A MOBILE WORKER MANAGEMENT SYSTEM FOR A LOGISTICS COMPANY

A Thesis

Presented to the Institute of Science and Engineering

of IŞIK University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

in

The Department of Computer Engineering

by
Uğur Güven

October 2005

Approval of the Institute of Science and Engineering

_____
(Prof. Hüsnü Ata Erbay)

Director

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

_____
(Prof. Selahattin Kuru)

Head of Department

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

_____
(Prof. Selahattin Kuru)

Supervisor

Examining Committee Members.

.....	_____
.....	_____
.....	_____
.....	_____
.....	_____

ABSTRACT

A MOBILE WORKER MANAGEMENT SYSTEM
FOR LOGISTICS COMPANY

GÜVEN, Uğur

Mobile worker management system is an application which aims to manage a company's site workers whose staff is on a specific area. The goal of the system is to assign daily tasks, receive completed tasks and in formations, inform workers about task changes, and send new tasks to the workers.

Mobile Workers send their completed/updated tasks, and GPS positions, which is received from Bluetooth GPS receiver, to Server-Side application, which is stored in company center by using Client-Side Mobile Application on PDA devices at predefined times over GPRS Connection.

Server-Side Application receives new/updated/completed tasks, makes assignments and changes and sends them to the mobile workers. The application makes all assignments and needed updates automatically by using application parameters.

Key words: Server-Side Application, Client-Side Application, Mobile Application, PDA, GPS, GPRS, Bluetooth

ÖZET

MOBİL ÇALIŞAN YÖNETİM SİSTEMİ

Güven, Uğur

Mobil Çalışan Yönetim Sistemi, belli bir sahada çalışanları bulunan bir işletmenin, saha çalışanlarını yönetmesini sağlayamayı hedefleyen bir uygulamadır. Uygulamanın amaçı çalışanlara günlük görevlerin dağıtmak, tamamlanan görevlere ilişkin bilgileri almak, görev değişikliklerini bildirmek, yeni gelen görevleri anlık olarak çalışanlara iletmektir.

Sistemde PDA cihazları kullanan mobil çalışanlar mobil uygulama ile belli aralıklarda, bluetooth GPS alıcılarından aldıkları kordinat bilgilerini, tamamlanan veya güncellenen görev bilgilerini GPRS vasıtası ile işletme merkezinde çalışmakta olan Sunucu uygulamasına aktarır.

Sunucu uygulaması, yeni gelen, değiştirilen ve tamamlanan görev bilgilerini kabul eder ve gerekli atamaları, değişklikleri yaparak mobil kullanıcılara değişiklikleri bildirir. Sunucu uygulaması bütün görev atamalarını, değişiklikleri kendisine bildirilen parametreler doğrultusunda otomatik olarak yapar.

Anahtar Kelimeler: Sunucu Uygulaması, Mobil Uygulama, PDA, GPS, GPRS, Bluetooth

ACKNOWLEDGEMENTS

Firstly, I would like to acknowledge my supervisor Prof. Dr. Selahattin Kuru for his endless suggestions, interest and helps. Many thanks to my dear friends, Zeki Özdemir, Ekrem Çağlar, Tuncay Çoruh, Özkan Hüray for their endless helps and supports.

TABLE OF CONTENTS

PAGE

ABSTRACT ii

ÖZET iii

ACKNOWLEDGEMENTS iv

TABLE OF CONTENTS v

LIST OF FIGURES vii

LIST OF TABLES viii

1. INTRODUCTION 1

2. COMPUTER-SUPPORTED COLLABORATIVE WORK, LOGISTICS AND MOBILE WORKER MANAGEMENT 3

2.1. Computer-Supported Collaborative Work 3

2.2. Worker Management in Logistics 4

2.3. Example of Existing Worker Management System 5

2.4. Requirements and Problems in Management for Logistics Companies 5

2.5. Main Characteristics of the Proposed system 6

3. TECHNOLOGIES USED 8

3.1. GPS (Global Positioning System) 8

3.2. Digital Maps 12

3.3. PDA 13

3.4. .NET 14

3.5. Microsoft SQL Server 2000 18

4. THE PROPOSED MOBILE WORKER MANAGEMENT SYSTEM 22

4.1. Architecture of The System 22

4.2. The Server-Side Application 24
4.3. The Client-Side Application 25
4.4. BackOffice Application 27
4.5. Database Design 29
5. AN EXAMPLE USE OF THE PROPOSED SYSTEM 33
6. CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK 43
REFERENCES 44
APPENDIX A - XML FILES' SCHEMAS AND SAMPLE FILES 46
APPENDIX B - CD INCLUDING DOCUMENTS AND APPLICATION SOURCE CODES 51

LIST OF FIGURES

Figure 3.1 Segments of GPS 9
Figure 3.2 Multipath Error 12
Figure 4.1 Architecture of the system 23
Figure 4.2 Server-Side applications jobs UML use case diagram 24
Figure 4.3 Server-Side application's Flow Chart 26
Figure 4.4 Client-Side application's Update Task Flow Chart 27
Figure 4.5 Client-Side application's Flow Charts 28
Figure 4.6 BackOffice application's user, Call Center Employee's use Case 29
Figure 4.7 BackOffice application's Flow Charts 30
Figure 4.8 Relations between tables 32
Figure 5.1 Logon Back Office Application 33
Figure 5.2 Adress Type Insert Form 34
Figure 5.3 Adress Type List 34
Figure 5.4 User List 35
Figure 5.5 User Form 35
Figure 5.6 Active Tasks Form 36
Figure 5.7 Active Task Form 36
Figure 5.8 Customer Form 37
Figure 5.9 Client Application Authentication 38
Figure 5.10 Client Application Task List 39
Figure 5.11 Task Status Update 40
Figure 5.12 Receiver Information 40
Figure 5.13 Client Application Parameter Request Form 41
Figure 5.14 Server-Side Applications 42

LIST OF TABLES

Table 3.1 System namespace major subsections…… 16
Table 4.1 Task Statuses Tables …….. 23
Table 4.2 Database Tables…….. 30

CHAPTER 1

INTRODUCTION

In today's world technology has been developing speedy and this time, companies which are following the developing technologies, and so they increase their competition power to other companies.

PDA (Personal digital assistant) devices are used for personal usage in earlier time, but today these devices' usage area are changed and it become more useful for the companies to the personal users. We can also see that some companies, which have mobile workers or hot selling market, are also using the PDA devices professionally in their application solution. At this point we saw that, the importance of the mobile devices is increased.

Our goal is develop an application system, named as Mobile Worker Management System (MWMS), for minimize personal errors, decrease the costs and accelerate works. This application will manage mobile workers in an area, who use client application of the system on PDA devices, and automate work flow in logistic market.

MWMS is formed three sub-applications, Client-Side application, Server-Side application, and BackOffice Application. Client-Side application is used on PDA device. It sends new, updated and completed tasks and GPS positions at predefined times. Server-Side application will communicate with Client-Side application and send new tasks and updated tasks to the Client application. Back office application is used for back office operations, like enter new tasks, take reports, etc.

The second chapter is about logistics, worker management in logistics, management problems, requirements and solutions of these problems in our proposed application.

The third chapter is basically explaining used technologies and development tools, which are GPS, PDA, Microsoft .Net and Microsoft SQL Server 2000.

The fourth chapter is deeply examines MWMS, its sub-applications, database design and systems software designs. Also communication between these applications and applications UML diagrams and flow charts are described and explained in this chapter.

In fifth Chapter, we realize an example scenario on proposed system and show results.

CHAPTER 2

COMPUTER –SUPPORTED COLLABORATIVE WORK, LOGISTICS AND MOBILE WORKER MANAGEMENT

Computer-Supported Collaborative Work (CSCW) is a generic term which combines the understanding of the way people work in groups with the enabling technologies of computer networking, and associated hardware, software, services and techniques[1].

Logistics' dictionary definition is that; the branch of military science having to do with procuring, maintaining, and transporting material, personnel, and facilities. But business objectives and activities differ from those of the military, so this definition does not capture the essence of business logistics management.

2.1. Computer-Supported Collaborative Work

In 1984, Greif and Cashman coined the term 'Computer-Supported Collaborative Work' at a workshop attended by individuals interested in using technology to support people in their work. Since then, CSCW has been interpreted and understood in a number of different ways. Some researchers use the term to express the idea of collaboration among a group of people using computers. CSCW is also referred to as 'software for groups of people' or 'groupware'. [1].

We accept Bannon & Schmidt's (1989) definition of CSCW: "an endeavor to understand the nature and characteristics of cooperative work with the objective of designing adequate computer-based technologies" (pp. 3-5). This definition of CSCW combines an understanding of how people work in groups and how

computer networking technologies can be designed to support activities. CSCW systems are collaborative environments that support dispersed working groups so as to improve quality and productivity [1].

2.2. Worker Management in Logistics

A representation of logistics is that the process of planning, implementing and controlling the efficient, cost-effective flow and purchasing, storage of raw materials, in-process inventory, finished goods and related information from point of origin to point of consumption for the purpose of conforming to customer requirements throughout the logistical channels that consist of suppliers, manufacturers, distributors, wholesalers, retailers, consumers [2].

The fundamental mission of the logistic is that to get right good or services and to provide the customer this good or services according the right time, desired conditions and the most efficient way of the wants of the customer and sender. Our interested area in this logistical supply chain is that to take the goods or services from the sender and to reach this good or services to customer.

The worker who work this part of the supply chain, is named mobile workers. The main purpose of these workers is that to get the goods and services in a right time, and to provide goods and services to customers according to their needs and requirements in the most efficient manner possible. Today's logistical chain, the communication with the mobile workers and officers obtained that with the radios and cellular phones. In this control system is always running time to time communication with the office workers and the mobile workers. When a task reached the office workers with a phone or on the internet, office workers, with the help of the radios, declare the task all the mobile workers to learn the nearest mobile worker to new task area. Then, again the new announcement from the radio, chosen worker announced and other workers continues to their work. At this point,

always mobile workers realize an important problem which who is the nearest the task or who get the task. This work flow is always not providing the efficiency for the customer because many logistical firms met many time errors and these errors always affect the customers.

As was previosly mentioned one of the main mision of the logistics is that realize the task at the right time but this communucation network does not provide this situation. To conclude that our interest area logistics have two parts which are the mobile workers and central office workers who make the work sharing between the mobile workers and get the task from the customer with a communucation network. To work of the communacition network efficiently provide the work flow and the reletionship between the customers better.

2.3. Example of Existing Worker Management System

Existing logistics companies don't exactly use worker management system as an application. They generally manage workers buy using known communication systems (phones, radios etc...). They give the works to the mobile workers from the central workers. There is no computer application for work sharing between the mobile workers.

2.4. Requirements and Problems in Management for Logistics Companies

Logistics companies want to manage their workers in an area efficiently. Today, they manage them by radio or cell phone. In this management style somebody in the center office must give information to the worker. In this communication style some problems have been occurred and these problems are generally about the communication network and the sharing of the works between the mobile workers. The fundamental problems about the communication network are that not certain control of the mobile workers, high cost of the communication,

taker and sender problems about the where is the package, and as a conclusion of these problems loss of the customers and loss of the market share.

- Not certain control of the mobile worker; central workers can not exactly know that where is the mobile workers right now and the truth of this information is only between the centre workers and mobile workers.
- High cost of the communication; in today's communication market the wage of the cellular phone is so high and this increase the cost of services and also the radios communications incomprehensible are not always exact.
- Taker and sender problems; are always about the where is the package and the control of their package when they want.
- Sharing of the task between the mobile workers; with the announcement of the task, in a same time two or more workers can near the same task and problem is occurred about task become whose.
- Conclusion of all above problems, happiness of the customers not exactly provided and this can cause the loss of customers and loss of the market share in the long term.

2.5. Main Characteristics of the Proposed system

Logistics companies try to make their work in an efficient way and meet above problems. The main purpose of the proposed application system is minimizing these problems, decrease the cost of work, and to increase the company's profit. To realize this efficient work flow diagram, the application system serves new solutions for the above problems.

- Controlling of the mobile workers; when the central workers want to learn a mobile worker, where he /she now, the system serves this

information with the GPS system, without any communication with the mobile worker.

- Cost of communication; communication cost will also decrease with using this application because when task reached the office, program will choose the nearest mobile worker to that task and give the task to worker by itself. Therefore, especially cellular phone and radios communication not used and also the number of central workers can go down because the giving of the task will realized from the application.
- Taker and sender problems; is also reach a solution because when they want they can learn the where is the package with its point wise place. In today's usage area this information giving customer only package last duration office but this system gives the place of the package time to time.
- Sharing of the work; as was mentioned, task are sent to mobile workers from the computer application and every worker has own place, and our proposed system will choose the nearest worker to that task and give this task to that worker. Other mobile workers do not need to learn every new task; they learn the new tasks which are interested to whom.
- Conclusion of these solutions for the above mentioned problems is that customer satisfaction will be achieved. This also provides the logistics firms a competition power with the other firms and also can give the greater market share to the logistical firms.

CHAPTER 3

TECHNOLOGIES USED

3.1. GPS (Global Positioning System)

GPS (Global Positioning System) is a satellite-based positioning and navigation system owned and operated by US Department of Defense. It was originally intended for military applications, but in the 1980s, the government made the system available for civilian use. GPS works in any weather conditions, anywhere in the world, 24 hours a day. There are no subscription fees or setup charges to use GPS [3].

How GPS works

GPS satellites circle the earth twice a day in a very precise orbit and transmit signal information to earth. GPS receivers take this information and use triangulation to calculate the user's exact location. Essentially, the GPS receiver compares the time a signal was transmitted by a satellite with the time it was received. The time difference tells the GPS receiver how far away the satellite is. Now, with distance measurements from a few more satellites, the receiver can determine the user's position and display it on the unit's electronic map.

A GPS receiver must be locked on to the signal of at least three satellites to calculate a 2D position (latitude and longitude) and track movement. With four or more satellites in view, the receiver can determine the user's 3D position (latitude, longitude and altitude). Once the user's position has been determined, the GPS unit

can calculate other information, such as speed, bearing, track, trip distance, distance to destination, sunrise and sunset time and more.

The Parts of GPS

GPS consists of three main segments as shown in Figure 3.1.

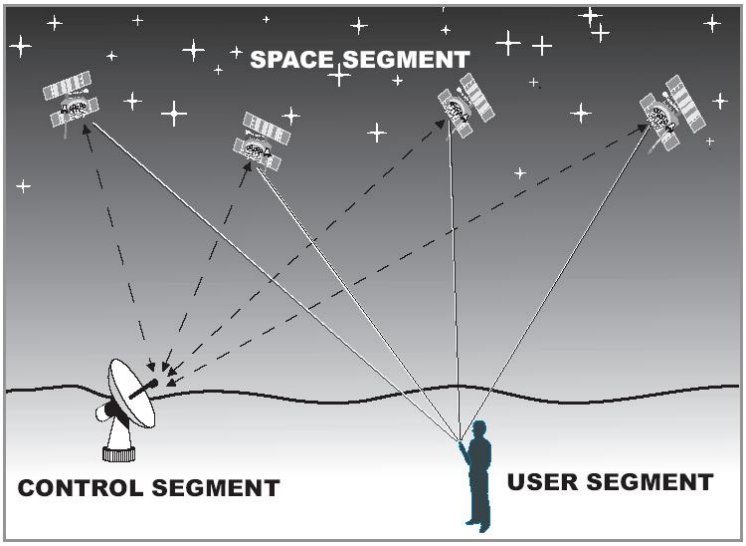

Figure 3.1 Segments of GPS

- **The Space Segment:** This part consists of 24 satellites, manufactured by Rockwell International, which are launched into space by rockets, from Cape Canaveral, Florida. They are about the size of a car, and weigh about 19,000lbs. Each satellite is in orbit above the earth at an altitude of 11,000 nautical miles (12,660 miles), and takes 12 hours to orbit one time. There are 6 orbital planes each having 4 satellites. The orbits are tilted to the equator of the earth by 55° so that there is coverage of the Polar Regions. The satellites continuously orient themselves to ensure that their solar panels stay pointed towards the sun, and their antennas point toward the earth. Each satellite carries 4 atomic clocks.

- **The Control Segment:** This part consists of 5 worldwide unmanned base-stations that monitor the satellites to track their exact position in space, and to make sure that they are operating correctly. The stations constantly monitor the orbits of the satellites and use very precise radar to check altitude, position and speed. Transmitted to the satellites are ephemeris constants and clock adjustments. The satellites in turn, use these updates in the signals that they send to GPS receivers
- **The User Segment:** This part consists of user receivers which are hand-held or, can be placed in a vehicle. All GPS receivers have an almanac programmed into their computer, which tells them where each satellite is at any given moment. The GPS receivers detect, decode and process the signals received from the satellites. The receiver is usually used in conjunction with computer software to output the information to the user in the form of a map. As the user does not have to communicate with the satellite there can be unlimited users at one time.

Sources of GPS signal errors

Factors that can degrade the GPS signal and thus affect accuracy include the following:

- **Ionosphere and troposphere delays —** The satellite signal slows as it passes through the atmosphere. The GPS system uses a built-in model that calculates an average amount of delay to partially correct for this type of error.
- **Signal multipath —** This occurs when the GPS signal is reflected off objects such as tall buildings or large rock surfaces before it reaches

the receiver. This increases the travel time of the signal, thereby causing errors.

- **Receiver clock errors —** A receiver's built-in clock is not as accurate as the atomic clocks onboard the GPS satellites. Therefore, it may have very slight timing errors.
- **Orbital errors —** Also known as ephemeris errors, these are inaccuracies of the satellite's reported location.
- **Number of satellites visible —** The more satellites a GPS receiver can "see," the better the accuracy. Buildings, terrain, electronic interference, or sometimes even dense foliage can block signal reception, causing position errors or possibly no position reading at all. GPS units typically will not work indoors, underwater or underground.
- **Satellite geometry/shading —** This refers to the relative position of the satellites at any given time. Ideal satellite geometry exists when the satellites are located at wide angles relative to each other. Poor geometry results when the satellites are located in a line or in a tight grouping.
- **Intentional degradation of the satellite signal —** Selective Availability (SA) is an intentional degradation of the signal once imposed by the U.S. Department of Defense. SA was intended to prevent military adversaries from using the highly accurate GPS signals. The government turned off SA in May 2000, which significantly improved the accuracy of civilian GPS receivers.

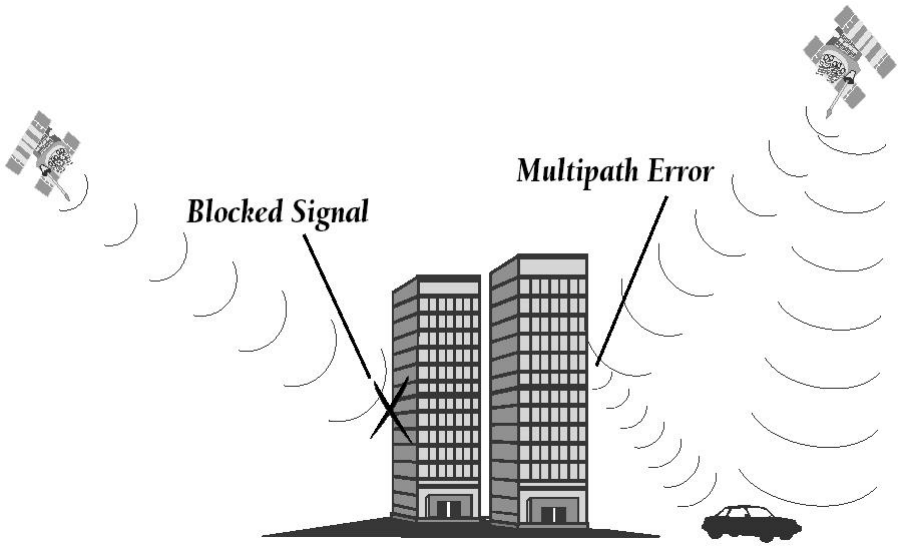

Figure 3.2 Multipath Error

3.2. Digital Maps

Digital map is a machine-readable representation of a geographic phenomenon stored for display or analysis by a digital computer; contrast with analog map [4].

Many different kinds of mapping programs are available; map programs are classified in two types: consumer programs and Geographic Information System (GIS) software.

Consumer programs

A consumer mapping program is software that displays street maps, topographic maps, marine charts, or aeronautical charts. Such mapping programs are easier to use (and much less expensive) than their professional counterparts, meeting most computer users' mapping needs [5].

GIS (Geographic Information System)

A Geographic Information System (GIS) is an information system that analyzes inputs, manipulates, outputs, retrieves, and stores spatial data. GIS is mostly used by governments; large corporations; and engineering and GIS consulting firms for land, natural resources, transportation, environmental, and urban planning and management [4].

Digital mapping software offers all sorts of enhancements over paper maps, including these capabilities:

- Finding street addresses quickly
- Interfacing with GPS receivers to see where you are or where you were
- Showing driving directions to just about anywhere
- Displaying terrain three-dimensionally
- Annotating maps with pop-up information
- Creating custom maps
- Printing a hard copy map

3.3. PDA

Personal digital assistant (PDA) is a handheld device that combines computing, telephone/fax, and networking features. A typical PDA can function as a cellular phone, fax sender, and personal organizer. Unlike portable computers, most PDAs began as pen-based, using a stylus rather than a keyboard for input. This means that they can incorporate handwriting recognition features. Some PDAs can also react to voice input by using voice recognition technologies. PDAs of today are available in either stylus or keyboard versions. PDAs are also called palmtops, handheld computers and pocket computers [5].

3.4. .NET

Microsoft explain .NET like this ".NET is the Microsoft Web services strategy to connect information, people, systems, and devices through software. Integrated across the Microsoft platform, .NET technology provides the ability to quickly build, deploy, manage, and use connected, security-enhanced solutions with Web services. .NET-connected solutions enable businesses to integrate their systems more rapidly and in a more agile manner and help them realize the promise of information anytime, anywhere, on any device." [6].

.NET is also referred to as the .NET Framework.

3.4.1. Microsoft .Net Framework

The .NET Framework is an enormous environment that represents much of the functionality contained in Visual Studio .NET. The base of the framework consists of the Common Language Runtime (CLR), the engine that drives the entire system. The CLR provides a standard system of data types and objects, and makes cross language and cross-platform development possible. The middle layer of the Framework includes next generation system services such as ADO.NET, ASP.NET, and XML. Unlike previous versions of these tools, the Framework manages these tools, making their availability universally consistent across languages. The top-most layer includes user and program interfaces. Windows Forms provides the interface to create standard Win32 applications. Web Forms provides a powerful server side visual development tool wrapped around ASP.NET. Web Services provides a mechanism for programs to communicate directly over the Internet. Also, a console interface allows creation of simple text-based applications [7].

CLR: The purpose of the CLR is to create an easier and faster development system with automatic handling of low-level details such as memory

management and process communication. Other important features for the CLR include a simpler and safer deployment environment and scalability. The CLR transparently supports any number of languages, not limited to those Microsoft develops. This integration enables many languages to take advantage of the built-in debugging system and memory management. The integration with the CLR required the VB language to change significantly, but gives it the ability to work with other languages much more effectively [6].

The .NET Framework provides an enormous class library that contains much of the functionality available to .NET languages. The .NET Class Framework libraries implement the functions, we can use in .NET enabled languages. The Class Framework consists of hundreds of classes and interfaces. It wraps class functionality around most of the Windows system libraries and provides extra functionality not available in standard Windows libraries.

The .NET class framework includes functions over a broad set of needs. These functions include:

- Data access and manipulation: data access through ADO.NET, robust XML support, and file and directory tools
- Standard language functions: math commands, arrays, collections, and strings
- Creation and management of threads: free-threading libraries and process management
- Interfaces to Windows functions: Windows Forms, Web Forms, Web Services, and console applications
- Management and enforcement of application security
- Application configuration

- Working with Windows Directory Services, event logs, message queues, and timers
- Support for a number of network protocols

Table 3.1 System namespace major subsections

Namespace	Description
`System.Collections`	Classes representing various sorts of collections.
`System.Data`	Classes that constitute the ADO.NET architecture.
`System.Diagnostics`	Classes that enable you to debug your application and to trace execution of your code.
`System.Drawing`	Provides access to the GDI+ basic graphics functionality.
`System.Drawing.Printing`	Provides printing functions and printer and page settings.
`System.IO`	Contains types for reading and writing data streams and files.
`System.Net`	Provides an interface to many common Web protocols.
`System.Resources`	Manage the creation and storage of various culture-specific resources in an application.
`System.Security`	Provides the underlying security structure of the CLR.
`System.Security. Cryptography`	Provides data encoding and decoding routines, and functions like hashing, random number generation, and message authentication.
`System.Text`	Contains classes representing ASCII, Unicode, other character encodings, and classes for conversion between formats.
`System.Threading`	Enables multi-threaded programming. Includes classes to manage individual threads, pools of threads, and synchronization issues.
`System.Web`	Supplies classes that enable browser and server communication for ASP.NET applications.
`System.Windows.Forms`	Provides classes to create complete Windows applications.
`System.Xml`	Provides standards-based support for processing XML.

3.4.2. Microsoft Visual Studio.NET

Visual Studio .NET is an environment for developing Windows and Web applications. Visual Basic .NET is just one of the language we can use to program our applications. Actually, Visual Studio .NET was designed to host any language, and many companies are working on languages that will be integrated in Visual Studio .NET. Some people will develop Windows applications in Visual Studio .NET with COBOL, or FORTRAN [8].

Visual Studio .NET is the environment that provides all the necessary tools for developing applications. The language is only one aspect of a Windows application. The visual interface of the application isn't tied to a specific language, and the same tools we'll use to develop our application's interface will also be used by all programmers, regardless of the language they'll use to code the application.

The tools we'll use to access databases are also independent of the language. Visual Studio provides tools that allow us to connect to a database inspect its objects, retrieve the information we're interested in, and even store it in objects that can be accessed from within any language.

There are many visual tools in the IDE, like the Menu Designer. This tool allows us to visually design menus and to set their names and basic properties (such as checking, enabling, or disabling certain options). Designing a menu doesn't involve any code, and it's carried out with point-and click operations. Of course, we will have to insert some code behind the commands of our menus, and (again) we can use any language to program them.

To simplify the process of application development, Visual Studio .NET provides an environment that's common to all languages, which is known as integrated development environment (IDE). The purpose of the IDE is to enable the

developer to do as much as possible with visual tools, before writing code. The IDE provides tools for designing, executing, and debugging your applications.

3.4.3. Visual Basic .NET

Visual Basic .NET is part of Microsoft Visual Studio .NET, the latest development environment from Microsoft. With Visual Basic .NET, or VB .NET for short. With VB.NET, Visual Basic received advanced language features on par with languages like C++, C#, and Java. Powerful new features give VB .NET the simplicity of its predecessors, but with the power of creating full-fledged applications that take advantage of the computing field's latest innovations [9].

3.5. Microsoft SQL Server 2000

SQL Server 2000 is a Relational Database Management System (RDBMS), which uses Transact-SQL to send requests between a client computer and a SQL Server 2000 computer. An RDBMS includes databases, the database engine, and the applications that are necessary to manage the data and the components of the RDBMS. The RDBMS organizes data into related rows and columns within the database [9].

The RDBMS is responsible for enforcing the database structure, including the following tasks:

- Maintaining the relationships among data in the database
- Ensuring that data is stored correctly and that the rules defining data relationships are not violated

- Recovering all data to a point of known consistency in case of system failures

The database component of SQL Server 2000 is a Structured Query Language (SQL)-compatible, scalable, relational database with integrated XML support for Internet applications. SQL Server 2000 builds upon the modern, extensible foundation of SQL Server 7.0.

SQL Server 2000 Tools

SQL Server 2000 includes many graphical and command-prompt utilities that help users, programmers, and administrators to perform a variety of tasks, including the following:

- Administering and configuring SQL Server
- Determining the catalog information in a copy of SQL Server
- Designing and testing queries for retrieving data
- Copying, importing, exporting, and transforming data
- Providing diagnostic information
- Starting and stopping SQL Server

- Command Prompt Tools
- User Interface Tools
    - SQL Server Enterprise Manager
    - SQL Server Agent
    - SQL Profiler
    - SQL Server Client Network Utility
    - SQL Server Network Utility
    - SQL Server Service Manager

- SQL Query Analyzer
- SQL Server 2000 Built-In Wizards

SQL Server Enterprise Manager

SQL Server Enterprise Manager is the primary administrative tool for SQL Server 2000 and provides a Microsoft Management Console (MMC)–compliant user interface that helps us to perform a variety of administrative tasks:

- Defining groups of servers running SQL Server
- Registering individual servers in a group
- Configuring all SQL Server options for each registered server
- Creating and administering all SQL Server databases, objects, logins, users, and permissions in each registered server
- Defining and executing all SQL Server administrative tasks on each registered server
- Designing and testing SQL statements, batches, and scripts interactively by invoking SQL Query Analyzer
- Invoking the various wizards defined for SQL Server

MMC is a tool that presents a common interface for managing different server applications in a Microsoft Windows network. Server applications include a component called a snap-in that presents MMC users with a user interface for managing the server application. SQL Server Enterprise Manager is the Microsoft SQL Server 2000 MMC snap-in.

SQL Query Analyzer

SQL Server 2000 SQL Query Analyzer is a graphical tool that helps us to perform a variety of tasks:

- Creating queries and other SQL scripts and executing them against SQL Server databases
- Creating commonly used database objects from predefined scripts
- Copying existing database objects
- Executing stored procedures without knowing the parameters
- Debugging stored procedures
- Debugging query performance problems
- Locating objects within databases or viewing and working with objects
- Inserting, updating, or deleting rows in a table
- Creating keyboard shortcuts for frequently used queries
- Adding frequently used commands to the Tools menu

SQL Query Analyzer can be start directly from the Start menu or from inside SQL Server Enterprise Manager.

CHAPTER 4

THE PROPOSED MOBILE WORKER MANAGEMENT SYSTEM

We call our proposed system as Mobile Worker Management System (MWMS). It is used to manage mobile workers of logistics companies and their tasks. It is developed in object oriented approach by using Visual Basic.Net programming language and Microsoft .NET environment, Visual Studio.NET ans Microsoft SQL Server 2000.

It is formed with 3 sub systems, Server-Side Application, Client-Side application, and BackOffice application. BackOffice application an Server-Side application are windows based applications. Client-Side application is a Mobile device application for Windows Mobile 2003.

4.1. Architecture of the System

MWMS is formed with 3 sub systems; Server-Side application is worked on application server, Client-Side application on PDA devices, and BackOffice application on office desktop computers. Applications data is stored on Microsoft SQL Server 2000. Architecture of the system is shown in Figure 4.1.

BackOffice application is used by office workers to create program users, declare program parameters and create new tasks, update and delete waiting tasks. Server-Side application is used to communicate with Client-Side applications. Communication between these two applications is started by Client-Side application by using GPRS connection type, port 9999. Applications communicate with each other by sending data via XML files.

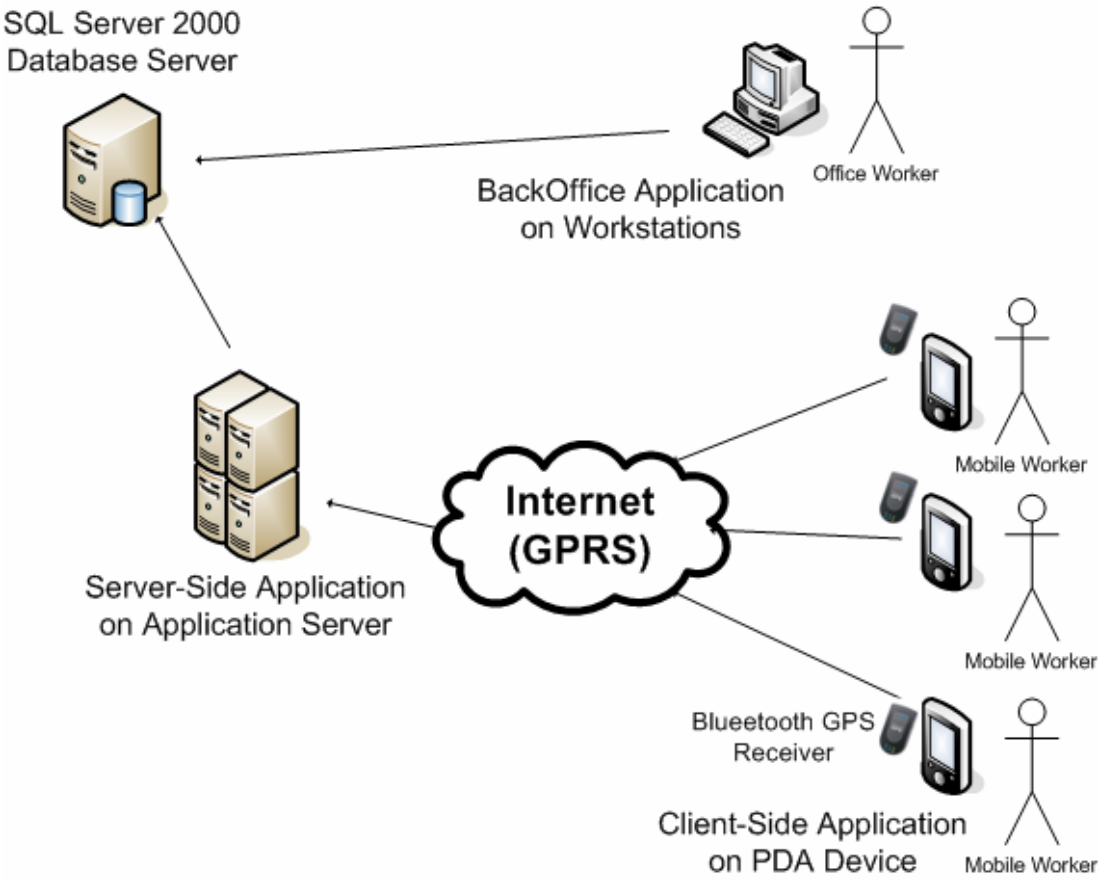

Figure 4.1 Architecture of the system

In these systems, a Task created for each job and managed by the system. Tasks are followed by Task status. System's task statuses are list in Table 4.1.

Table 4.1 Task Statuses Tables

Task Status ID	Description
0	New
1	Received
2	Office
3	Sent
4	Delivered
-1	Cancelled

4.2. The Server-Side Application

Server application works on server on 7/24. The application waits to Client applications requests to communication. It can be talk more than one Client application at a time. It creates a new threat for each client connection. It receives a connection and analyses received XML data. It parses xml tags and inner data of this XML tags.

It receives updated tasks, client's GPS positions, username and password for client-side application authentication and sends user's task list, requested parameter, new tasks and updated tasks. It's Use Case Diagram shown in Figure 4.2.

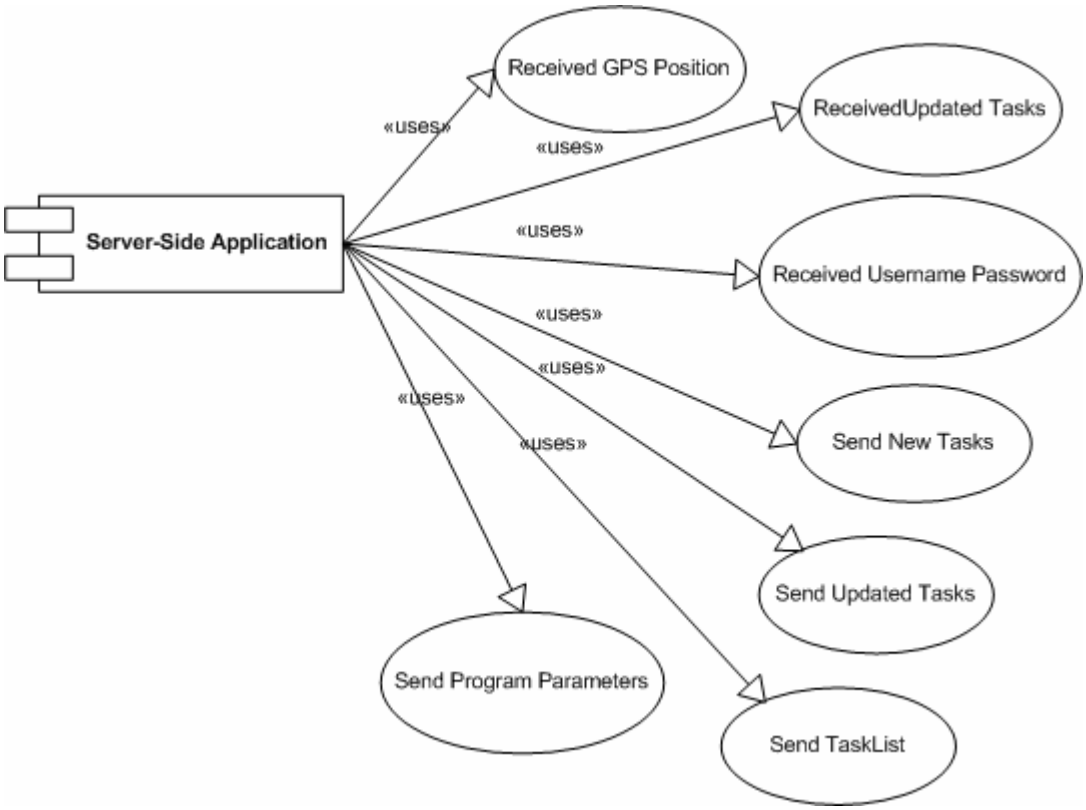

Figure 4.2 Server-Side applications jobs UML use case diagram

When the Server application is started, it starts to listening port 9999 for client connection. The client application can be request authentication, program parameters or task list via XML files. Client GPS positions and user identification number (userID) (except authentication request) always sent with these XML files. If received XML file has authentication request, firstly application finds userID. And then it writes received coordinates to database with userID. If client user have got new and updated tasks, requested parameters or authenticated userID, Server application sends these data to client and waits "OK" messages to close connection. Otherwise send "OK" message and close connections. The Server application sends return data via XML file. Server application's Flow Chart is in Figure 4.3.

4.3. The Client-Side Application

Client application is worked on PDA, that's operating system is Windows Mobile 2003. Client application needs .Net Compact Framework, GPRS and Bluetooth support on PDA device. IT communicates with Server Application by using GPRS connection on port 9999.

Client Application is used by mobile worker. Mobile User makes 3 things (authentication, update task, receives program parameters) in this program (Figure 4.5). It starts communication by sending XML document, which has got authentication request (username and password). If username or password is correct server sends user identification number (userID) and user task list as an XML file. After userID is received, application connects server by using this userID. Client application sends worker coordinates (latitude, longitude), which is received from Bluetooth GPS receiver by using serial port at predefined times (1 minute). Received task list record PDA's memory as an XML file.

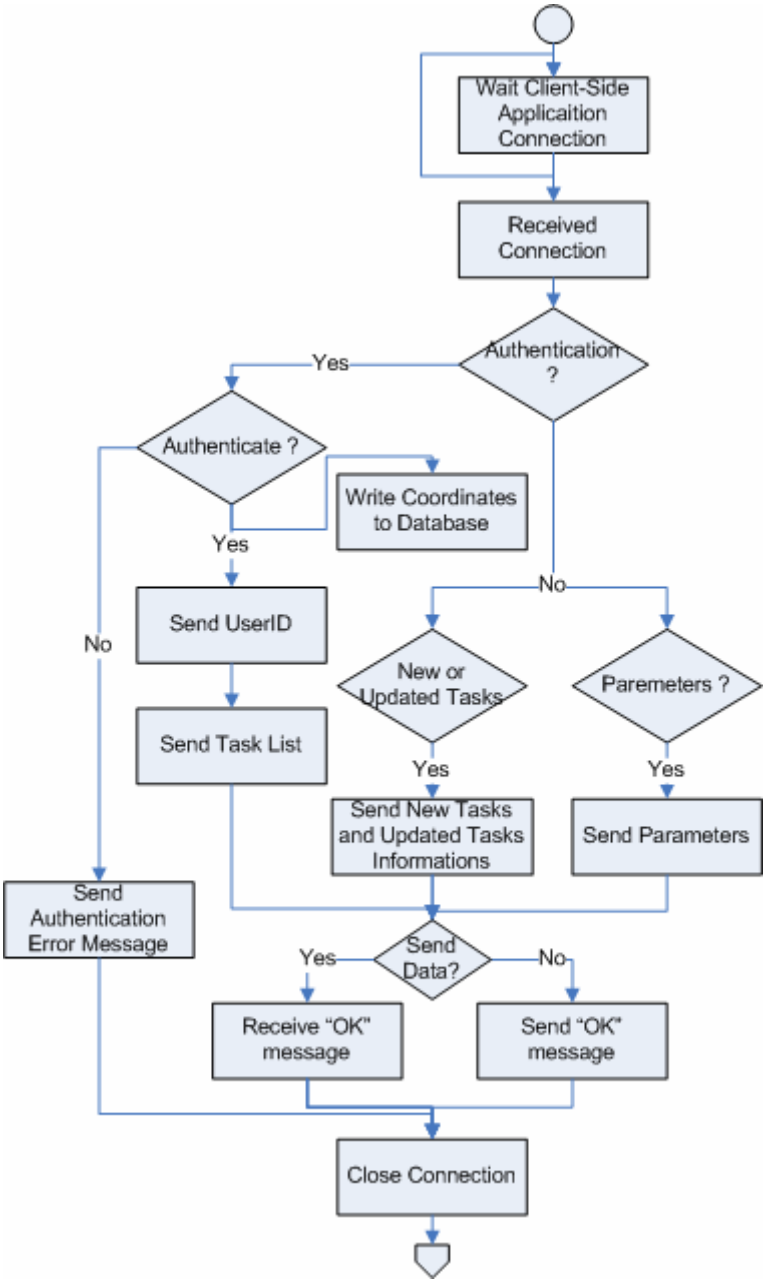

Figure 4.3 Server-Side Application's Flow Chart

When application can not find needed program parameters, it request needed parameters list automatically from server. Application user also can request application's parameters by using buttons on parameter tab.

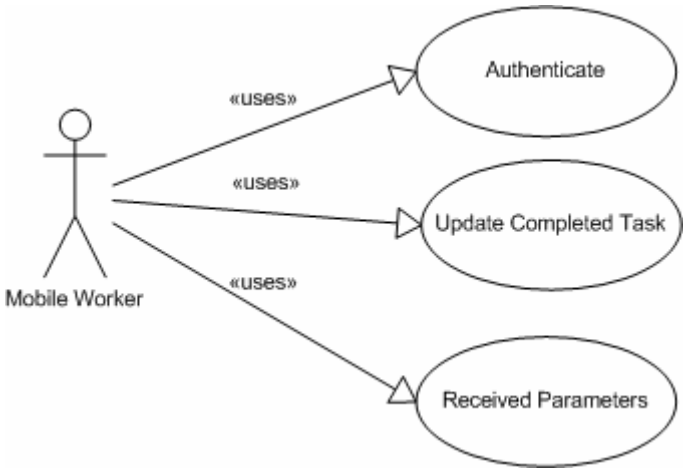

Figure 4.4 Mobile Worker use Case

Mobile worker follows tasks by using the application. He/she can not change task order and must do queued task. When he/she completes the task, he/she updates task status (Figure 4.4). If task status is became "Received", worker must enter receiver information (receiver name, receiver address). If receiver is recorded Customer, its name and address can be select from recorded addresses. The application sends task's new status and new information. If receiver and its address are recorded, application sends address's identification number (AddressID). Otherwise all other address data (mahalle, sokak, etc...) is sent to server. The application sends all these data as an XML file to Server application.

4.4. BackOffice Application

BackOffice application is used to manage system. BackOffice Application user can declare program parameters (task types, package types, address types, etc..) and manage users and task (Figure 4.6).

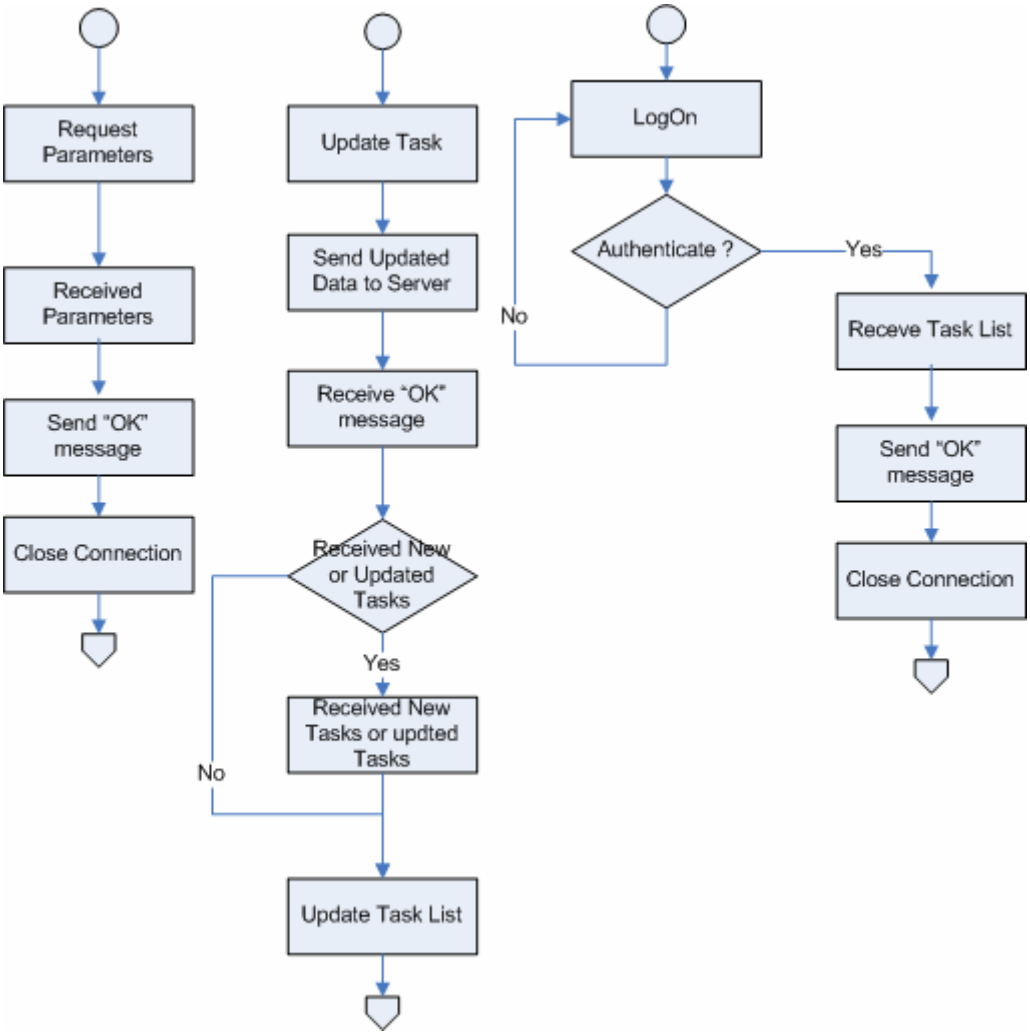

Figure 4.5 Client-Side application's Flow Charts

Officer can create new tasks. When he/she wants to inserts a new task, sender of the task is recorded as a customer with a default address. Two type of customer (company or personal) can record to system. Each customer can have more than one address. All recorded address's GPS coordinates are found from "CityAddresses" database, which has got GPS coordinates of all address in our application area. When task is recorded to database, after customer address insertion or selection (for recorded addresses) application assigns mobile worker to task according to task type and address's coordinates. Mobile user task list is updated and sent to mobile user by

the server application. Flow Chart of the application is shown in Figure 4.7. Only new tasks can be updated. Other tasks can only cancelled.

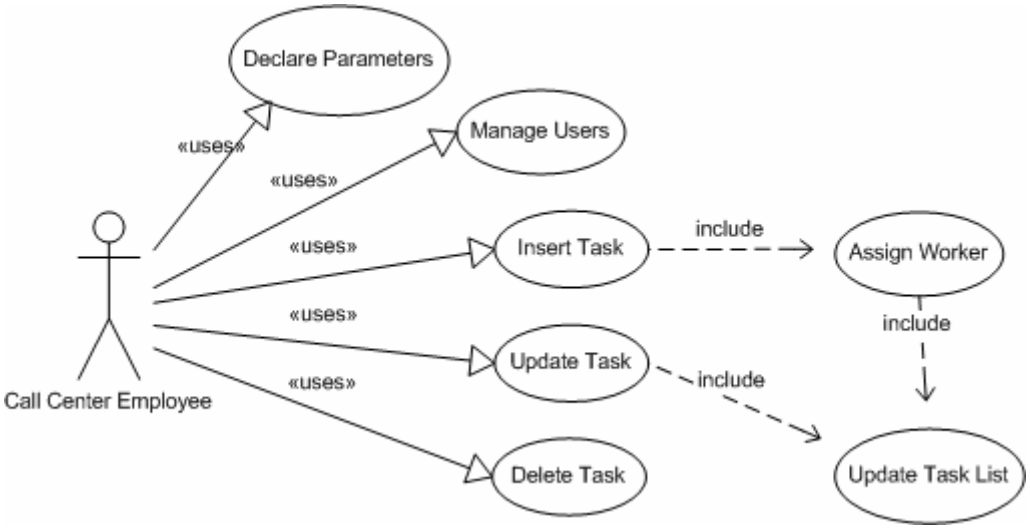

Figure 4.6 BackOffice application's user, use Case

Officer also can report task history, mobile worker's daily activities.

4.5. Database Design

In this application, we use Microsoft SQL Server 2000 database management system to store data. Two sub-systems (BackOffice and Server-Side applications) connect SQL Server by using ADO.Net connection type. The programs use stored procedures, which are compiled T-SQL strings, on SQL Server.

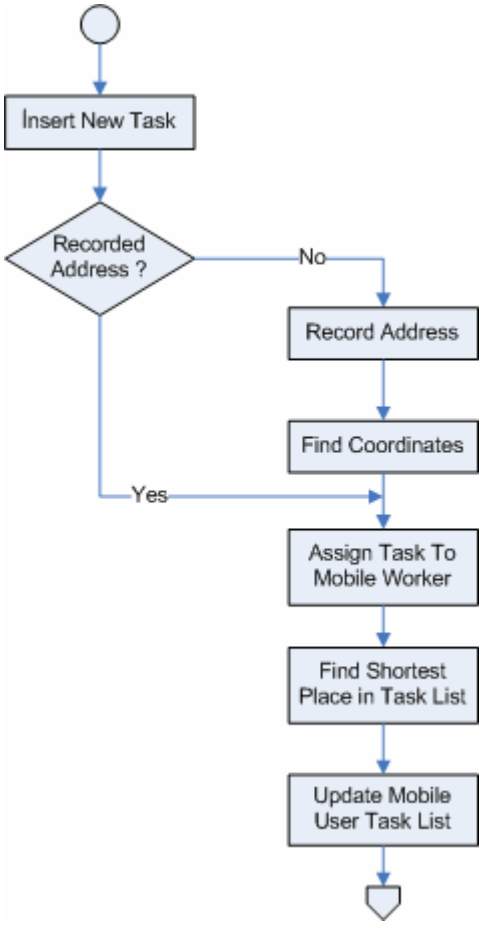

Figure 4.7 BackOffice application's Flow Charts

We will use 2 databases in this system. The first database is "Cargo", it is used to store our system's data. Other database is "CityAddresses", which is stored formatted addresses' GPS coordinates.

Main tables and data stored in our system's database CARGO are shown in Table 4.2.

Table 4.2 Database Tables

Table Name	**Description**
ActiveTask	Active Tasks (When a task is completed or cancelled, it is deleted from this table)
Address	Customer's addresses.

AddressType	Address Type of customers, like office Address, Factory Address, Branch Address.
City	All Cities and codes
Country	Countries
Customer	Personal and Company customers
CustomerAddress	Address of Customers
CustomerCompany	Customer Companies' details, Name, Tax Office, Tax Number, etc.
CustomerPerson	Personal Customers' details, Name, Surname, etc.
District	All Districts of Cities in City Table.
Logon	User Logon History
MobileUser	Declared mobile users, which use Client Application on PDA.
MobileUserType	Mobile User Types
MobileUserType_TaskType	Matched Task types and mobile user types.
Package	Tasks' package in formations.
Part	Part of the District
PackageType	Package types, like box, envelope, bag, etc.
Task	Each task is recorded in this table.
TaskStatus	Task status, like Completed, Cancelled, Waiting, etc.
TaskType	Task types.
TaxOffice	Tax Offices
UserGroups	User Groups
Users	Program Users.

Primary keys, columns and relations between these tables are shown in Figure 4.8.

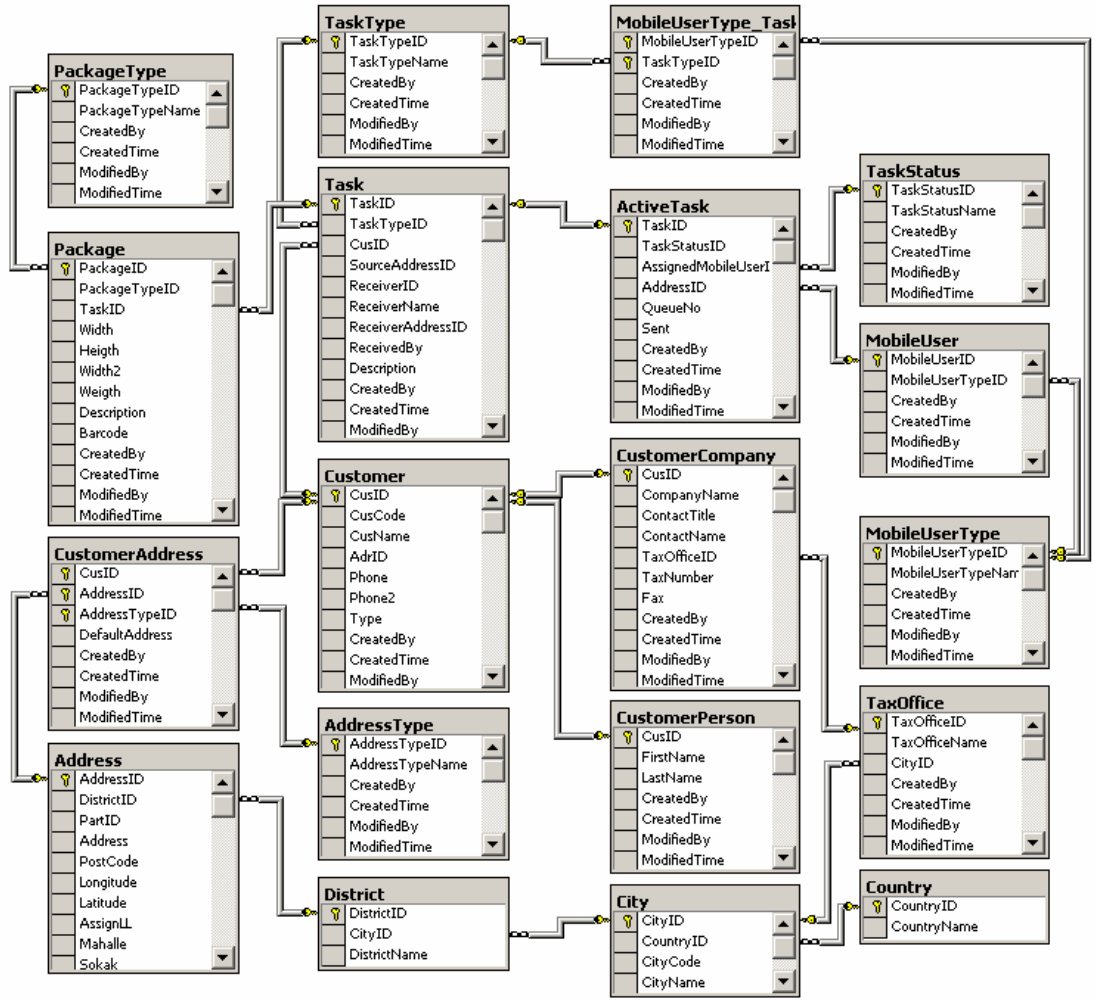

Figure 4.8 Relations between tables

CHAPTER 5

AN EXAMPLE USE OF THE PROPOSED SYSTEM

In this Chapter, we will constitute a scenario, and show the program interfaces in this scenario. In this scenario, we have got 3 actors, Tuncay as a office worker, Fatih and Uğur as mobile workers. Tuncay will logon to system by using Back Office application, declares program parameters and creates Uğur's and Fatih's user accounts.

- Tuncay logon to system by using his username (tuncay) and password (123).

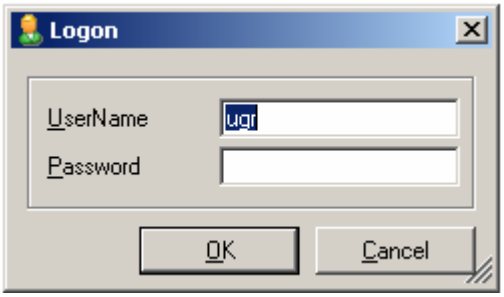

Figure 5.1 Logon Back Office Application

- Firsty Tuncay declares some program parameters, Address Types, Task Types, Package Types, Mobile User Types, User Groups. Tuncay declares "Depo Adresi" as an address type like in Figure 5.2.

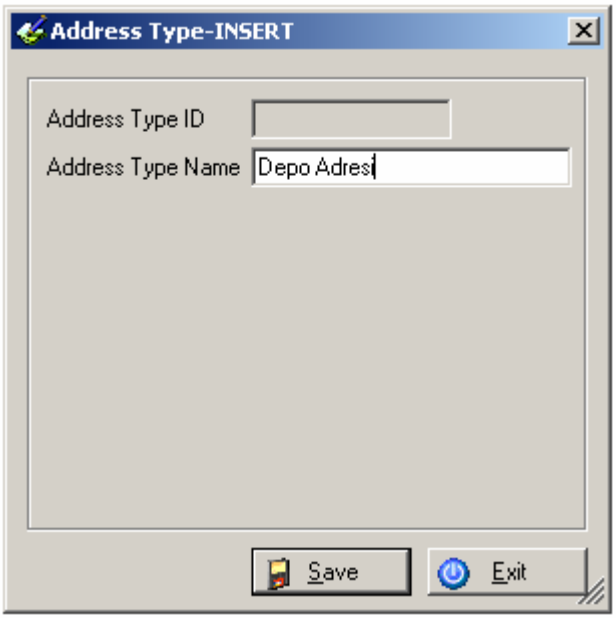

Figure 5.2 Adress Type Insert Form

After creatining new AddressType, new created address type will be listed in Address Types form (Figure 5.3). Tuncay also can make changes to other records by select record, press **Update** button and make changes on opened Address Type form and **Save** changes .

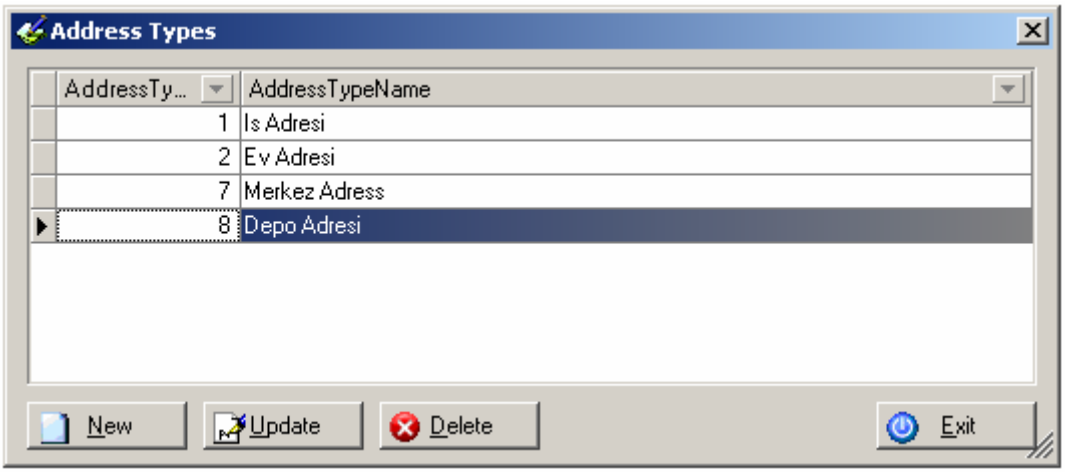

Figure 5.3 Adress Type List

- Tuncay opens Users form to create mobile user accounts for Uğur and Fatih (Figure 5.4). System has got 2 built-in user, “ugr” and “test”. He presses **New** button and opens User-Insert form (Figure 5.5) to create new users.

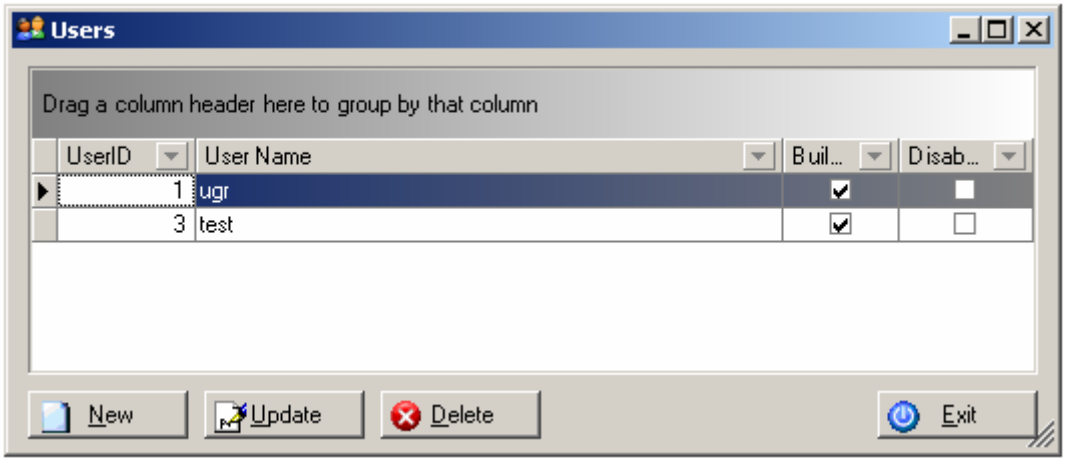

Figure 5.4 User List

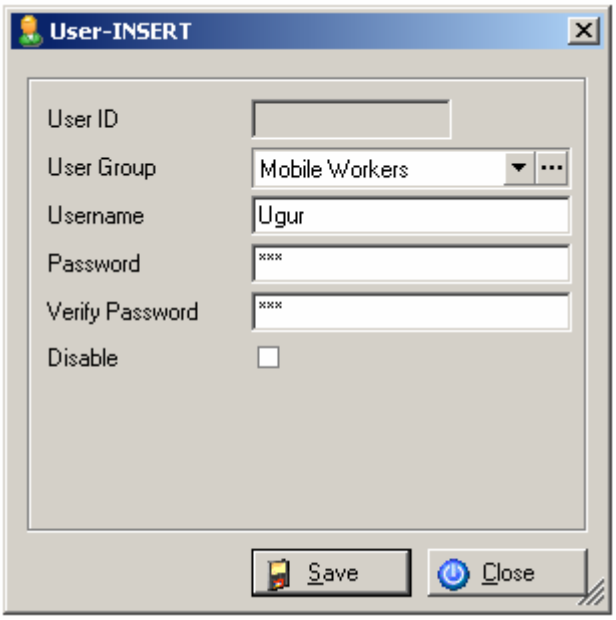

Figure 5.5 User Form

- Tuncay can list active tasks by using Active Tasks form (Figure 5.6).

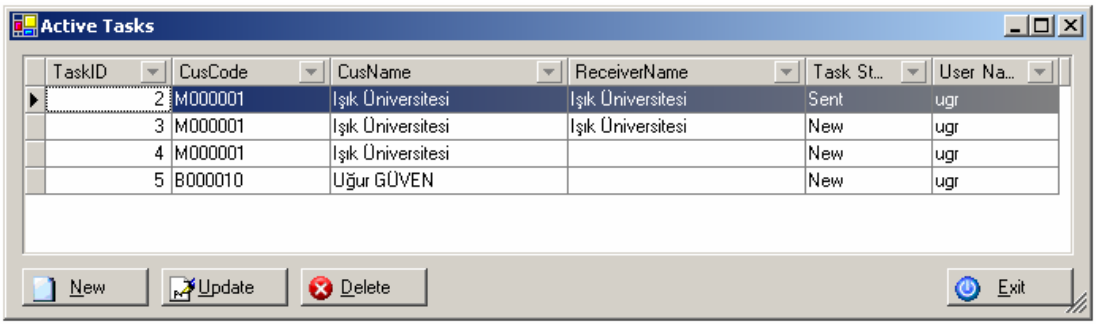

Figure 5.6 Active Tasks Form

He can enter new task by using **New** button. **New** button opens Active Task form to create new task(Figure 5.7). In this form he writes task description, and selects Task Type, Customer and Customers Address.

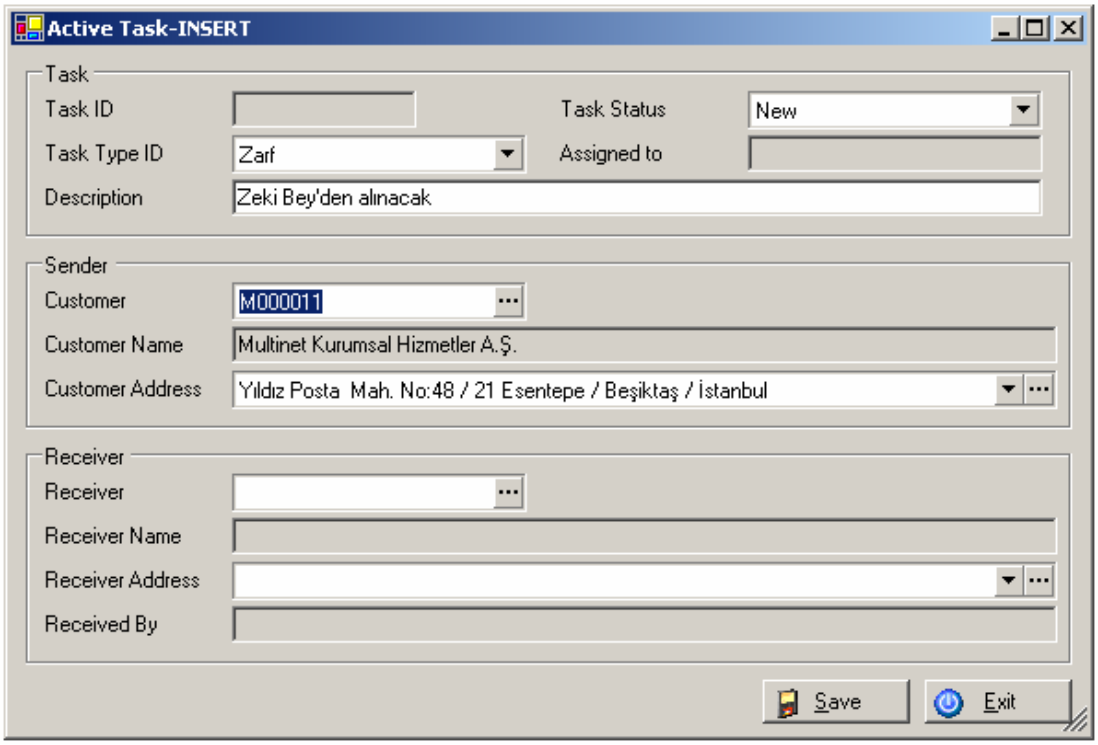

Figure 5.7 Active Task Form

He uses "**...**" button to list and selects customer from Customers Form. If customer is new he can create from opened Customers form to press **New** button. We can create two types of customer, company or personal. He creates a company customer Multinet, like in figure 5.8. He can also add extra addresses to customer by using "**...**" button at customer address line. For personal customers, Customer form has personal area, which includes firstname and lastname informations, instead of Company area

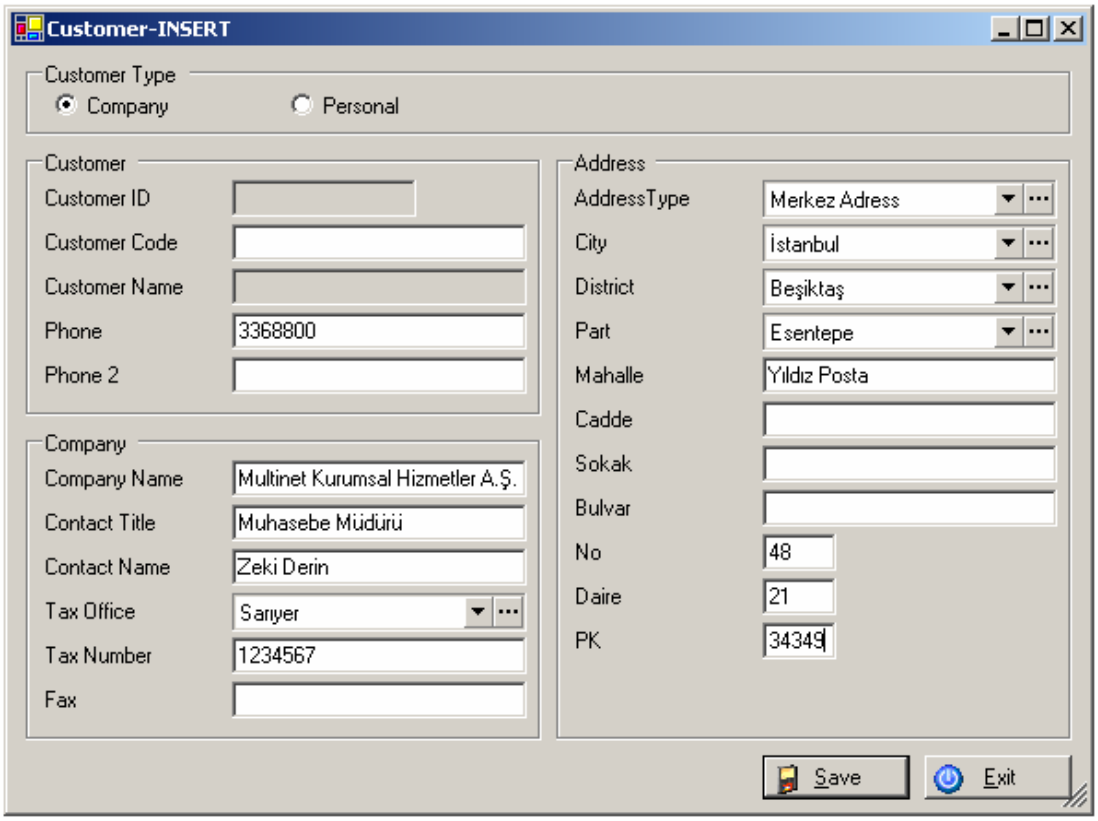

Figure 5.8 Customer Form

When Tuncay saves Task, application finds coordinates of the task address and finds assigned mobile worker and update his task list. We see that new task assign to mobile worker "fatih".

- Fatih can login Client application on their PDA devices. When application is started, logon form is appeared. He logons to system with correct username and password, and receives task list (Figure 5.9).

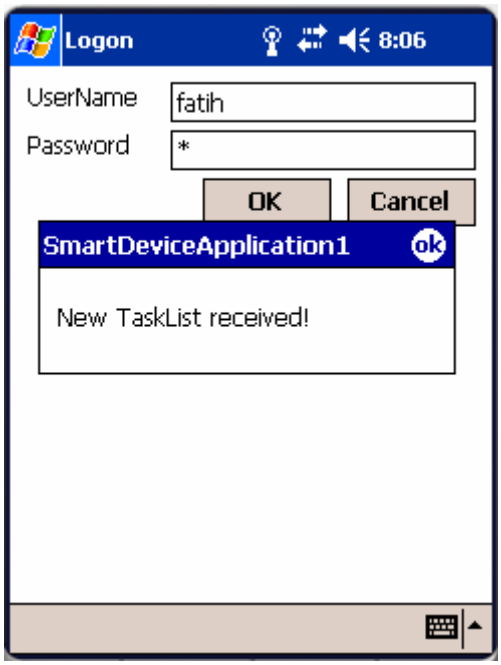

Figure 5.9 Client Application Authentication

- Client Application's main display is Tasks tab. Tasks tab shows worker's tasks like in Figure 5.10, which will be done. Fatih can open task to see all details by click the task record on list

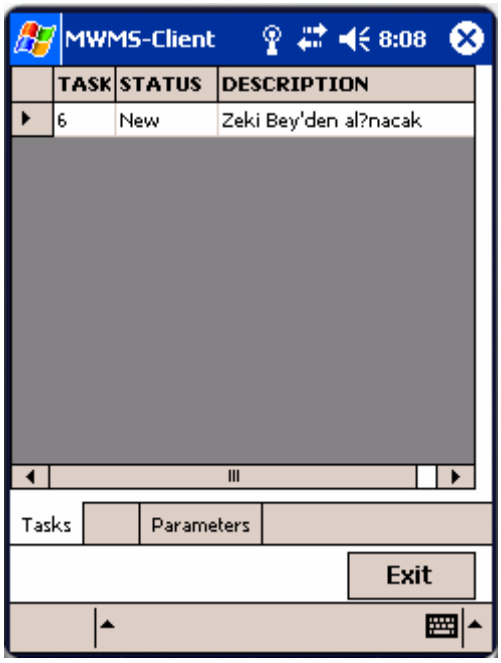

Figure 5.10 Client Application Task List

Fatih makes task status New-> Received and saves the task. When task status is made "Received", user must enter or select receiver information, Receiver, receiver Address (Figure 5.11). If receiver is our customer, he can find receiver customer by using "**…**" and select customer from customers form. He can not create customer account in here. He can only select before records. He can also select address from recorded addresses.

- When he clicks "**Save"** button, the updated task is sent to Server application and delete from user task list.
- When he makes task Status "Sent"-> "Delivery", User also must enter "Received by" data and finalize the task.
- He can receive program parameters by parameters tab of the application.

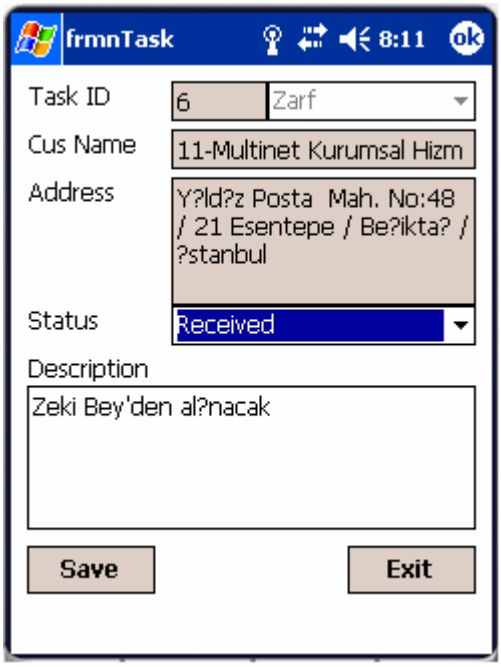

Figure 5.11 Task Status Update

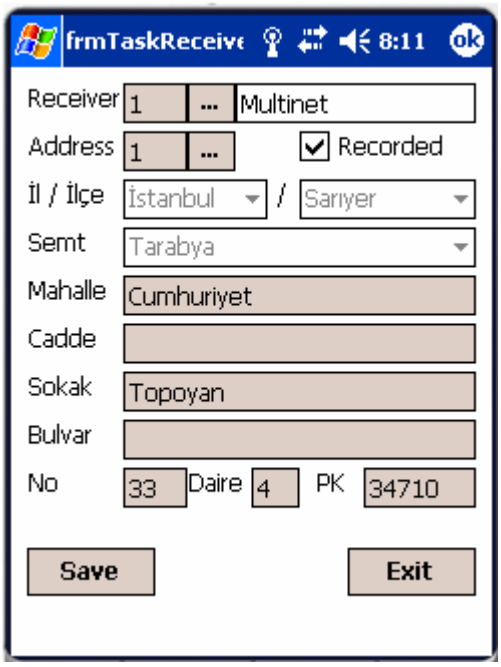

Figure 5.12 Receiver Informations

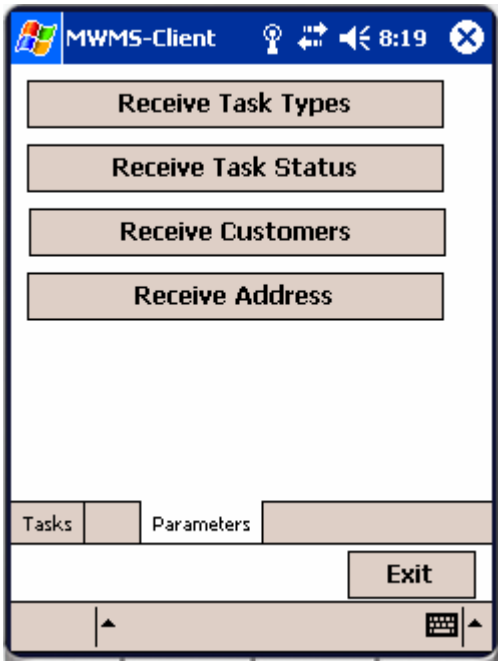

Figure 5.13 Client Application Parameter Request Form

- Server application works on Application server of the company. It starts to listen port 9999 after start button pressed. It saves mobile workers position informations to SQL server. Find new addresses' GPS positions and assign updated tasks to mobile workers. In these scenario, Server application received fatih's task updates and receiver address. It finds GPS positon of the receiver address and assign this task to user "ugur".

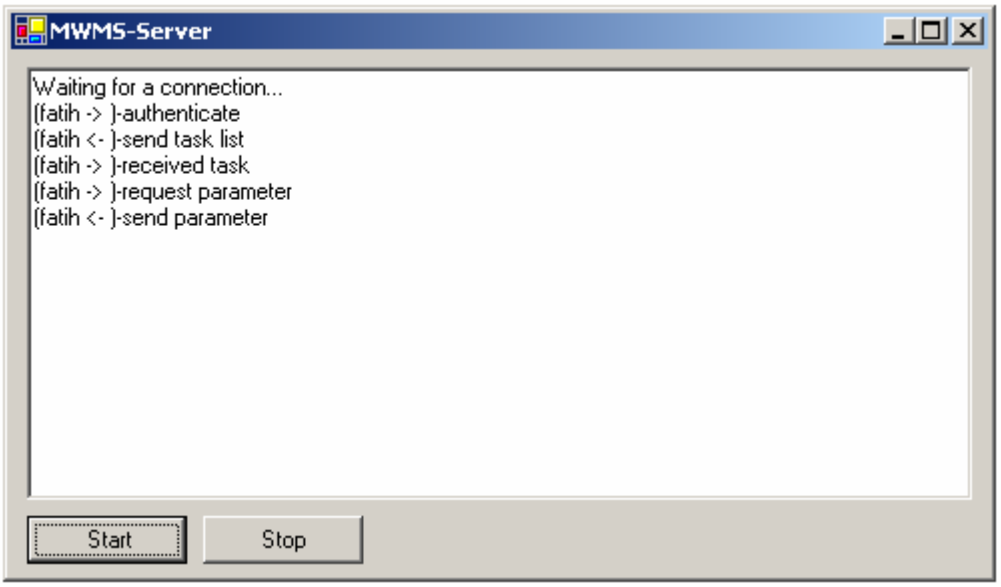

Figure 5.14 Server-Side Application

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK

In Today's world, technology has been developing rapidly. Companies start to use new technologies to increase their competition power and market share. Logistics market also needs to use these new technologies, mobile devices, collaborative systems to manage their workers. Today, they manage their workers without a system. Office workers orientated them by using radio announcement and cellular phones.

In this thesis, we proposed a mobile worker management system (MWMS) for a logistics company, which use GPS, mobile devices and .NET technologies. The system aims are assigning tasks to mobile workers automatically, less using radio or cell phone to manage workers. It decreases personal errors, costs and accelerates works. System sends and receives data to mobile workers, who use mobile application on their mobile devices via GPRS.

In future works, the system can be integrated with a barcode system. Web site can be developed for Customers' follow-ups. A digital Map can be integrated with Client-Side application to help mobile workers. Task assigning algorithm also can be developing with new algorithm.

REFERENCES

[1] Deniz Eseryel, Radha Ganesan, Gerald S. Edmonds, *"Review of Computer-Supported Collaborative Work Systems"* in Educational Technology & Society 5 (2) 2002

[2] Council of Supply Chain Management Professionals Web Site, http://www.cscmp.org/

[3] Garmin Web Site, www.garmin.com

[4] McNamara, J. "*GPS for Dummies*", Wiley Publishing , 2004, p.11-15

[5] Canadian Heritage Information Network Web Site, http://www.chin.gc.ca

[6] Microsoft Web Site, http://www.microsoft.com

[7] Bowman, R. "*Visual Basic.NET*", Hungry Minds, 2002, p. 10

[8] Petroutsos, E. "*Mastering Visual Basic.NET*", Sybex, 2002, p.3-4

[9] Utley, C. "*A Programmer's Introduction to Visual Basic.NET*", Sams Publishing, 2001 p. 3-4

[10] Gosnell, D.; Reynolds, M. and Forgey, B., "*Beginning Visual Basic .NET Database Programming*", Wrox Press, 2001, p.1-3

[11] Bruegge, B. and Dutoit, A., "*Object-Oriented Software Engineering*", Prentince Hall,2000

[12] Grundgeiger D., "*Programming VisualBasic .NET*", O'Reilly Publication, 2002

[13] Roman, S.; Petrusha, R. and Lomax, P.,"*VB .NET Language in a Nutshell*", O'Reilly Publication, August 2001

[14] GPS World Web Site, http://www.gpsworld.com

[15] Pocket GPS World Web Site, http://www.pocketgpsworld.com/

[16] Christine A. Halverson, Activity theory and distributed cognition: or What does CSCW need to do with theories?, IBM Research, 2001

[17] Gerhard Fischer, Leysia Palen, Computer-Supported Cooperative Work (CSCW), April 7, 1999

[18] Eberhard Bluemel, Leonid Novitsky, Svetlana Vinichenko, Egils Ginters, Web-Based and Mobile Solutions for collaborative work environment with logistics and maritime applications, 2002

APPENDIX A

XML FILES' SCHEMAS AND SAMPLE FILES

A.1. Schema of Logon XML

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="LogOn">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="USERNAME" type="xs:string" minOccurs="0" />
        <xs:element name="PASSWORD" type="xs:string" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="tr-TR">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="LogOn" />
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A.2. Sample of Logon XML

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<LogOn>
   <USERNAME>uguven</USERNAME>
   <PASSWORD>123</PASSWORD>
</LogOn>
```

A.3. Schema of Receive Task XML

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<xs:schema id="TASKS" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="TASKS" msdata:IsDataSet="true" msdata:Locale="tr-TR">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="TASK">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="TASKTYPEID" type="xs:string" minOccurs="0" msdata:Ordinal="0" />
              <xs:element name="CUSID" type="xs:string" minOccurs="0" msdata:Ordinal="1" />
              <xs:element name="DESC" type="xs:string" minOccurs="0" msdata:Ordinal="2" />
              <xs:element name="ADDRESS" type="xs:string" minOccurs="0" msdata:Ordinal="3" />
              <xs:element name="LATITUDE" type="xs:string" minOccurs="0" msdata:Ordinal="4" />
              <xs:element name="LONGTUDE" type="xs:string" minOccurs="0" msdata:Ordinal="5" />
            </xs:sequence>
            <xs:attribute name="TYPE" type="xs:string" />
            <xs:attribute name="ID" type="xs:string" />
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A.4. Sample of Receive Task XML

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<TASKS>
  <TASK TYPE="INS" ID="1">
    <TASKTYPEID>1</TASKTYPEID>
    <CUSID>1</CUSID>
    <DESC>Description</DESC>
    <ADDRESS>Cumhuriyet Mahallesi Topoyan Sokak No:33 Daire 4 Tarabya</ADDRESS>
    <LATITUDE>N 50 50.567</LATITUDE>
    <LONGTUDE>E 04 21.583</LONGTUDE>
```

```
  </TASK>
  <TASK TYPE="UPD" ID="2">
    <TASKTYPEID>1</TASKTYPEID>
    <CUSID>1</CUSID>
    <DESC>Description_2</DESC>
  </TASK>
</TASKS>
```

A.5. Schema of Sending Coordinates XML

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="NewDataSet" xmlns="" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-microsoft-com:xml-msdata">
  <xs:element name="POSITION">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="USERID" type="xs:string" minOccurs="0" />
        <xs:element name="LATITUDE" type="xs:string" minOccurs="0" />
        <xs:element name="LONGTUDE" type="xs:string" minOccurs="0" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="NewDataSet" msdata:IsDataSet="true" msdata:Locale="tr-TR">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element ref="POSITION" />
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A.6. Sample of Sending Coordinates XML

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<POSITION>
    <USERID>1</USERID>
    <LATITUDE>N 50 50.567</LATITUDE>
    <LONGTUDE>E 04 21.583</LONGTUDE>
</POSITION>
```

A.7. Schema of Task List XML

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="TASKS" xmlns=""
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:msdata="urn:schemas-
microsoft-com:xml-msdata">
  <xs:element name="TASKS" msdata:IsDataSet="true" msdata:Locale="tr-TR">
    <xs:complexType>
      <xs:choice maxOccurs="unbounded">
        <xs:element name="TASK">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="ID" type="xs:string" minOccurs="0" />
              <xs:element name="TASKTYPEID" type="xs:string" minOccurs="0" />
              <xs:element name="CUSID" type="xs:string" minOccurs="0" />
              <xs:element name="DESC" type="xs:string" minOccurs="0" />
              <xs:element name="ADDRESS" type="xs:string" minOccurs="0" />
              <xs:element name="LATITUDE" type="xs:string" minOccurs="0" />
              <xs:element name="LONGTUDE" type="xs:string" minOccurs="0" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

A.8. Sample of Task List XML

```
<?xml version="1.0" encoding="ISO-8859-9"?>
<TASKS>
  <TASK>
    <ID>1</ID>
    <TASKTYPEID>1</TASKTYPEID>
    <CUSID>1</CUSID>
    <DESC>Description</DESC>
    <ADDRESS>Cumhuriyet Mahallesi Topoyan Sokak No:33 Daire 4
Tarabya</ADDRESS>
    <LATITUDE>N 50 50.567</LATITUDE>
    <LONGTUDE>E 04 21.583</LONGTUDE>
  </TASK>
  <TASK>
```

```
    <ID>2</ID>
    <TASKTYPEID>1</TASKTYPEID>
    <CUSID>1</CUSID>
    <DESC>Description_2</DESC>
    <ADDRESS>Cumhuriyet Mahallesi Topoyan Sokak No:33 Daire 4
Tarabya</ADDRESS>
    <LATITUDE>N 50 50.567</LATITUDE>
    <LONGTUDE>E 04 21.583</LONGTUDE>
  </TASK>
</TASKS>
```

APPENDIX B

CD INCLUDING DOCUMENTS AND APPLICATION SOURCE CODES