine whether specifications are met.

5. *Operation and Maintenance*: The system is installed and put into service.

Normally these stages are not considered to be separate stages. One stage does not start after the previous one is finished. Instead of this, there is iteration between the stages during the development. For example, during coding, requirements may be changed and therefore coding may be changed. However, iterations are very costly. Because of this reason, after few iterations, some parts of the life cycle are frozen such as requirements analysis and the development process continue with the later stages (Somerville, 2001).

In database design and implementation, there are mainly 3 substeps. These are conceptual database design, logical database design and the implementation of the database.

### **3.3 Database Design and Implementation**

In this section, first the conceptual database design is covered. In conceptual database design, entities, attributes and relationships are determined and then they are put into EER. Then the logical design or data model mapping is discussed and then, the database system implementation is given.

#### **3.3.1 Conceptual Database Design**

In database design the first thing which should be performed is collection of requirements. After the requirements are collected, the conceptual schema is created by using high-level conceptual model (Elmasri and Navathe, 2000).

There are several high-level conceptual data models and one of the most popular one is the entity-relationship data model. On the other hand object data model is a high level implementation data model that is close to conceptual data model (Elmasri and Navathe, 2000).

In this study, enhanced entity-relationship data model which is extended version of the entity-relationship model is preferred rather than object data model. In addition for logical model, relational database is used instead of object oriented database. There are two main reasons for choosing relational database systems:

- Object oriented database system is still new and growing concept. Just because of this, it has not standards. For example, there is not standard ad hoc query language like relational database management system.
- Because it is a new concept, the implementation cost of these systems is high which is undesirable for the public establishments like municipalities.

While deciding the entities, attributes and relationships, the interviews made with the municipality personnel and excel and word sheets taken from the municipality are the primary reference point. The data which are in the excel and word format are given in Appendix A under the heading of data taken from municipality. In addition to these file cadastral procedure in Turkey is also considered while designing the conceptual model. The conceptual model established for this study is given in Figure 3-2.

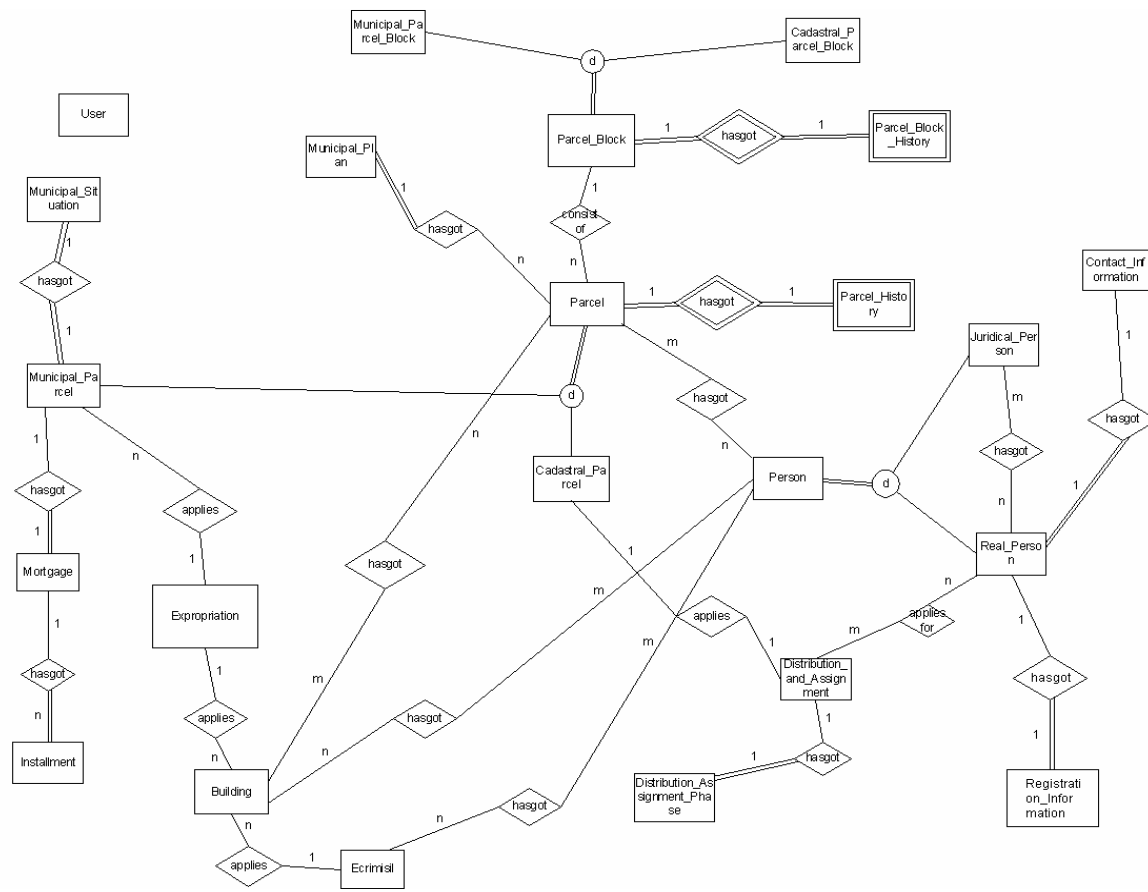


Figure 3-2 The EER model of parcel-based information system

### **3.3.2 Logical Database Design**

In logical design the conceptual model is mapped to a logical model. In this section the enhanced entity-relationship model is mapped into relational database model. After that the relations in database are formed. At last the normalization procedure is applied for the relations according to functional dependencies.

### **3.3.3 Database Implementation**

After normalizing the database relations, they are implemented on a database management system. In this study Microsoft SQL Server 2000 is chosen as a database management system.

Microsoft SQL Server is a relational database management system that supports client/server architecture. It can carry a relational database. The main features of SQL Server are;

- It can be hosted by Win9x and NT operating systems.
- It is scalable. It can run both on small scale desktop computer and large server system that has 32 processors.
- It has replication ability.
- The OLAP (On-Line Analytical Processing) services can be given.
- Importing and exporting data can be possible by data transformation services.
- The SQL2 is supported (Web Reference 2).

There are two ways of implementing these relations on SQL Server. The first one is to use the new table dialog box which is available under the tables menu of any databases. By using it, the tables are created in the design view. First of all, table columns are defined and then primary key of

tables are set. Then inside the diagrams menu, the relationships between the tables are created.

The second way of implementing relations on SQL Server is to use data definition language statements.

In this study the first way is preferred, but data definition language statements are given in Appendix B under the heading of Data Definition Languages.

### **3.4 System Implementation**

In this section, the implementation history of the parcel-based information system is demonstrated. First of all, the application architecture of the system is explained.

#### **3.4.1 Application Architecture**

Application architecture is a kind of language to express the construction style of your application to your group and your customer (Rofail and Martin, 1999). It defines what an application looks like at the end of the development process. Because of this, application architecture is very similar to the blueprints. In this study two-tier application architecture is used.

Two-tier application architecture is also called as client/server architecture and consists of two layers. One of the layers is client which request service and the other layer is the server which processes the requests that come from client. In this architecture, the client part is called as a fat client because all the presentation service and the business logic are running on this part (Hussey, 1998). The server side only maintains the database operations.

The main advantage of the two tier application architecture is the centralized database.



The server side implementation is explained in section 3.3.3, therefore in the next section, the client one is given.

### **3.4.2 Client Side Implementation**

The client side implementation of the parcel based information system is developed on the Visual Basic .NET which is a component of the Visual Studio .NET.

Visual Studio .NET is the complete set of development tools for rapid development of enterprise-scale ASP web applications, high performance desktop applications, XML web services and mobile applications (Web Reference 3). It includes component-based development tools such as Visual C#, Visual Basic and Visual C++. Moreover, it includes additional tools in order to simplify the team-based design, development and deployment of solutions. All of the tools share the same integrated development environment (IDE), which allow them to share the tools and facilities to create the mixed-language solutions.

In the development environment, first, all graphical user interfaces which are called as forms are created without writing any code. During this procedure, mainly the use cases, data insertion and update operations are being considered.

After graphical user interfaces are created, the classes are formed. Classes have services or methods those are implemented as functions. These functions are public so that they can be reached from everywhere in the code.

By the help of the functions, the data insert, update and selection operations are performed. During this, ActiveX Data Objects, which is one of the Microsoft technologies, is used.

In coding strategy, forms are directly used in order to implement database operations instead of defining attributes and setting and getting value of them. The overall coding strategy can be seen in the Figure 3.3.

```

Imports ADODB
Public Class RegistrationInformation
    Protected Const SQL_CONNECT_IN_STRING As String = "DSN=MSIS;Description=SQL
Server;UID=sa;PWD=;APP=Visual Basic;WSID=EMRAH;DATABASE=MUNICIPALITY"
    Private ConnectionString As String = SQL_CONNECT_IN_STRING
    Dim cnn As New Connection()
    Dim cmd As New Command()

    Public Function IRegistrationInformation(ByVal frmInRegistrationInformation As
frmRealPerson) As String
        On Error GoTo errorHandler

        cnn.Open(ConnectionString)
        Dim strSQLRI As String = "insert into REGISTRATION_INFORMATION values (' &
frmInRegistrationInformation.txtRPIdentityNumber.Text & ',' &
frmInRegistrationInformation.txtRPRegInfoProvince.Text & ',' &
frmInRegistrationInformation.txtRPRegInfoSubProvince.Text & ',' &
frmInRegistrationInformation.txtRPRegInfoDistrict.Text & ',' &
frmInRegistrationInformation.txtRPRegInfoPN.Text & ',' &
frmInRegistrationInformation.txtRPRegInfoVolume.Text & ',' &
frmInRegistrationInformation.txtRPRegInfoFamilyOrder.Text & ',' &
frmInRegistrationInformation.txtRPName.Text & ',' &
frmInRegistrationInformation.txtRPSurname.Text & ',' &
frmInRegistrationInformation.txtRPFatherName.Text & ',' &
frmInRegistrationInformation.txtRPMotherName.Text & ',' &
frmInRegistrationInformation.txtRPBirthPlace.Text & ',' &
IIf(frmInRegistrationInformation.txtRPBirthDay.Text = vbNullString, "NULL", "" &
frmInRegistrationInformation.txtRPBirthDay.Text & "')) & ',' &
IIf(frmInRegistrationInformation.txtRPDeathDay.Text = vbNullString, "NULL", "" &
frmInRegistrationInformation.txtRPDeathDay.Text & "')) & ',' &
frmInRegistrationInformation.txtRPSex.Text & ',' &
frmInRegistrationInformation.txtRPCitizenship.Text & '))"

        cnn.Execute(strSQLRI)
        cnn.Close()
        errorHandler:
        IRegistrationInformation = Err.Description
    End Function

    Public Function URegistrationInformation(ByVal frmUpdRI As frmUpdateRP) As String
        On Error GoTo errorHandler

        cnn.Open(ConnectionString)

        Dim strSQLUpdRI As String = "update REGISTRATION_INFORMATION set RI_Province='" &
Trim(frmUpdRI.txtRPRegInfoProvince.Text) & "',RI_Subprovince='" &
Trim(frmUpdRI.txtRPRegInfoSubProvince.Text) & "',RI_District='" &
Trim(frmUpdRI.txtRPRegInfoDistrict.Text) & "',RI_Village='" &
Trim(frmUpdRI.txtRPRegInfoPN.Text) & "',RI_Volume='" &
Trim(frmUpdRI.txtRPRegInfoVolume.Text) & "',RI_Family_Order='" &
Trim(frmUpdRI.txtRPRegInfoFamilyOrder.Text) & "',Name='" &
Trim(frmUpdRI.txtRPName.Text) & "',Surname='" & Trim(frmUpdRI.txtRPSurname.Text) &
"','Father_Name='" & Trim(frmUpdRI.txtRPFatherName.Text) & "',Mother_Name='" &
Trim(frmUpdRI.txtRPMotherName.Text) & "',Birth_Place='" &
Trim(frmUpdRI.txtRPBirthPlace.Text) & "',BirthDay='" &
Trim(IIf(frmUpdRI.txtRPBirthDay.Text = vbNullString, "NULL", "" &
frmUpdRI.txtRPBirthDay.Text & "')) & "',Death_Date='" &
Trim(IIf(frmUpdRI.txtRPDeathDay.Text = vbNullString, "NULL", "" &
frmUpdRI.txtRPDeathDay.Text & "')) & "',Sex='" & Trim(frmUpdRI.txtRPSex.Text) &
"','Citizenship='" & Trim(frmUpdRI.txtRPCitizenship.Text) & "' where
Identity_Number=" & (Trim(frmUpdRI.txtRPIdentityNumberValue.Text)) & " "
        cmd.ActiveConnection = cnn
        cmd.CommandText = strSQLUpdRI
        cmd.Execute()
        cnn.Close()
        errorHandler:
        URegistrationInformation = Err.Description
    End Function
End Class

```

Figure 3-3 The coding strategy

In Figure 3-3, the class RegistrationInformation is demonstrated. First, all the references ADODB is imported to the system, so that the components of the ActiveX Data Objects can be used such as; connection and command

components. Then the class is declared and connection string is created to connect the database. After that the functions are declared and implemented.

In this class, there are two methods, IRegistrationInformation and URegistrationInformation which are used to insert and update the registration information of a real person respectively. During these operations, the related form or graphical user interface is put inside the object as an identifier of a function, and the data on that form are used to perform the database operations. For example, in IRegistrationInformation function, form frmRealPerson is put into function as frmInRegistrationInformation and each value of textboxes on that form is used in SQL string in that function. So that in each attribute in database is set by the values of textboxes.

Besides common database operations, there are some map operations or functionalities. To achieve these functionalities MapInfo Professional version 6.5 is used.

MapInfo Professional is a mapping solution that allows performing a location-based data analysis (Web Reference 4). By using MapInfo professional;

- a detailed map can be created,
- the pattern and trend of a spatial data can be observed,
- a detailed and sophisticated data analysis can be performed,
- the demographics of customers and marketplace can be understood,
- geographically based assets, such as stores, people and property can be managed,
- logistics can be planned and a preparation can be made for emergency response.

While using MapInfo in this study, the coding strategy which is called as integrated mapping is used. In this coding technique, MapInfo is started in

the background of the program by using the createobject() statement of Visual Basic .NET. Then by sending MapBasic statements to the program, the map operations are performed. This strategy can be seen in the Figure 3.4.

```
Public Function mapload(ByRef frmMap As frmMapWindow) As
String

    mapinfo = CreateObject("mapinfo.application")
    mapinfo.do("Set Application Window " &
    frmMap.pctMapWindow.Handle.ToInt32)
    mapinfo.do("Set Next Document Parent " &
    frmMap.pctMapWindow.Handle.ToInt32 & " Style 1")
    mapinfo.do("run application
    ""D:\works\TEZ\Data\THESIS\Thesis.WOR""")

End Function

Public Function mapQueryAdress(ByRef frmMap As frmMapWindow,
ByRef dlgExpAddress As DialogExpAddress) As String

    Dim i As Object
    mapinfo.do("select * from Cankaya_Street order by
    mahalle group by mahalle into sel")
    mapinfo.do("fetch first from sel")
    For i = 1 To mapinfo.eval("tableinfo(sel,8) ")
        mapinfo.do("fetch next from sel")

        dlgExpAddress.lstDistrict.Items.Add((mapinfo.eval
        ("sel.mahalle")))
    Next
    dlgExpAddress.lstDistrict.SetSelected(0, True)

    mapinfo.do("select * from Cankaya_Street where mahalle
    = " & Chr(34) & dlgExpAddress.lstDistrict.Text &
    Chr(34) & " order by yoladi into sel")
    mapinfo.do("fetch first from sel")
    For i = 1 To mapinfo.eval("tableinfo(sel,8) ")
        mapinfo.do("fetch next from sel")

        dlgExpAddress.lstStreet.Items.Add((mapinfo.eval("
        sel.yoladi")))
    Next
End Function
```

Figure 3-4 The implementation details of map functions.

## **CHAPTER 4**

### **PARCEL-BASED INFORMATION SYSTEM FOR ÇANKAYA MUNICIPALITY**

#### **4.1 Software Design and Implementation**

In this section, first three parts of the software life cycle are covered. While the system and software design is explaining, the software developed for the municipality is demonstrated.

##### **4.1.1 Requirement Analysis**

In this section packages, use case diagrams and activity diagrams are given.

###### **i. Packages**

Analyzing the user's necessities and cadastral procedure, it is decided that the system has three different user types. First one is the citizen who is interacting with the municipality and the second one is a municipality personnel. The third and the last user is the planning officer who designs a municipal plan for either citizens or municipality. Therefore there are three packages in the system which are given in Figure 4.1.

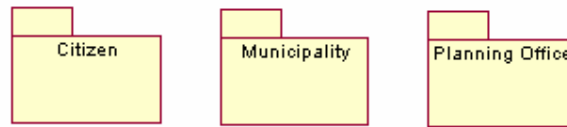


Figure 4-1 Packages in the system

## ii. Use Cases

After constructing the packages, the user's needs in each package are decided. While deciding the necessities, interviews made with users of the system are taken into consideration. In addition to interviews, the cadastral procedure is also considered. When the needs are determined, then the actors and use cases are constructed in each package. In package citizen and planning office there is only one actor who is citizen and city planner in planning office respectively. On the other hand, there are seven actors in municipality. These are classified according to department in which actors are working. Therefore the actors in municipality are, building land assignment department employee, expropriation department employee, hiring department employee, map department employee, map engineer in map department. Moreover there are two more actors who are city planner in municipality and municipal work department head.

The use case diagrams related with actors in each package are given in Figures 4.2 - 4.5.

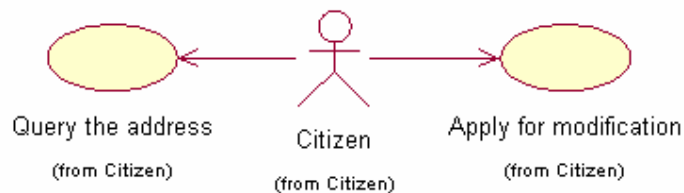


Figure 4-2 Use case diagrams for package citizen



Figure 4-3 Use case diagrams for package planning office

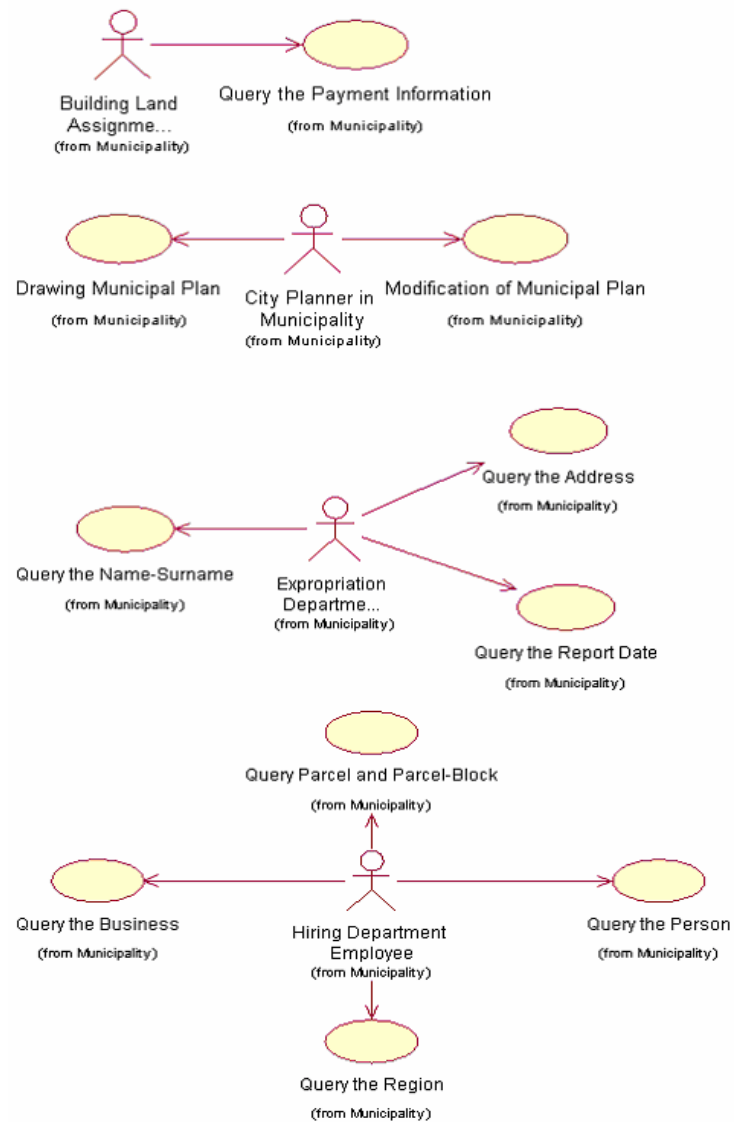


Figure 4-4 Use case diagrams for package municipality

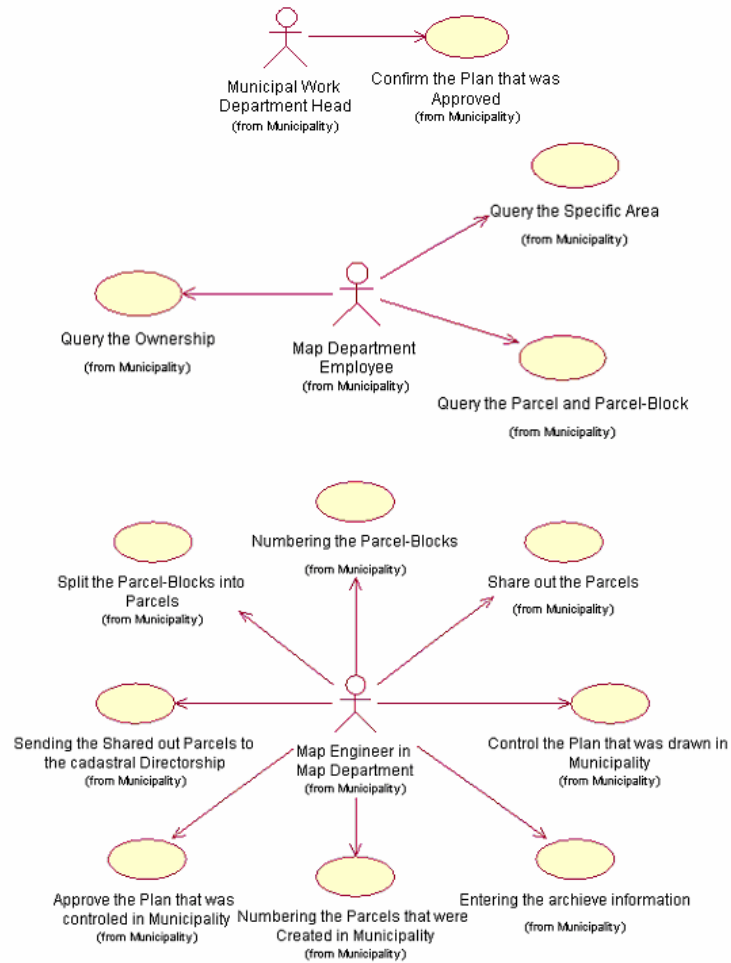


Figure 4-5 Use case diagrams for package municipality (continued)

### iii. Activity Diagrams

Use case diagrams define user and system requirements. After drawing them, they need to be detailed. In fact, the detailing process explains the user and the system interaction that means how a requirement is performed by using the system. The detailing is done by the help of the activity diagrams. Therefore for each use case, there is one or more activity diagram.

In this study the use case and activity diagrams are designed for the query operations which reflect the primary user needs. Therefore for inserting



new data or updating existing data, the use case and activity diagrams are not drawn.

#### 4.1.2 System and Software Design

In this section the general view of the system is designed by the help of the class and sequence diagrams.

##### i. Class Diagram

Class diagram is used to show services of the class and relationship between the classes. In this study classes do not have attribute because of the implementation detail. In implementation, instead of defining an attribute and use the set and get the method which is generally used to set the value of an attribute and get the value of an attribute, forms are used. This is discussed in section 3.4.2. Class diagram of the system is given in Figure 4-6.

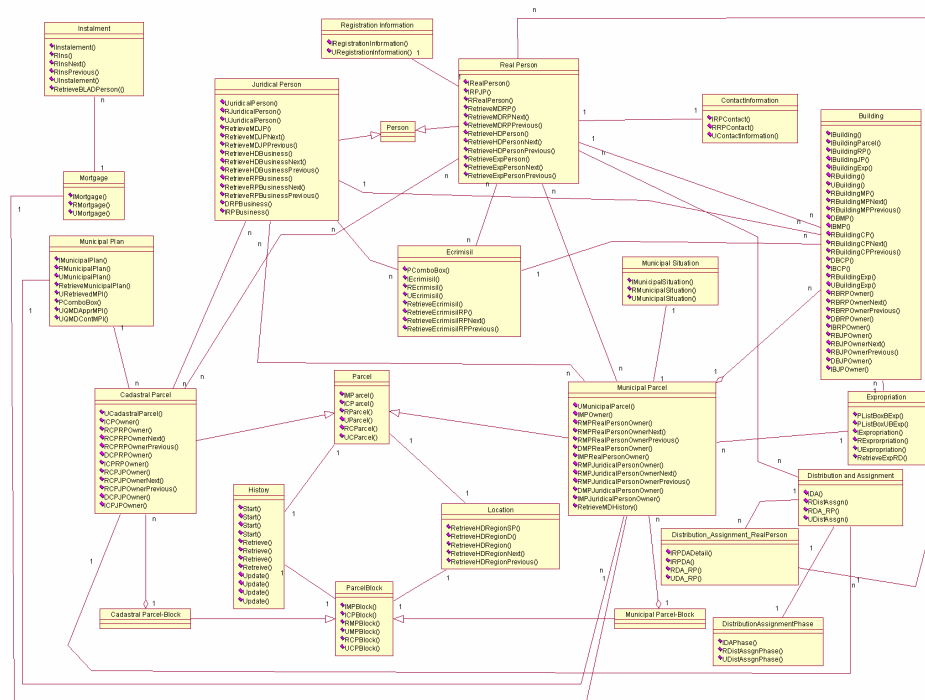


Figure 4-6 Class diagram of the system

## **ii. Sequence Diagram**

In UML, sequence diagram is used to define the timeline of the objects. During performing a set of operation, services of objects which are used by other objects are shown by the sequence diagram. Generally they are drawn or defined for each use case. For one use case, there can be one or more sequence diagrams just like activity diagrams. In this study, sequence diagrams are also defined at the use case level. Therefore they are also drawn for query operations only.

## **4.2 Overview of the System**

Parcel-based information system of Çankaya Municipality is a tool that is designed and developed to meet the needs of the staff of Çankaya Municipality. This tool can be separated into two parts if the functionalities are concerned. The menu and forms that manipulate basic database operations and the graphical user interface that manipulate map operations.

The login mechanism is introduced into the application so that the security and integrity of the data can be provided. After the user is starting the program, he/she should be logged on to enable the menus and buttons on the map window. The buttons and menus which will be enabled are decided according to the department of the user in which he/she belongs to. For example if the user does not work in the map department, the drawing buttons of the map window are not enabled at the end of the successful logon. This is illustrated in Figure 4-7.

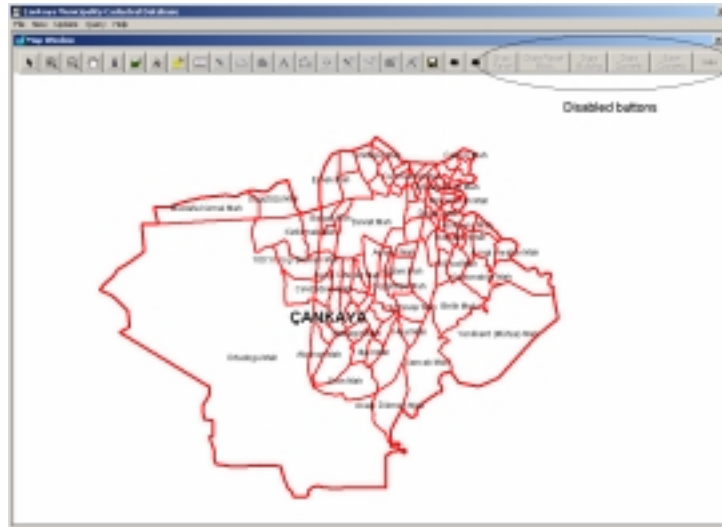


Figure 4-7 The logon mechanism

## 4.2 Map Handling

In parcel-based information system of Çankaya Municipality, some important map functionalities which are necessary to manipulate the map of Çankaya which consists of several layers are provided. These functionalities include zoom out, zoom in, pan, layer control, geometric feature selection tool, and point, line, polygon, text editing tools. These are supplied by the help of the buttons which are located at the map window of the system. This is illustrated in Figure 4.8.

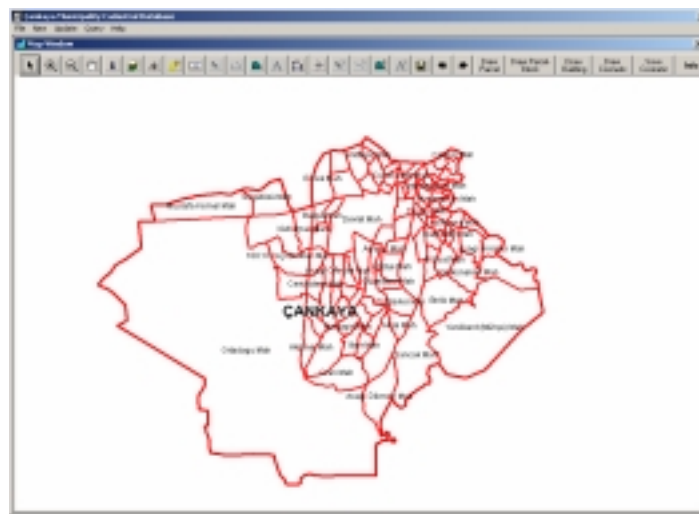


Figure 4-8 The map window of the system

The graphical data that are used in this study were acquired from the Çankaya Municipality and Başar Computer Systems. In this study the spatial data are arranged in seven layers. These are:

- Cankaya\_StreetNames
- Cankaya\_Street
- Cankaya\_Building
- Cankaya\_Parcel
- Cankaya\_ParcelBlock
- Cankaya\_District
- Cankaya\_SubProvince

The parcels and parcel-blocks that form the Cankaya\_Parcel and Cankaya\_ParcelBlocks layers respectively are obtained from municipality. The original format of these files was dwg and ncz. These file formats are for AutoCAD and NetCAD respectively. Therefore they were converted to the TAB files so that MapInfo can access them directly. At the time the conversion is performed, the database of the MapInfo files is also created.

The data that forms the other layers are acquired from Başar Computer System. The information on that layer is used for address queries primarily.

The “Draw Parcel-Block”, “Draw Parcel” and “Draw Building” buttons is added to the map window, so that the interactivity between the user and the system is increased. Using these buttons, the user doesn’t need to know which layer’s edit property should be on. He just clicks the button and the related layer becomes editable.

### i. Map Manipulation Capabilities

**Map Information Button:** By using map information button, user of the system can obtain information from map (See Figure 4-9).

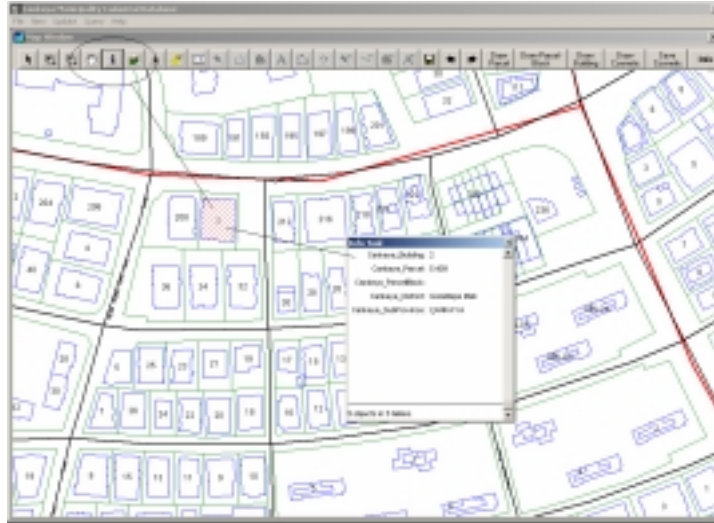


Figure 4-9 The map information button and dialog box which retrieve information from map.

**Layer Control Button:** By the help of the layer control button, user of a system can access the layer information of a system which are layer names, layer's display properties, layer's label properties etc. and he/she can change them easily (See Figure 4-10).

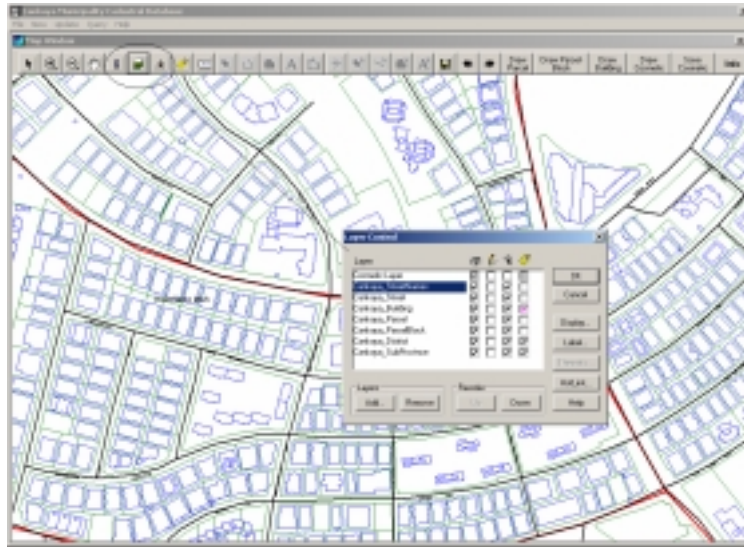


Figure 4-10 The layer control dialog box

*Geometric Selection Button:* By this button, the geometric features from map window can easily be selected (See Figure 4-11).

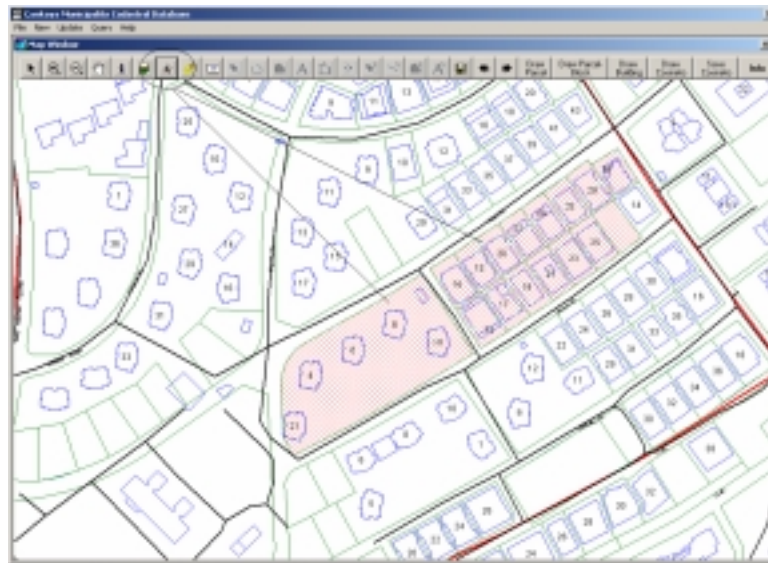


Figure 4-11 The geometric selection tool

*Label button:* By using label button, user can add a label which is retrieved from the database of a geometric feature (See Figure 4-12).

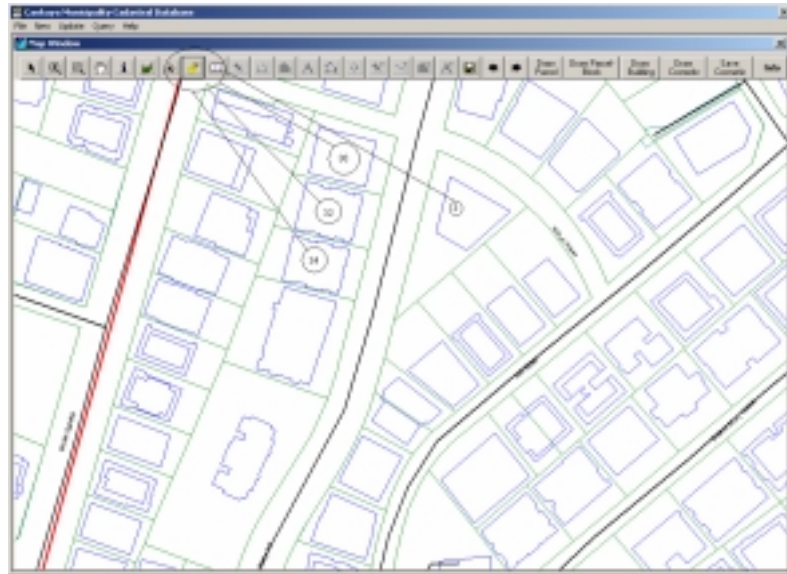


Figure 4-12 The label addition properties of a system

*Import button:* By the help of the import button, the user can import a map to the system easily (See Figure 4-13).

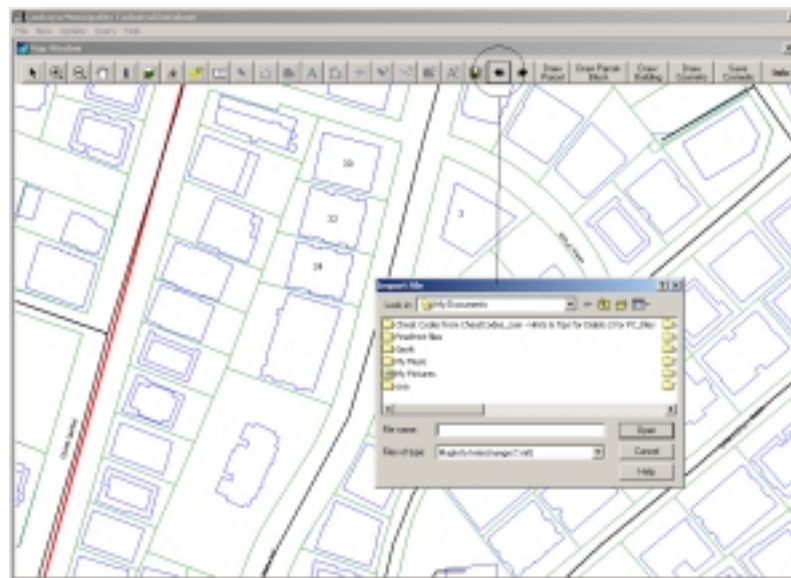


Figure 4-13 Import button and import file dialog box

*Export Button:* By the help of the export button, the user can export a layer from the system easily (See Figure 4-14).







*Buttons related with Inserting new geometric unit:* In case of changing a municipal plan, user needs to create a new parcel or modify an existing parcel. By the help of these draw parcel button and modify geometric unit buttons, these operations can be completed without dealing with any paper documentation, it provides time saving. These are illustrated in Figure 4-16 and 4-17.

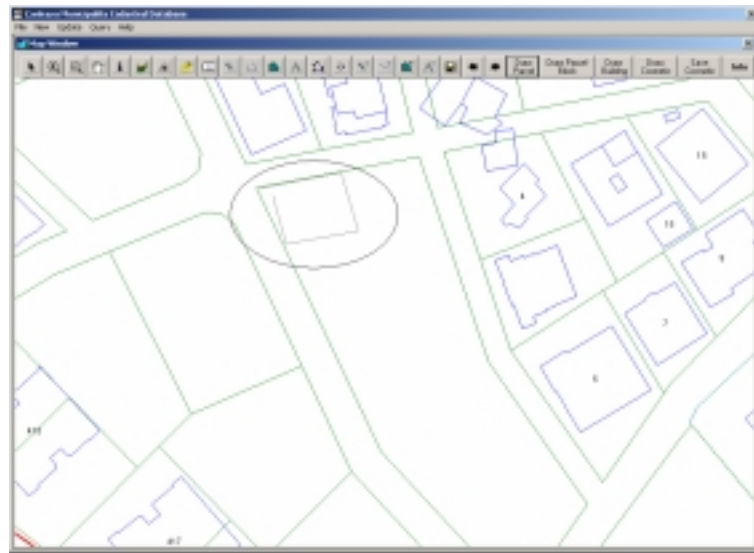


Figure 4-16 Creating new parcel

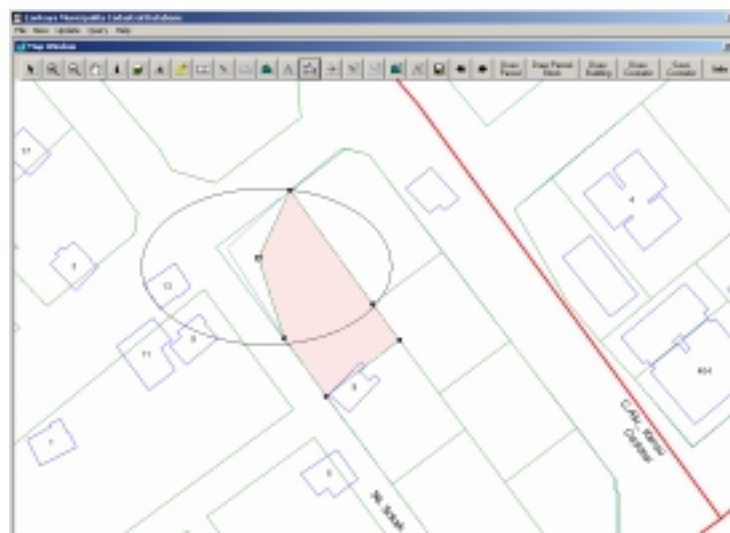


Figure 4-17 Modifying an existing parcel

*Info button:* Info button is also used in order to get information from map. The difference of this button from map information button is that, it provides an information which is retrieved from the relational database to the user. In addition user can update these information without calling them from update menus of a system. So if the user of a system wants to update an information related with the parcel, parcel-block or building, and he/she does not know the identifier of that geometric unit but he/she knows the location of the geometric unit, he/she can access information of that geometric unit by the help of the info button (See Figure 4-18).

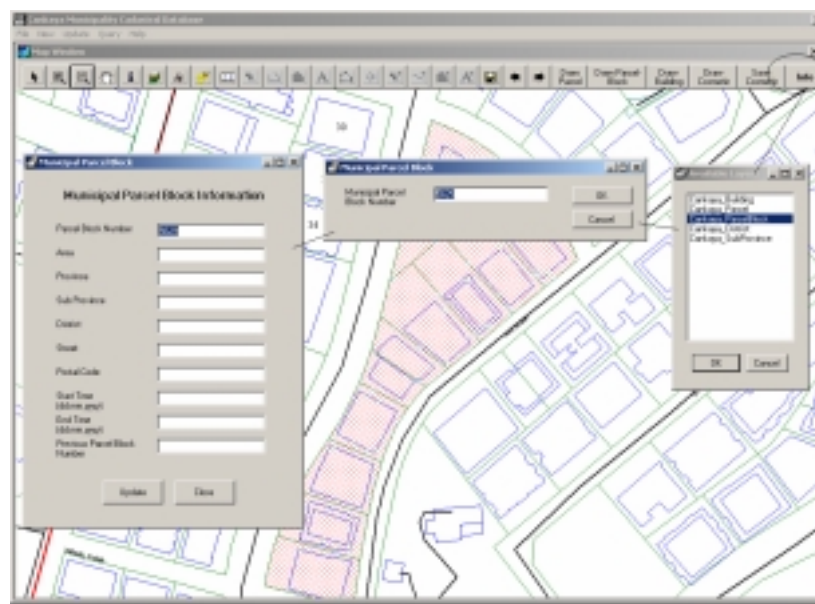


Figure 4-18 The info button, available layers, municipal parcel block dialog box and municipal parcel block update form.

### 4.3 Database Operations Manipulations

The database is manipulated on the SQL Server and then the data is inserted into tables. For table `municipal_parcel` and `municipal_parcelblock` the data is acquired from the geographic data which form the `Cankaya_Parcel` and `Cankaya_ParcelBlock` layers. The parcel block and parcel numbers are exported from MapInfo and imported to the SQL Server.

However during this operation, there are two important things that were noticed;

- Because there is not primary key concept of geographic data, when the exported data, which are from MapInfo, were tried to be imported to the SQL server, it was seen that there were many primary key conflicts.
- During the production stage of the geographic data in municipality, attention was not given to the numbering procedure of the parcels and parcel blocks. Because when the data were examined, it was seen that, there can be a parcel number whose parcel block number does not exist in the Cankaya\_ParcelBlock layer. Because of these data, the importing procedure to the SQL Server gives a referential integrity error.

As a result of these two situations, about 8000 parcel numbers were eliminated and municipality personnel will be informed about this problem.

For other information, the data which were given from the municipality in the Microsoft excel and word format are tried to be used. But primary problem of that data is that, they have missing information. For example they record data related with the person such as name and surname and they have never used a mechanism which should be used for separating one person from the other. Therefore in municipality, it is very difficult to separate people from each other whose names are identical. In this study, in order to solve this conflict, identity numbers of the people are used. However, because person related data which were taken from municipality do not contain the identity numbers of the people, they cannot be inserted into the database.

In the system, the database operations are implemented through three different menus. These menus are New, Update and Query.

*New:* By using submenus which are under the new menu, new data to database can be inserted. The available menus under New are demonstrated

in Figure 4-19. However not all of the data that is stored on the database are inserted into database by using these menus. For some data, there are forms that are called by buttons which are located in another form. For example, expropriation information that is related to specific building can be inserted by using a form that is called from Building form. This is illustrated in Figure 4-20.

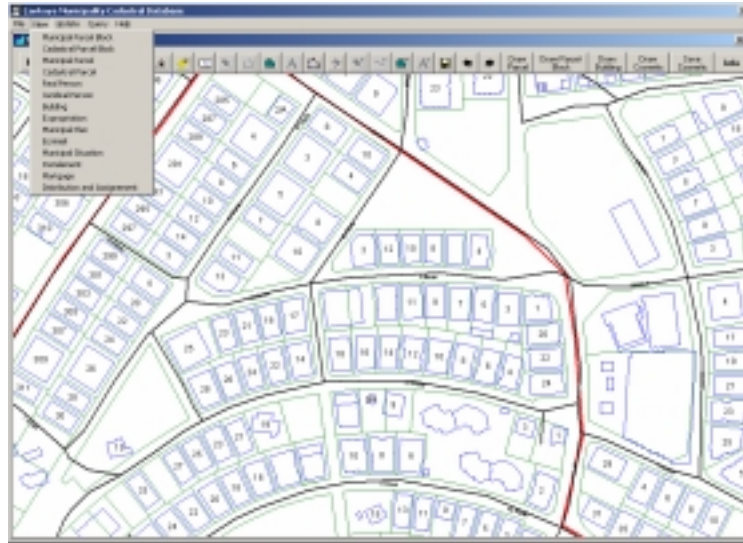


Figure 4-19 The New menu and submenus of New menu.

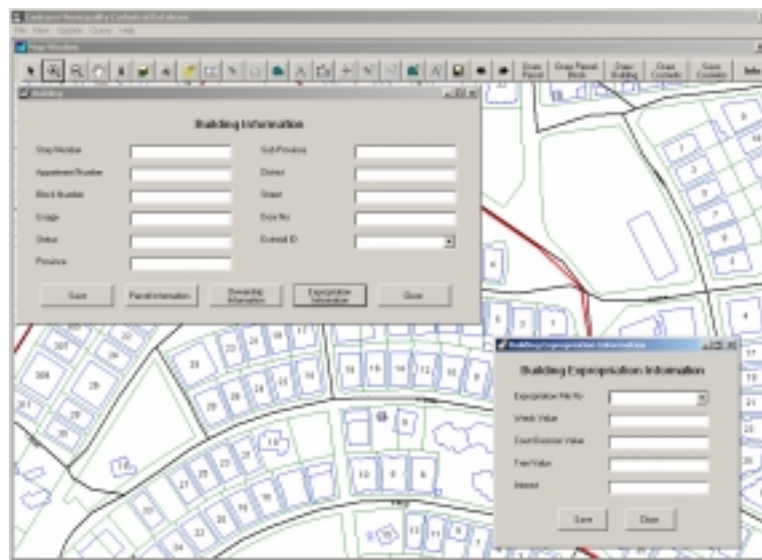


Figure 4-20 The Building and Expropriation information that is related to that building.

*Update:* Under the Update menu, there are several submenus which are used to update data in the database. In update operations, a user first finds the data desired to update, then he/she gives a new value to that data. The available menus under the update menu are given in Figure 4-21. In order to update data, user should first find the data which is desired to be updated. This is possible by using identifier of that data and this property is given by the help of the dialog boxes (See Figure 4-22).

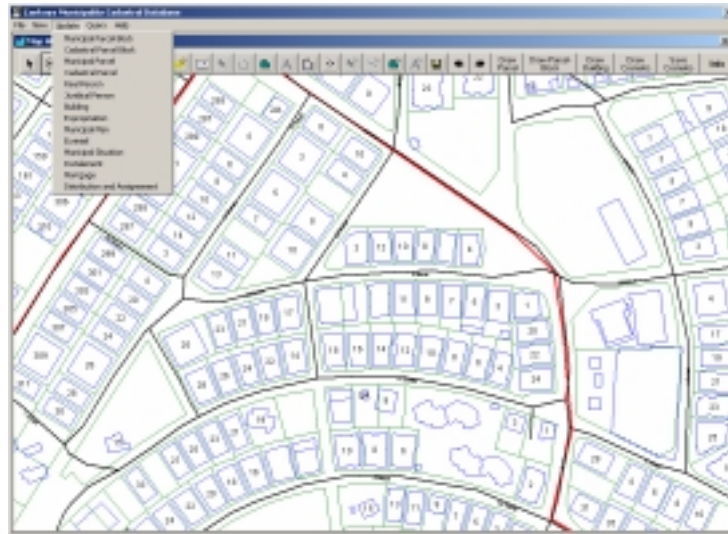


Figure 4-21 The Update menu and submenus of Update menu.

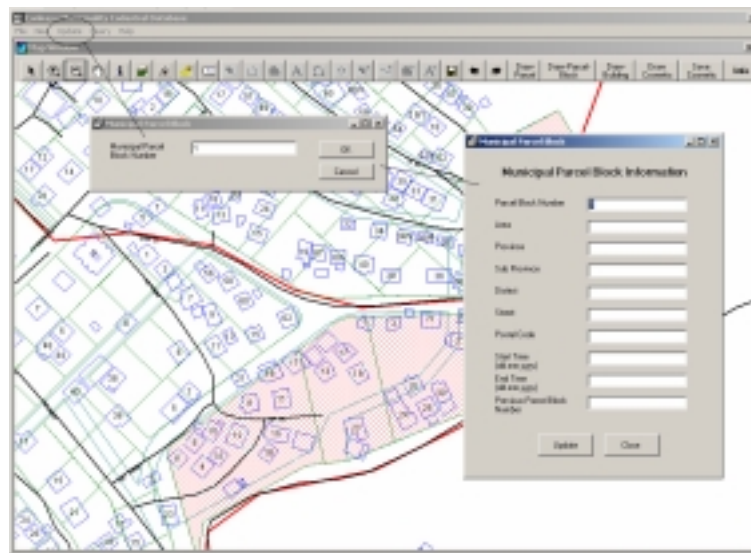


Figure 4-22 The dialog box that is used to call parcel block data

*Query:* Under query menu, submenus are classified according to departments in the municipality. Because, personnel in each department wants to make a different query.

- Municipal Works Department

*Municipal Plan Query:* This query is inserted into the system to show the unconfirmed municipal plan to the municipal works department head and allowed him/her to confirm that plan. The confirmation procedure is given by the help of the activity diagram (See Figure 4-23).

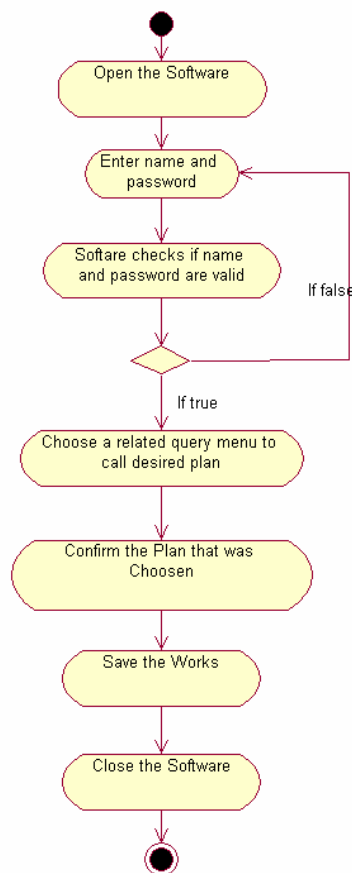


Figure 4-23 The activity diagram of actor “municipal work department head” and use case “confirms the plan that was approved” from package municipality.

In order to perform this query, system uses one or more classes and the methods of those classes. These are given in Figure 4-24.

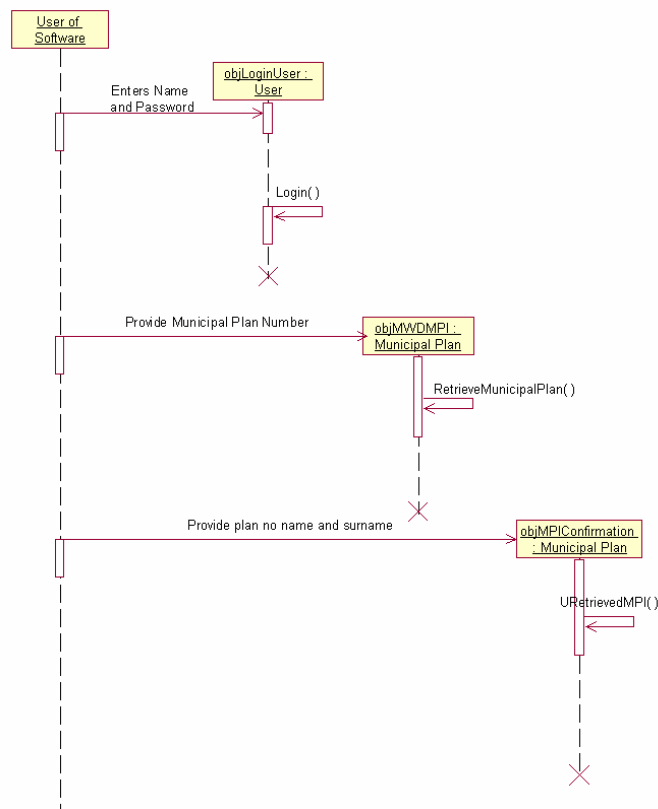


Figure 4-24 Sequence diagram of use case “confirm the plan that was approved” for an actor municipal work department head from package municipality.

According to Figure 4-24, the user of the system first provides his/her username and password information to the system. Then the objLogonUser object is created from the User class and by the help of the Login methods of the User class, the logon information is validated. Then user of software provides municipal parcel number to the system and objMWDHMPI is created from the Municipal Plan class. The RetrieveMunicipalPlan method is executed and the desired municipal plan is accessed. At last the user give his/her name and surname to the system and by the help of the URetreivedMPI method of a objMPIConfirmation object, the municipal plan is

confirmed by municipal works department head. This procedure is demonstrated in Figure 4-25.

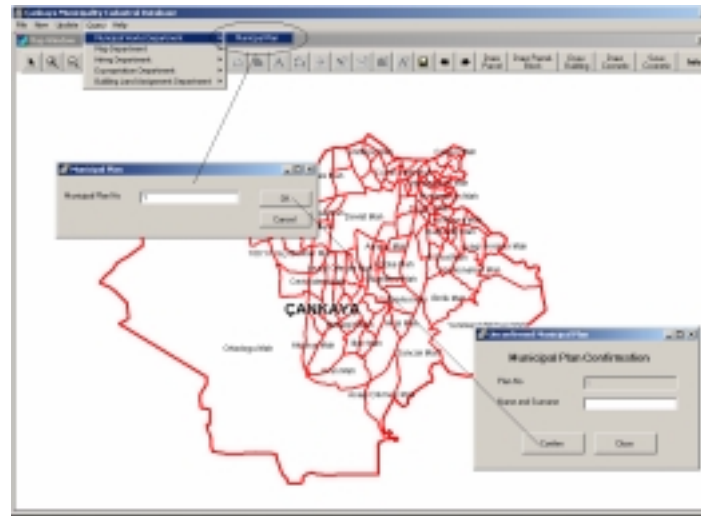


Figure 4-25 The confirmation procedure of a municipal plan

- Map Department

*Ownership Query:* This query is put into the system in order to find the ownership information of municipal parcel. The ownership can be either real person or juridical person or both of them. By the help of the query, the access time to the ownership information is a few second which is a great improvement. Because without the ownership query, this information is retrieved at least in a few hours. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-26 – 4-28.



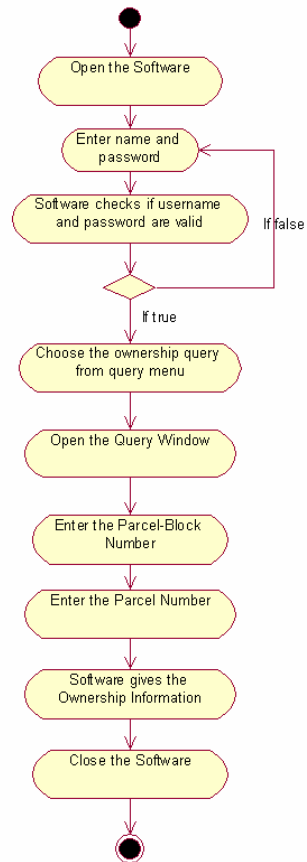


Figure 4-26 The activity diagram of actor “map department employee” and use case “ownership query” from package municipality.

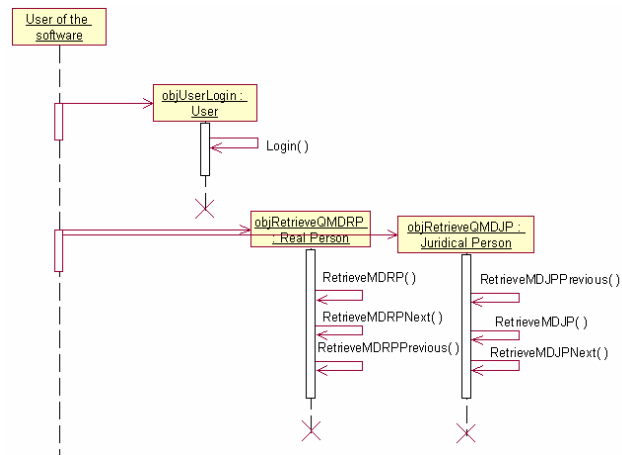


Figure 4-27 Sequence diagram of use case “query an ownership” for an actor map department employee from package municipality.

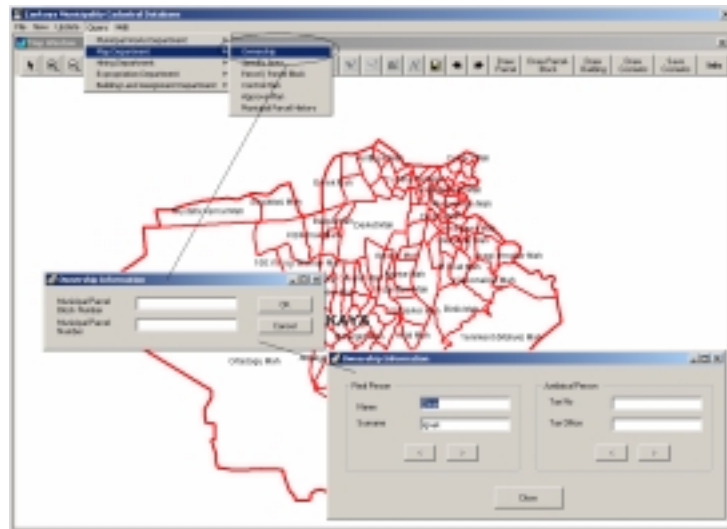


Figure 4-28 The ownership query for map department

***Specific Area Query:*** This query is used to locate the specific area such as park in the municipality. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-29 – 4-31.

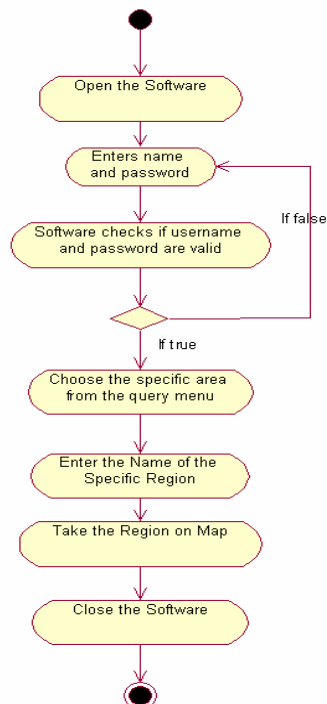


Figure 4-29 The activity diagram of actor “map department employee” and use case “specific area query” from package municipality.

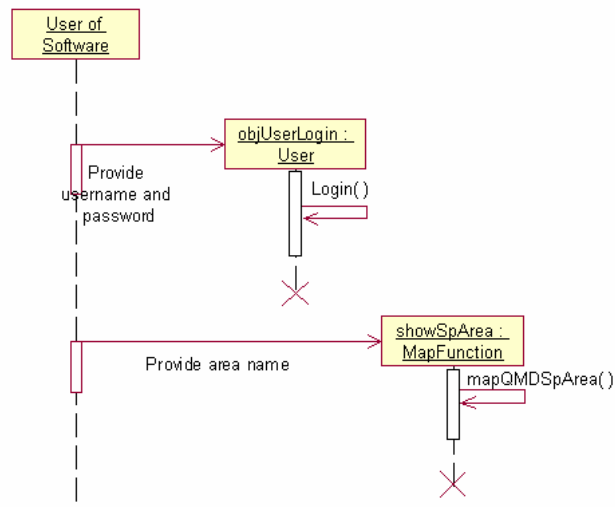


Figure 4-30 Sequence diagram of use case “query a specific area” for an actor map department employee from package municipality.



Figure 4-31 The specific area query for map department

*Parcel / Parcel Block Query:* This query provides the location of the municipal parcel to the user of the system. The system gives a result in a few seconds. If the system is not used and the user does not know about the map, finding the specific parcel on map takes lots of hours. Therefore system provides great convenience to the user. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-32 – 4-34.

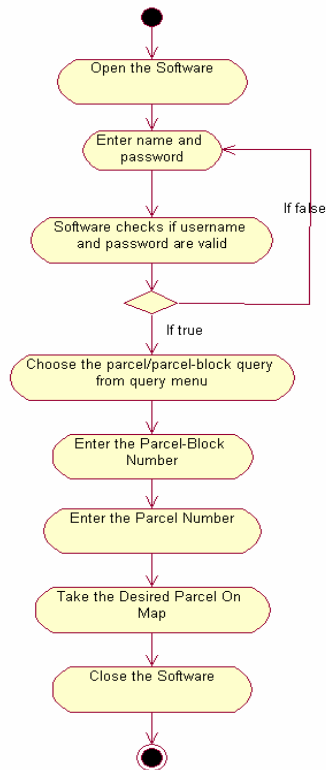


Figure 4-32 The activity diagram of actor “map department employee” and use case “parcel, parcel-block query” from package municipality.

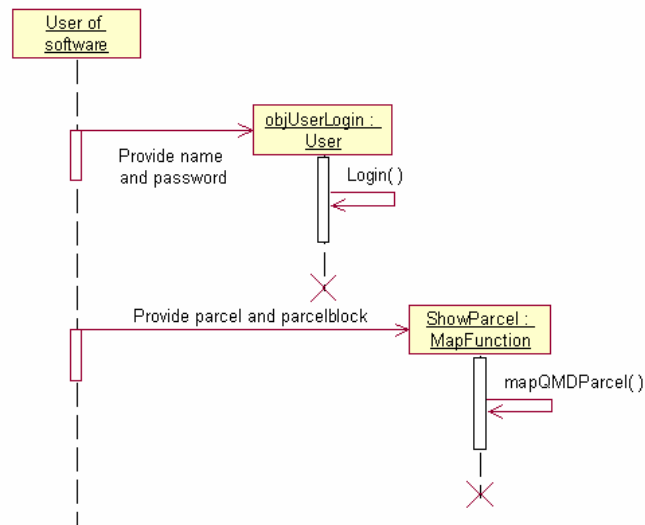


Figure 4-33 Sequence diagram of use case “query a parcel and parcel-block” for an actor map department employee from package municipality.

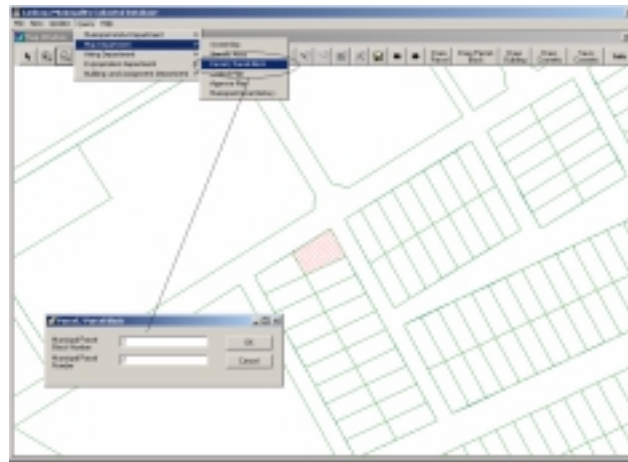


Figure 4-34 Municipal parcel query

*Control Plan Query.* This query is inserted into the system to show the uncontrolled municipal plan to the map department employee and allowed him/her to insert a control approval for that plan. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-35 – 4-37.

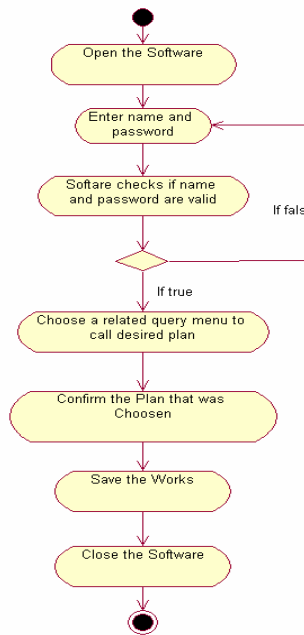


Figure 4-35 The activity diagram of actor “municipal work department head” and use case “confirms the plan that was approved” from package municipality.

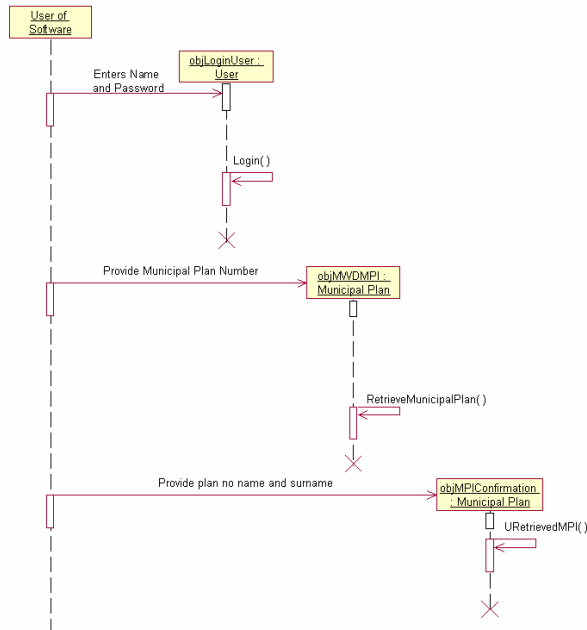


Figure 4-36 Sequence diagram of use case “confirm the plan that was approved” for an actor map department employee from package municipality.

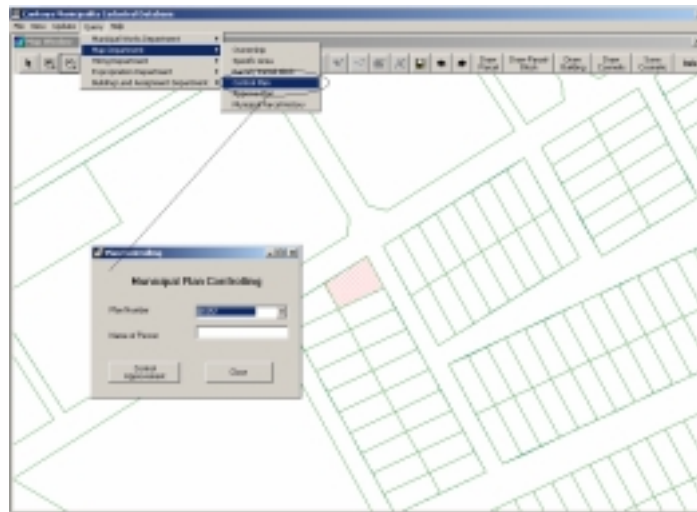


Figure 4-37 The municipal plan control approval dialog box

- Hiring Department

*Tenant Query:* This query is put into the system to retrieve the tenants of the municipality. Instead of searching paper documentation to find the tenants of one of the municipal parcel, the use of this system reduces time which is

required to retrieve the desired information. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-38 – 4-40.

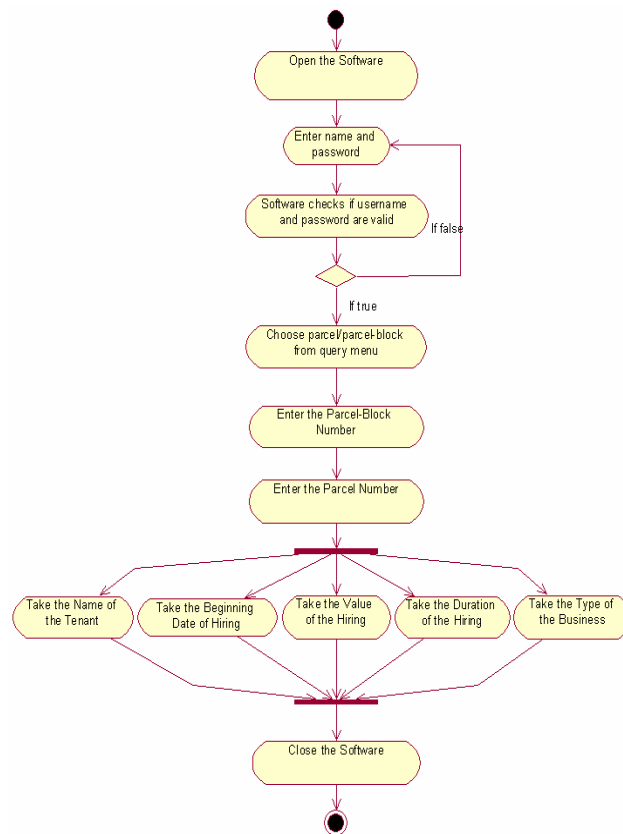


Figure 4-38 The activity diagram of actor “hiring department employee” and use case “parcel-block/parcel query” from package municipality.

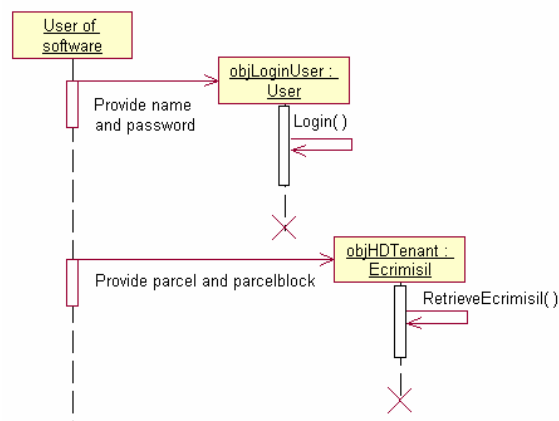


Figure 4-39 Sequence diagram of use case “query a parcel and parcel-block” for an actor hiring department employee from package municipality.

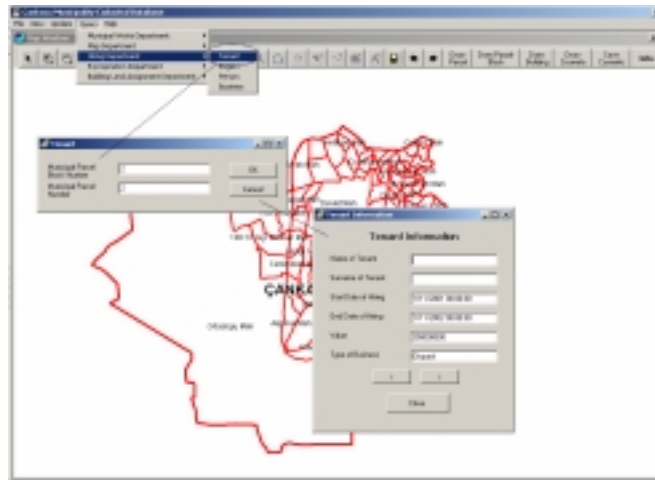


Figure 4-40 The tenant information query

*Region Query:* This query is inserted into the system to find the hired places in a specific region. The result of the query is either building ID or municipal parcel number. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-41 – 4-43.

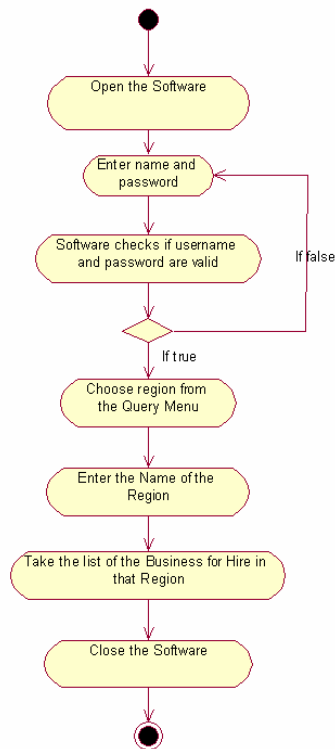


Figure 4-41 The activity diagram of actor “hiring department employee” and use case “region query” from package municipality.



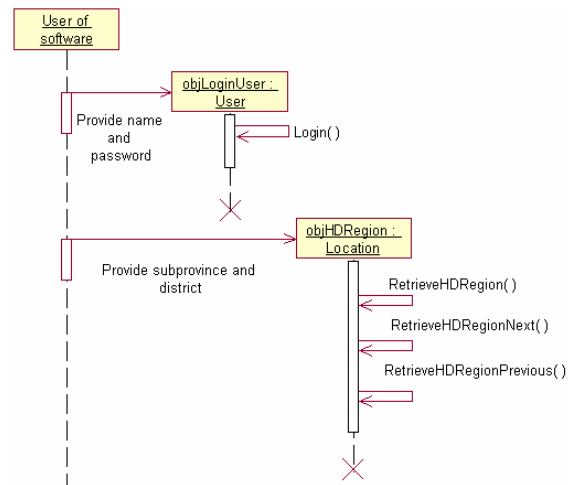


Figure 4-42 Sequence diagram of use case “query a region” for an actor hiring department employee from package municipality.

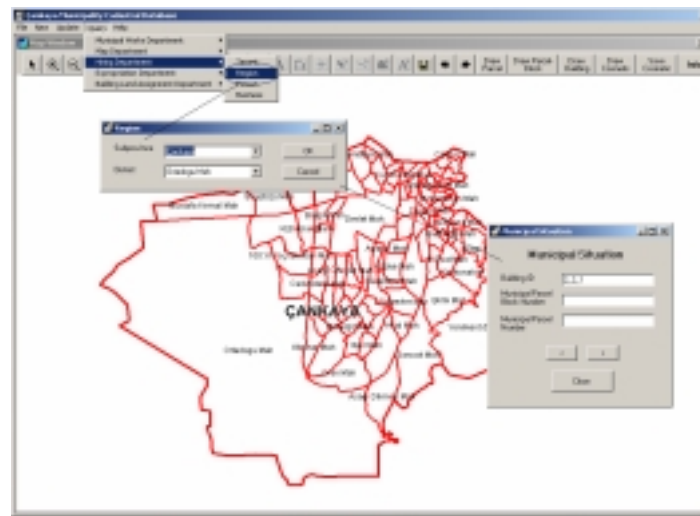


Figure 4-43 The region query demonstration for hiring department

*Person Query:* This query is put into the system in order to retrieve the list of business of a specific person and its parcel and building information. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-44 – 4-46.

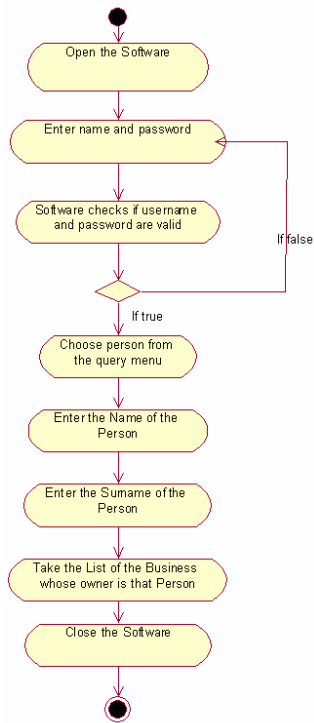


Figure 4-44 The activity diagram of actor “hiring department employee” and use case “person query” from package municipality.

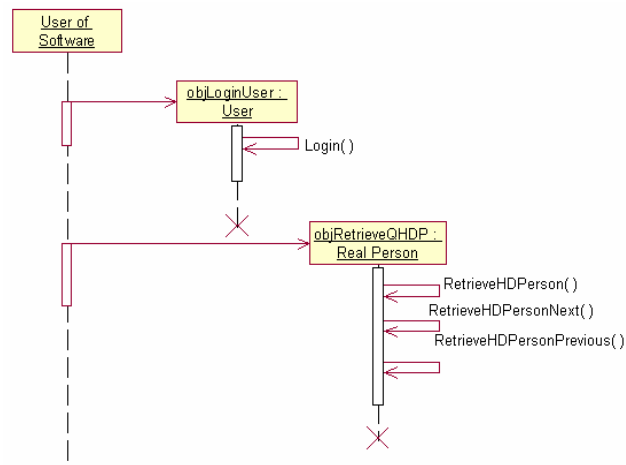


Figure 4-45 Sequence diagram of use case “query a person” for an actor hiring department employee from package municipality.

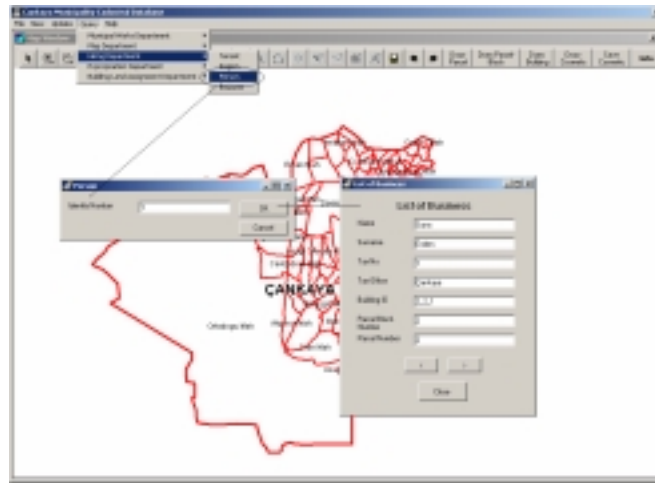


Figure 4-46 The person query for hiring department

*Business Query:* This query is designed for finding information related with a specific business such as its name, building ID, parcel number etc. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-47 – 4-49.

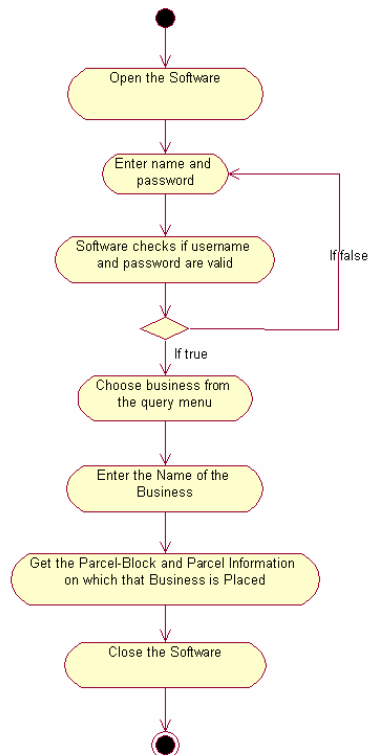


Figure 4-47 The activity diagram of actor “hiring department employee” and use case “business query” from package municipality.

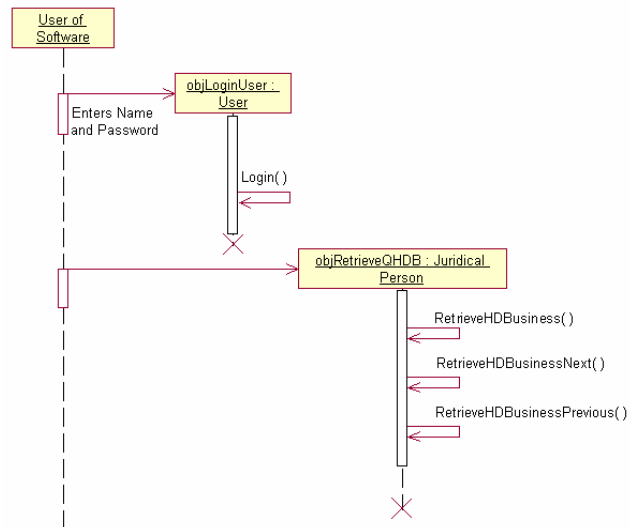


Figure 4-48 Sequence diagram of use case “query a business” for an actor hiring department employee from package municipality.

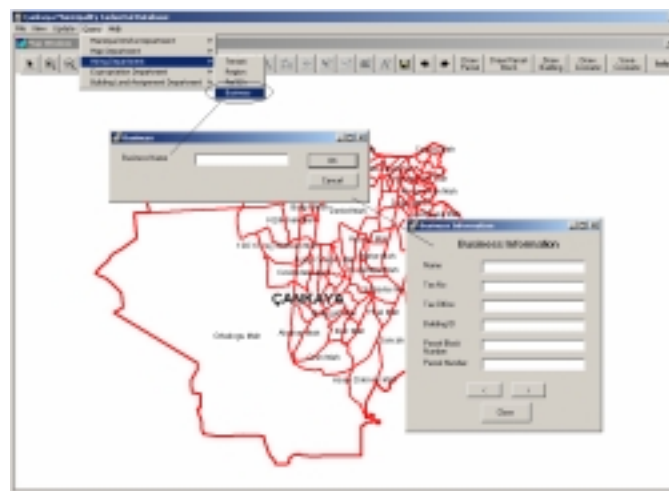


Figure 4-49 The business query dialog boxes for hiring department

- Expropriation Department

*Person Query:* This query is put into the system to retrieve the person related information. Such as reporting date of an expropriation and contact address of a person. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-50 – 4-52.

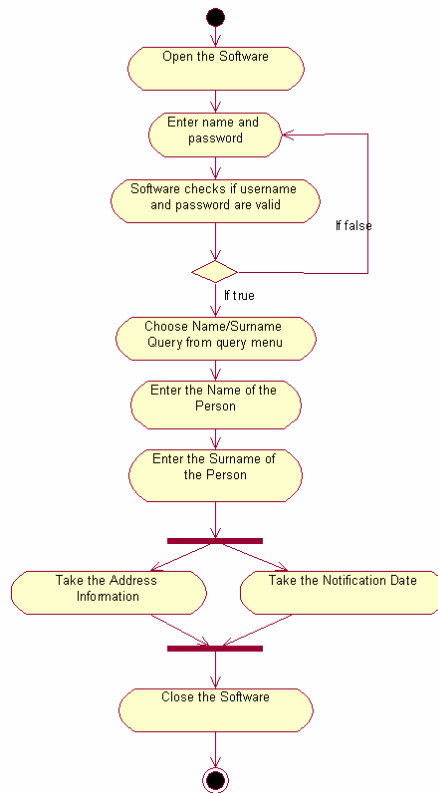


Figure 4-50 The activity diagram of actor “expropriation department employee” and use case “name-surname query” from package municipality.

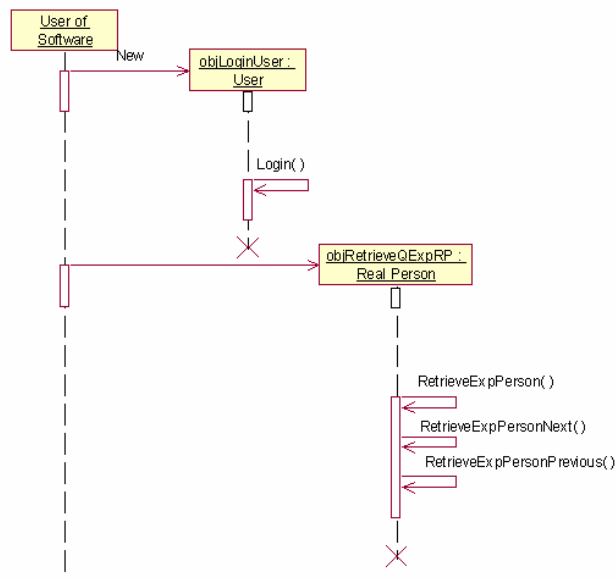


Figure 4-51 Sequence diagram of use case “query a name and surname” for an actor expropriation department employee from package municipality.

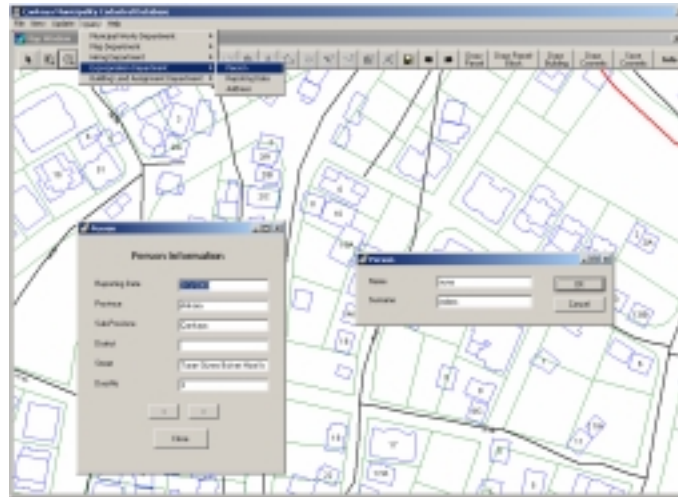


Figure 4-52 Person query dialog boxes for expropriation department

*Reporting Date Query:* This query is designed to retrieve the person information to whom the expropriation notification is given within the specified date. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-53 – 4-55.

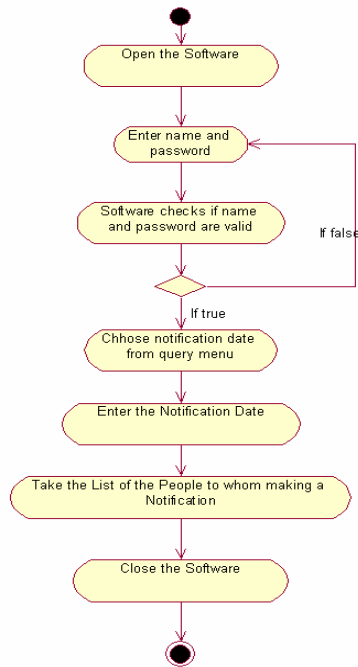


Figure 4-53 The activity diagram of actor “expropriation department employee” and use case “report date query” from package municipality.

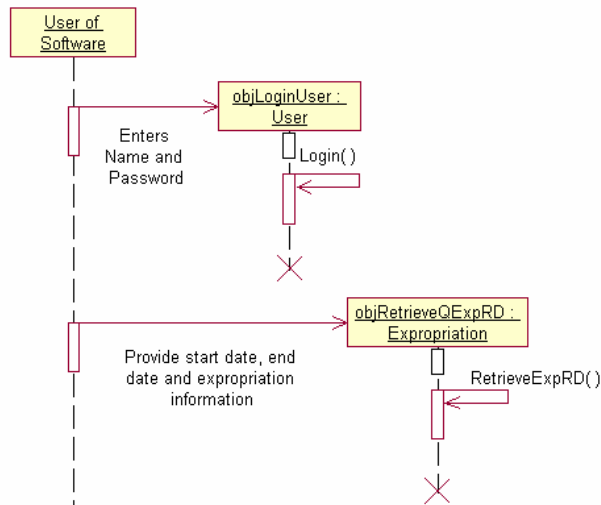


Figure 4-54 Sequence diagram of use case “query a report date” for an actor expropriation department employee from package municipality.

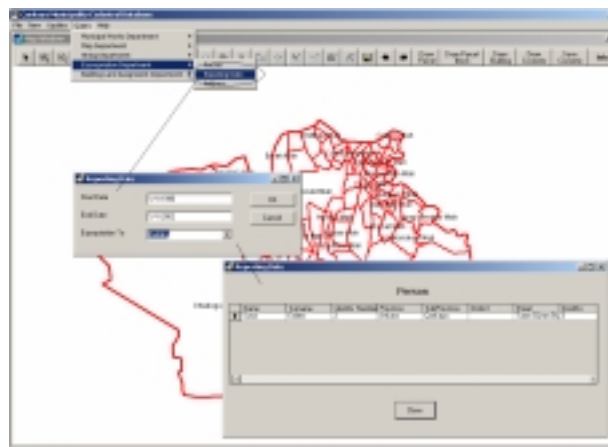


Figure 4-55 The reporting date query dialog boxes

*Address Query:* The address query is one of the queries that expropriation department employees want. In this query the streets are classified according to the districts and then these streets are queried. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-56 – 4-58.

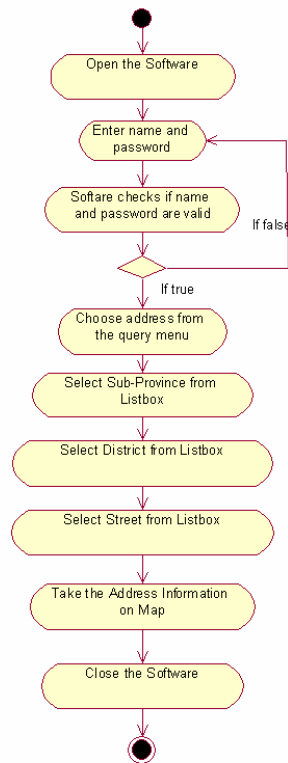


Figure 4-56 The activity diagram of actor “expropriation department employee” and use case “address query” from package municipality.

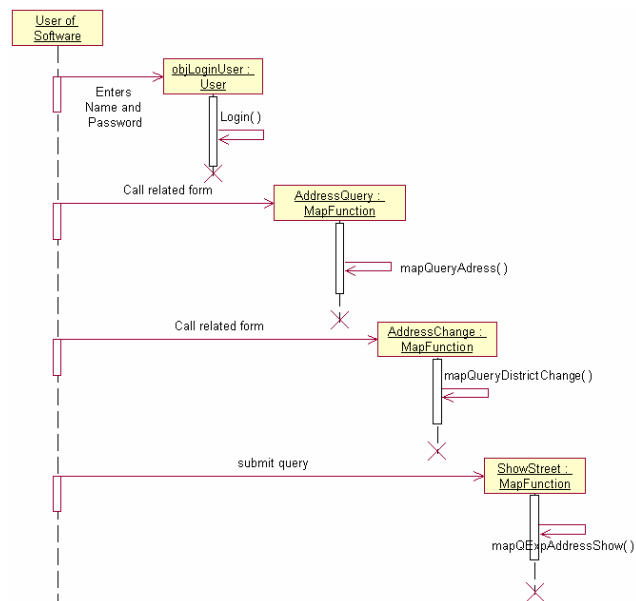


Figure 4-57 Sequence diagram of use case “query an address” for an actor expropriation department employee from package municipality.





Figure 4-58 The address query dialog boxes

- Building Land Assignment Department

*Person Query:* At the end of the distribution and assignment procedure, the municipal parcel is sold to people. The value of a parcel can be paid by cash or instalment. The building land assignment department employee can see the instalment plan by making query. The related activity diagram, sequence diagram and query demonstration are given in Figures 4-59 – 4-61.

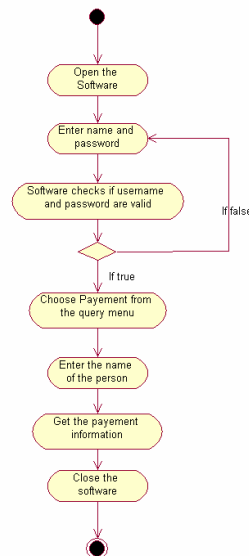


Figure 4-59 The activity diagram of actor “building land assignment department employee” and use case “query the payment information” from package municipality.

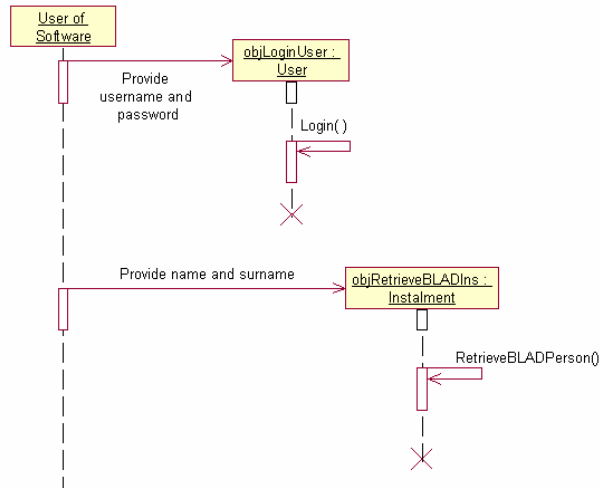


Figure 4-60 Sequence diagram of use case “query the payment information” for an actor building land assignment department employee from package municipality.

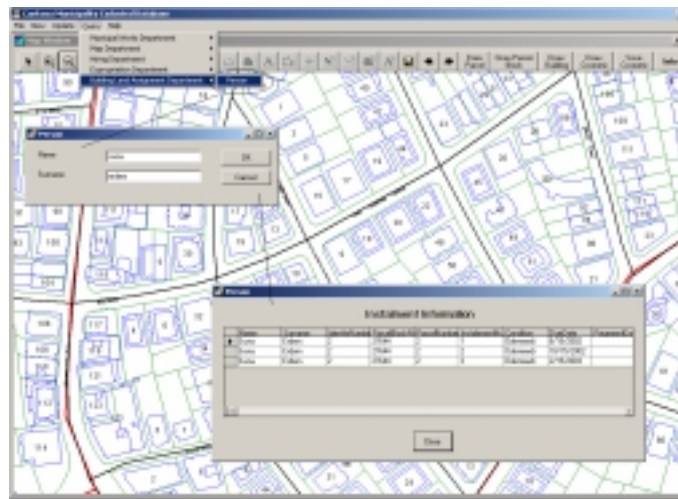


Figure 4-61 Instalment plan query

## **CHAPTER 5**

### **CONCLUSION AND RECOMMENDATION**

#### **i. Conclusion**

Today, fast access to information becomes very critical for all kind of establishments. In addition to fast access, the accuracy of information is also important. Because of these two concerns, very well management of information is very critical and important, and this can be achieved by well-designed information system.

In recent years, both the geographic data production capability and volume of geographic data have increased dramatically. In addition to these two improvements, volume of attribute data related with geographic data have also increased. The progress of computer technology plays a very important role for these improvements. Again due to the rapid growing of computer technology and industry, the management of geographic data and its attribute data are getting easier and easier.

Geographic information system is a tool that helps us to handle geographic data and their attributes. Parcel-based information system is a kind of geographic information system which aims to solve the data management problem on cadastral basis.

In this context, the central data storage area is created which is used to manage the parcel related data for Çankaya Municipality. Relational

database management system which is a common and well known standard for database design is used for the data storage area. It is supported by many software companies so the cheaper solution can be found and it has a query language namely structured query language (SQL). SQL is also a world wide standard.

In addition to parcel related data or attribute data, geographic data are also stored in central location. In Çankaya Municipality, the geographic data have been stored in computer aided design (CAD) environment so far. Therefore the geographic data are imported into GIS environment so that the geographic data become wiser. As a result of this, the database concept is supported by the geographic data.

Then in order to insert new data, update and retrieve available data, the software is developed. In design and development process, the object-oriented analysis and design is used.

The two tiers or client/server application architecture is proposed in this study. The major advantages of the system are the central data management and low implementation cost. By central database, the data redundancy can be prevented and instead of recording the data on the excel or word sheets, which is the current situation in municipality, the database management system is used. In addition, by the help of the application which is built on the database, direct access of the data is prevented, so that the security of the data is guaranteed.

Conclusions which are retrieved from this study are useful for both municipalities and developers. While developing the parcel-based information system, one of the major guides is cadastral laws. However applying just cadastral laws makes the contents of the system very extensive. Therefore another guide should be followed to narrow the contents of the system. The user requirements analysis is a very useful mechanism to limit the content. The needs of the user help the developer or designer that, he/she can make a decision about what should be done. In this study, the user requirements

are also collected and used during the development stage. Especially, it helps to the database design. Besides, the types of the queries that are inserted into the system are decided by analyzing user requirements.

Using EER is useful for conceptual modeling. Because by using EER, one can use the inheritance and generalization-specialization mechanisms which allow a better modeling and understanding of the system. In this study, these mechanisms are also used during the modeling of the parcel entity. Besides, by applying database normalization procedure and making all relations to the boyce-codd normal form eliminates the common database anomalies. Moreover, relational database management system makes the attribute data retrievable from one or more location at the same time. In other words, more than one user can manipulate the data at the same time. But the same thing is not true for geographic data. The geographic data can be changed by only one user at a time, but other users access the data as read-only.

Object-oriented analysis and design makes the system extendibility much easier. Because system is developed to have independent parts. Therefore if different functionality is desired to be added into the system in the future, it does not take too much effort or time. Adding a service to one class increases the functionality. Besides, the complexity of a system is decreased. If the same implementation were performed by classical approach or procedural language, then system would become much more complex and adding or changing functionality would have taken more time.

During application development, it is observed that, using UML and waterfall methodology together provides great control over development steps. Because the designer can plan the main functionalities of the system and the tasks which should be performed in each step. Therefore it is very difficult to skip a task during the development stage.

Moreover the system is software independent. In this study the geographic and attribute data are managed separately. Geographic data are

managed by MapInfo and attribute data are managed by SQL Server. In attribute data part, the software can be changed easily. Instead of SQL Server, MySQL, which is a cheaper solution, can be used, and the same thing is valid for geographic data part. As far as the data format is changed from MapInfo file format to appropriate file format, another software is easily adapted to the system.

Another conclusion gained from the study is that; the system can be easily used by low-level or non-technique municipality staff. In order to query the relational database, one should know the query language syntax but in municipality such users are not available. Therefore, by the help of the developed system, complex queries can be performed by a button click.

The last conclusion is about the time concept in cadastral data. Generally cadastral data is time dependent which means as the time passes, both the geometric and attribute data about cadastre is changed. Therefore time concept should be introduced into cadastral data management. In this study, time is recorded for the archive purpose because of municipality staff's requirements. Instead of holding geometric changes in the parcels or parcel-blocks with respect to time, changing time in geometric properties is recorded so that the historical information about one parcel or parcel-block can be accessed.

The main contribution of the parcel-based information system developed in this study is that it is one of the first studies that have been developed for parcel related data in municipalities so far. Therefore it is the first step of the automated and computer aided information system for municipalities in Turkey. In addition the system is developed by using object-oriented analysis and design. This methodology does not contribute the parcel-based information system anything new, rather, by the help of it, the existing concepts can be used much more effectively.

## ii. Recommendation

After giving a brief explanation of what is being done in this study, it is time to mention that what can be performed to improve this study one step further. The purpose of this study is to develop a parcel-based information system in order to manage the parcel related data of any municipality and system is aimed to be implemented on Çankaya Municipality. The improvements can be divided into three parts; data management, application architecture and integration of this system to other systems.

- **Data Management:** In this study geographic data management is performed by using MapInfo Professional, and non-geographic data management is done by Microsoft SQL Server and one of the major disadvantages of a system is this separation. Because, by this way, the data integrity between the geographic and non-geographic data becomes difficult to maintain. This separation can be combined and implemented in one system which means, all the spatial and non-spatial data can be stored on database management system. Then data integration can be provided much more easily.
- **Application Architecture:** The two tiers application architecture was selected in this study. The major drawback of the two-tier architecture as well as the parcel-based information system developed in this study is that, if one user wants to see any changes to the data which is performed by another user, he should restart the application. In other words, the updating available data or inserting new data can not be followed by all users, the restarting of the application is necessary. At this time, the municipality can tolerate this problem, because number of clients is very few. However, as the user of the system increases, this will be a real problem. In order to achieve this, instead of two tiers model, three tiers architecture can be used.
- **Integration of this system to other systems:** In Turkey, parcel related data are produced and used by several establishments. If the public

one is considered, Title Deed and Cadastral Works Directorship (TKGM) is the most important one. The city planning offices are private establishments which use the cadastral data. The integrated web-based system can be developed to collect these establishments on the same system so that data exchange between them will be easier. Moreover by adding web capabilities, the citizenship also uses the system to learn necessary information from internet.

- In this study the archive information of the municipal plan is stored in database. For example which plan is stored in the warehouse by which archive file number. In addition to this, the image of these plans can be inserted into the database and by designing a well indexing mechanisms, they can be reached much more easily.



## REFERENCES

- [1] Coad, P., and Yourdon, E., Object Oriented Analysis, Prentice-Hall Inc., U.S.A., 1991
- [2] Elmasri, R. and Navathe, S.B., Fundamentals of Database Systems, Addison-Wesley, USA, 2000
- [3] Enemark, S. and Sevetdal, H. "Cadastres, Land Information System and Planning- is decentralization a significant key to sustainable development?" UN-FIG Conference on the Land Tenure and Cadastral Infrastructures for Sustainable Development, 1999, Melbourne, Australia.
- [4] Farago, Z. "BIIR-Cadastral Information System for Budapest Land Office" FIG Commision 3 Annual Meeting and Seminer, 1999, Budapest, Hungary.
- [5] Hussey, P., "Designing Efficient Applications for Microsoft SQL Server", White Paper, Microsoft, 1998.
- [6] İmar Yasası 3194, 1985.
- [7] Kapan, G. İmar Mevzuatı, Ankara, 1998
- [8] Keleş, R., Kentleşme Politikası, İmge Kitabevi, Ankara, 2000
- [9] Kısa, A. Yerel Yönetimlerde Kent Bilgi Sistemleri Oluşturulması Bursa Yıldırım Belediyesi Bilgi Sisitemleri Çalışmaları, Coğrafi Bilgi Sistemleri Bilişim Günleri, Fatih Üniversitesi, 13-14 Kasım 2001.

- [10] Kocaman, S. Developing a Cadastral Information System with a Spatiotemporal Modeling Approach, MSc. Thesis, Middle East Technical University, Geodetic and Geographic Information Technologies Department, Ankara, 2001
- [11] Kösters, G., Pagel, B. and Six, H. GeoOOA: Object-Oriented Analysis for GIS Applications, University of Hagen, 1996
- [12] Laudon, K.C., and Laudon, J.P., Management Information Systems: New Approaches to Organization and Technology, Prentice Hall International Inc., U.S.A., 1998
- [13] Liou, j., Temporal Support for Land Information Systems in Object-Oriented Modeling, PhD Thesis, Royal Institute of Technology, School of Architecture, Surveying and Civil Engineering, Stocholm, 1999
- [14] Loerntzos, N.A, Sideridis, A.B., Yialouris, C.P., Kollias, V. An integrated spatiotemporal system, Computers and Electronics in Agriculture, Vol:22, 1999. pp: 233-242
- [15] Oh, K. Landscape Information System: A GIS approach to managing urban development, Landscape and Urban Planning, Vol:54, 2001, pp: 79-89
- [16] Rofail, A. and Martin, T., Building N-Tier Applications with COM and Visual Basic 6.0, John Wiley & Sons Inc., New York, 1999
- [17] Somerville, I. Software Engineering, Addison-Wesley, Massachusetts, 2001
- [18] Turkstra, J., Amemiya, N., Murgia, J. Local spatial data infrastructure, Trujillo-Peru, Habitat International, Vol:27, 2003, pp. 669-682
- [19] The Bogor Declaration, United Nations Interregional Meeting of Experts on the Cadastre, Bogor Indenosia, March 1996

- [20] Yalın, D. Türkiye Koşullarında Çok Amaçlı Kadastro Sistemi ve Sistemin Otomasyonu, PhD Thesis, İstanbul Teknik Üniversitesi, Jeodezi ve Fotogrametri Mühendisliği Bölümü, İstanbul, 1986.
- [21] Yomralıoğlu, T. and Mataracı, O. Tapu-Kadastro Bilgilerinin Kent Bilgi Sistemine Entegrasyonu: Arcview Örneği, Yerel Yönetimlerde Kent Bilgi Sistemi Uygulamaları Sempozyumu, 1999, KTÜ, Trabzon
- [22] Zhi, Y.C. and Wang, P. Designing a City-wide Information System in the People's Republic of China, International Journal of Information Management, Vol:15, No:5, 1995, pp. 377-388

Online References:

[1] Web Reference 1

<http://turk.internet.com/haber/yazigoster.php3?yaziid=5900> visited on 20.08.2003

[2] Web Reference 2: SQL Server web site

<http://www.microsoft.com/sql/evaluation/overview/default.asp> visited on 10.09.2003

[3] Web Reference 3: Visual Studio Web Site:

<http://msdn.microsoft.com/vstudio/> visited on 10.09.2003

[4] Web Reference 4: MapInfo Web Site : <http://www.mapinfo.com/> visited on 10.09.2003

[5] Web Reference 5: Kugo, T. ,Karube, T. and Koshizawa, A., The Development of Urban Information Systems in Japanese City Planning, <http://gis.esri.com/library/userconf/proc01/professional/papers/pap414/p414.htm> visited on May 2002

[6] Web Reference 6 Alfabim Web Site: [www.alfabim.com.tr](http://www.alfabim.com.tr) visited on 24.10.2003

[7] Web Reference 7 Erdoğan, T. and Şahin, N. Land Registry and Cadastre Activities in Turkey

<http://www.sli.unimelb.edu.au/fig7/Brighton98/Comm7Papers/TS65-Erdogan.html>, 27.11.2002

[8] Web Reference 8 TKGM Web Site:

<http://www.tkgm.gov.tr/tapunet/atkgmgovtr/takbis/takbis.htm> visited on 20.12.2003

## APPENDIX A

### DATA TAKEN FROM MUNICIPALITY

Ecrimisil and Çiçekçiler Liste:

They represent the renting of the real property of the municipality and the data inside these files are given Table A-1.

S.NO	ADI/SOYADI	ADRESİ	ECRİMİSİL BAŞL.TARİHİ
CİNSİ	ECR. SÜR.	ECR.BAŞL.VE BİTİŞ TARİHİ	ECR.BEDELİ

Table A-1Ecrimisil and Çiçekçiler Liste

Hilal:

This file represents how expropriation procedure is going on for one parcel for Hilal region.

ADA-PARSEL	S.N	ADI SOYADI	OLUR TARİHİ	P.NO
SONUÇ	ADRESİ	SOKAK NUMARASI	TEBLİGAT TARİHİ	NİTELİK
KIYMET TAKDİRİ	ENCÜMENE SUNUŞ TARİHİ VE SAYISI	ENCÜMEN KARAR TARİHİ VE SAYISI	TEBLİGAT TEBLİĞ TARİHİ	

Table A-2 Hilal

#### Dağıtım ve Tahsis:

This file explains the distribution and assignment details of a parcel as a result of the improvement procedure. It contains following information:

MAHALLE	ISLAH ADA / PARSEL	YÜZÖL ÇÜMÜ	MALİKLERİ	HİSSESİ m2
KULLANIM TÜRÜ	DEVİR HAK KINDA	GECEKONDU MALİKLERİ	TAHSİS EDİLEN m2	KORU NAN
KORUNAMAYAN	MÜRACAAT TARİH/SAYISI	KADASTRO GELDİSİ	KADASTRO MÜLKİYETİ	DOSYA NO
KALAN HİSSEMİZ	ADA / PARSEL	YÜZÖLÇÜMÜ	MALİKLERİ	HİSSESİ
PLAN NO	DOSYA NO	HİLAL.1 M. SOYAD	ADA PARSEL	HİSSE
PARSEL M2'Sİ	İPOTEK KALKAN	İPOTEK DEVAM	ENC. ÇIKAN	EVRAK DEVAM
HAKSAHİBİ DEĞİL	PLAN DIŞI	GELMEYEN	MÜRACAAT TARİH/SAYISI	YAPININ DURUMU
YÖTB	İLK TAKSİT TARİHİ			

Table A-3 Dağıtım ve Tahsis

#### İpotek Formu:

This file represents the debt of a person to the municipality because of the improvement procedure. It contains following data

TAHSİS SAHİBİNİN	
ADI VE SOYADI	
BABA ADI	
DOĞUM YERİ VE YILI	
İKAMETGAH ADRESİ	

Table A-4 İpotek Formu: Tahsis Sahibi Bilgileri

TAHSİS EDİLEN TAŞINMAZIN	
MEVKİİ	
ADA VE PARSEL NO	
ARSANIN YÜZÖLÇÜMÜ	
TAHSİS EDİLEN MİKTAR	
M2 BİRİM FİYATI	
TOPLAM BEDELİ	
ENKAZ BEDELİ	

Table A-5 İpotek Formu: Tahsis Edilen Taşınmazın Bilgileri

TAHSİS EDİLEN TAŞINMAZIN ÖDEME ŞEKLİ	
PEŞİN FİYATI	
TAKSİT 1/12	
MAKBUZ TARİH - NO:	
MÜTEBAKİ MİKTAR	

Table A-6 İpotek Formu: Ödeme Bilgileri

TAKSİTLE İLGİLİ İŞLEMLER				
	ÖDEME TARİHİ	MİKTAR	ÖDEDİĞİ TARİH	FİŞ NO
1.TAKSİT				
2.TAKSİT				
3.TAKSİT				
4.TAKSİT				
5.TAKSİT				
6.TAKSİT				
7.TAKSİT				
8.TAKSİT				
9.TAKSİT				
10.TAKSİT				
11.TAKSİT				

Table A-7 İpotek Formu: Taksit Bilgileri

ParalarıÖdenen:

This file represents the money given by the municipality to the person whose property is destructed and has following data:

Mahalle	Adalar Arası	Adı-Soyadı	M	Sokak
KapıNo	Enc. Karar Tarih ve No	Yıkım Tarihi	Enkaz Bedeli	Mahkeme Kar. Bed.
Tezyidi Ağaç bedeli	Faiz	Toplam		

Table A-8 Paraları Ödenenler



## APPENDIX B

### DATA DEFINITION LANGUAGES

```
CREATE TABLE [BUILDING] (  
    [Building_ID] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS  
NOT NULL ,  
    [Story_Number] [int] NULL ,  
    [Appartment_Number] [int] NULL ,  
    [Block_Number] [int] NULL ,  
    [Usage] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS  
NULL ,  
    [Status] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS  
NULL ,  
    [Province] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS  
NULL ,  
    [SubProvince] [char] (50) COLLATE  
SQL_Latin1_General_CP1_CI_AS NULL ,  
    [District] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS  
NULL ,  
    [Street] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS  
NULL ,  
    [DoorNo] [int] NULL ,  
    [Ecrimisil_ID] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS  
NULL ,  
    [Expropriation_FileNo] [char] (50) COLLATE  
SQL_Latin1_General_CP1_CI_AS NULL ,  
    CONSTRAINT [PK_BUILDING] PRIMARY KEY CLUSTERED  
    (  
        [Building_ID]  
    ) ON [PRIMARY] ,  
    CONSTRAINT [FK_BUILDING_ECRIMISIL] FOREIGN KEY  
    (  

```

```

        [Ecrimisil_ID]
    ) REFERENCES [ECRIMISIL] (
        [Ecrimisil_ID]
    ) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT [FK_BUILDING_EXPPAYEMENT_BUILDING]
FOREIGN KEY
    (
        [Expropriation_FileNo],
        [Building_ID]
    ) REFERENCES [EXPPAYEMENT_BUILDING] (
        [ExpropriationFileNo],
        [BuildingID]
    ) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT [FK_BUILDING_EXPROPRIATION] FOREIGN KEY
    (
        [Expropriation_FileNo]
    ) REFERENCES [EXPROPRIATION] (
        [FileNo]
    ) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

CREATE TABLE [BUILDING_CADASTRAL_PARCEL] (
    [Building_ID] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [Cadastral_Parcel_Block] [int] NOT NULL ,
    [Cadastral_Parcel] [int] NOT NULL ,
    CONSTRAINT [PK_BUILDING_CADASTRAL_PARCEL] PRIMARY
KEY CLUSTERED
    (
        [Building_ID],
        [Cadastral_Parcel_Block],
        [Cadastral_Parcel]
    ) ON [PRIMARY] ,
    CONSTRAINT [FK_BUILDING_CADASTRAL_PARCEL_BUILDING]
FOREIGN KEY
    (
        [Building_ID]
    ) REFERENCES [BUILDING] (
        [Building_ID]
    ) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT
[FK_BUILDING_CADASTRAL_PARCEL_CADASTRAL_PARCEL] FOREIGN
KEY
    (
        [Cadastral_Parcel],
        [Cadastral_Parcel_Block]
    )

```

```

) REFERENCES [CADASTRAL_PARCEL] (
    [Cadastral_Parcel_Number],
    [Cadastral_Parcel_Block_Number]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [BUILDING_JURIDICALPERSON] (
    [Building_ID] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [Tax_No] [int] NOT NULL ,
    [Tax_Office] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    CONSTRAINT [PK_BUILDING_JURIDICALPERSON] PRIMARY KEY
CLUSTERED
(
    [Building_ID],
    [Tax_No],
    [Tax_Office]
) ON [PRIMARY] ,
    CONSTRAINT [FK_BUILDING_JURIDICALPERSON_BUILDING]
FOREIGN KEY
(
    [Building_ID]
) REFERENCES [BUILDING] (
    [Building_ID]
) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT
[FK_BUILDING_JURIDICALPERSON_JURIDICAL_PERSON] FOREIGN
KEY
(
    [Tax_No],
    [Tax_Office]
) REFERENCES [JURIDICAL_PERSON] (
    [TaxNo],
    [Tax_Office]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [BUILDING_MUNICIPAL_PARCEL] (
    [Building_ID] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [Municipal_Parcel_Block_Number] [int] NOT NULL ,
    [Municipal_Parcel_Number] [int] NOT NULL ,
    CONSTRAINT [PK_BUILDING_MUNICIPAL_PARCEL] PRIMARY
KEY CLUSTERED

```

```

(
    [Building_ID],
    [Municipal_Parcel_Block_Number],

    [Municipal_Parcel_Number]
) ON [PRIMARY] ,
CONSTRAINT [FK_BUILDING_MUNICIPAL_PARCEL_BUILDING]
FOREIGN KEY
(
    [Building_ID]
) REFERENCES [BUILDING] (
    [Building_ID]
) ON DELETE CASCADE ON UPDATE CASCADE ,
CONSTRAINT
[FK_BUILDING_MUNICIPAL_PARCEL_MUNICIPAL_PARCEL] FOREIGN
KEY
(
    [Municipal_Parcel_Number],
    [Municipal_Parcel_Block_Number]
) REFERENCES [MUNICIPAL_PARCEL] (
    [Municipal_Parcel_Number],
    [Municipal_Parcel_Block_Number]
)
) ON [PRIMARY]

```

```

CREATE TABLE [BUILDING_REALPERSON] (
    [Identity_Number] [float] NOT NULL ,
    [Building_ID] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    CONSTRAINT [PK_BUILDING_REALPERSON] PRIMARY KEY
CLUSTERED
(
    [Identity_Number],
    [Building_ID]
) ON [PRIMARY] ,
CONSTRAINT [FK_BUILDING_REALPERSON_BUILDING]
FOREIGN KEY
(
    [Building_ID]
) REFERENCES [BUILDING] (
    [Building_ID]
) ON DELETE CASCADE ON UPDATE CASCADE ,
CONSTRAINT [FK_BUILDING_REALPERSON_REAL_PERSON]
FOREIGN KEY
(
    [Identity_Number]
) REFERENCES [REAL_PERSON] (

```

```

        [Identity_Number]
    ) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [CADASTRAL_PARCEL] (
    [Cadastral_Parcel_Number] [int] NOT NULL ,
    [Cadastral_Parcel_Block_Number] [int] NOT NULL ,
    [Area] [float] NULL ,
    [Value] [float] NULL ,
    [Landuse] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [Street] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [PostalCode] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [Expropriation_Notification] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Municipal_PlanNo] [char] (25) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    CONSTRAINT [PK_CADASTRAL_PARCEL] PRIMARY KEY
CLUSTERED
    (
        [Cadastral_Parcel_Number],
        [Cadastral_Parcel_Block_Number]
    ) ON [PRIMARY] ,
    CONSTRAINT
[FK_CADASTRAL_PARCEL_CADASTRAL_PARCEL_BLOCK] FOREIGN
KEY
    (
        [Cadastral_Parcel_Block_Number]
    ) REFERENCES [CADASTRAL_PARCEL_BLOCK] (
        [Cadastral_Parcel_Block_Number]
    ) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT
[FK_CADASTRAL_PARCEL_CADASTRAL_PARCEL_HISTORY] FOREIGN
KEY
    (
        [Cadastral_Parcel_Block_Number],
        [Cadastral_Parcel_Number]
    ) REFERENCES [CADASTRAL_PARCEL_HISTORY] (
        [Parcel_Block_Number],
        [Parcel_Number]
    ) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT [FK_CADASTRAL_PARCEL_LOCATION] FOREIGN
KEY
    (
        [PostalCode]
    )

```

```

    ) REFERENCES [LOCATION] (
        [Postal_Code]
    ) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT [FK_CADASTRAL_PARCEL_MUNICIPAL_PLAN]
FOREIGN KEY
(
    [Municipal_PlanNo]
) REFERENCES [MUNICIPAL_PLAN] (
    [PlanNo]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [CADASTRAL_PARCEL_BLOCK] (
    [Cadastral_Parcel_Block_Number] [int] NOT NULL ,
    [Area] [float] NULL ,
    [Street] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [Postal_Code] [char] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    CONSTRAINT [PK_CADASTRAL_PARCEL_BLOCK] PRIMARY KEY
CLUSTERED
(
    [Cadastral_Parcel_Block_Number]
) ON [PRIMARY] ,
    CONSTRAINT [FK_CADASTRAL_PARCEL_BLOCK_LOCATION]
FOREIGN KEY
(
    [Postal_Code]
) REFERENCES [LOCATION] (
    [Postal_Code]
)
) ON [PRIMARY]

```

```

CREATE TABLE [CADASTRAL_PARCEL_BLOCK_HISTORY] (
    [Parcel_Block_Number] [int] NOT NULL ,
    [Start_Date] [datetime] NULL ,
    [End_Date] [datetime] NULL ,
    [Previous_Parcel_Block_Number] [int] NULL ,
    CONSTRAINT [PK_CADASTRAL_PARCEL_BLOCK_HISTORY]
PRIMARY KEY CLUSTERED
(
    [Parcel_Block_Number]
) ON [PRIMARY] ,
    CONSTRAINT
[FK_CADASTRAL_PARCEL_BLOCK_HISTORY_CADASTRAL_PARCEL_B
LOCK] FOREIGN KEY

```

```

(
    [Parcel_Block_Number]
) REFERENCES [CADASTRAL_PARCEL_BLOCK] (
    [Cadastral_Parcel_Block_Number]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [CADASTRAL_PARCEL_HISTORY] (
    [Parcel_Block_Number] [int] NOT NULL ,
    [Parcel_Number] [int] NOT NULL ,
    [Start_Date] [datetime] NULL ,
    [End_Date] [datetime] NULL ,
    [Previous_Parcel_Block_Number] [int] NULL ,
    [Previous_Parcel_Number] [int] NULL ,
    CONSTRAINT [PK_CADASTRAL_PARCEL_HISTORY] PRIMARY
KEY CLUSTERED
(
    [Parcel_Block_Number],
    [Parcel_Number]
) ON [PRIMARY]
) ON [PRIMARY]

```

```

CREATE TABLE [CADASTRAL_PARCEL_JURIDICAL_PERSON] (
    [Parcel_Number] [int] NOT NULL ,
    [Parcel_Block_Number] [int] NOT NULL ,
    [Tax_No] [int] NOT NULL ,
    [Tax_Office] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    CONSTRAINT [PK_CADASTRAL_PARCEL_JURIDICAL_PERSON]
PRIMARY KEY CLUSTERED
(
    [Parcel_Number],
    [Parcel_Block_Number],
    [Tax_No],
    [Tax_Office]
) ON [PRIMARY] ,
    CONSTRAINT
[FK_CADASTRAL_PARCEL_JURIDICAL_PERSON_CADASTRAL_PARCE
L] FOREIGN KEY
(
    [Parcel_Number],
    [Parcel_Block_Number]
) REFERENCES [CADASTRAL_PARCEL] (
    [Cadastral_Parcel_Number],
    [Cadastral_Parcel_Block_Number]
) ON DELETE CASCADE ON UPDATE CASCADE ,

```

```

CONSTRAINT
[FK_CADASTRAL_PARCEL_JURIDICAL_PERSON_JURIDICAL_PERSON]
FOREIGN KEY
(
    [Tax_No],
    [Tax_Office]
) REFERENCES [JURIDICAL_PERSON] (
    [TaxNo],
    [Tax_Office]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [CADASTRAL_PARCEL_REAL_PERSON] (
    [Parcel_Number] [int] NOT NULL ,
    [Parcel_Block_Number] [int] NOT NULL ,
    [Identity_Number] [float] NOT NULL ,
    CONSTRAINT [PK_CADASTRAL_PARCEL_REAL_PERSON]
PRIMARY KEY CLUSTERED
(
    [Parcel_Number],
    [Parcel_Block_Number],
    [Identity_Number]
) ON [PRIMARY] ,
CONSTRAINT
[FK_CADASTRAL_PARCEL_REAL_PERSON_CADASTRAL_PARCEL]
FOREIGN KEY
(
    [Parcel_Number],
    [Parcel_Block_Number]
) REFERENCES [CADASTRAL_PARCEL] (
    [Cadastral_Parcel_Number],
    [Cadastral_Parcel_Block_Number]
) ON DELETE CASCADE ON UPDATE CASCADE ,
CONSTRAINT
[FK_CADASTRAL_PARCEL_REAL_PERSON_REAL_PERSON] FOREIGN
KEY
(
    [Identity_Number]
) REFERENCES [REAL_PERSON] (
    [Identity_Number]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [CONTACT_INFORMATION] (
    [Identity_Number] [float] NOT NULL ,
    [Province] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS

```



```

NULL ,
    [SubProvince] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [District] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [Street] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [DoorNo] [int] NULL ,

    CONSTRAINT [PK_CONTACT] PRIMARY KEY CLUSTERED
    (
        [Identity_Number]
    ) ON [PRIMARY] ,
    CONSTRAINT [FK_CONTACT_INFORMATION_REAL_PERSON]
FOREIGN KEY
    (
        [Identity_Number]
    ) REFERENCES [REAL_PERSON] (
        [Identity_Number]
    ) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

CREATE TABLE [DISTRIBUTION_AND_ASSIGNMENT] (
    [Municipal_ParcelBlock_Number] [int] NOT NULL ,
    [Municipal_Parcel_Number] [int] NOT NULL ,
    [File_No] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [PlanNo] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [YOTB] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    CONSTRAINT [PK_DISTRIBUTION_AND_ASSIGNMENT] PRIMARY
KEY CLUSTERED
    (
        [Municipal_ParcelBlock_Number],
        [Municipal_Parcel_Number],
        [File_No]
    ) ON [PRIMARY] ,
    CONSTRAINT
[FK_DISTRIBUTION_AND_ASSIGNMENT_DISTRIBUTION_ASIGNEMENT
_PHASE] FOREIGN KEY
    (
        [Municipal_ParcelBlock_Number],
        [Municipal_Parcel_Number],
        [File_No]
    ) REFERENCES [DISTRIBUTION_ASIGNEMENT_PHASE] (
        [Municipal_ParcelBlock_Number],

```

```

        [Municipal_Parcel_Number],
        [File_No]
    ) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT
[FK_DISTRIBUTION_AND_ASSIGNMENT_MUNICIPAL_PARCEL]
FOREIGN KEY
(
    [Municipal_Parcel_Number],
    [Municipal_ParcelBlock_Number]

) REFERENCES [MUNICIPAL_PARCEL] (
    [Municipal_Parcel_Number],
    [Municipal_Parcel_Block_Number]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [DISTRIBUTION_ASIGNEMENT_PHASE] (
    [Municipal_ParcelBlock_Number] [int] NOT NULL ,
    [Municipal_Parcel_Number] [int] NOT NULL ,
    [File_No] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [Plan_Situation] [char] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Rights] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [DocumentCompleteness] [char] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [CouncilCompleteness] [char] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Apropriation] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [MortgageSituation] [char] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    CONSTRAINT [PK_ASIGNEMENT_DISTRIBUTION_PHASE]
PRIMARY KEY CLUSTERED
(
    [Municipal_ParcelBlock_Number],
    [Municipal_Parcel_Number],
    [File_No]
) ON [PRIMARY]
) ON [PRIMARY]

```

```

CREATE TABLE [ECRIMISIL] (
    [Ecrimisil_ID] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [Decision_Date] [datetime] NULL ,

```

```

[Type] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
[Start_Date] [datetime] NULL ,
[End_Date] [datetime] NULL ,
[Value] [float] NULL ,
CONSTRAINT [PK_ECRIMISIL] PRIMARY KEY CLUSTERED
(
    [Ecrimisil_ID]
) ON [PRIMARY]
) ON [PRIMARY]

```

```

CREATE TABLE [ECRIMISIL_JURIDICALPERSON] (
    [Tax_No] [int] NOT NULL ,
    [Tax_Office] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [Ecrimisil_ID] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    CONSTRAINT [PK_ECRIMISIL_JURIDICALPERSON] PRIMARY
KEY CLUSTERED
(
    [Tax_No],
    [Tax_Office],
    [Ecrimisil_ID]
) ON [PRIMARY] ,
    CONSTRAINT [FK_ECRIMISIL_JURIDICALPERSON_ECRIMISIL]
FOREIGN KEY
(
    [Ecrimisil_ID]
) REFERENCES [ECRIMISIL] (
    [Ecrimisil_ID]
) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT
[FK_ECRIMISIL_JURIDICALPERSON_JURIDICAL_PERSON] FOREIGN
KEY
(
    [Tax_No],
    [Tax_Office]
) REFERENCES [JURIDICAL_PERSON] (
    [TaxNo],
    [Tax_Office]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [ECRIMISIL_REALPERSON] (
    [Identity_Number] [float] NOT NULL ,
    [Ecrimisil_ID] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,

```

```

        CONSTRAINT [PK_ECRIMISIL_REALPERSON] PRIMARY KEY
CLUSTERED
(
    [Identity_Number],
    [Ecrimisil_ID]
) ON [PRIMARY] ,
    CONSTRAINT [FK_ECRIMISIL_REALPERSON_ECRIMISIL]
FOREIGN KEY
(
    [Ecrimisil_ID]
) REFERENCES [ECRIMISIL] (
    [Ecrimisil_ID]
) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT [FK_ECRIMISIL_REALPERSON_REAL_PERSON]
FOREIGN KEY
(
    [Identity_Number]
) REFERENCES [REAL_PERSON] (
    [Identity_Number]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [EXPPAYEMENT_BUILDING] (
    [ExpropriationFileNo] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [BuildingID] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [WreckValue] [float] NULL ,
    [ValuebyCourt] [float] NULL ,
    [TreeValue] [float] NULL ,
    [Interest] [float] NULL ,
    CONSTRAINT [PK_EXPPAYEMENT_BUILDING] PRIMARY KEY
CLUSTERED
(
    [ExpropriationFileNo],
    [BuildingID]
) ON [PRIMARY]
) ON [PRIMARY]

```

```

CREATE TABLE [EXPROPRIATION] (
    [FileNo] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS NOT
NULL ,
    [OK_Date] [datetime] NULL ,
    [Map_Number] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,

```

```

[State] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
[Reporting_Date] [datetime] NULL ,
[Presentation_Date_to_Council] [datetime] NULL ,
[Council_Decision_Date] [datetime] NULL ,
[Council_Decision_Number] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
[Notification_Date_of_Reports] [datetime] NULL ,
CONSTRAINT [PK_EXPROPRIATION] PRIMARY KEY CLUSTERED
(
[FileNo]
) ON [PRIMARY]
) ON [PRIMARY]

```

```

4
CREATE TABLE [INSTELEMENT] (
[Municipal_ParcelBlock_Number] [int] NOT NULL ,
[Municipal_Parcel_Number] [int] NOT NULL ,
[Number_of_Instelement] [int] NOT NULL ,
[Condition] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
[DueDate] [datetime] NULL ,
[PaymentDate] [datetime] NULL ,
CONSTRAINT [PK_INSTELEMENT] PRIMARY KEY CLUSTERED
(
[Municipal_ParcelBlock_Number],
[Municipal_Parcel_Number],
[Number_of_Instelement]
) ON [PRIMARY] ,
CONSTRAINT [FK_INSTELEMENT_MORTGAGE] FOREIGN KEY
(
[Municipal_ParcelBlock_Number],
[Municipal_Parcel_Number]
) REFERENCES [MORTGAGE] (
[Municipal_Parcel_Block_Number],
[Municipal_Parcel_Number]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [JURIDICAL_PERSON] (
[TaxNo] [int] NOT NULL ,
[Tax_Office] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
[Name] [char] (100) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
[Type] [char] (100) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,

```

```

        [Start_Date] [datetime] NULL ,
        [End_Date] [datetime] NULL ,
        CONSTRAINT [PK_JURIDICAL_PERSON] PRIMARY KEY
CLUSTERED
    (
        [TaxNo],
        [Tax_Office]
    ) ON [PRIMARY]
) ON [PRIMARY]

```

```

CREATE TABLE [LOCATION] (
    [Postal_Code] [char] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NOT NULL ,
    [Province] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [SubProvince] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [District] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    CONSTRAINT [PK_LOCATION] PRIMARY KEY CLUSTERED
    (
        [Postal_Code]
    ) ON [PRIMARY]
) ON [PRIMARY]

```

```

CREATE TABLE [MORTGAGE] (
    [Municipal_Parcel_Block_Number] [int] NOT NULL ,
    [Municipal_Parcel_Number] [int] NOT NULL ,
    [Unit_Price] [float] NULL ,
    [Total_Price] [float] NULL ,
    [Way_of_Payment] [char] (25) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    CONSTRAINT [PK_MORTGAGE] PRIMARY KEY CLUSTERED
    (
        [Municipal_Parcel_Block_Number],
        [Municipal_Parcel_Number]
    ) ON [PRIMARY] ,
    CONSTRAINT [FK_MORTGAGE_MUNICIPAL_PARCEL] FOREIGN
KEY
    (
        [Municipal_Parcel_Number],
        [Municipal_Parcel_Block_Number]
    ) REFERENCES [MUNICIPAL_PARCEL] (
        [Municipal_Parcel_Number],
        [Municipal_Parcel_Block_Number]
    )

```

```

) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [MUNICIPAL_PARCEL] (
    [Municipal_Parcel_Number] [int] NOT NULL ,
    [Municipal_Parcel_Block_Number] [int] NOT NULL ,
    [Area] [float] NULL ,
    [Value] [float] NULL ,
    [Landuse] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [Street] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [PostalCode] [char] (10) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [Archive_FileNo] [char] (25) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Archive_MapNo] [char] (25) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [StudyNo] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [PlanFrom] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [Municipal_PlanNo] [char] (25) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Expropriation_FileNo] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    CONSTRAINT [PK_MUNICIPAL_PARCEL] PRIMARY KEY
CLUSTERED
(
    [Municipal_Parcel_Number],
    [Municipal_Parcel_Block_Number]
) ON [PRIMARY] ,
    CONSTRAINT [FK_MUNICIPAL_PARCEL_EXPROPRIATION]
FOREIGN KEY
(
    [Expropriation_FileNo]
) REFERENCES [EXPROPRIATION] (
    [FileNo]
) ON UPDATE CASCADE ,
    CONSTRAINT [FK_MUNICIPAL_PARCEL_LOCATION] FOREIGN
KEY
(
    [PostalCode]
) REFERENCES [LOCATION] (
    [Postal_Code]
) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT
[FK_MUNICIPAL_PARCEL_MUNICIPAL_PARCEL_BLOCK] FOREIGN KEY

```

```

(
    [Municipal_Parcel_Block_Number]
) REFERENCES [MUNICIPAL_PARCEL_BLOCK] (
    [Municipal_Parcel_Block_Number]
) ON DELETE CASCADE ON UPDATE CASCADE ,
CONSTRAINT [FK_MUNICIPAL_PARCEL_MUNICIPAL_PLAN]
FOREIGN KEY
(
    [Municipal_PlanNo]
) REFERENCES [MUNICIPAL_PLAN] (
    [PlanNo]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

CREATE TABLE [MUNICIPAL_PARCEL_BLOCK] (
    [Municipal_Parcel_Block_Number] [int] NOT NULL ,
    [Area] [float] NULL ,
    [Street] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [Postal_Code] [char] (10) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    CONSTRAINT [PK_MUNICIPAL_PARCEL_BLOCK] PRIMARY KEY
CLUSTERED
(
    [Municipal_Parcel_Block_Number]
) ON [PRIMARY] ,
    CONSTRAINT [FK_MUNICIPAL_PARCEL_BLOCK_LOCATION]
FOREIGN KEY
(
    [Postal_Code]
) REFERENCES [LOCATION] (
    [Postal_Code]
),
    CONSTRAINT
[FK_MUNICIPAL_PARCEL_BLOCK_MUNICIPAL_PARCEL_BLOCK_HISTO
RY] FOREIGN KEY
(
    [Municipal_Parcel_Block_Number]
) REFERENCES [MUNICIPAL_PARCEL_BLOCK_HISTORY] (
    [Parcel_Block_Number]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

CREATE TABLE [MUNICIPAL_PARCEL_BLOCK_HISTORY] (
    [Parcel_Block_Number] [int] NOT NULL ,
    [Start_Date] [datetime] NULL ,

```



```

        [End_Date] [datetime] NULL ,
        [Previous_Parcel_Block_Number] [int] NULL ,
        CONSTRAINT [PK_PARCEL_BLOCK_HISTORY] PRIMARY KEY
CLUSTERED
    (
        [Parcel_Block_Number]
    ) ON [PRIMARY]
) ON [PRIMARY]

```

```

CREATE TABLE [MUNICIPAL_PARCEL_HISTORY] (
    [Parcel_Block_Number] [int] NOT NULL ,
    [Parcel_Number] [int] NOT NULL ,
    [Start_Date] [datetime] NULL ,
    [End_Date] [datetime] NULL ,

    [Previous_Parcel_Block_Number] [int] NULL ,
    [Previous_Parcel_Number] [int] NULL ,
    CONSTRAINT [PK_PARCEL_HISTORY] PRIMARY KEY
CLUSTERED
    (
        [Parcel_Block_Number],
        [Parcel_Number]
    ) ON [PRIMARY] ,
    CONSTRAINT [FK_PARCEL_HISTORY_MUNICIPAL_PARCEL]
FOREIGN KEY
    (
        [Parcel_Number],
        [Parcel_Block_Number]
    ) REFERENCES [MUNICIPAL_PARCEL] (
        [Municipal_Parcel_Number],
        [Municipal_Parcel_Block_Number]
    ) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [MUNICIPAL_PARCEL_JURIDICAL_PERSON] (
    [Parcel_Number] [int] NOT NULL ,
    [Parcel_Block_Number] [int] NOT NULL ,
    [Tax_No] [int] NOT NULL ,
    [Tax_Office] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    CONSTRAINT [PK_PARCEL_JURIDICAL_PERSON] PRIMARY KEY
CLUSTERED
    (
        [Parcel_Number],
        [Parcel_Block_Number],
        [Tax_No],

```

```

        [Tax_Office]
    ) ON [PRIMARY] ,
    CONSTRAINT
[FK_MUNICIPAL_PARCEL_JURIDICAL_PERSON_MUNICIPAL_PARCEL]
FOREIGN KEY
(
    [Parcel_Number],
    [Parcel_Block_Number]
) REFERENCES [MUNICIPAL_PARCEL] (
    [Municipal_Parcel_Number],
    [Municipal_Parcel_Block_Number]
) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT
[FK_PARCEL_JURIDICAL_PERSON_JURIDICAL_PERSON] FOREIGN
KEY
(
    [Tax_No],
    [Tax_Office]
) REFERENCES [JURIDICAL_PERSON] (
    [TaxNo],
    [Tax_Office]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [MUNICIPAL_PARCEL_REAL_PERSON] (
    [Parcel_Number] [int] NOT NULL ,
    [Parcel_Block_Number] [int] NOT NULL ,
    [Identity_Number] [float] NOT NULL ,
    CONSTRAINT [PK_PARCEL_REAL_PERSON] PRIMARY KEY
CLUSTERED
(
    [Parcel_Number],
    [Parcel_Block_Number],
    [Identity_Number]
) ON [PRIMARY] ,
    CONSTRAINT
[FK_MUNICIPAL_PARCEL_REAL_PERSON_MUNICIPAL_PARCEL]
FOREIGN KEY
(
    [Parcel_Number],
    [Parcel_Block_Number]
) REFERENCES [MUNICIPAL_PARCEL] (
    [Municipal_Parcel_Number],
    [Municipal_Parcel_Block_Number]
) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT [FK_PARCEL_REAL_PERSON_REAL_PERSON]
FOREIGN KEY

```

```

(
    [Identity_Number]
) REFERENCES [REAL_PERSON] (
    [Identity_Number]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [MUNICIPAL_PLAN] (
    [PlanNo] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    [StudyNo] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [Included_Cadastral_Parcel] [char] (1000) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Excluded_Cadastral_Parcel] [char] (1000) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [DOP] [float] NULL ,
    [KOP] [float] NULL ,
    [Person_Organizing] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Person_Controlling] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Chief_of_Person_Controlling] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Person_Conforming] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Approvement_Date] [datetime] NULL ,
    CONSTRAINT [PK_MUNICIPAL_PLAN] PRIMARY KEY
CLUSTERED
(
    [PlanNo]
) ON [PRIMARY]
) ON [PRIMARY]

```

```

CREATE TABLE [MUNICIPAL_SITUATION] (
    [Municipal_Parcel_Block_Number] [int] NOT NULL ,
    [Municipal_Parcel_Number] [int] NOT NULL ,
    [Block_Number] [int] NULL ,
    [Appartment_Number] [int] NULL ,
    [Construction_Area] [float] NULL ,
    [Sketch_FileNo] [char] (25) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Sketch_Date] [datetime] NULL ,
    [Sketch_No] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [Kind_of_Construction] [char] (50) COLLATE

```

```

SQL_Latin1_General_CP1_CI_AS NULL ,
    [Number_of_Storey] [int] NULL ,
    [Building_Height] [float] NULL ,
    [From_Front_Road] [float] NULL ,
    [From_Neighbour] [float] NULL ,
    [KAKS] [float] NULL ,
    [TAKS] [float] NULL ,
    CONSTRAINT [PK_MUNICIPAL_SITUATION] PRIMARY KEY
CLUSTERED
(
    [Municipal_Parcel_Block_Number],
    [Municipal_Parcel_Number]
) ON [PRIMARY] ,
    CONSTRAINT [FK_MUNICIPAL_SITUATION_MUNICIPAL_PARCEL]
FOREIGN KEY
(
    [Municipal_Parcel_Number],
    [Municipal_Parcel_Block_Number]

) REFERENCES [MUNICIPAL_PARCEL] (
    [Municipal_Parcel_Number],
    [Municipal_Parcel_Block_Number]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [REAL_PERSON] (
    [Identity_Number] [float] NOT NULL ,
    CONSTRAINT [PK_REAL_PERSON] PRIMARY KEY CLUSTERED
(
    [Identity_Number]
) ON [PRIMARY]
) ON [PRIMARY]

```

```

CREATE TABLE [REALPERSON_DIST_ASSGN] (
    [Identity_Number] [float] NOT NULL ,
    [Municipal_Parcel_Block_Number] [int] NOT NULL ,
    [Municipal_Parcel_Number] [int] NOT NULL ,
    [File_No] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    CONSTRAINT [PK_REALPERSON_DIST_ASSGN] PRIMARY KEY
CLUSTERED
(
    [Identity_Number],
    [Municipal_Parcel_Block_Number],
    [Municipal_Parcel_Number],
    [File_No]

```

```

        ) ON [PRIMARY] ,
        CONSTRAINT
[FK_REALPERSON_DIST_ASSGN_DISTRIBUTION_AND_ASSIGNMENT]
FOREIGN KEY
(
        [Municipal_Parcel_Block_Number],
        [Municipal_Parcel_Number],
        [File_No]
) REFERENCES [DISTRIBUTION_AND_ASSIGNMENT] (
        [Municipal_ParcelBlock_Number],
        [Municipal_Parcel_Number],
        [File_No]
) ON DELETE CASCADE ON UPDATE CASCADE ,
CONSTRAINT [FK_REALPERSON_DIST_ASSGN_REAL_PERSON]
FOREIGN KEY
(
        [Identity_Number]
) REFERENCES [REAL_PERSON] (
        [Identity_Number]
) ON DELETE CASCADE ON UPDATE CASCADE ,
CONSTRAINT
[FK_REALPERSON_DIST_ASSGN_REALPERSON_DISTASSG_DETAIL]
FOREIGN KEY
(
        [File_No]
) REFERENCES [REALPERSON_DISTASSG_DETAIL] (
        [FileNo]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [REALPERSON_DISTASSG_DETAIL] (
        [FileNo] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS NOT
NULL ,
        [Share] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
        [ApplicationNo] [char] (25) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
        [Application_Date] [datetime] NULL ,
        [Protection] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
        CONSTRAINT [PK_REALPERSON_DISTASSG_DETAIL] PRIMARY
KEY CLUSTERED
(
        [FileNo]
) ON [PRIMARY]
) ON [PRIMARY]

```

```

CREATE TABLE [REALPERSON_JURIDICALPERSON] (
    [Identity_Number] [float] NOT NULL ,
    [Tax_No] [int] NOT NULL ,
    [Tax_Office] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
    CONSTRAINT [PK_REALPERSON_JURIDICALPERSON] PRIMARY
KEY CLUSTERED
    (
        [Identity_Number],
        [Tax_No],
        [Tax_Office]
    ) ON [PRIMARY] ,
    CONSTRAINT
[FK_REALPERSON_JURIDICALPERSON_JURIDICAL_PERSON]
FOREIGN KEY
    (
        [Tax_No],
        [Tax_Office]
    ) REFERENCES [JURIDICAL_PERSON] (
        [TaxNo],
        [Tax_Office]
    ) ON DELETE CASCADE ON UPDATE CASCADE ,
    CONSTRAINT
[FK_REALPERSON_JURIDICALPERSON_REAL_PERSON] FOREIGN KEY
    (
        [Identity_Number]
    ) REFERENCES [REAL_PERSON] (
        [Identity_Number]
    ) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [REGISTRATION_INFORMATION] (
    [Identity_Number] [float] NOT NULL ,
    [RI_Province] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [RI_Subprovince] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [RI_District_Village] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [RI_Page_Number] [char] (50) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [RI_Volume] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
    [RI_Family_Order] [char] (25) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
    [Name] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,

```

```

        [Surname] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
        [Father_Name] [char] (25) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
        [Mother_Name] [char] (25) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
        [Birth_Place] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
        [Birthday] [datetime] NULL ,
        [Death_Date] [datetime] NULL ,
        [Sex] [char] (5) COLLATE SQL_Latin1_General_CP1_CI_AS NULL ,
        [Citizenship] [char] (50) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
        CONSTRAINT [PK_REGISTRATION_INFORMATION] PRIMARY
KEY CLUSTERED
(
        [Identity_Number]
) ON [PRIMARY] ,
        CONSTRAINT
[FK_REGISTRATION_INFORMATION_REAL_PERSON] FOREIGN KEY
(
        [Identity_Number]
) REFERENCES [REAL_PERSON] (
        [Identity_Number]
) ON DELETE CASCADE ON UPDATE CASCADE
) ON [PRIMARY]

```

```

CREATE TABLE [USERS] (
        [UserName] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NOT NULL ,
        [Password] [char] (25) COLLATE SQL_Latin1_General_CP1_CI_AS
NULL ,
        [Department] [char] (150) COLLATE
SQL_Latin1_General_CP1_CI_AS NULL ,
        CONSTRAINT [PK_USER] PRIMARY KEY CLUSTERED
(
        [UserName]
) ON [PRIMARY]
) ON [PRIMARY]

```