

OCCUPATIONAL HEALTH AND SAFETY
MANAGEMENT TOOL

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

BY

ALPER GÜNGÖR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN
THE DEPARTMENT OF MINING ENGINEERING

January 2004

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 Doctor of Philosophy.

Prof. Dr. Ümit Atalay
Head of 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 Doctor of Philosophy.

Prof. Dr. Tevfik Güyagüler
Supervisor

Examining Committee Members

Prof. Dr. Tevfik Güyagüler

Prof. Dr. Naci Bölükbaşı

Prof. Dr. Nurkan Karahanoğlu

Prof. Dr. Erhan Tercan

Assoc. Prof. Dr. Levent Tutluoğlu

ABSTRACT

OCCUPATIONAL HEALTH AND SAFETY MANAGEMENT TOOL

Güngör, Alper

Ph. D. Department of Mining Engineering

Supervisor: Prof. Dr. Tevfik Güyagüler

January 2004, 168 Pages

Labor protection, that is prevention of occupational disease and reducing the frequency of accident, has always been a matter of major concern of mining industry. Management and the government should promote and maintain high safety standards through some measures and tools to reduce frequency of accidents and occupational diseases.

This thesis describes the development of a national occupational health and safety management tool that is composed of educational, statistical and database interfaces for mine safety and health administration.

The detailed analysis of an accident requires knowledge of many parameters such as location, time, type, cost of the accident, victim information, nature of injury, result of the accident etc. that can be obtained from a standard coded accident report form. So, database interface of the management tool is developed with this sense to realize

collecting accident data in a nationally used format to produce a common safety reporting system.

Prepared database maintains user-friendly environment on Internet to submit accident information. Dynamic structure and ease of use of the developed database allow administered user to expand it without detailed computer programming knowledge. This was achieved by prepared modules to change or register new data fields within the database.

Created database is also secure since only gives data input access rights to registered users. Database administrator is able to create registered users. Registered users could be safety engineer or manager of a mine who is responsible from the submission of data to the ministry of labor. So, standard and secure accident data collected rapidly through Internet connection.

The other feature of the database is that, it is open to all people to query accidents with many aspects.

Prepared management tool also includes educational interface, content of which can also be enlarged, as the new ideas, information or solutions for accidents are improved. This information is also open to all people since educating workers and managers about accidents and prevention techniques can improve working conditions and increase awareness.

Knowing the fact that submitted accident data is still collected in hard copies in folders, the need for a kind of management tool, which is completed in this study, is obvious. Application of this kind of management tool will be able to prevent the collection of accident data in dusty shelves and share the accident data information with all people who are interested in with this subject.

Key words: Accident Report Form, Accident Database, Occupational Diseases and Mine Accidents, Accident Reporting System, Health and Safety Management Tool.

ÖZ

İŞÇİ SAĞLIĞI VE İŞ GÜVENLİĞİ
YÖNETİM ARACI

Güngör, Alper

Doktora, Maden Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Tevfik Güyagüler

Ocak 2004, 168 Sayfa

İş güvenliği, diğer bir deyişle meslek hastalıklarından korunma ve kaza oranlarının azaltılması, madencilik endüstrisinin en önemli konularından birisidir. Yönetim ve devlet, yüksek iş güvenliği standartları oluşturarak ve bazı araçlar kullanarak kaza oranları ve meslek hastalıklarının azaltılmasını sağlamakla yükümlüdürler.

Bu tezde, başka; eğitim, istatistik ve veritabanı arabirimlerinden oluşan ve maden endüstrisi için işgüvenliği ve sağlığı yönetim aracı olarak çalışan ulusal iş sağlığı ve iş güvenliği yönetim aracının geliştirilmesi anlatılmıştır.

Kazaların detaylı analizi, yer, zaman, tip, maliyet, kazaya uğrayanın kişisel bilgileri, kazanın doğası, sonuçları gibi bir çok parametrenin bilinmesini gerektirir. Bunlar standart bir kaza bildirim formu aracılığıyla toplanabilir. Bu sebepten dolayı, geliştirilen yönetim aracının veritabanı arabirimi, standart bir raporlama sistemi oluşturacak şekilde kaza bilgilerini toplayan bir yapıda hazırlanmıştır.

Hazırlanan veritabanı, kolay kullanımlı yapısıyla Internet üzerinden kaza bilgilerinin gönderilmesini sağlar. Dinamik yapısı ve kullanım kolaylığı sayesinde veritabanı yöneticisinin detaylı bir bilgisayar programlama bilgisine ihtiyaç duymadan veritabanını geliştirmesini olanaklı kılar. Bu, veritabanına yeni bilgi alanları eklenmesi veya halihazırda girilmiş bulunan bilgi alanlarının değiştirilmesini sağlayacak şekilde hazırlanan modüller aracılığıyla gerçekleştirilmiştir.

Oluşturulan veritabanı güvenlidir, çünkü sadece onaylanmış kullanıcılara kaza bilgileri girme hakkı tanır. Veritabanı yöneticisi yeni kullanıcılar yaratabilir. Onaylı kullanıcılar, oluşan kazayı Çalışma ve Sosyal Güvenlik Bakanlığı'na bildirmekle sorumlu fenni nezaretçi veya o madenden sorumlu yöneticiler olabilir. Böylece, kaza bilgileri standart ve güvenli bir şekilde Internet üzerinden hızla toplanır.

Veritabanının diğer bir özelliği de, kaza sorgulamaları için sistemin herkese açık olmasıdır. Yönetim aracının eğitim arabirimi, kazalar hakkında yeni fikirler, bilgiler ve çözümler geliştirildikçe içeriği büyüyebilecek şekilde hazırlanmıştır. Bu bilgiler, yönetici ve çalışanların kazalar hakkında eğitiminin çalışma koşullarını geliştirebileceği ve bu konuya ilgiyi arttıracığından dolayı herkesin kullanımına açıktır.

Kaza bilgilerinin hala kağıtlar üzerinde toplanarak dosyalarda saklanması gerçeğinin farkında olarak, bu çalışmada hazırlanan şekilde bir yönetim aracına ihtiyaç olduğu aşikardır. Bu şekilde bir yönetim aracı sayesinde, kaza bilgilerinin tozlu raflarda toplanmasının önüne geçilerek, bu konuyla ilgilenen herkesin kullanımına açılması sağlanacaktır.

Anahtar Kelimeler: Kaza Rapor Formu, Kaza Veritabanı, Meslek Hastalıkları ve Maden Kazaları, Kaza Raporlama Sistemi, Sağlık ve Güvenlik Yönetim Aracı

To my dear family

ACKNOWLEDGEMENTS

I would like to express my great appreciation to my supervisor Prof. Dr. Tevfik Güyagüler for his guidance, concern, friendship and encouragement throughout my thesis. I wish to express my appreciation to committee members; Prof. Dr. Naci Bölükbaşı and Prof. Dr. Nurkan Karahanoğlu for their suggestions and valuable comments.

I would like to sincerely thank my family for their support, help and patience throughout this tedious job.

I wish to thank my colleagues for sharing their ideas with me and their friendship. I wish also express my special thank to my best friend Koray Önal and Hakan Başarır, who is my roommate in Mining Engineering Department, for their support.

TABLE OF CONTENTS

ABSTRACT	iii
ÖZ	v
ACKNOWLEDGEMENT	viii
TABLE OF CONTENTS	ix
LIST OF TABLES	xii
LIST OF FIGURES	xiii
CHAPTER	
1. INTRODUCTION	1
1.1. General Remarks	1
1.2. Methodology of the Thesis	4
1.3. Objective of the Thesis	5
1.4. Thesis Outline	6
2. MINE ACCIDENT ANALYSES	7
2.1. Overview	7
2.2. Accident Occurrence	9
2.3. Legal Framework	10
2.4. System Dynamics of the Mine Accidents	11
2.5. Cost of the Accidents	18
2.6. Standard Format	21
2.6.1. The Inquiry	21
2.6.2. The Investigation	22
2.7. General Information about the Accident Coding System	23
2.8. Data Analysis	24
2.9. Calculation of Measurement Rates	25
2.10. Risk Analysis	26

3. COMPUTER PROGRAMMING AND TOOLS USED TO CREATE	
OHSMT.....	28
3.1. Database	28
3.2. Relational Database Model	28
3.3. Perl	29
3.4. Common Gateway Interface	30
3.5. World Wide Web	31
3.6. Fundamentals of SQL Server Architecture	32
3.7. Structured Query Language (SQL)	33
3.8. System Requirements.....	34
3.9. Required Program and Operating System Licenses.....	35
3.10. SQL Server™ Authentication	35
3.11. Windows NT and SQL Server Authentication Mixed Mode.....	36
4. CONNECTION OF THE SYSTEM TO THE DATABASE.....	38
4.1. Overview	38
4.2. Interfaces	38
5. ACCIDENT CODING STANDARDS CREATED FOR OHSMT	42
5.1. Accident Information Data.....	42
5.1.1. Company Information Data Items.....	42
5.1.2. Personal Information of the Employee Subjected to an Accident	43
5.1.3. Accident Details	43
5.1.4. Outcome of the Accident	43
5.2. Coding the Mechanism of Injury/Disease.....	43
5.3. Bodily Location of Injury/Disease Classification.....	44
5.4. Mechanism of Injury/Disease Classification	45
5.5. Agency Classification	45
6. USAGE OF MANAGEMENT TOOL.....	46
7. CONCLUSIONS AND RECOMMENDATIONS	63
REFERENCES.....	66
APPENDICES	
A. Accident Information Data Details.	69
B. Mechanism of Injury/Disease Details.	83

C. Bodily Location of Injury/Disease Classification Details.....	85
D. Mechanism of Injury/Disease Classification Details.	88
E. Agency Classification Details.....	92
F. Written PHP Codes	95
G. Apache Server Configuration File	143
H. PHP Server Configuration File	150
I. PHP Modules.	162
VITA	168

LIST OF TABLES

TABLE

1.1. Accident and occupational disease statistics of Turkey	2
1.2. SSK statistics (2002) by activities	3
1.3. SSK statistics (2002) by province	3

LIST OF FIGURES

FIGURES

1.1. TTK injury statistics (DIE 1977-2001).....	2
1.2. TTK fatality statistics (DIE 1977-2001).....	2
2.1. Accident and possible consequences	9
2.2. Accident investigation.....	10
2.3. Legislative framework	12
2.4. Safety framework	14
2.5. Safety parameters	14
2.6. System dynamics of the occupational health and safety system.....	17
2.7. Relationship between the safety expenses and cost of the accident.....	20
3.1. Accessing a database using a web browser.....	31
3.2. Company data in SQL table	34
3.3. Windows NT authentication	37
3.4. SQL Server security decision tree.....	37
4.1. Accident database tables in SQL server.....	39
4.2. Accident mechanism fields	40
4.3. Field details for an accident	40
4.4. Database administration through a web interface	41
4.5. Web interface access to the database	41
6.1. National safety mining database index page.....	47
6.2. National safety mining database, database link	47
6.3. National safety mining database, education link	48
6.4. National safety mining database, legal issues link.....	48

6.5. National safety mining database, bulletin board link.....	49
6.6. National safety mining database, links	49
6.7. National safety mining database, news link.....	50
6.8. Ordinary user search in database.....	50
6.9. Ordinary user search results for fatality in database	51
6.10. Calculation of parameters	51
6.11. Calculation of parameters company selection.....	52
6.12. Calculation of parameters selection of site belong to company.....	52
6.13. Calculation results for selected site of company.....	52
6.14. Administrator login page.....	53
6.15. Administrator main page.....	53
6.16. Administrator new agency record page.....	54
6.17. Administrator new sub-agency record page.....	54
6.18. Administrator new company record page	55
6.19. Administrator mine record page first step.....	55
6.20. Administrator new mine record page second step	56
6.21. Administrator new mine record completed.....	56
6.22. Administrator new user record first step page	57
6.23. Administrator new user record second step page.....	57
6.24. Administrator new personnel record page first step	57
6.25. Administrator new personnel record page second step.....	58
6.26. Administrator new personnel record completed page.....	58
6.27. User login page	59
6.28. Unauthorized user login name or password result	59
6.29. Authorized user login.....	59
6.30. Authorized user accident report insurance number input	59
6.31. Authorized user accident report insurance number not valid warning	60
6.32. Authorized user accident report valid insurance number input	60
6.33. Authorized user accident report input first step	60
6.34. Authorized user accident report input second step	61
6.35. Authorized user accident report input third step.....	61
6.36. Authorized user accident report input fourth step.....	62
6.37. Authorized user accident report input final step	62

CHAPTER I

INTRODUCTION

1.1 General Remarks

Historically, mining has been the industry with the highest fatal and nonfatal injury rates. Mining still has the highest fatal injury rate. Underground work locations exhibited both the highest numbers and rates of fatalities, and preparation plants and mills exhibited the lowest fatality rates. Because of the accompanying social toll and public pressures for action, data on fatalities and injuries became available far earlier in the mining industry than in many others. State and Federal agencies began collecting data in the 1870s, and reliable information has been available for an entire century in USA.

Labor protection, that is prevention of occupational disease and reduction in the frequency of accident, has always been a matter of major concern of mining industry. Management and the government should promote and maintain high safety standards through some measures to reduce frequency of accidents and occupational disease.

Unfortunately, in undeveloped countries the problems related with health and safety are still unsolved causing high number of serious accidents and occupational diseases. When mine accidents and the occupational diseases are concerned, Turkey is still one of the leading countries. The related statistics for 1999 are given in Table 1.1. Also, Figure 1.1 and Figure 1.2 show the injury and fatality rates respectively for Turkish Hard Coal Foundation (TTK).

Table 1.1. Accident and occupational disease statistics of Turkey (SSK, 1999)

Branch of Activity	Employment Accident	Occupational Diseases	Permanent Incapacity for work	Death
Coal mining	5428	629	985	49
Other mining activities	423	4	19	15
Total	5851	633	1004	64

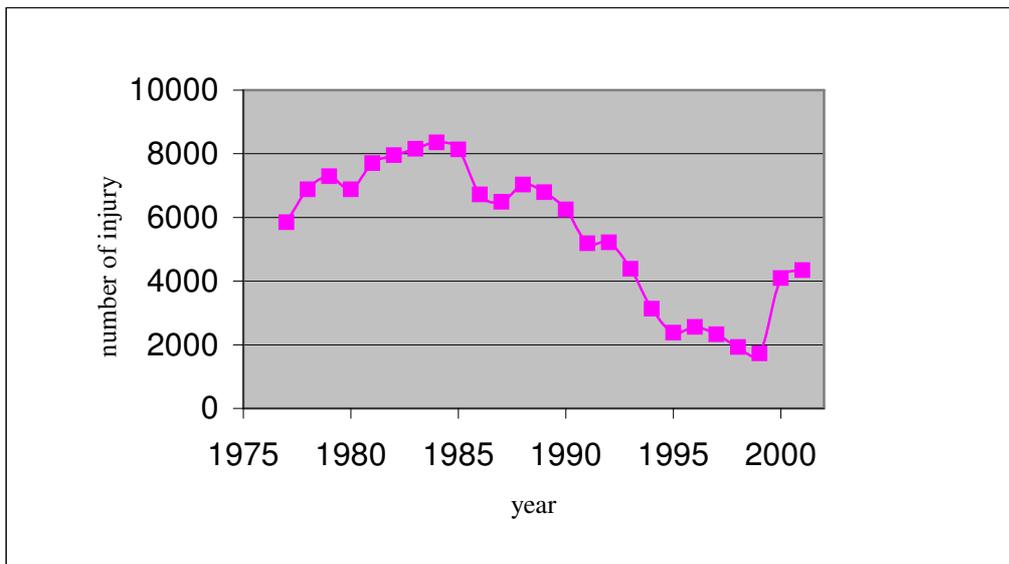


Figure 1.1. TTK injury statistics (DIE 1977-2001)

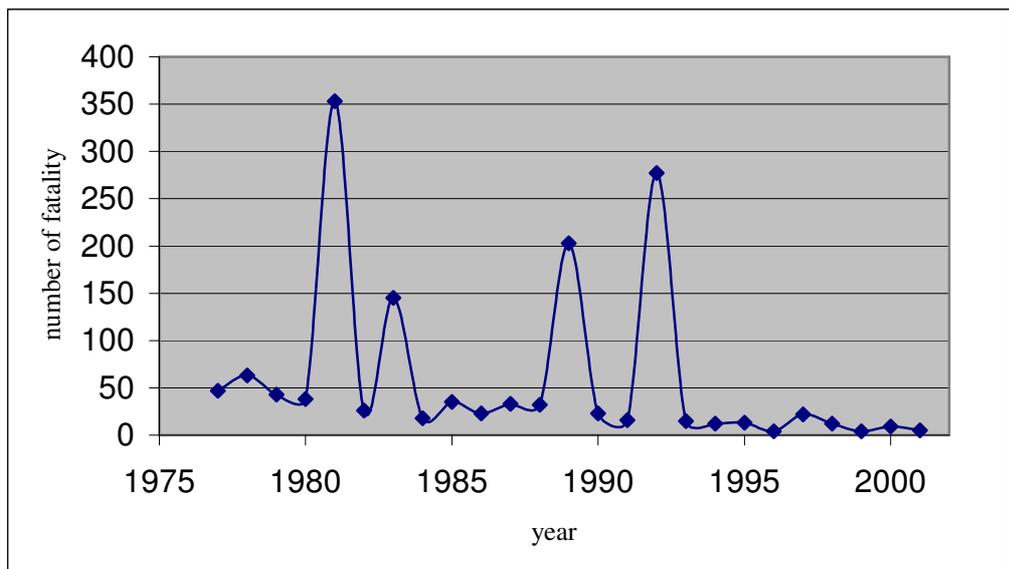


Figure 1.2. TTK fatality statistics (DIE 1977-2001)

As it is seen from the Table 1.1, Figure 1.1 and Figure 1.2, reducing mine accidents and occupational diseases is vital in mining industry of Turkey (Guyaguler, T., 2000).

Social Insurance and Security Foundation (SSK) statistics including the number of accidents, occupational diseases, permanent incapacity for work, and death according to the industries for 2002 are given in the Table 1.2. Statistics show that mining and construction industries have the highest accident rates in 44 classified industries. Distribution for the same parameters by provinces is given in the Table 1.3.

Table 1.2. SSK statistics (2002) by activities.

Activity	Accident	Occupational Diseases	Permanent Incapacity	Death	Total Temporary Incapacity for work (days)
Coal Mining	7.104	426	342	61	141.117
Non-Coal Mining	367	19	6	10	10.821
Stone Quarries, clay and sand pits	358	0	28	20	17.102
Other, non-metallic material production	178	0	4	5	5.612
Mining Industry Total	8.007	445	380	96	174.652
Construction	8.459	11	517	341	311.019
TOTAL	72.367	883	2.183	1.008	1.852.502

Table 1.3. SSK statistics (2002) by province.

Province	Accident	Occupational Diseases	Permanent Incapacity	Death	Total Temporary Incapacity for work (days)
Ankara	2.893	251	116	367	101.292
Bursa	5.272	2	139	141	99.082
İstanbul	8.527	81	356	437	294.104
İzmir	10.683	0	185	185	265.956
Kocaeli	5.864	22	111	133	137.947
Zonguldak	4.656	422	324	23	109.403
TOTAL	72.367	883	2.183	1.008	1.852.502

Unsurprisingly, Zonguldak is in the first 6 provinces among 81 cities of Turkey having high rates probably due to coal mining industry.

If the accidents and safety considered together, cost is another factor. It plays an important role in all type of industries. Safety in relation to cost stands in a special position. Occurrence of accident causes to rise of cost to the company due to increase in compensation amounts, loss of a key worker for a long period or permanently, education costs of new workers, material or machinery losses, interruptions of work etc. All these results directly increase the cost. Also disorganisation within the workplace or psychological problems after the accident or near misses may decrease efficiency and productivity indirectly. It is not easy to calculate the total cost of the accident. Direct costs may be calculated with some costs models but indirect costs can not be calculated accurately most of the time. Besides, the worth of human life could not be evaluated with money.

1.2. Methodology of the Thesis

Preventing occupational injuries and illnesses depends on our ability to quantify and track them. Through occupational safety and health surveillance, one can provide ongoing and systematic collection, analysis, interpretation, and dissemination of data for the purposes of prevention. Surveillance increases the effectiveness of prevention activities by targeting them to industries, workplaces, and occupations that have the greatest needs. Surveillance also expands knowledge about which prevention programs are effective.

It is also important to gather information on health and safety so that it may be possible to make informed decisions about specific issues in the workplace. The information, for example, can enable people to identify hazards, take action to prevent injuries, or to control conditions that may affect the health of themselves and other workers.

This may be achieved by using the science and technology to improve the working condition, educating workers and managers about accidents and prevention technique (Guyaguler, T., 2000).

Using computer technology it is possible to build a Management Tool composed of database, multimedia and Internet technology satisfying these needs to prevent accidents. So, it is possible to make existing data more accessible to provide a variety of occupational safety and health surveillance and educational information on Internet.

Management Tool computer program includes educational and statistical interfaces for Mine Safety and Health Administration. To achieve this, programming languages and visual environments, discussed in Chapter three, are used.

1.3. Objective of the Thesis

The objectives of this study are as follows;

- To create a database to collect accident information data in a standard form,
- To create a standard accident report form which is dynamic. That means, it can be developed by administered user and instantly available to registered users,
- To provide accident data which can be queried any time by all people,
- To develop a management tool that collect all information about accident at one centre,
- To create an environment to share ideas and information that could help to increase awareness about accidents,
- To create a standard national health and safety database to speed up the collection of a data in a secure way.

1.4. Thesis Outline

Chapter one gives general remarks about the importance of the mine accidents and methodology, objectives and the outline of the thesis.

The second chapter introduces a critical survey about mine accidents. System dynamic approach prepared for mine accidents also given in this chapter. Importance of the cost of the accidents, general outline of the accident data and accident measurement rates introduced.

In chapter three, computer program languages, tools and system requirement for server that are used to create occupational health and safety management tool outlined.

The working mechanism of computer programs within database is given in the fourth chapter.

The basis of the created standard accident data format for collection of data within database is introduced in chapter five. Detailed information of this chapter is given in Appendix A.

In chapter six, usage and capabilities of the management tool is demonstrated.

Finally, conclusions derived and discussions on developed system are given in chapter seven. Recommendations for future works are also included in this chapter.

CHAPTER II

MINE ACCIDENT ANALYSES

2.1. Overview

As a word, accident is defined as “an unexpected happening causing loss or injury” “an unforeseen and unplanned event or circumstance” (Merriam Webster Dictionary). Accident is an undesired event that results in a certain length of disability or stoppage of work or time loss due to the effect of a production related dangerous factor or a combination of such factors. Accidents may cause death, injury and loss of material (Guyaguler T., 2000). According to Mine Safety and Health Administration U.S. Department of Labor (MSHA), an accident is any unplanned event that results in personal injury or in property damage. When the personal injury requires little or no treatment, it is minor. If it results in a fatality or in a permanent total, permanent partial, or temporary total (lost-time) disability, it is serious. Similarly, property damage may be minor or serious. All accidents should be investigated regardless of the extent of injury or damage. Sometimes, the meaning is specified in detail considering insurance expenses. The definition of an accident used in WAAS (World Aircraft Accident Summary) is that; any event which causes death or serious injury and/or substantial damage to the aircraft from the time the first person boards with intention of flight until the last person leaves after flight. Airclaims Limited has excluded: non-operational accidents such as hangar fires, damage contained entirely within an engine and deaths and injuries caused by slips and falls, food poisoning, onboard machinery etc.

In Turkey, accident is defined in the 11th article of Social Insurance Law, numbered as Law 501, as an incident which related with harmful effect to the worker immediately or later, physically or mentally. Occupational injuries are defined as all employment-related injuries which are the result of a single traumatic event occurring while a person is on duty or during a recess period and where there was a short or non-existent latency period. So, the injury is an external damage to the human body resulted from an accident.

Occupational diseases are defined as all employment-related diseases which result from repeated or long-term exposure to an agent(s) or event(s) or which are the result of a single traumatic event where there was a long latency period (for example, the development of deafness following a continuous exposure to the sound). Briefly, the occupational disease is an unhealthy condition caused to person by exposure to unsafe working conditions.

Occupational disease is defined by 11th article of Social Insurance Law as the workers temporary or permanent illness, amputation or mental disorder caused by the repeated condition of the work, procedure of the work or the environment.

The occupational poisoning is partial case of occupational disease. Continuous and long exposure to relatively small amounts of poisonous substances may cause to a chronic poisoning. Instant poisoning called the acute intoxication is caused by penetration into the body of a large amount of noxious substances. This type of poisoning should be reported as an accident.

The term “reportable injury” as defined by MSHA includes all incidents that require medical treatment or result in death, loss of consciousness, inability to perform all job duties, or temporary assignment or transfer to another job.

Information reported mainly includes titles like; demographics of the injured or ill worker such as age, sex, years of total mining experience, years of experience at current mine, where the incident occurred (i.e., underground, surface, plant/mill),

days away from work, days of restricted work activity, source of the injury, body part(s) injured, and a narrative description of the incident.

According to these definitions, it is possible to prepare an accident report form. Standard report form is necessary for the statistical and economic analyses that can be used as an on-line database.

Considering all of these, there is a need to prepare safety data in a common format to produce a common safety reporting system.

2.2. Accident Occurrence

Often people interchange accident (event) and injury (consequence). However, not every accident leads to injury as given in Fig 2.1.

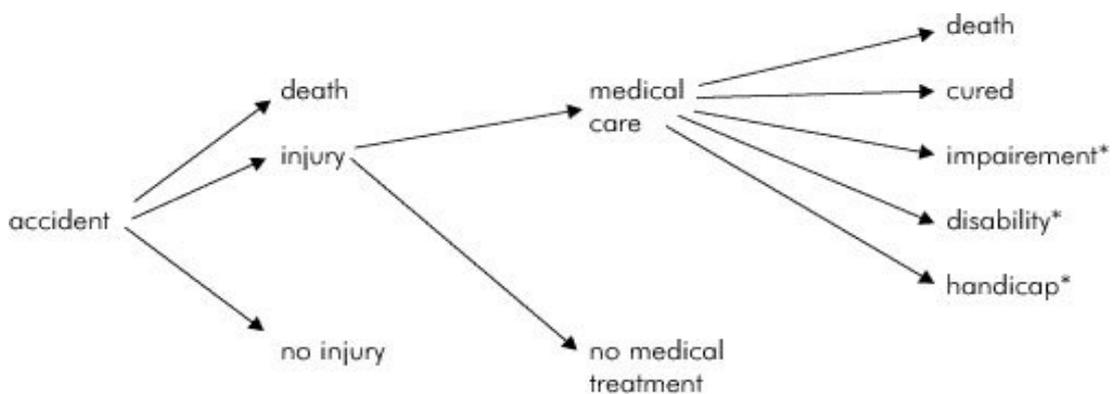
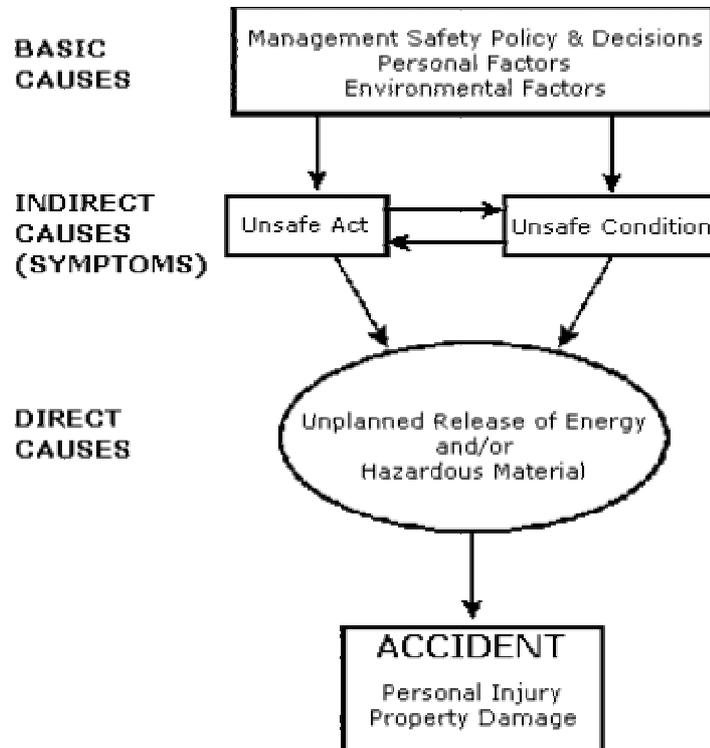


Figure 2.1. Accident and possible consequences (MSHA, 1999)

Accidents are usually complex. An accident may have 10 or more events that can be causes. A detailed analysis of an accident will normally reveal three cause levels: basic, indirect, and direct as given in Figure 2.2.

At the lowest level, an accident results only when a person or object receives an amount of energy or hazardous material that cannot be absorbed safely. This energy or hazardous material is the direct cause of the accident. The direct cause is usually

the result of one or more unsafe acts or unsafe conditions, or both. Unsafe acts and conditions are the indirect causes or symptoms. In turn, indirect causes are usually traceable to poor management policies and decisions, or to personal or environmental factors named as basic causes (MSHA).



A detailed analysis of an accident will normally reveal three cause levels: basic, indirect, and direct.

Fig 2.2. Accident investigation (MSHA, 1999)

2.3. Legal Framework

Health and safety law is governed by a framework of acts, regulations and support material including codes of practice and standards, as illustrated in the Figure 2.3.

An act or statute is law made by parliament and enforced by government departments. Health and Safety Acts set out legal rules that govern workplaces to ensure persons in workplaces do not suffer injury or illness. Health and Safety acts should be applied at any place where people are required to work. A workplace is not necessarily a building. It can be a factory or vehicle or anywhere else defined by the Act as a workplace and can include outdoors. Small businesses are not exempt from health and safety law.

A Regulation is made under the principal acts governing occupational health and safety legislation. Regulations support a principal act by outlining how the general obligations of an act will be applied in a workplace. Regulations are usually made in relation to a particular type of health and safety issue, such as asbestos, first aid, or a dangerous chemical. Not complying with a regulation is considered an offence and can result in a fine, issuing of an improvement or prohibition notice or imprisonment.

State and territory governments are able to approve codes of practice through the powers of the principal occupational health and safety act. They are supporting material additional to acts and regulations. They give practical advice and guidance on how to comply with the general obligations set out in the act and regulations. Sometimes the guidance provided in a code may only cover specific areas of the principal act or regulation.

There are two main sources of standards relevant to health and safety: National standards produced by the Ministry of Labor and Social Security in consultation with the state/territory health and safety authorities, employee unions and employer associations. Turkish Standards produced by Turkish Standards Institution (TSE), a non-government, non-profit organisation that makes standards in consultation with overseas standards bodies and Turkish working parties. The standard is OHSAS 18001 (Occupational Health and Safety Administration System).

2.4. System Dynamics of the Mine Accidents

Human beings are quick problem solvers. From an evolutionary standpoint, this makes sense. We quickly determine a cause for any event that we think is a problem. Usually we conclude that the cause is another event. For example, if sales are poor (the event that is a problem), then we may conclude that this is because the sales force is insufficiently motivated (the event that is the cause of the problem). This approach works well for simple problems, however, it does not work if the problems get more complex, for example an accident case involving many parameters.



Figure 2.3. Legislative framework (Nohsc, 1998)

System dynamics is a methodology for studying and managing complex feedback systems, as found in business and other social systems. In fact it has been used to address practically every sort of feedback system. While the word system has been applied to all sorts of situations, feedback is the differentiating descriptor here. Feedback refers to the situation of X affecting Y and Y in turn affecting X perhaps through a chain of causes and effects. One cannot study the link between X and Y and, independently, the link between Y and X and predict how the system will behave.

The methodology;

- Identifies a problem,
- Develops a dynamic hypothesis explaining the cause of the problem,
- Builds a computer simulation model of the system at the root of the problem,
- Tests the model to be certain that it reproduces the behavior seen in the real world,
- Devises and tests in the model alternative policies that alleviate the problem, and
- Implements solution.

System dynamics encompasses work in;

- Corporate planning and policy design,
- Public management and policy,
- Biological and medical modeling,
- Energy and the environment,
- Theory development in the natural and social sciences,
- Dynamic decision making,
- Complex nonlinear dynamics.

In spite of their complexity, most accidents are preventable by eliminating one or more causes. Accident investigations determine not only what happened, but also how and why. The information gained from these investigations can prevent recurrence of similar or perhaps more disastrous accidents. Accident investigators are interested in each event as well as in the sequence of events that led to an accident. The accident type is also important to the investigator. The recurrence of accidents of a particular type or those with common causes shows areas needing special accident prevention emphasis.

A number of factors are taken into account to determine what would be reasonable and practical. These factors include the:

- Nature and severity of the hazard;
- Knowledge of severity of the hazard;
- Knowledge of solutions;
- Availability of solutions;
- Common standards of practice;
- Cost of solutions.

In summary, employers, manufacturers, designers, suppliers, persons in control of workplaces and persons who erect or install plant and equipment must ensure:

- Safe property, which includes premises (safe access and exit), safe plant and equipment, materials and substances (raw materials, chemicals, products, stock etc).

- Safe systems of work, which includes your work practices, manufacturing processes, standard operating procedures and your administration procedures.
- Safe people, which includes providing them with suitable information, instruction, training and supervision. For example the safe use of plant and equipment, materials and chemicals and information about the working environment and health and safety generally.

So, if the duties is thought as pieces of a jigsaw puzzle as given in Figure 2.4. and Figure 2.5, it's not complete if a piece is missing.

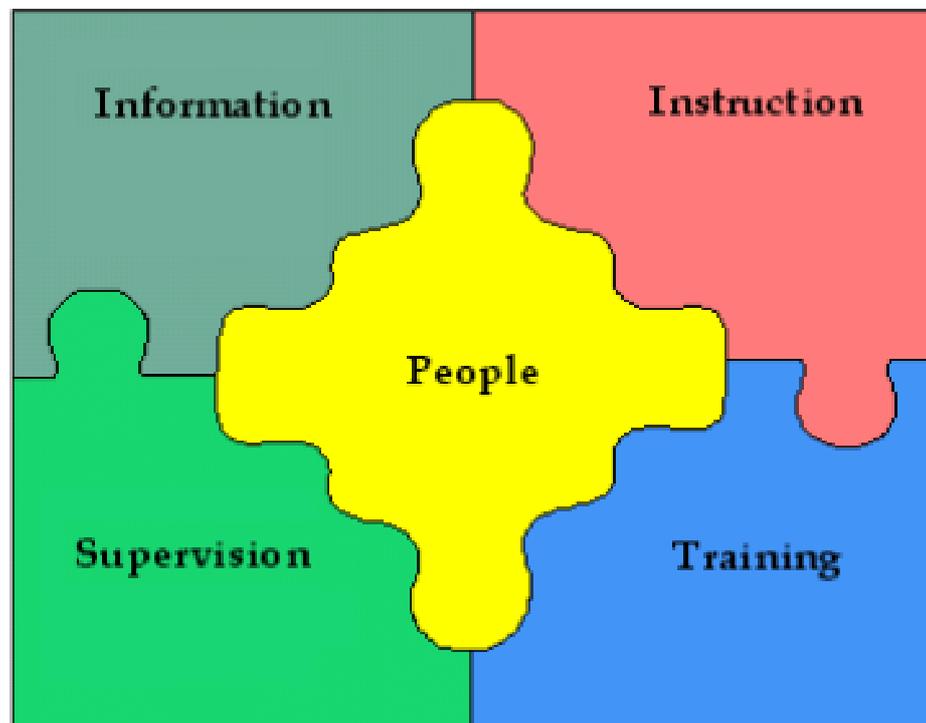


Figure 2.4. Safety framework (Nohsc, 1999)

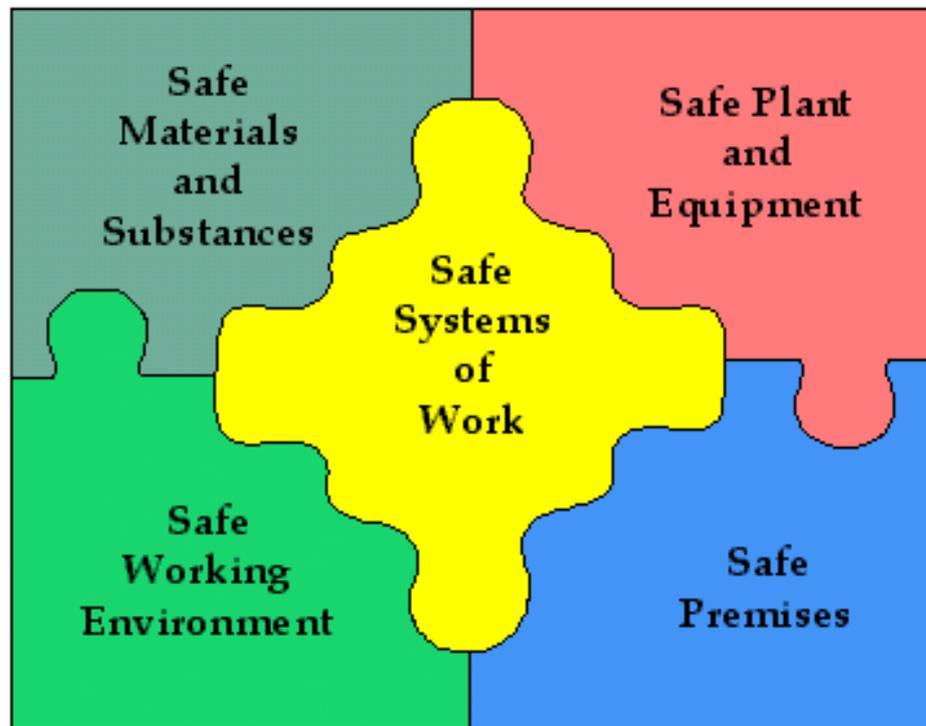


Figure 2.5. Safety parameters (Nohsc, 1999)

Environmental factors, which influence people's ability to work effectively and safely, are composed of the factors of physical environment, social and economical environment, biological environment and psychological environment. Therefore, positive progress for any of these environments is reflected to health and safety of the worker in a positive manner.

Training and education equip people to work safely due to the fact that education increases the awareness of occupational diseases and accidents and the importance of to create safe and healthy environment in working sites. Education also creates better personal characteristics, which help to create healthy environment.

It is a known fact that when the number of occupational disease and mine accidents are decreased, production increases. Therefore efficiency and profit is directly related with health and safety standards, which consequently will result in creating better environment factors and better personal characteristics.

Above all of these, Occupational Health and Safety Management Tool for Mines has the capability and capacity to influence education, laws, regulations, instructions, agreements and standards, and personal characteristics directly. So within this kind of dynamic environment, containing many parameters affecting each other, even small changes may be resulted in much more better conditions.

Generally, in system dynamics, 20% of the parameters create 80% of the problems. So, precautions for that 20% of parameters should be taken as soon as possible. With this approach, OHSMT will be a good tool to minimize and solve problems related with occupational health and safety.

“System Dynamics of the Occupational Health and Safety System” is prepared with this approach and given in Figure 2.6. It shows how decisions taken according to the results of the program could interact with the accident cause variables. Feedbacks in this figure show the importance of application of the Management Tool.

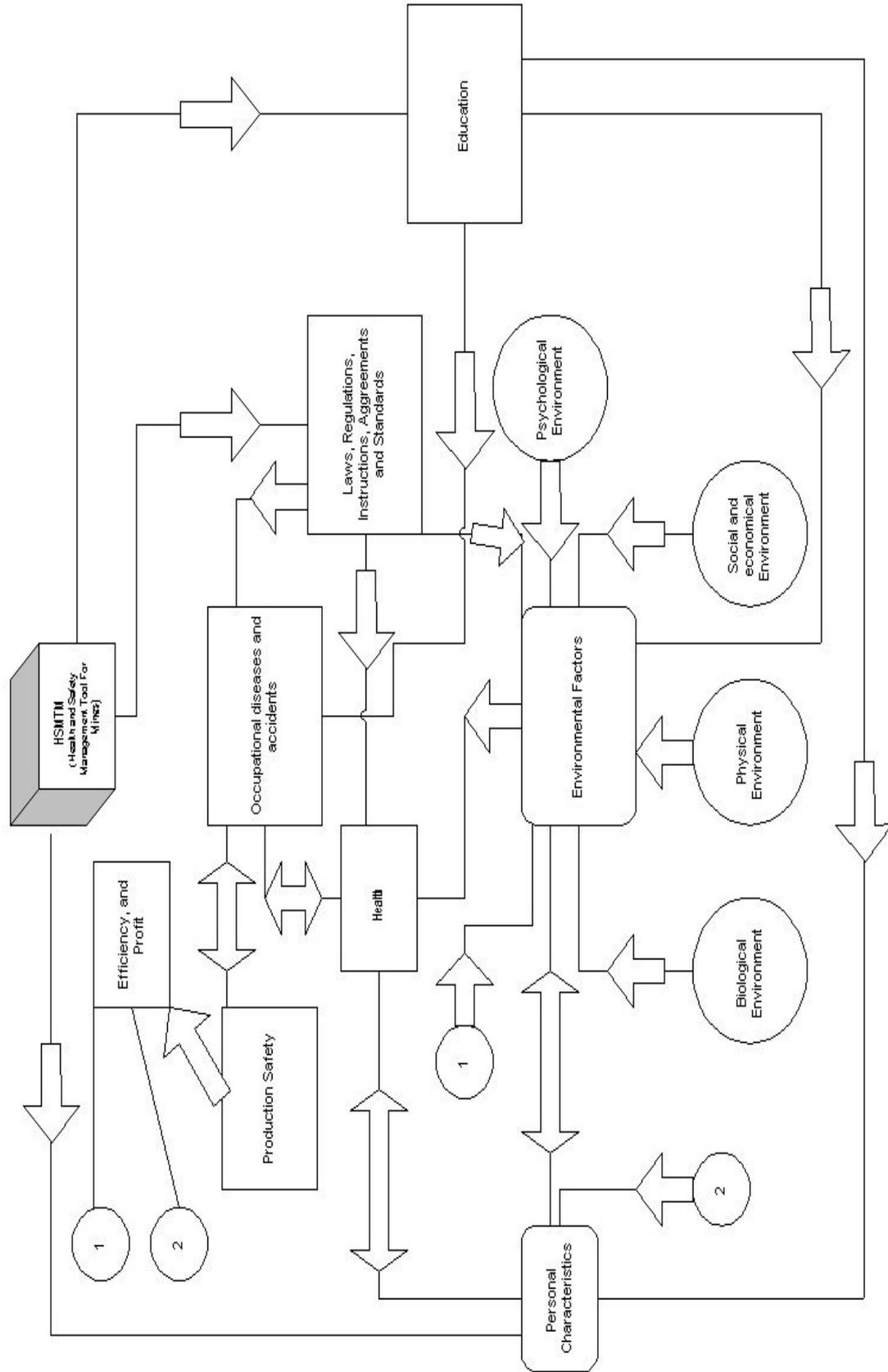


Figure 2.6. System dynamics of the occupational health and safety system

2.5. Cost of the Accidents

It has been noted that there are two major categories of cost resulting from accidents, usually referred as direct and indirect costs (Guyaguler T., 2000).

The direct cost of the accidents consists of:

- Cash expenses payable under the related laws,
- Medical benefits,
- Disablement expenses payable under the social insurance scheme,
- Other benefits payable under the company's own scheme.

The indirect costs of mine accidents can be outlined as follows:

- Cost of lost time of injured employee.
- Cost of time lost by other employees who stop work out of curiosity, out of sympathy, to assist injured employee, and for other reasons.
- Cost of time lost by foremen, supervisors, or other executives when assisting the injured person(s), investigating the cause of accident, arranging for the injured employee's production to be continued by some other worker, selecting, training or breaking-in a new worker to replace the injured man, and preparing accident reports required by law, or attending hearings related to compensation claims.
- Cost of time spent by the first-aid attendant and hospital department staff.
- Cost due to damage to machine, tools or other property or to the spoilage of material.
- Incidental cost due to interference with production, failure to fulfill orders in time, payment forfeits, and other similar causes.
- Cost of employer in continuing the wages of the injured worker who is not yet fully recovered may, for a time, be worth only about half of the normal value.
- Cost due to loss of profit or the injured worker's efficiency and production.

- Cost of the subsequent injuries that occurs in consequence of the excitement or weakened morale due to the original accident.
- Cost of light, rent and other such items, which continue to be incurred while the injured employee continues to be non-productive.

Mining having the highest accident ratios in industry creates the highest accident costs. It has been noted (Handley W., 1977) that in order to determine the cost of the accidents, it is necessary to obtain reliable data. If there is not sufficient data, the statistical results will not be true. In some cases the indirect cost of the accidents may be four times of the direct cost (Heinrich H. W., 1980).

The Australian Bureau of Statistics (ABS) National Accounts data estimate the direct cost of workers' compensation in 1992-1993 as \$4.83 billion. Using the most conservative overall measure of 1:1, as the indirect to direct cost ratio and factoring in under coverage, total costs translate into \$8 billion per year or, using a frequently quoted compromise measure of 4:1, total annual costs could range up to \$20 billion. Unfortunately, exact costs are not available from Bureau of Labor Statistics, U.S. Department of Labor (BLS) injuries, illnesses, and fatalities statistical program.

Direct costs can be calculated from standard accident report form with the cost models (Guyaguler T., 2000). So, the indirect costs can also be calculated with the compromise measure of 4:1.

Non-fatal Accident Cost Model is given below:

$$\text{Cost} = [L_{SN}, (b_S(100) + D (b_{SNM} + b_{SND})/5] \quad (2.1)$$

where:

L_{SN} = the maximum benefit limit for non-fatal injury

b_S = 1 time benefit loss of member in amputation (0 for all other injuries)

D = number of qualified work days on disability.

b_{SNM} = the basic weekly swcb (state worker compensation benefit)

b_{snd} = the incremental benefit for dependants

Fatal Accident Cost Model is given below:

$$\text{Cost} = [L_{sf} \cdot (b_s(LS)) + W_e (b_{sfm} + b_{sfd})] \quad (2.2)$$

where:

$$W_e = 52 (65 - Y_m)$$

L_{sf} = the maximum swcb (state worker compensation benefit) for fatality.

$b_s(LS)$ = one time burial benefit

b_{sfm} = maximum weekly benefit to surviving spouse (if married)

b_{sfd} = maximum weekly benefit to children

W_e = remaining expected work life of miner (weeks)

Y_m = the age of the miner at the time of accident.

Cost of the accidents shows the effectiveness of the accident prevention program. Required fields to perform cost calculations are being prepared for OHSMT. There is a relationship between the money spent for safety and cost of the accident as given in Figure 2.7 to prevent or reduce the accident or occupational diseases and cost of accident. Output from OHSMT program may help management in the determination of optimum expenditure to reduce accident and occupational diseases.

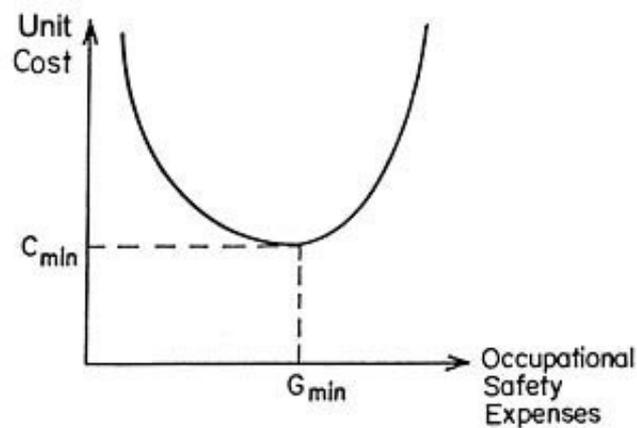


Figure 2.7. Relationship between the safety expenses and cost of the accident

2.6. Standard Format

It is important to prepare comprehensive coding guidelines and rules; otherwise it is impossible to provide comprehensive and accurate accident data. The data should include all details of the circumstances surrounding the accident or injury so that all information necessary to make conclusions and find out statistical data is available. So, all reported cases involve straightforward coding. If there is some uncertainty in assigning the most appropriate code to any field, it should not be coded. Procedure that should be followed in such a case is discussed in coding part, Chapter III.

An example given below gives an idea about the differences between the safety data in common and uncommon format. It can be recognized that, the result of the accident may be concluded completely different from the real case if there is no standard format for accident data.

This accident, which happened at a mine in South Africa resulted in the death of a newly recruited fitter, who was struck by a moving train. It is used as an example to highlight the difficulties detailed above. Two very different conclusions were made. One as a result of an inquiry held by the state, and one as a result of an investigation carried out by an independent investigation team, under the guise of legal privilege.

2.6.1. The Inquiry

The locomotive driver and guard were two key witnesses to the accident. No one else was on haulage road at that time. The locomotive was pushing a man-carriage. This is against company procedure. The guard stated that he sat in the man-carriage, diligently looking forward, shining his cap lamp in front of the train. When the train approached the cubby, the driver as a witness stated that the guard blew his pea-whistle as a precaution, but neither of them saw the deceased in the cubby.

During the inquiry, the locomotive driver stated that he did not see the deceased at all, until he appeared from a small cubby and walked straight into the path of the

moving train that is too late for the driver to stop before striking the deceased. Tests and simulations showed that the loco was traveling relatively slowly, as was stated by the driver, physical inspection of the site revealed poor footwall conditions for both side of the rail tracks. It was suggested that this was the prime reason that caused the deceased to stumble in front of the moving train, after exiting the sidewall cubby. The deceased had only been working on the mine for approximately two weeks, and had limited underground experience.

The main conclusion drawn at the end of the inquiry process, was that the deceased, unfamiliar with underground operations, was unaware of the hazards of moving rolling stock, and as a result had inadvertently stumbled in front of the moving train. Poor footwall conditions had contributed to the accident. It was found that the loco driver and guard, whilst in breach of company procedure, had acted reasonably, and had taken reasonable precautions under the circumstances. Disciplinary action was taken against the loco crew by the mine involved.

Recommendations were made which largely concentrated on improving the mine's workplace induction procedures for new employees, and the improvement of conditions underfoot in this area

2.6.2. The Investigation

The picture painted following a comprehensive investigation was quite different. The inquiry process had been focused on discovering breaches, and had missed some pertinent facts. When given indemnity from having their statements used in evidence against them, the witnesses told quite a different story:

It seems that the loco guard was simply traveling in the man-carriage, and was not "diligently looking forward, shining his cap lamp in front of the train". The view of the driver was almost totally obscured by the leading man-carriage. The fitter was simply walking away from the oncoming train, along the railway sleepers, as a lot of us have done many times. He was simply run over by the train.

Investigations showed that the cooling unit in the cubby was un-silenced, and at the time of the accident was operating at noise levels over 108 dB. It seems that even if the guard had blown his pea-whistle, no one could have heard it, let alone the deceased having any audible warning of the locomotive's presence. The light on the front of the train was measured at 2 lux at 10 meters, thereby negating any visual warning. This was due to intermittent problems with the charging of loco lamps in the lamp room.

The investigation concluded that the accident had occurred as a result of a number of key factors, including conditions in respect of the occupational environment.

Recommendations were made about issues such as: Fan silencing, improvements to the planned maintenance system in the lamp room, signaling systems in haulages where pedestrian and locomotive traffic co-exist, etc.

In conflict with the Inquiry, the investigation did not find out that the workplace induction program was deficient, and most importantly, it did not seek to apportion blame, or recommend disciplinary action. In terms of the recommendations however, it does seek to assign accountabilities for remedial measures.

2.7. General Information about the Accident Coding System

There are many items for an accident to be coded. As described, one of the aim of coding is to clarify the situation. Besides, on database point of view, coding system has got many advantages. First of all, when coding system is used on database, users do not have to know the code given for each case. He will be choosing the appropriate options that will be converted to codes by the program. This makes database much more user friendly.

The other benefit is the ease of making statistical conclusions using already coded data. For example, when the user input sex of the person subjected to the accident, he will just make his choice between the "male" and "female" buttons. User does not

have to know the code that will be used for sex in database. If the user chooses male as an accident person's gender, this data will be input as "m" in gender field. Then it will be very easy to count how many "m" exists in gender field to make comparisons between the genders accident ratio for a given period.

Another benefit of the coding will be the speed of the database. It is much more speedy to count "1" instead of "male" in database. Also, the database will be smaller which also increases the speed of the database.

The objective of the database is to assist in the prevention of occupational accidents by the production of uniform national and nationally comparable indicators of occupational health and safety data.

2.8. Data Analysis

There are some items in database, which are not going to be analyzed. These are the data to be used to clarify the situation. For example, it may be meaningless to make any analysis over "witness name". This is just complementary information for the accident. So the data that are not going to be analyzed are recorded on a different database part to make sure that the analyzing process will be faster.

Data could be analyzed in a variety of ways by a number of different users. Some of the more important types of analysis are:

- Causal factors, to analyse the cause of the accident,
- Occurrence time, to take some extra precautions in dangerous shifts,
- Company name, to enable investigation for distribution of occupational injury/disease for each company.
- Number of employees working, to enable investigation of size of business factors on occupational injury/disease experience, etc.

Analysis of the type of occurrence data items, that is, nature of injury/disease, bodily location of injury/disease, mechanism of injury/disease, and agency helps in the

identification of the nature and causes of the problem, enabling research, resources and risk reduction strategies to be better targeted.

Analysis that could be performed are explained in chapter six in detail.

2.9. Calculation of Measurement Rates

Various accident statistics are used to determine trends when evaluating an accident prevention program.

Unless the total hours of exposure or tonnage remain constant from one time interval (say one year) to the next, total number of accidents, injuries and fatalities can be misleading. For example, a mine with 100 accidents in 2003 and 150 accident in 2004 may still have an effective accident prevention program if the number of employees or the amount of production doubled in 2004.

Therefore, trends are usually evaluated in terms of accidents and injuries per man-shift, per million man-hours, per million tons of material moved etc. (Guyaguler T., 2000). One of them is the Disabling Injury Frequency Rate (DIFR) that is given by the equation;

$$D.I.F.R. = \frac{(\text{Number of disabling injuries}) \times (1.000.000)}{\text{Hours of exposure (or tonnage)}} \quad (2.3)$$

Another trend tool is the Fatality Injury Frequency Rate (FIFR) which is given by the equation;

$$F.I.F.R. = \frac{(\text{Number of fatalities}) \times (1.000.000)}{\text{Hours of exposure (or tonnage)}} \quad (2.4)$$

Accident frequency rate stands for number of accidents per hour for each million worker and calculated from the formula given below:

$$\text{A.F.R.} = \frac{(\text{Total Number of Accidents}) \times (1.000.000)}{(\text{Number of workers}) \times (\text{Average working period})} \quad (2.5)$$

To have better idea of the situation the Accident Severity Rate (ASR) also has to be calculated. ASR is the total number of days lost per thousand man-hours of exposure. Theoretically, for each fatality case 7500 days are considered as days lost.

$$\text{A.S.R.} = \frac{(\text{loss of working days due to accidents}) \times (1.000)}{(\text{Number of workers}) \times (\text{Average working period})} \quad (2.6)$$

Accident frequency rates, together with accident severity rates can be used as comparative standards for safety measurements either between the years in the same mine or between the mines in the same company in the same period and between the company and the industry average in the same period.

2.10. Risk Analysis

Risk Analysis is “a systematic use of available information to determine how often specified events may occur and what is the magnitude of their consequences”. The reporting of accidents and high potential incidents is an important step in the investigation, analysis and ultimately the future prevention of further occurrence of such accidents and high potential incidents, thus contributing to improvements in mine safety.

The detailed analysis of an accident requires knowledge of many factors such as location, time, type, cost of accident, victim, nature of the injury, personal and environmental factors etc. that can be obtained from the standard coded accident report form.

A complete study of accident trends also includes a study of causes, activity at the time of the accident, location, job experience, occupation of injured, years at the mine, and time of day. These data can be used to evaluate hazards, prepare job safety guides, formulate new policies, redesign equipment, modify operating procedures and develop training programs.

All risk analyses follow a general scheme that can be described as follows:

- Describe the system under analysis (including equipment, personnel, procedures, work environment, management and supervisory systems etc.)
- Identify loss scenarios (i.e. sequences of events leading up to potential or actual losses i.e. incidents or accidents) in the form of hazards, potential productivity interruptions, asset damage events, environmental issues etc.
- Evaluate the risks of each loss scenario by determining the relative likelihood of each event, and the relative consequence of each event.
- Evaluate the currently planned controls, barriers and safeguards.
- Identify additional, potential controls, barriers and safeguards.

It is possible to perform risk analyses for the management with the data supplied from OHSMT database prepared in this thesis

CHAPTER III

COMPUTER PROGRAMMING AND TOOLS USED TO CREATE OHSMT

3.1. Database

Database systems store information in every conceivable business environment. From large tracking databases such as airline reservation systems to a child's baseball card collection, database systems store and distribute the data that we depend on. Until the last few years, large database systems could be run only on large mainframe computers. These machines have traditionally been expensive to design, purchase, and maintain. However, today's generation of powerful, inexpensive computers enables programmers to design software that maintains and distributes data quickly and inexpensively (Amundsen, M. et al., 1996).

3.2. Relational Database Model

A relational Database Management System (DBMS) must be able to manage databases entirely through its relational capabilities in order to satisfy the "Management Tool" computer program requirements:

- Information rule: All information in a relational database (including table and column names) is represented explicitly as values in tables.
- Guaranteed access: Every value in a relational database is guaranteed to be accessible by using a combination of the table name, primary key value, and column name.

- Systematic null value support: The DBMS provides systematic support for the treatment of null values (unknown or inapplicable data), distinct from default values, and independent of any domain.
- Active, on-line relational catalogue: The description of the database and its contents are represented at the logical level as tables and can therefore be queried using the database language.
- Comprehensive data language: At least one supported language must have a well-defined syntax and be comprehensive. It must support data definition, manipulation, integrity rules, authorization, and transactions.
- View updating rule: All views that are theoretically updateable can be updated through the system.
- Physical data independence: Application programs and ad hoc programs are logically unaffected when physical access methods or storage structures are altered.
- Logical data independence: Application programs and ad hoc programs are logically unaffected, to the extent possible, when changes are made to the table structures.
- Integrity independence: The database language must be capable of defining integrity rules. They must be stored in the on-line catalogue, and they cannot be bypassed.
- Distribution independence: Application programs and ad hoc requests are logically unaffected when data is first distributed or when it is redistributed.

"Management Tool" computer program includes both standard database and relational database. Standard database logic is going to be used in retrieveing and sharing accident data, whereas, relational database is used in matching accidents with precautions and legislation databases.

3.3. Perl

As a programming language, PERL (Practical Extraction and Report Language) was chosen and used. Because PERL is efficient, flexible, extensible, and easy to maintain when programming a wide range of tasks, in particular those involving the

manipulation of text files. Another advantage is its security. Security is a major issue when writing system administrative programs and on the Internet in general. Using PERL for scripting on your Web server, you can easily guard against users trying to sneak commands through for the server to execute on their behalf. Another facet of PERL that is of particular interest to many Web server managers is that PERL works very well with standard UNIX DBM files and support for proprietary databases is growing. This is a significant consideration if you plan to allow users to query database material over the Web (Wall, L. et al., 2000).

3.4. Common Gateway Interface

The Common Gateway Interface (CGI) is an accepted standard for interfacing Web servers and external applications. Web servers were originally designed to serve static HTML documents along with other associated static files. A Web browser that communicates with a Web server that limits its functionality to serving static pages displays only documents whose contents will not change between requests or during page visualization.

A Web server is generally installed on a powerful computer, and it would be very frustrating not to be able to offer many more interesting and dynamic things to remote users, using the computer power available. The CGI specifications were created to answer this problem. CGI establishes a standard way of information exchange between Web servers and browsers (also called clients). It allows the passing of information between a browser or server to an external program that performs some actions and then outputs its results back to the user's browser. The external program is generally known as the CGI program, CGI script, CGI application, or simply gateway, because it makes use of the CGI specification and is specially designed for functioning on a Web platform as given in Figure 3.1. It is executed in real-time, by initiative of the user even if sometimes nothing is noticed, and it can output dynamic on-the-fly information.

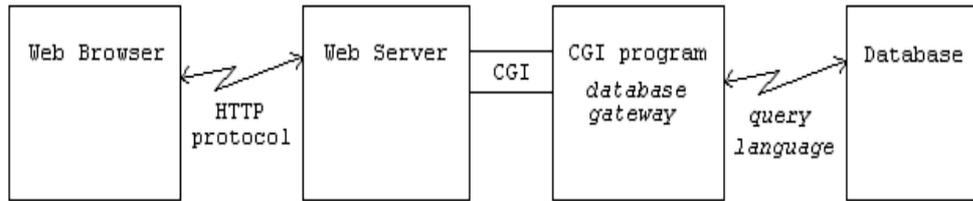


Figure 3.1. Accessing a database using a web browser

CGI is an interface specification. It does not define how a Web server works or how a program is expected to produce results, but it establishes a set of guidelines that both must follow in order to inter-operate.

The CGI specification is implemented on Web servers, as well as on programs built for use over the Web. It is not part of the HTTP, but most Web servers choose to implement this useful feature. Therefore, you are able to use CGI applications in most known Web servers, including NCSA httpd, CERN httpd, Apache httpd, and many other commercial servers. For this "Management Tool" computer program "Apache httpd" was chosen and installed on a server. Apache Web Server is free of charge. Also, CGI Star is installed as an interpreter program on the same server. Both of them are free of charge shareware programs (Christiansen, T. et al., 1998).

3.5. World Wide Web

The World Wide Web (www) is a vast collection of information that is spread across hundreds of thousands of computers around the world. When you access a document on the www, there's a lot going on behind the scenes. Very simple and brief description is "The www is a network of thousands of computers, all of which fall neatly into two categories: *clients* and *servers*. Through the use of special software, they form a kind of network called a *client-server network*" (Arpajian,S. et al., 1996).

Servers store information and process requests from clients. Then they send the requested information to the clients. This information covers all kinds of data, including images, sounds, and text. Servers also send instructions to the client on

how to display all this information. These instructions are sent in the form of Hypertext Mark-up Language (HTML).

Clients make requests for information and then handle the chore of displaying that information to the end user. When using a Web browser to navigate the www, browsing software is acting as a client. Users navigate the World Wide Web through the use of hypertext links. When you select or click on a hypertext link, you go to another area on the Internet. Almost all of the documents on the Web are interconnected through the use of hypertext links. HTML provides instructions for the client software on how the document should be displayed. HTML also contains information about how to link up to other documents on the Web. Hypermedia are hypertext systems which use other formats in addition to text and much more efficient and effective for info sharing: Images- gif, jpeg, Video- mpeg, avi, asf, wmv, quicktime, Audio- wav, mp3, etc. Media is shared through ftp (file transfer protocol) server.

3.6. Fundamentals of SQL Server Architecture

Microsoft® SQL Server™ is a Structured Query Language (SQL) based, client/server relational database. Each of these terms describes a fundamental part of the architecture of SQL Server (Delaney, K., 2000).

It is also a full-featured relational database management system (RDBMS) that offers a variety of administrative tools to ease the burdens of database development, maintenance and administration.

A database is similar to a data file in that it is a storage place for data. Like a data file, a database does not present information directly to a user; the user runs an application that accesses data from the database and presents it to the user in an understandable format.

Database systems are more powerful than data files. The data is more highly organized. In a well-designed database, there are no duplicate pieces of data that the user or application has to update at the same time. Related pieces of data are grouped

together in a single structure or record, and relationships can be defined between these structures and records (Henderson, K., 2000).

When working with data files, an application must be coded to work with the specific structure of each data file. In contrast, a database contains a catalog that applications use to determine how data is organized. Generic database applications can use the catalog to present users with data from different databases dynamically, without being tied to a specific data format.

Server applications are usually capable of working with several clients at the same time. SQL Server can work with thousands of client applications simultaneously. The server has features to prevent the logical problems that occur if a user tries to read or modify data currently being used by others (Jacobson, R., 2000).

In a client/server database architecture, the database files and DBMS software reside on a server. A communications component is provided so applications can run on separate clients and communicate to the database server over a network. The SQL Server communication component also allows communication between an application running on the server and SQL Server.

While SQL Server is designed to work as a server in a client/server network, it is also capable of working as a stand-alone database directly on the client. The scalability and ease-of-use features of SQL Server allow it to work efficiently on a client without consuming too many resources.

3.7. Structured Query Language (SQL)

To work with data in a database, you must use a set of commands and statements (language) defined by the DBMS software. There are several different languages that can be used with relational databases; the most common is SQL. Standards for SQL have been defined by both the American National Standards Institute (ANSI) and the International Standards Organization (ISO).

Tables are database objects that contain all the data in a database. A table definition is a collection of columns. In tables, data is organized in a row-and-column format similar to a spreadsheet. Each row represents a unique record, and each column represents a field within the record (Andrews, C. et al., 2003). For example, as given in Figure 3.2, table contains company data for a company contain a row for each company and columns representing company details such as name of the company, address of the company, number of workers for underground and open pit.

company_id	company_name	company_address	workers_undergrou	workers_openpit
1	Administrator	Ankara	1	1
2	Etibank	Istanbul	50	150
3	TTK	Izmir	100	90
4	Soda Sanayii	Mersin	0	150
6	TKI	Ankara	250	250
10	MTA	Ankara	100	150
*				

Figure 3. 2. Company data in SQL table

3.8. System Requirements

Microsoft SQL Server 2000 requires the following minimum system configuration:

Processor: Intel Pentium or compatible 166-megahertz (MHz) or higher processor

Operating System: Windows 2000 Advanced Server

Memory: Enterprise Edition: 64 megabytes (MB) of RAM; 128 MB recommended

Hard Disk: 95–270 MB of available hard disk space for the server; 250 MB for a typical installation.

- 50 MB of available hard disk space for a minimum installation of Analysis Services; 130 MB for a typical installation.

- 80 MB of available hard disk space for query.

Drive: CD-ROM drive

Display VGA or higher-resolution monitor

Other Devices: Microsoft Internet Explorer version 5.0 or later

3.9. Required Program and Operating System Licenses

SQL Server installations that are part of your day-to-day operations require production licenses. SQL Server is currently available under three licensing options:

- Processor license: Requires a single license for each CPU in the operating system instance running SQL Server. This license does not require any device or user client access licenses (CALs).
- Server plus device CALs: Requires a license for the computer running the Microsoft server product, as well as a CAL for each client device.
- Server plus user CALs: Requires a license for the computer running the Microsoft server product, as well as a CAL for each user.

Retail pricing for “Enterprise Edition” is \$ 19,999 US.

As an operating system “Windows 2000 Advanced Server” is used (with 25 client access licenses) and retail pricing is \$ 3999 US.

3.10. SQL Server™ Authentication

Microsoft® SQL Server™ can operate in one of two security (authentication) modes:

- Windows NT authentication mode (Windows NT authentication)
- Mixed mode (Windows NT authentication and SQL Server authentication)

Mixed mode allows users to connect using Windows NT authentication or SQL Server Authentication. Users who connect through a Microsoft Windows NT® user account can make use of trusted connections (connections validated by Windows NT) in either Windows NT authentication mode or mixed mode. After successful connection to SQL Server, the security mechanism is the same for both modes.

Security systems based on SQL Server logins and passwords (SQL Server Authentication) may be easier to manage than security systems based on Windows

NT user and group accounts, especially for databases that are not mission critical and applications without sensitive and confidential information. For example, a single SQL Server login and password can be created for all users of an application, rather than creating all the necessary Windows NT user and group accounts. This does, however, remove the ability to track and control the activities of individual users and is therefore not recommended.

3.11. Windows NT and SQL Server Authentication Mixed Mode

When a user connects through a Windows NT user account, SQL Server verifies that the account name and password were validated when the user logged on to Microsoft® Windows NT as shown in Figure 3.3.

SQL Server achieves login security integration with Windows NT by using the security attributes of a network user to control login access. A user's network security attributes are established at network login time and are validated through a sophisticated password encryption mechanism. When a network user tries to connect, SQL Server uses Windows NT-based facilities to determine the validated network username. SQL Server then permits or denies login access based on that network username alone, without requiring a separate login name and password.

Login security integration operates over network protocols that support authenticated connections between clients and servers. Such connections are referred to as trusted connections and are supported by SQL Server using any available network library.

Windows NT authentication has certain benefits over SQL Server authentication, primarily due to its integration with the Windows NT security system. Windows NT security provides more features, such as secure validation and encryption of passwords, auditing, password expiration, minimum password length, and account lockout after multiple invalid login requests.

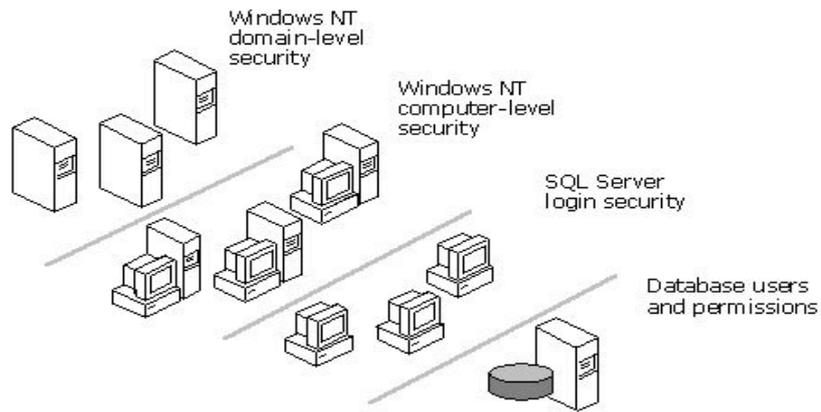


Figure 3. 3. Windows NT authentication

Since Windows NT users and groups are maintained only by Windows NT, SQL Server reads information about a user's membership in groups when the user connects. If changes are made to the accessibility rights of a connected user, the changes become effective the next time the user connects to SQL Server or logs on to Windows NT depending on the type of change.

SQL Server authentication is required when SQL Server is running. Security decision three is given in Figure 3. 4.

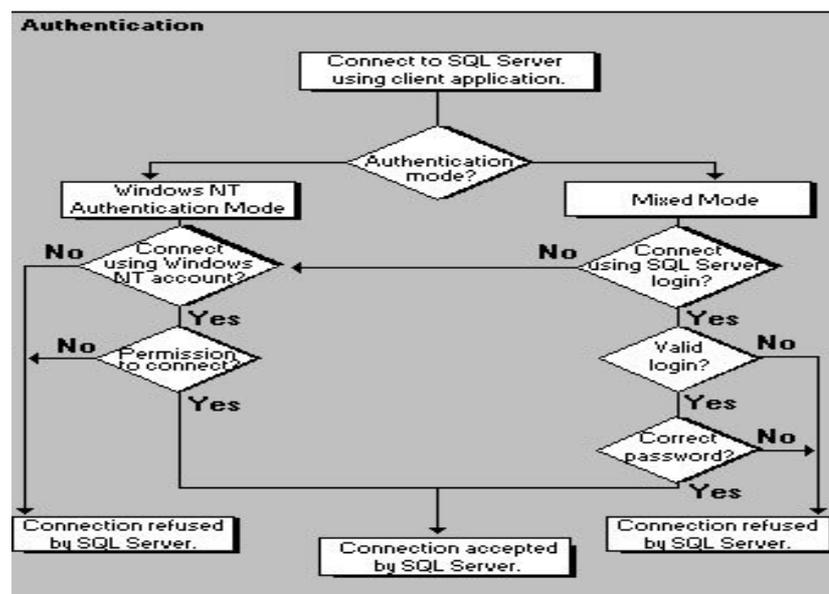


Figure 3.4. SQL Server security decision tree

CHAPTER IV

CONNECTION OF THE SYSTEM TO THE DATABASE

4.1. Overview

The detailed analysis of an accident requires knowledge of many factors such as location, time, type, cost of accident, victim, nature of the injury, personal and environmental factors etc. that can be obtained from the standard coded accident report form.

4.2. Interfaces

The system is designed to code both injuries and diseases. An injury is the result of a single traumatic event where the harm or hurt is immediately apparent, for example, a broken leg resulting from an accident of hit by a truck. A disease, on the other hand, results from repeated or long-term exposure to an agent or event, for example, loss of hearing as a result of long-term exposure to noise. Figure 4.1. shows the accident database tables prepared in SQL server. Each table has its own fields as given in Figure 4.2. A Field detail for an accident is shown in Figure 4.3.

These subtitles should include as many items as possible. Detailed coding system is used for on-line interface. But, it is not possible to give a code to each parameter at the beginning. So, accident report interface includes an administration part to add new codes. As shown Figure 4.4, web browser is connected to the database and administrator is capable of registering new mechanism field through web interface.

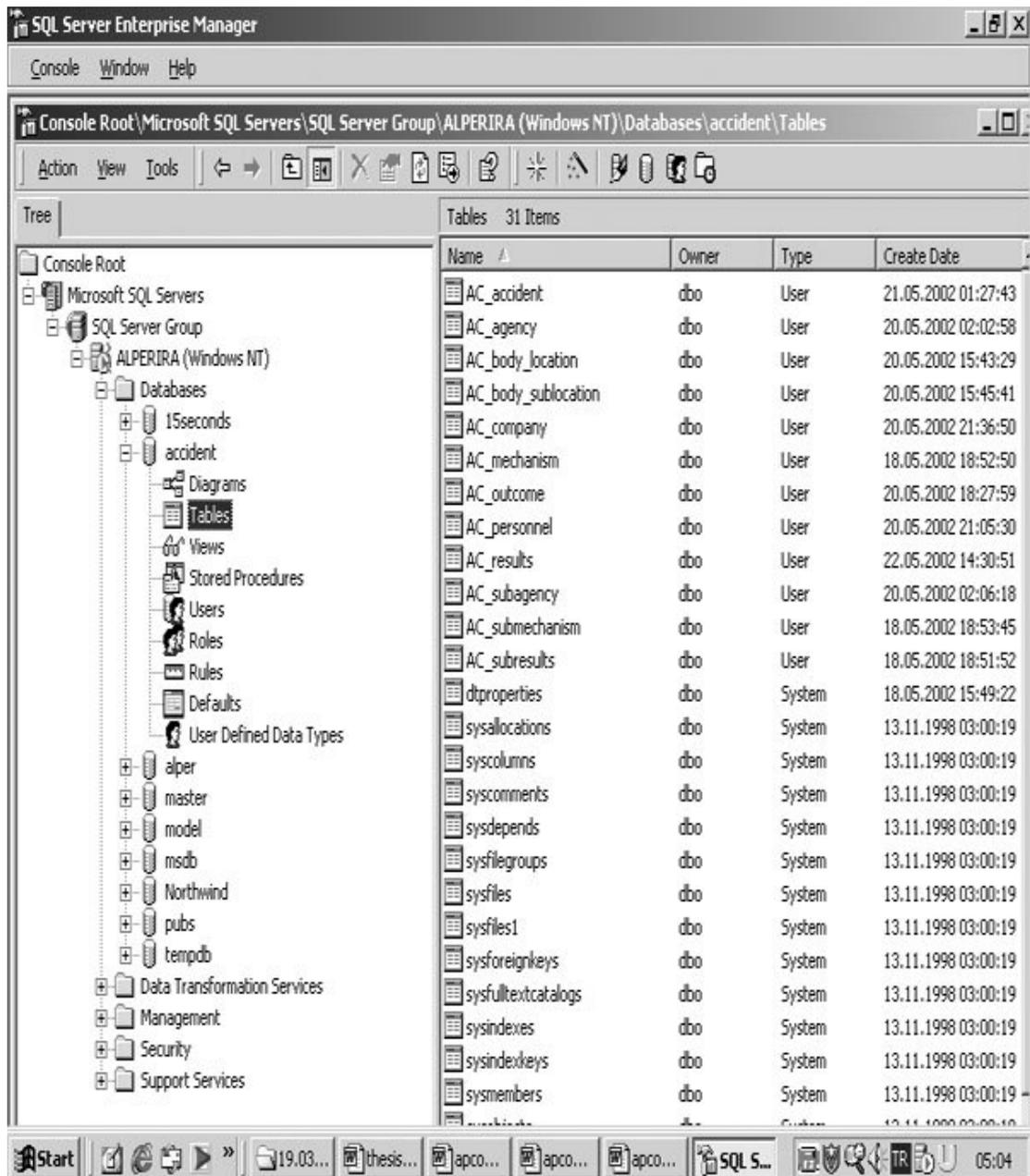


Figure 4.1. Accident database tables in SQL Server

mech_id	definition
1	Falls, trips and slips of a person
2	Hitting objects with a part of the body
3	Being hit by moving objects
4	Sound and Pressure
5	Heat and Electricity
6	Other and unspecified mechanisms
7	new mech

Figure 4.2. Accident mechanism fields

Column Name	Datatype	Length	Precision	Scale	Allow Nulls	Default Value	Identity	Identity Seed	Identity Increment	Is RowGuid
acc_id	int	4	10	0	<input type="checkbox"/>		<input checked="" type="checkbox"/>	1	1	<input type="checkbox"/>
personnel_id	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
accident_result_id	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
accident_subresult_id	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
agency_id	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
subagency_id	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
mechanism_id	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
submechanism_id	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
outcome_id	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
body_location_id	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
body_sublocation_id	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
period_unattended_d	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
replacement	char	1	0	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
description	text	16	0	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
conclusion	text	16	0	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
recommendations	text	16	0	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
job_experience_days	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
mining_experience_da	int	4	10	0	<input type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
accident_date	datetime	8	0	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
accident_time	datetime	8	0	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
shift_time_start	datetime	8	0	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
shift_time_finish	datetime	8	0	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
recorder_name	char	50	0	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
witness_name	char	50	0	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
mine_name	char	30	0	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>
district_name	char	50	0	0	<input checked="" type="checkbox"/>		<input type="checkbox"/>			<input type="checkbox"/>

Figure 4.3. Field details for an accident

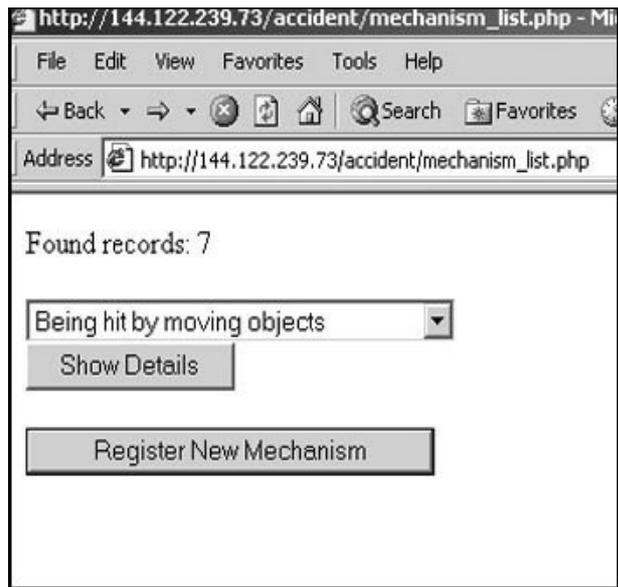


Figure 4.4. Database administration through a web interface

Accessing a database through a web browser is given in Figure 4.5.

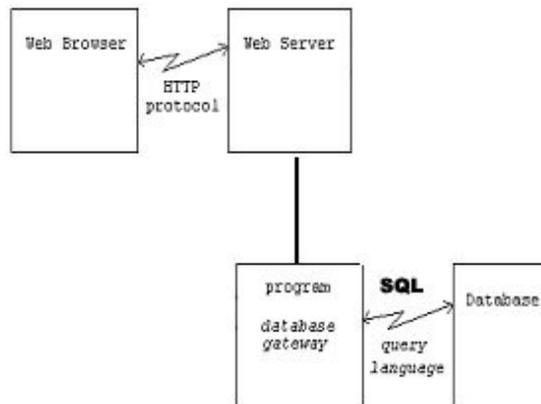


Figure 4.5. Web interface access to the database

CHAPTER V

ACCIDENT CODING STANDARDS CREATED FOR OHSMT

5.1. Accident Information Data

The system is designed to code both injuries and diseases. An injury is the result of a single traumatic event where the harm or hurt is immediately apparent, for example, a broken leg resulting from an accident of hit buy a truck. A disease, on the other hand, results from repeated or long-term exposure to an agent or event, for example, loss of hearing as a result of long-term exposure to noise. Considering all of the information given, a coding system is developed given below.

These subtitles should include as many items as possible. Detailed coding system is used for on-line interface. But, it is not possible to give a code to each parameter at the beginning. So, accident report interface includes an administration part to add new codes. Created coding system is divided into main parts as following. Prepared complete classification has been structured hierarchically and given in Appendices A.

5.1.1. Company Information Data Items

- a. Company name
- b. Address of the company
- c. Total employees working in underground
- d. Total employees working in open-pit.

5.1.2. Personal Information of the Employee Subjected to an Accident

- a. Date of birth
- b. Gender
- c. Occupation
- D. Duty status
- E. Number of hours usually worked each week
- F. Insurance number

5.1.3. Accident Details

- A. Date of occurrence/report
- B. Nature of injury/disease
- C. Bodily location of injury/disease
- D. Mechanism of injury/disease
- E. Agency of injury/disease
- F. Breakdown agency
- g. Location

5.1.4. Outcome of the Accident

- A. Time lost
- B. Disability percentage
- C. Payments made
- D. Compensation payments
- E. Non-compensation payments
- f. DIFR
- g. FIFR
- h. AFR
- i. ASR

5.2. Coding the Mechanism of Injury/Disease

The mechanism of injury/disease classification is intended to identify the mechanism or process by which the worker suffered an injury or disease. The mechanism of

injury/disease is most easily thought of in terms of an action, exposure or event. Different types of mechanisms are actions, such as being struck by, or striking against, an object, or lifting, handling or carrying objects. Other mechanisms can be exposures, such as to a dust, environmental factors, specific events such as vehicle accidents and cave-ins.

Administered users identify the action, exposure or event, which was the direct cause of the most serious injury or disease, as identified as a first step, and select it from combo box in database.

Since it is most unusual for someone to suffer any injuries just through the act of falling, the mechanism of injury categories for falls, trips and slips include the event immediately following the fall, trip or slip. For example, if a worker trips over and hits his or her head on the floor, the mechanism of injury is the trip, which includes the event immediately following it, that is, the hitting of his or her head on the floor. If the most serious injury is not the result of the event immediately following the fall, trip or slip, the event that directly caused the most serious injury should be coded as the mechanism of injury. For example, if a worker trips over, bumps a knee on a table causing a bottle of acid to tip over and splash him or her in the face, the most serious injury of burns to the face should have the mechanism of injury code should be 'contact with the acid' rather than the 'trip'. The classification has been structured hierarchically and given in Appendices B.

5.3. Bodily Location of Injury/Disease Classification

The Bodily Location of Injury/Disease Classification is intended to identify the part of the body affected by the most serious injury or disease. The classification has been structured hierarchically and given in Appendices C.

5.4. Mechanism of Injury/Disease Classification

The Mechanism of Injury/Disease Classification is intended to identify the action, exposure or event which was the direct cause of the most serious injury or disease. The classification has been structured hierarchically and given in Appendices D.

5.5. Agency Classification

The Agency Classification is to be used to code both Breakdown Agency and Agency of Injury or Disease. The Breakdown Agency is intended to identify the object, substance or circumstance that was principally involved in, or most closely associated with, the point at which things started to go wrong and which ultimately led to the most serious injury or disease. The Agency of Injury/Disease refers to the object, substance or circumstance directly involved in inflicting the injury or disease. The classification has been structured hierarchically and given in Appendices E.

CHAPTER VI

USAGE OF MANAGEMENT TOOL

There are three kinds of users who were created for the usage of the system. First of them is ordinary user. Ordinary users don't have the permission to input accident data. They can only access to the results of the accidents.

The other type is the administered users, who have the right to input accident data.

Administered users are the safety engineer or one of the managers of the mine who is responsible of submitting accident data to the Ministry of Labour and Social Security. Safety engineers could be responsible from at most 10 mine sites according to the Turkish Mine Law No: 3213. So using the required database link, they can fill accident information online.

Third type user is the administrator of the system. Only administrator has the right to create administered users, company and mine. Insurance number and personnel details are input by the administrator to the database. Change and editing of the accident fields is other rights given to the administrator.

General view of the created management tool as web page is given in Figure 6.1. Links of this prepared web page are given as database link in Figure 6.2, education link in Figure 6.3, legal issues link in Figure 6.4, bulletin board link in Figure 6.5, links in Figure 6.6, news link in Figure 6.7 of national safety mining database.

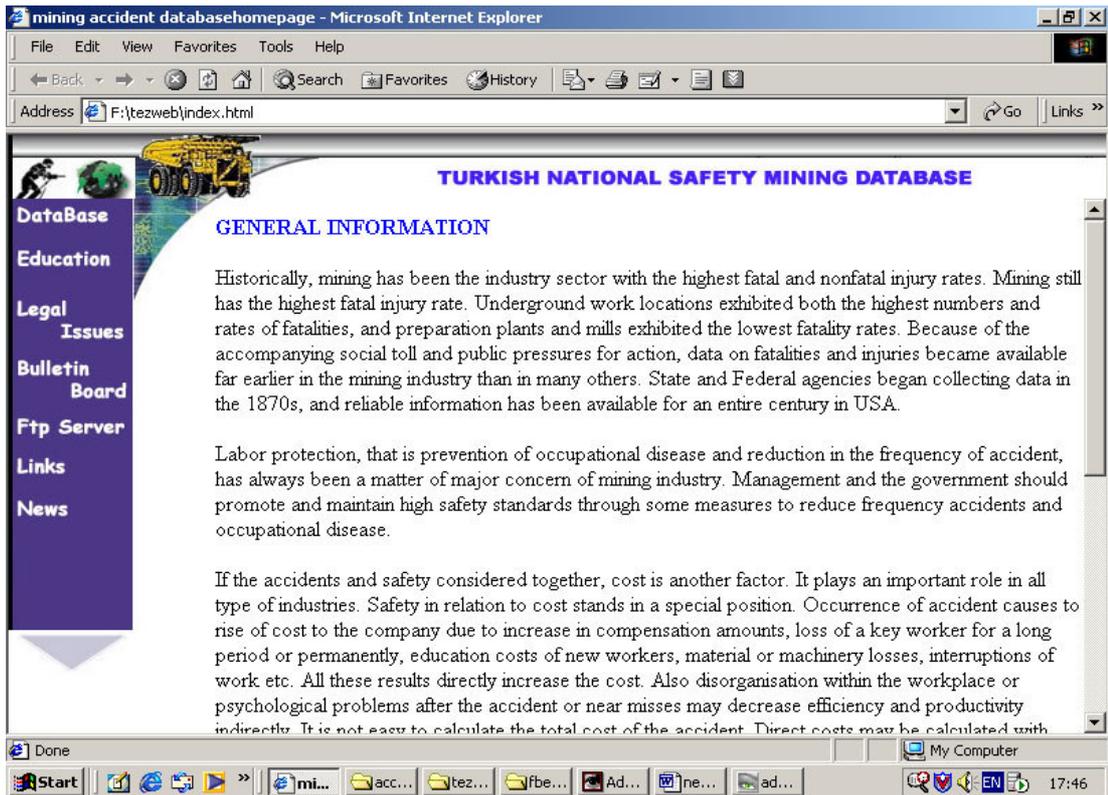


Figure 6.1. National safety mining database index page

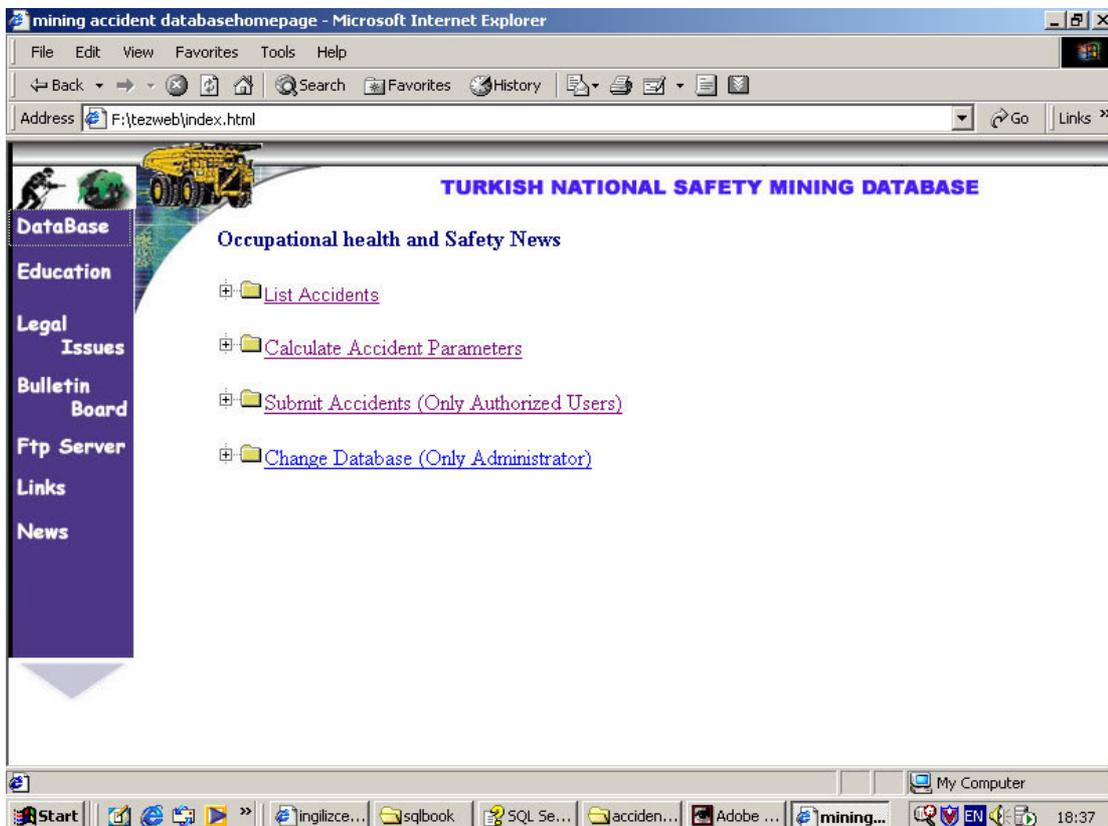


Figure 6.2. National safety mining database, database link

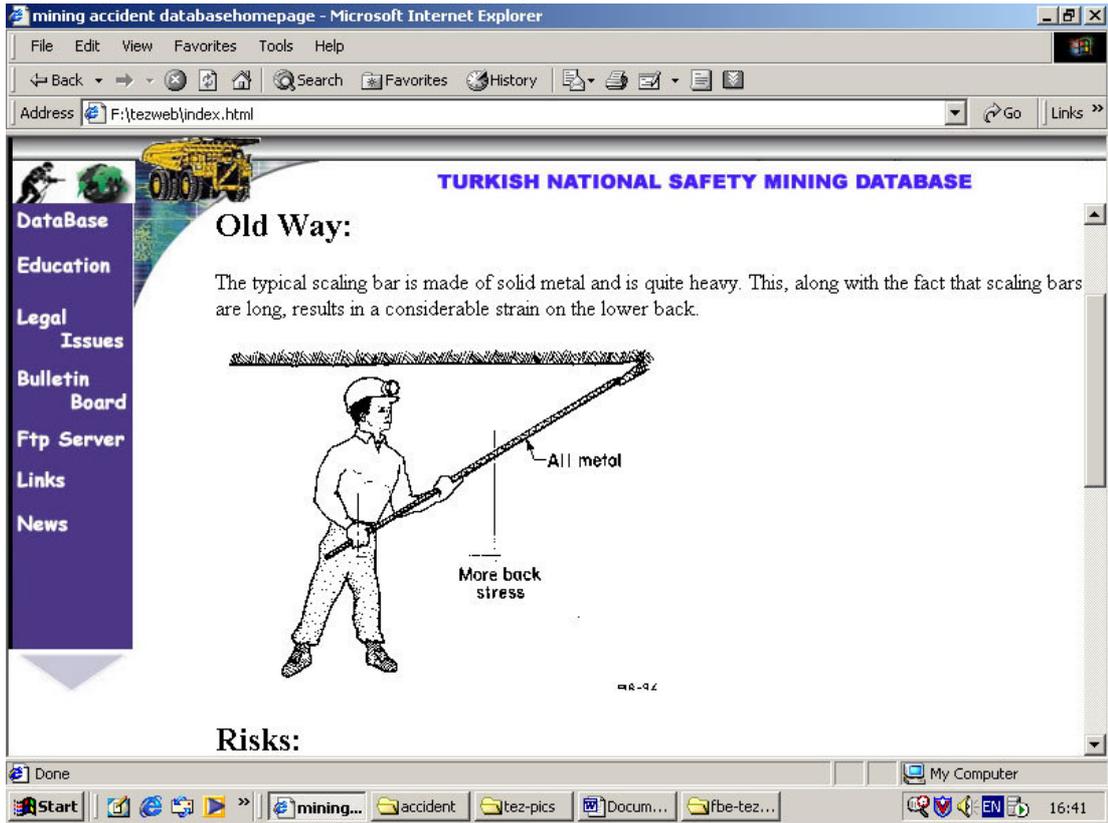


Figure 6.3. National safety mining database, education link

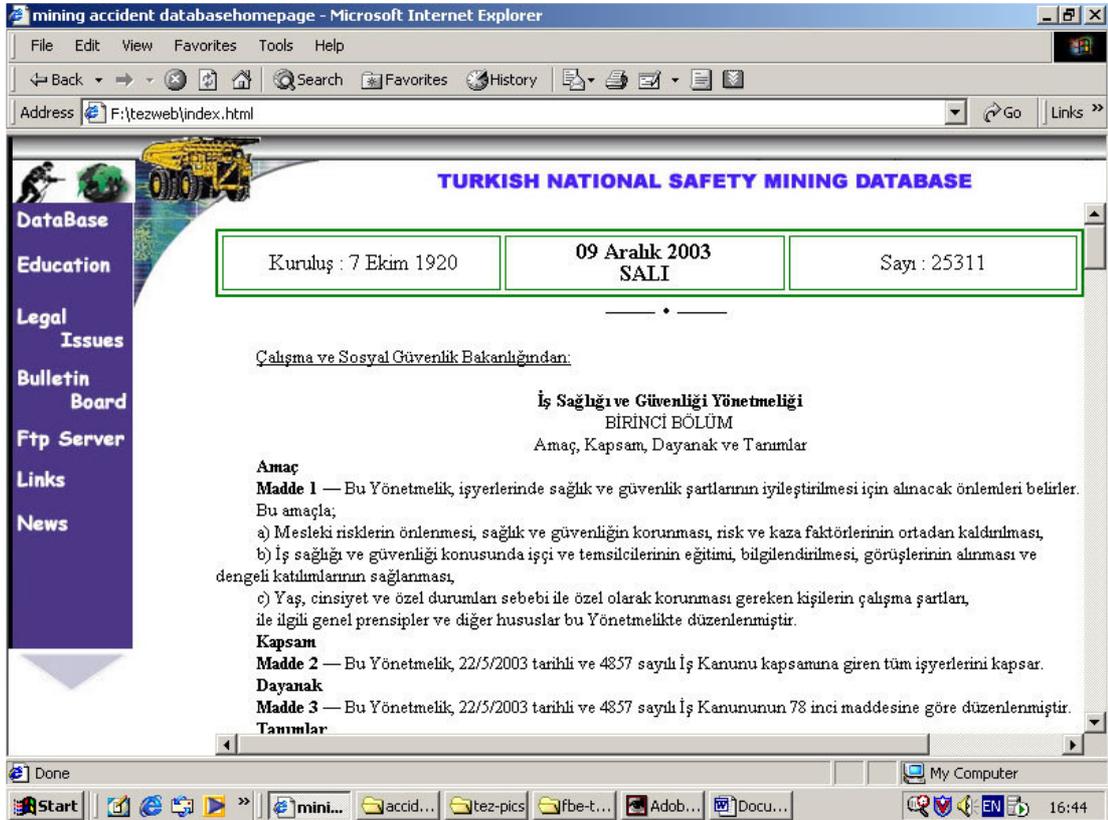


Figure 6.4. National safety mining database, legal issues link

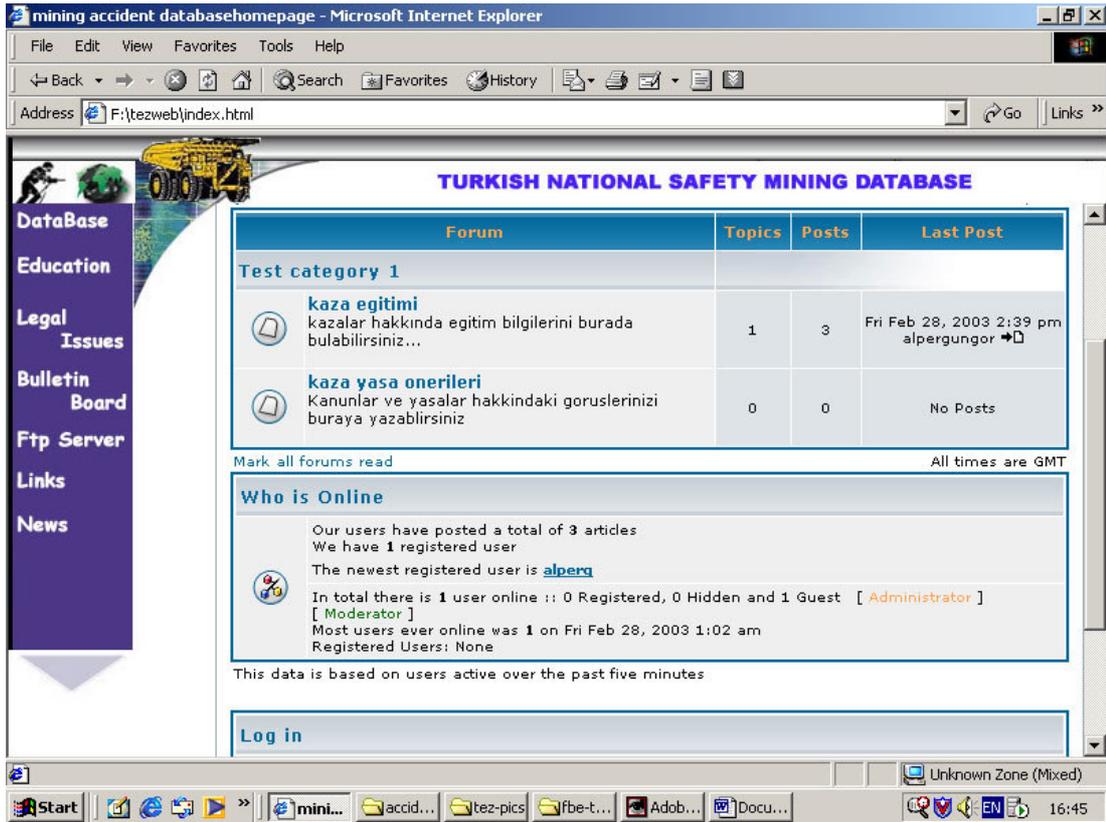


Figure 6.5. National safety mining database, bulletin board link

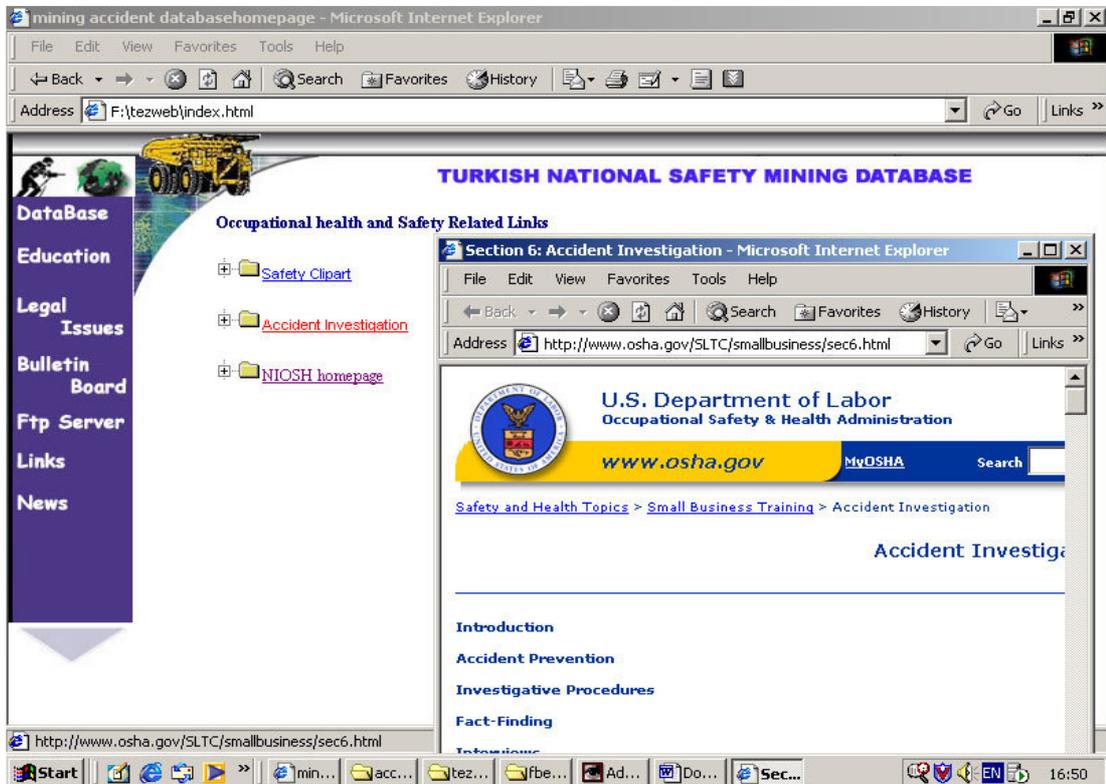


Figure 6.6. National safety mining database, links

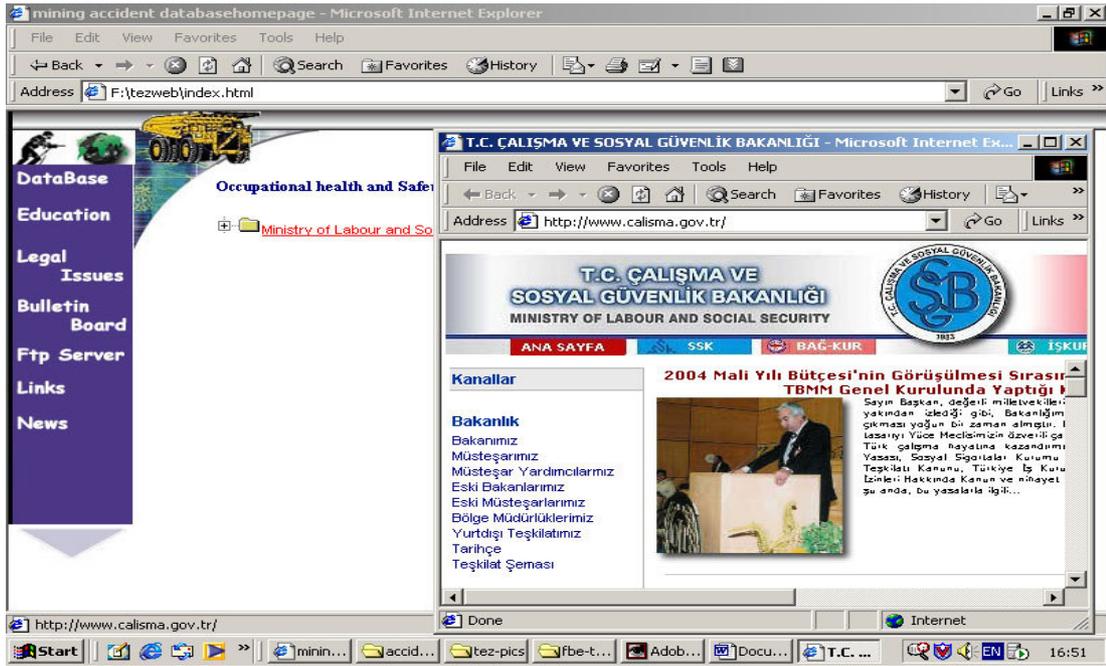


Figure 6.7. National safety mining database, news link

Ordinary users may perform search using “list accidents” link as shown in Figure 6.2. Accidents resulted in fatality search is given in Figure 6.8. and Figure 6.9. DIFR, FIFR, AFR, AND ASR values for each mine can be calculated within database by ordinary users. For example the procedure to calculate DIFR is given in Figure 6.10, Figure 6.11, Figure 6.12 and Figure 6.13.

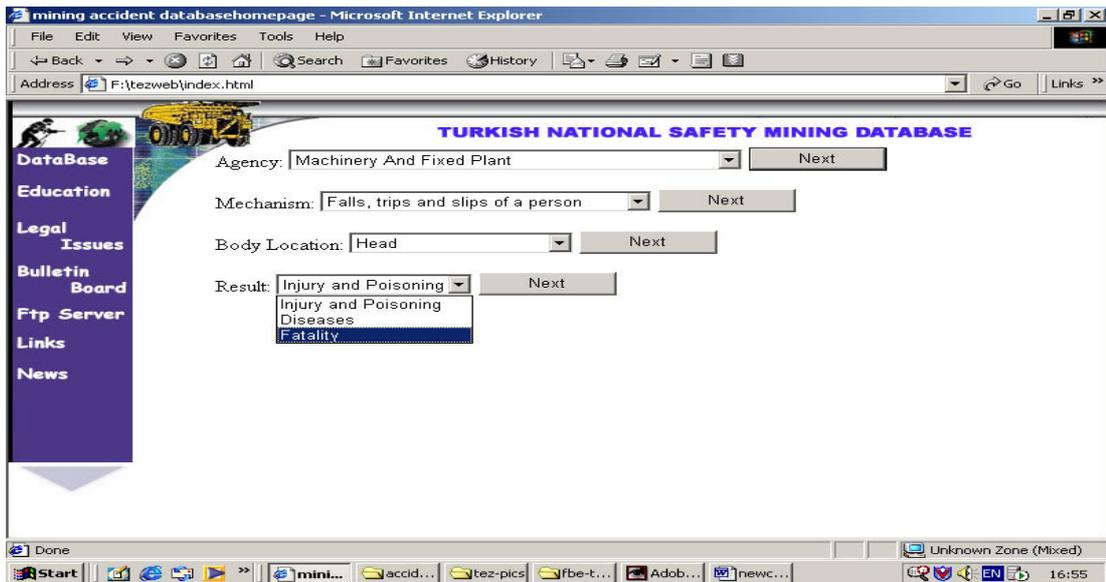


Figure 6.8. Ordinary user search in database

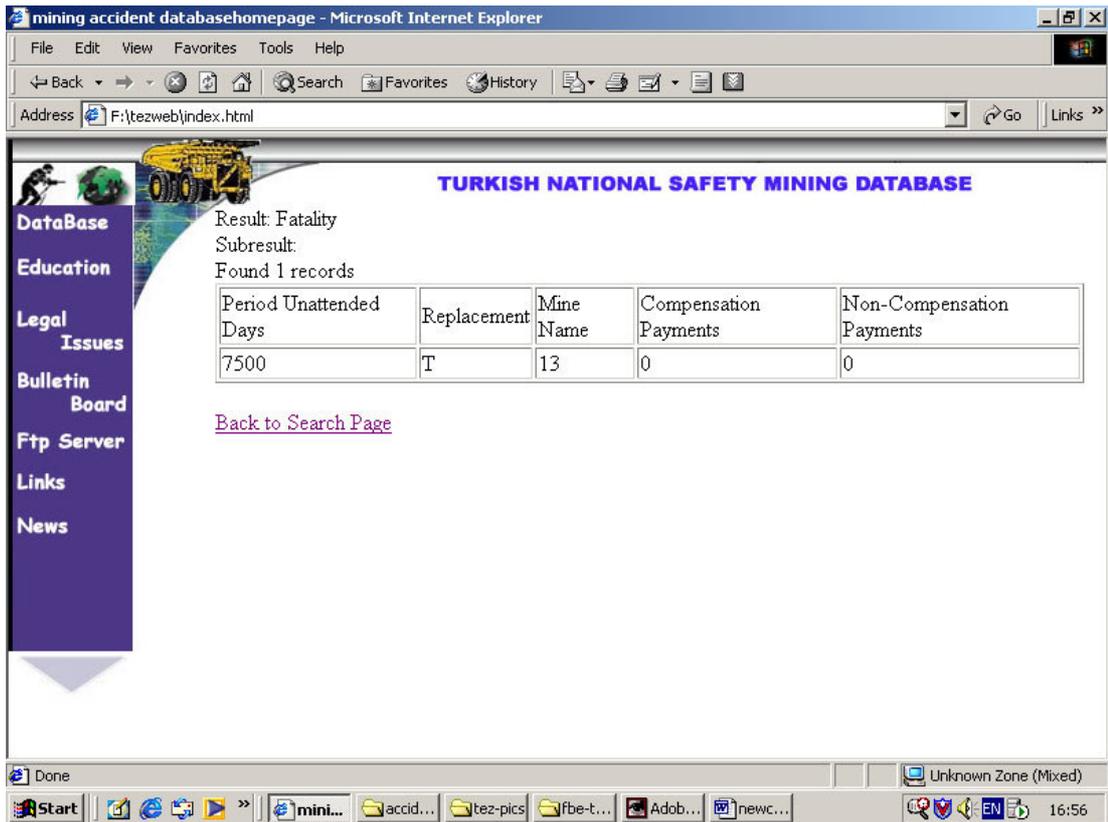


Figure 6.9. Ordinary user search results for fatality in database

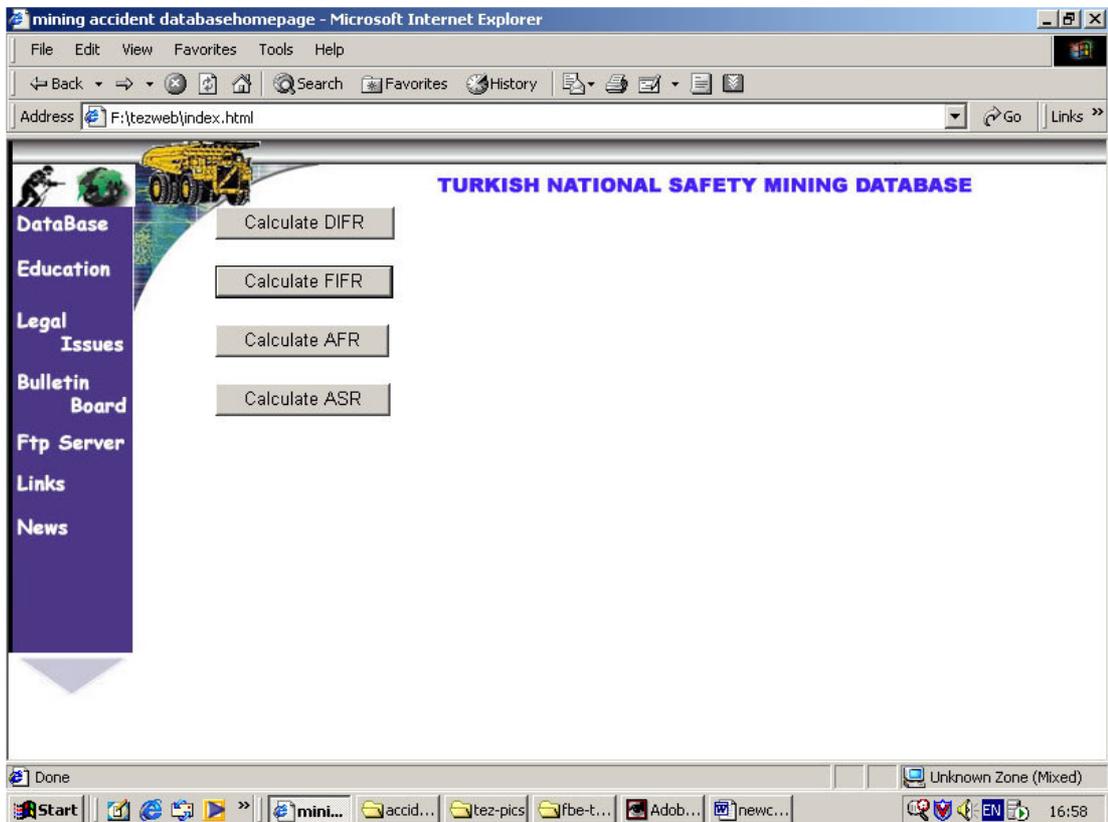


Figure 6.10. Calculation of parameters

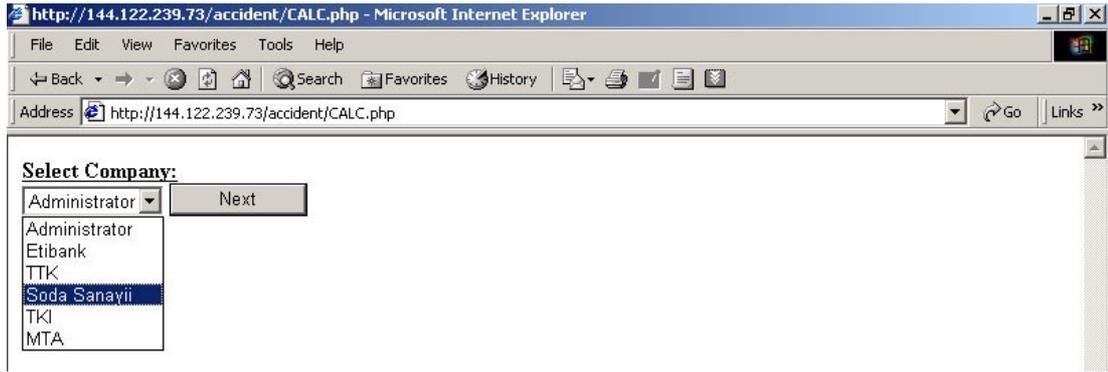


Figure 6.11. Calculation of parameters company selection



Figure 6.12. Calculation of parameters selection of site belong to company

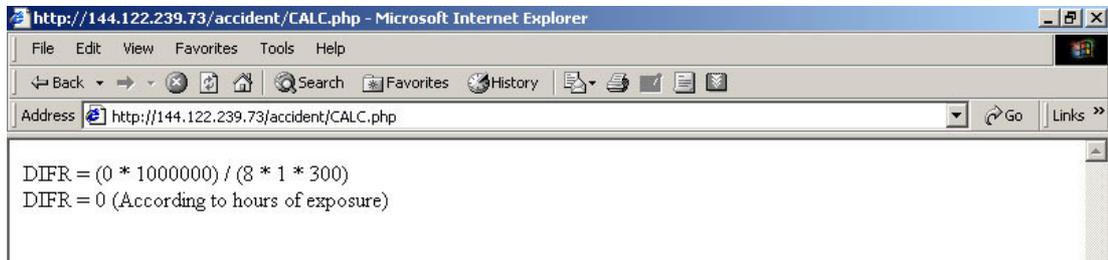


Figure 6.13. Calculation results for selected site of company

As described before, administrator has the right to change and add database fields. Login page for administrator of the database is given in Figure 6.14. After successful login, administrator reaches administration web page as shown in Figure 6.15. Registration of new agency and sub-agencies are given in Figure 6.16. and 6.17 respectively. The other registration procedures can be realized by the same way.

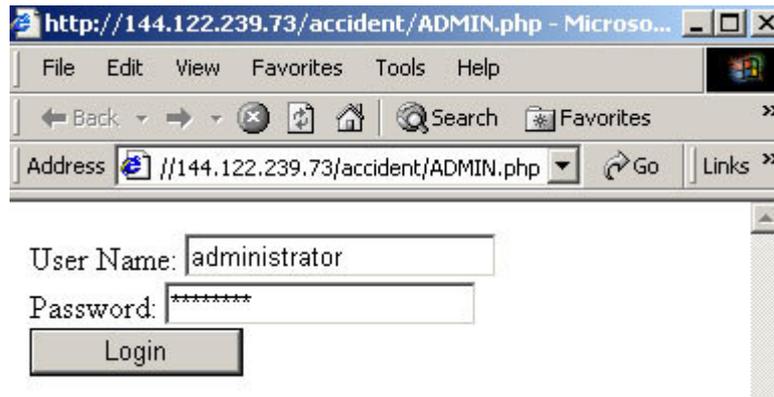


Figure 6.14. Administrator login page

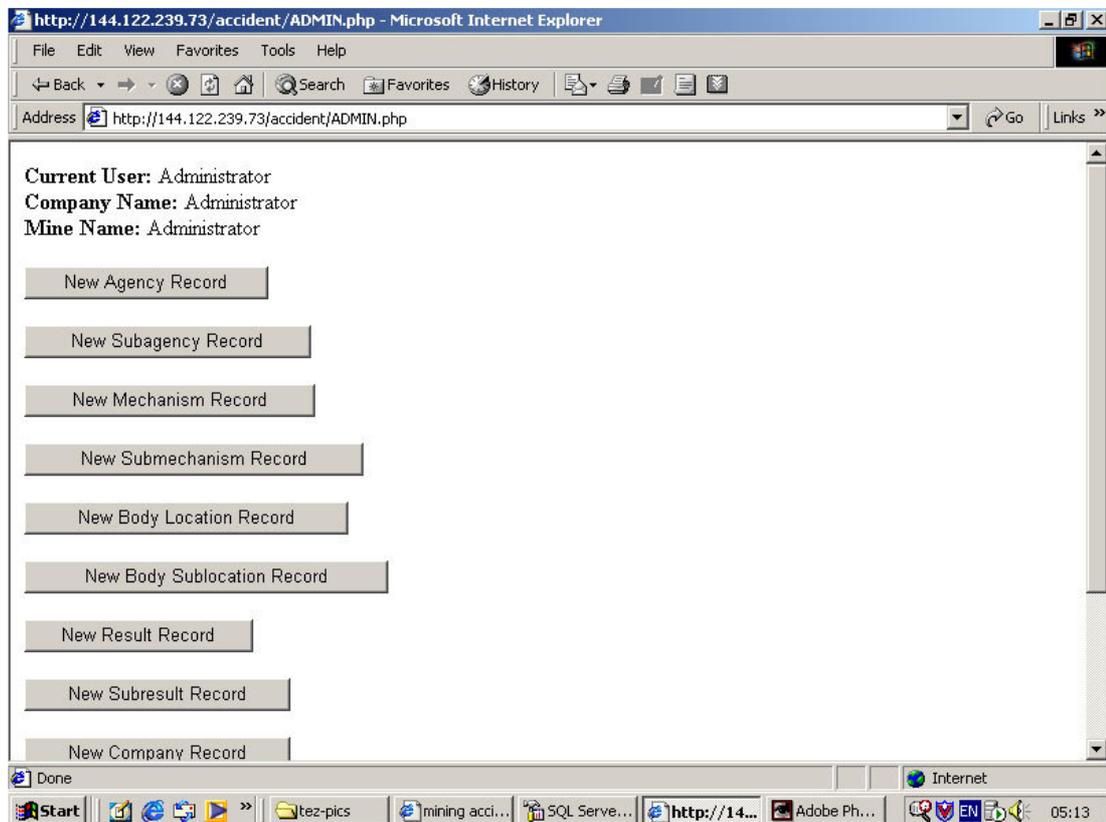


Figure 6.15. Administrator main page

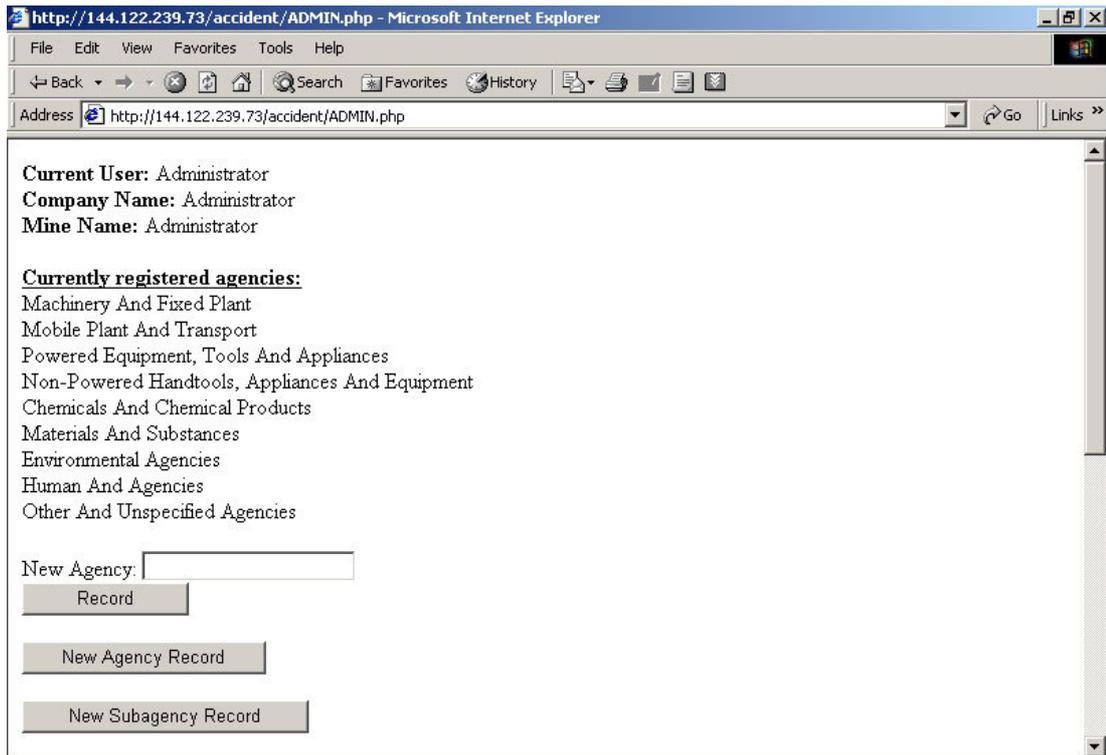


Figure 6.16. Administrator new agency record page

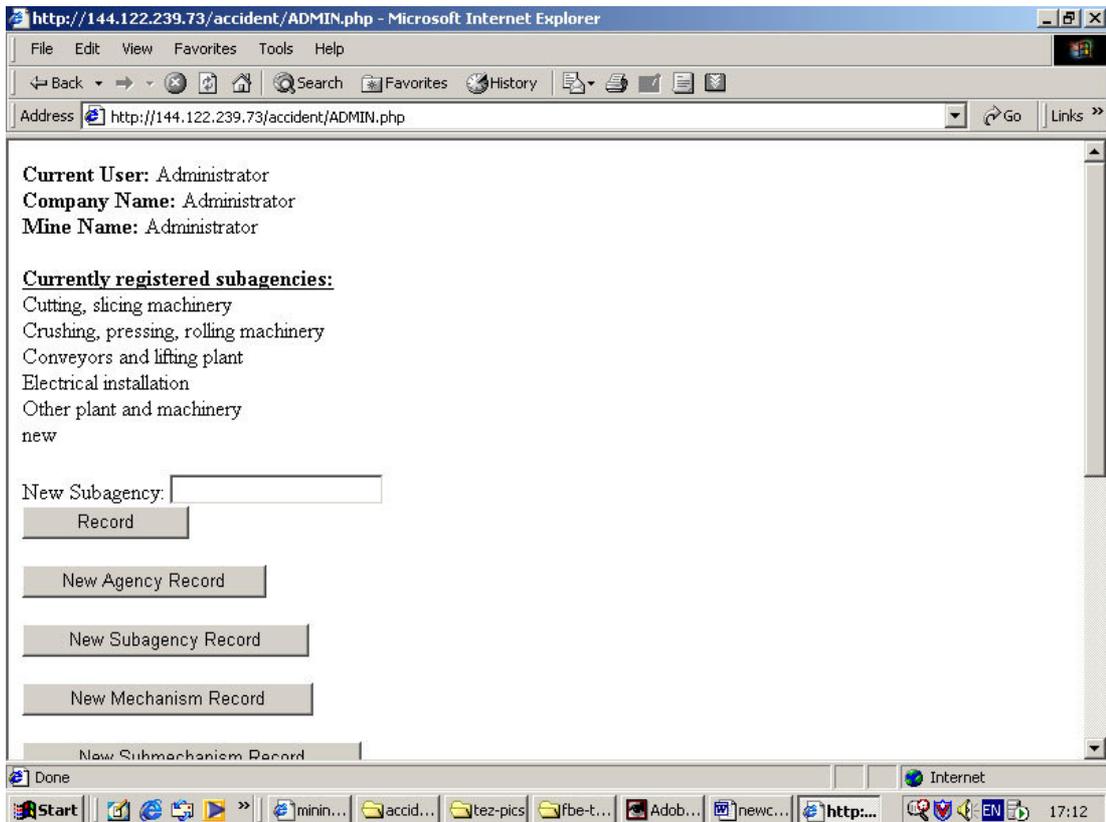


Figure 6.17. Administrator new sub-agency record page

Administrator performs the registration of new company and mine belongs to recorded company as shown in Figures 6.18. and 6.19. respectively. Mine information is recorded and record is finalized as given in Figures 6.20. and 6.21.

http://144.122.239.73/accident/ADMIN.php - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Copy Paste

Address http://144.122.239.73/accident/ADMIN.php Go Links >>

Current User: Administrator
Company Name: Administrator
Mine Name: Administrator

Currently registered companies:
Administrator
Etibank
TTK
Soda Sanayii
TKI
MTA

New company:
Company Name:
Company Address:
Number of Underground Workers:
Number of Open Pit Workers:

Figure 6.18. Administrator new company record page

http://144.122.239.73/accident/ADMIN.php - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites History Print Copy Paste

Address http://144.122.239.73/accident/ADMIN.php Go Links >>

Current User: Administrator
Company Name: Administrator
Mine Name: Administrator

Select Company:
Soda Sanayii
Administrator
Etibank
TTK
Soda Sanayii
TKI
MTA

Figure 6.19. Administrator mine record page first step

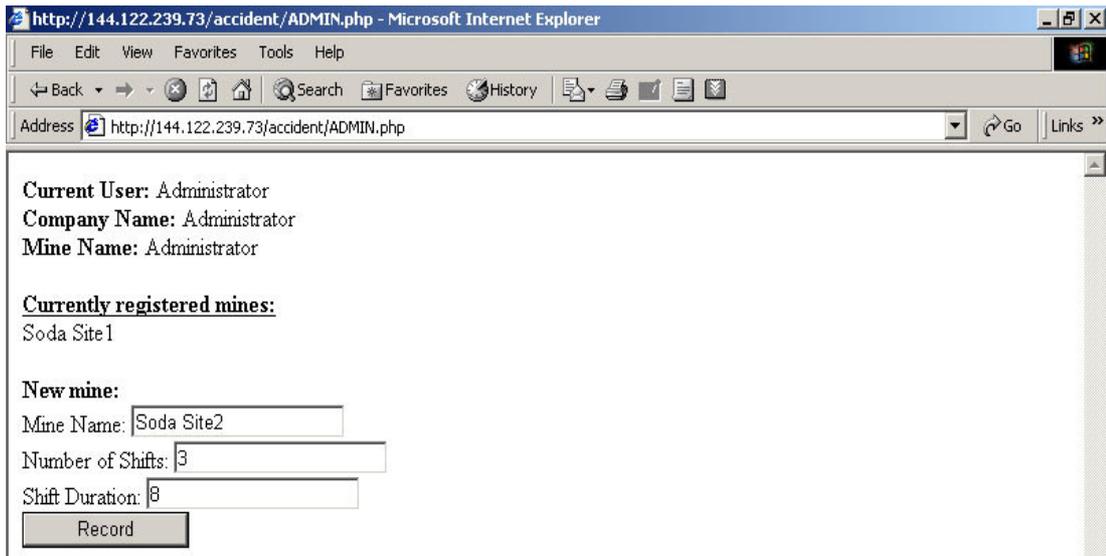


Figure 6.20. Administrator new mine record page second step

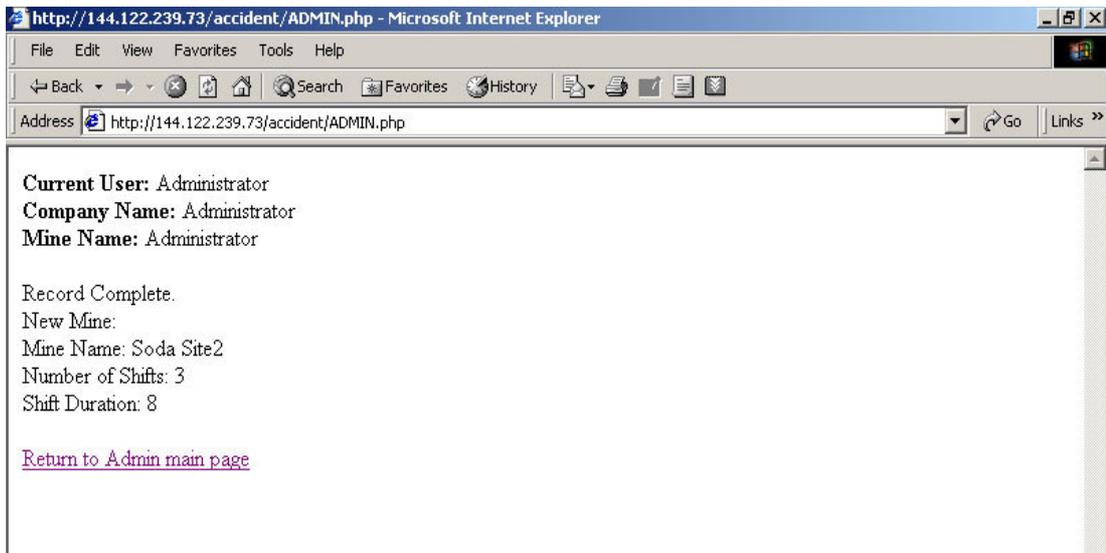


Figure 6.21. Administrator new mine record completed

Procedure to be followed by the administrator to record new administered user is performed in two steps. As shown in Figure 6.22, administrator's first step is to choose the company of the user to be created. Then the mine name is selected as given in Figure 6.23. Username and password is also created in this step.

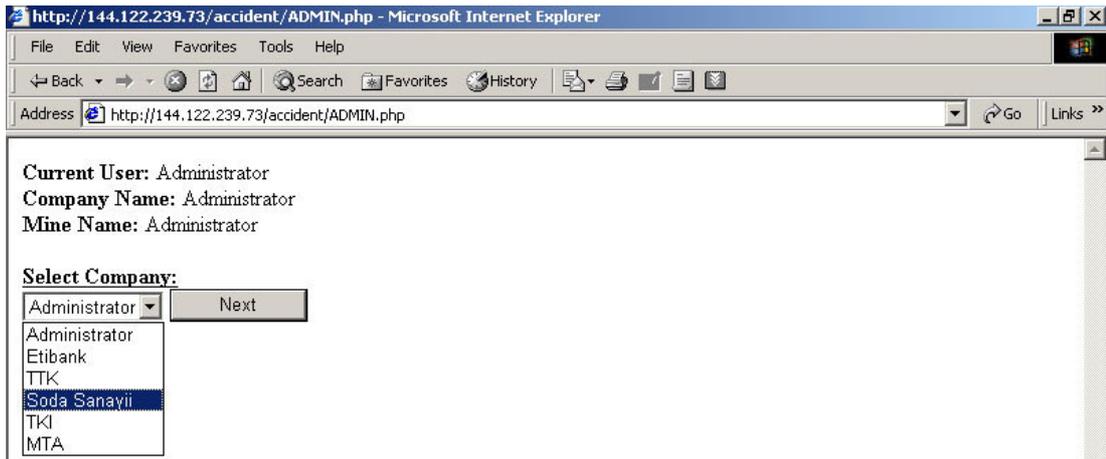


Figure 6.22. Administrator new user record first step page

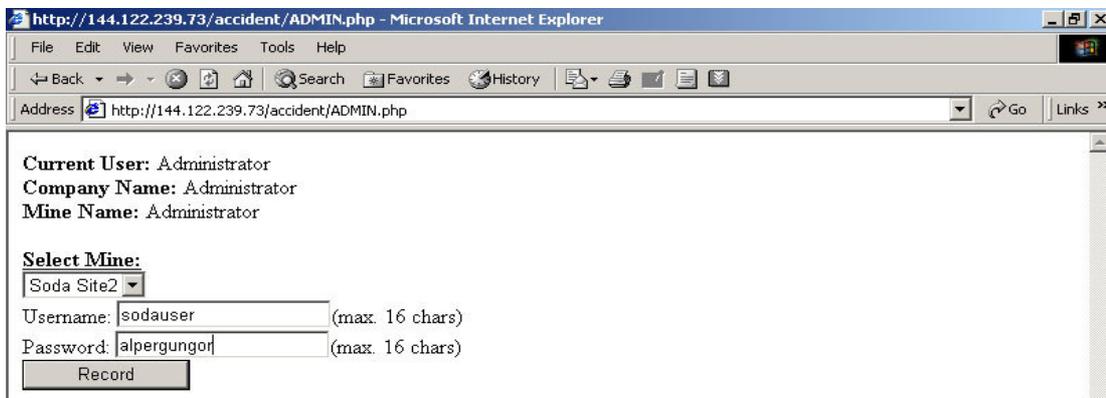


Figure 6.23. Administrator new user record second step page

Personnel information input carried out by the administrator starts with the selection of the ccompany, as shown in Figure 6.24. Then, Figure 6.25. and 6.26. illustrates how to input detailed information and finalizing the record in following steps.

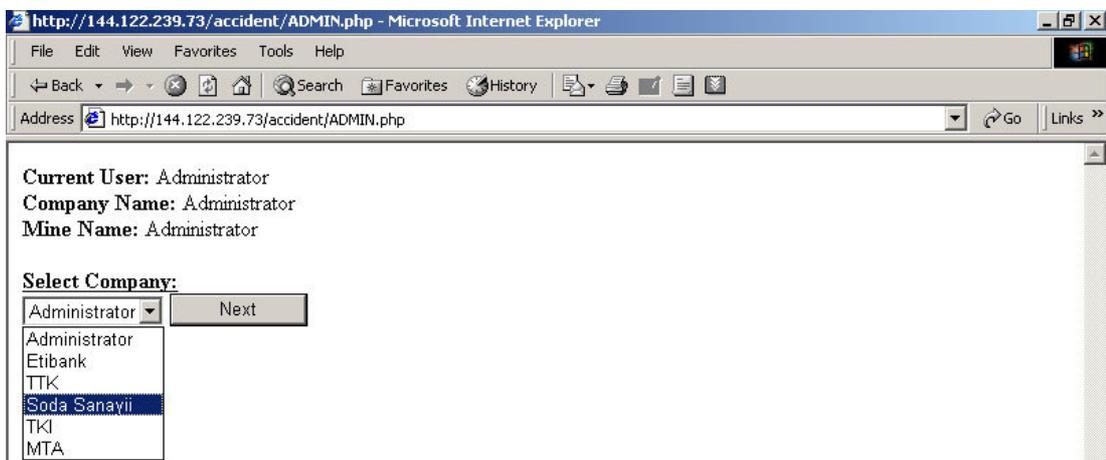


Figure 6.24. Administrator new personnel record page first step

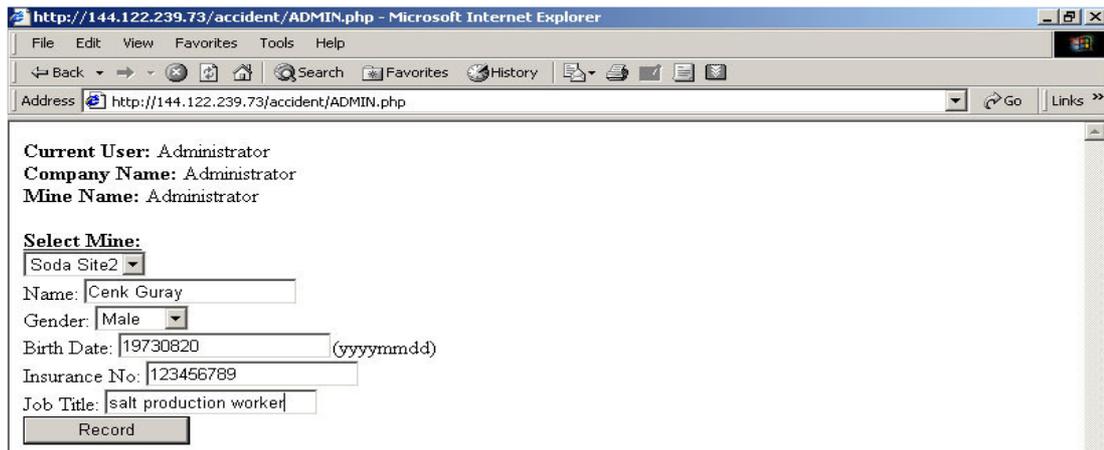


Figure 6.25. Administrator new personnel record page second step

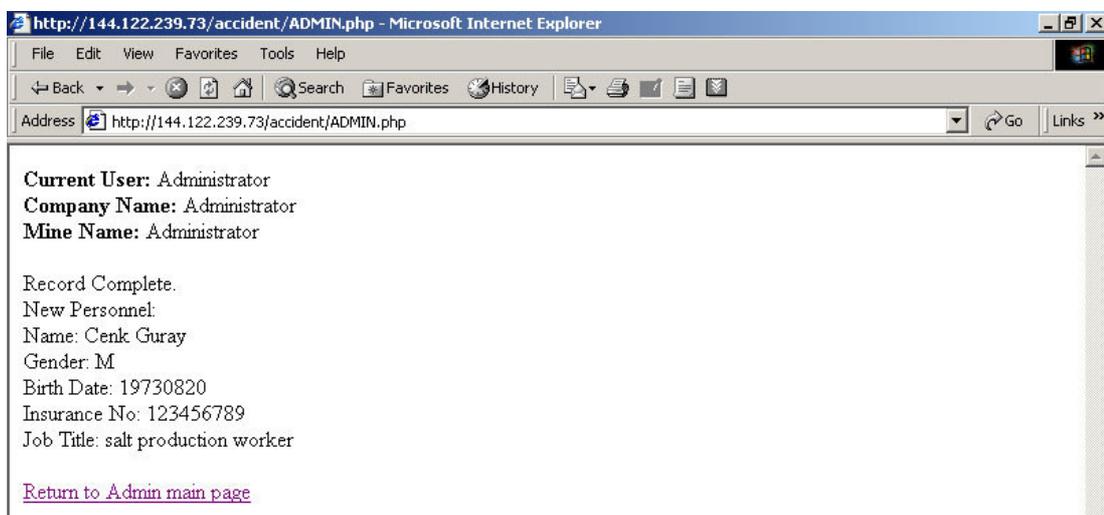


Figure 6.26. Administrator new personnel record completed page

Administered users login to database as shown in Figure 6.27. If the user is not authorized, which means username and password is wrong or not created, user gets unauthorized user notification as shown in Figure 6.28. In this case, user cannot input accident data. So, security of the database is supplied with this protection. If the user input correct information, user name, company name and mine name of that user is printed in screen as illustrated in Figure 6.29. In order to input accident data, administered user should input insurance number of the person, who involved in the accident as given in Figure 6.30.

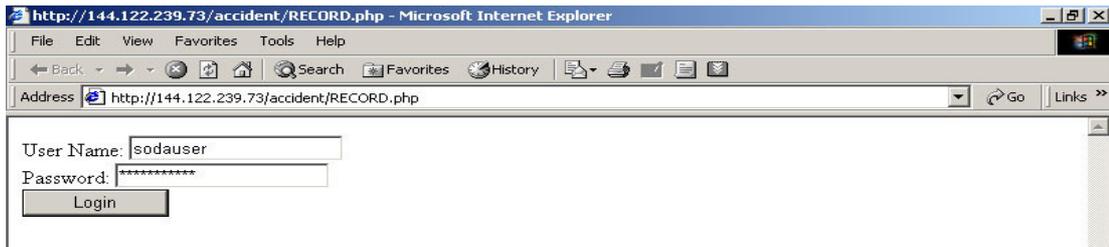


Figure 6.27. User login page



Figure 6.28. Unauthorized user login name or password result



Figure 6.29. Authorized user login

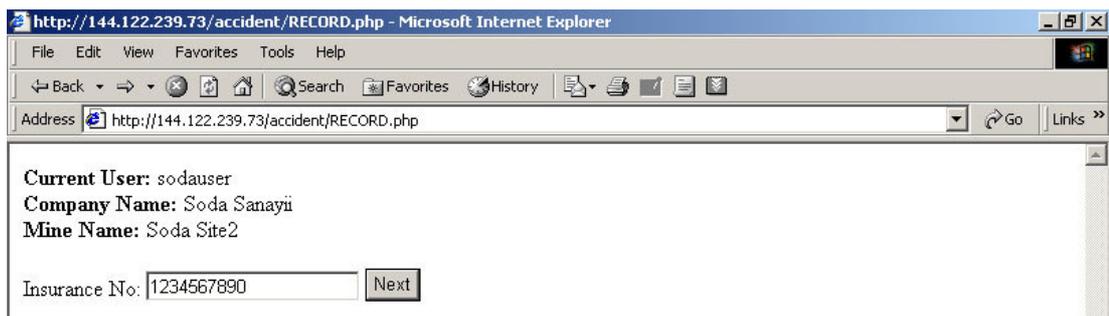


Figure 6.30. Authorized user accident report insurance number input

If the insurance number input is not belonging to the mine, from which the user is responsible, or wrong insurance number, input stops as illustrated in Figure 6.31. However, if the insurance number input is valid for that mine as shown in Figure

6.32., user ended up with detailed information of the person printed to the screen as given in Figure 6.33.

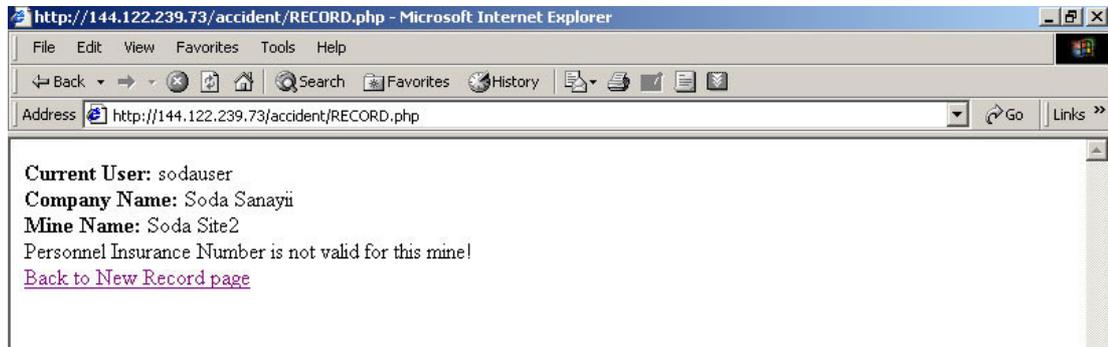


Figure 6.31. Authorized user accident report insurance number not valid warning

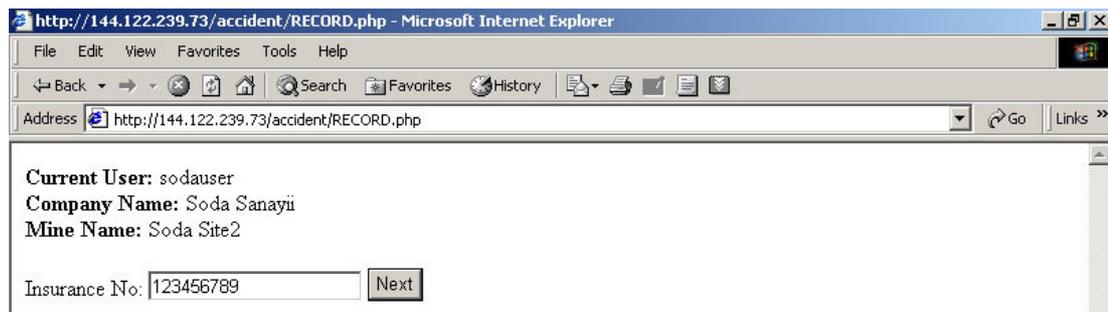


Figure 6.32. Authorized user accident report valid insurance number input

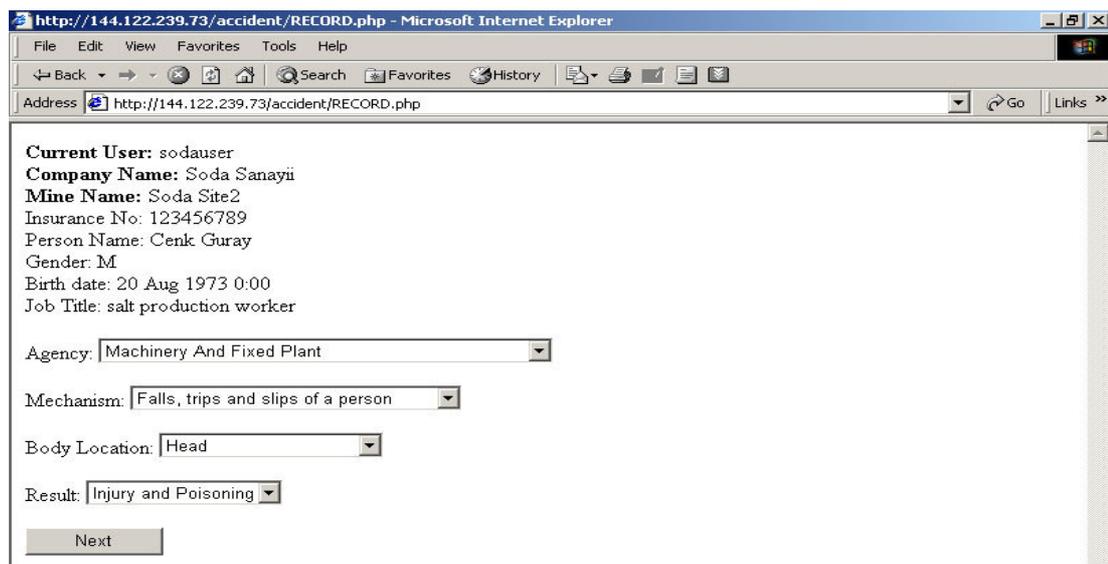


Figure 6.33. Authorized user accident report input first step

Then, the user starts to input accident data by using scroll bars as shown in Figure 6.34. and proceeds to following input by pressing the next button as given in Figure 6.35. Data in the scrollbars carries information of the related fields, which are prepared by the administrator. If the user cannot find appropriate information in the related field, he chooses “other or undefined” option, that specific case can be added by the administrator later on.

Current User: sodauser
Company Name: Soda Sanayii
Mine Name: Soda Site2
Insurance No: 123456789
Person Name: Cenk Guray
Gender: M
Birth date: 20 Aug 1973 0:00
Job Title: salt production worker

Agency: Powered Equipment, Tools And Appliances
Mechanism: Heat and Electricity
Body Location: Multiple Locations
Result: Fatality

Next

Figure 6.34. Authorized user accident report input second step

Current User: sodauser
Company Name: Soda Sanayii
Mine Name: Soda Site2
Insurance No: 123456789
Person Name: Cenk Guray
Gender: M
Birth date: 20 Aug 1973 0:00
Job Title: salt production worker

Agency: Powered Equipment, Tools And Appliances
Sub Agency: Other powered equipment, tools and appliances
Mechanism: Heat and Electricity
Sub Mechanism: Contact with electricity
Body Location: Multiple Locations
Body Sublocation: Multiple locations
Result: Fatality
Subresult: Fatality

Next

Figure 6.35. Authorized user accident report input third step

User continues to input accident detailed information as illustrated in Figure 6.36. and finalize the data input by pressing “finish” button . If he want to continue to input another accident, he chooses “new record” button as shown in Figure 6.37.

Replacement:

Description: There was slack in the cable and the victim assumed the cable hook was removed from the puller. The cable

Conclusion: A lack of communication between the victim and those operating the puller failed to ascertain the cable had

Recommendations: During work involving pulling with cables, chains, etc., all persons shall position themselves in a safe

Job Experience Days: 850

Mining Experience Days: 412

Accident Date: 20040126 (yyyymmdd hh:mm:ss)

Witness Name: koray onal

Compensation Payments:

Non-Compensation Payments:

Finish

Figure 6.36. Authorized user accident report input fourth step

Current User: sodauser
Company Name: Soda Sanayii
Mine Name: Soda Site2

Record complete.
[New Record](#)

Figure 6.37. Authorized user accident report input final step

CHAPTER VII

CONCLUSIONS AND RECOMMENDATIONS

One of the major aspects of this study is to develop a standard accident reporting system. To realize this aim, standard but dynamic classification system which is given in Appendix A is created. So, this system is an attempt to collect information about the wider circumstances in which accidents occur and to define more carefully every aspect of the accidents. This allows much more specific definition of factors that are common to a number of accidents. With this information, better strategies can be developed for accident prevention.

The described management tool is composed of educational, statistical and database interfaces prepared with PHP modules. Written PHP codes of the system are given in Appendix F. All of these are gathered in a web page coded with HTML. Media is shared with ftp server and knowledge transaction is carried on bulletin board. So, at the moment, a kind of rigid management tool for collecting accident information and tutoring occupational health and safety is achieved. This prepared tool can be easily adapted to a server with low cost. Apache server configuration and PHP server configuration to be used for that server is given in Appendix G and H respectively. PHP modules configuration is given in Appendix I.

One of the major contributions of this work is that, collected accident information data can be reached and acquired easily at any time, 24 hours a day, 7 days a week.

The second contribution of this system is; recommendations and conclusions prepared by experts for each accidents and other detailed information collected can

be used as a knowledge base system to prevent same kind of accidents to happen again.

The main contribution of this study to the literature is the prepared standard accident report form, which includes most of the aspects of an accident.

Other important items that should be emphasized for the created system are as follows:

- The system has a dynamic characteristic. This characteristic gives the system ability to grown by experts,
- The system is secure. Only registered users can submit information which prevents misleading information,
- Administration interface is user friendly and secure,
- The speciality of the system provides to input all accident data step by step, preventing error proof input environment,
- The capability of the system provides to contain millions of accident data and queries result in a few seconds.

So, created management tool stands successfully which contributes to data mining for accidents and supply educational information.

As a first recommendation, collected data from the database can be used to improve national cost model.

Secondly, created accident database could be related with legislative database, which could be developed by another study.

Thirdly, collected information could be used as a basis to perform risk analysis for mines.

Although this management tool is specifically prepared for mines, it can also be used by other industries, which also have high accident rates, especially construction and metal industries.

The earlier data collected for accidents in most countries were not analysed carefully, therefore such data become useless for developing specific strategies to prevent accidents. So, as a final recommendation, related institutions or government agencies should react and take the responsibility to use this kind of management tool.

To have an accident is unfortunate, however, to have an accident and learn nothing from it is unforgivable.

REFERENCES

1. Amundsen M. and Smith C., January 1996, Teach Yourself Database Programming With Visual Basic 4 in 21 Days, Sams Co., ASIN: 0672308320
2. Andrews C., Litchfield D., Grindlay B., 2003, SQL Server Security, McGraw-Hill Osborne Media, ISBN: 0072225157
3. Arpajian S. and Mullen R., November 1996, How to Use HTML 3.2, Ziff Davis Pr, ASIN: 1562764969
4. Bozkurt, R. 1993, Application of the Modern Accident Prevention Techniques to OAL Mines (TKI). METU, Ankara.
5. Brancoli, M., 1983, Accident Statistics. Encyclopedia of Occupational Health and Safety, Geneva.
6. Bureau of Labor Statistics U.S. Department of Labor
<http://www.bls.gov/>
7. Christiansen T., Torkington N., Perl Cookbook , August 1998, O'Reilly & Associates, ISBN: 1565922433
8. Delaney, K., 2000, Inside Microsoft® SQL™ Server 2000, Microsoft Press, ISBN: 0735609985

9. DIE, Turkish Republic Statistics Institute
<http://www.die.gov.tr/istatistikler.html>
10. Güyagüler T., September 2000, Occupational Health and Safety, Middle East Technical University, Lecture Book.
11. Handley W., August 1977, Industrial Safety Handbook, McGraw Hill Publishing, ISBN: 007084481X
12. Heinrich H.W., January 1980, Industrial Accident Prevention, McGraw Hill Publishing, ASIN: 0070280614
13. Henderson, K., 2000, The Guru's Guide to Transact-SQL, Addison-Wesley Pub Co, ISBN: 0201615762
14. Jacobson, R., 2000, Microsoft® SQL Server™ 2000 Analysis Services Step by Step, Microsoft Press ISBN: 0735609047
15. MSHA, Mine Safety and Health Administration U.S. Department of Labor
<http://www.msha.gov/>
16. Niosh, National Institute for Occupational Safety and Health
<http://www.cdc.gov/niosh/mining/>
17. Niosh, Common Information Service System (CISS)
<http://outside.cdc.gov:8000/ciss/Welcome.html>
18. Noshc, Australian Government National Occupational Health and Safety Commission
<http://www.nohsc.gov.au/SmallBusiness/BusinessEntryPoint/laws/>
19. Ratschiller T., Gerken T., Web Application Development with PHP 4.0, New Riders Publishing, ISBN: 0735709971

20. SSK, Turkish Social Insurance Institute Statistics

<http://www.ssk.gov.tr/wps/portal>

21. Wall L., Christiansen T., Orwant J., July 2000, Programming Perl (3rd Edition), O'Reilly & Associates, ISBN: 0596000278

22. World Health Organization (WHO).

<http://www.who.int>

APPENDIX A

Accident Information Data Details

5.1.1.a. Company Name

Description: Name of the company.

Purpose: To enable investigation for distribution of occupational injury/disease for each company.

Coding: Text format, variable character.

Comment: Each company may have more than one work area. Other parameters will be queried within related fields.

Length: 50.

5.1.1.b. Company Address

Description: Address of the company.

Purpose: Detailed information about the accident.

Classification: Address of the company center.

Coding: Text format, variable character.

Comment: Each company may have more than one work area. Other parameters will be queried within related fields.

Length: 150.

5.1.1.c. Total employees working in underground

Description: The size of the company is the number of workers employed by the enterprise. Total will be calculated by summing up the number of underground and open-pit workers by the program.

Purpose: To enable investigation of size of business factors on occupational injury/disease experience.

Classification: Number of workers employed by the company that work in underground.

Coding: Numeric format, variable integer.

Comment: The number of workers working in underground should be the total as at the time of reporting of the injury/disease occurrence.

Length: 4.

5.1.1.d. Total employees working at open-pit:

Description: Same as item 5.4.1.c

Purpose: Same as item 5.4.1.c

Classification: Number of workers employed by the company that work at open-pit area.

Coding: Numeric format, variable integer.

Comment: The number of workers working in open pit should be the total as at the time of reporting of the injury/disease occurrence.

Length: 4.

5.1.2.a. Date of Birth

Description: The date of birth of the worker subjected to injury or disease.

Purpose: To facilitate analysis of occupational injury/disease experience by age of worker.

Classification: Date of Birth.

Coding: Date Numeric format specified as YYYYMMDD.

Comment: Age will be derived from the date of birth and the date of occurrence/report.

Length: 8.

5.1.2.b. Gender

Description: The sex of the worker.

Purpose: To facilitate analysis of occupational injury/disease experience by sex of worker.

Coding: Alphanumeric format, Variable Character. To be recorded as male or female coded according to the following classification:

'M' Male
'F' Female

Length: 1.

5.1.2.c. Occupation

Description: The worker's occupation at the time of the injury or reporting of the occupational disease.

Purpose: To enable between-occupation and within-occupation analysis of occupational injury/disease experience.

Coding: Numeric format, variable character.

Length: 2.

Comment: Occupation data should not be confused with activity at the time of the incident or report. The activity at that time may have been entirely unrelated to the worker's occupation. Occupation database is gathered in another file, and can be selected within database.

5.1.2.d. Duty Status

Description: The worker's duty status at the time of the injury or disease.

Purpose: To enable separate analysis of injuries according to the level of control the employer had over the working environment.

Coding: Numeric format, variable integer. To be recorded using the following classification:

- 01 At work - working at normal workplace
- 02 At work - working away from normal workplace
- 03 At work - on break

Length: 2

Comment: Occupational diseases of long latency should be coded to duty status '01' unless sufficient information is available.

The definitions for the Classification are:

At work: Is where the worker's activity is under the control of the employer. That is, he/she is being paid but is not on an authorised break from work.

'01' *At work - working at normal workplace*: The injury or disease occurred while the worker was working at their normal workplace or base of operations.

'02' *At work - working away from normal workplace*: Injury or disease that occurred while the worker is working at a location other than the worker's normal workplace or base of operations.

'03' *At work - on break*: Refers to the paid or unpaid period within the shift during which the worker has taken an authorised break from work and is still covered for compensation.

5.1.2.e. Hours worked

Description: The number of hours and minutes usually worked each week by the injured worker.

Purpose: To enable separate analysis of injuries sustained by workers engaged in non-standard employment arrangements.

Coding: Date Format. The number of hours and minutes usually worked each week (including overtime) by the injured worker up to a maximum of 90 hours. Format numeric specified as HHMM. If number of hours usually worked each week are reported in whole hours and parts thereof then the decimal place should be converted to minutes (for example, 37.5 hours equates to 37 hours and 30 minutes). If minutes are not known the MM columns should be zero filled.

Length: 4.

Comment: Recording the number of hours and minutes usually worked each week will enable: (i) the comparability of the data across jurisdictions to be increased through the application of a standard method for treatment of part-time workers; and (ii) analyses based on varying risk exposure thresholds to be undertaken. For example, the OHS experience of employees who work less than 35 hours each week (part-timers) could be examined.

Overtime - Overtime (defined as work undertaken which is outside, or in addition to, ordinary working hours of the employee in their main job, whether paid or unpaid) which is usually worked each week should be included in the calculation of number of hours usually worked.

5.1.3.a. Date of Occurrence

Description: Date of occurrence. The date on which the injury occurred or the occupational disease was first reported to the employer.

Purpose: To enable provision of time series data and provide a basis of investigation of any seasonal variation of occurrences.

Coding: Date Format. Format specified as YYYYMMDD.

Length: 8.

5.1.3.b. Nature of Injury or Disease

Description: The nature of injury/disease is intended to identify the most serious injury or disease sustained or suffered by the worker.

Purpose: To enable the analysis of the distribution of types of injury or disease and to relate specific types of injury or disease to the processes leading to them.

Coding: Format numeric, variable integer.

Length: 3.

Comment: The classification should be used for coding both injuries and diseases.

The classification has been structured hierarchically and consists of divisions, which are divided into a number of codes. In order to provide a capacity to expand on the number of codes available, a third digit level has been incorporated.

The classification relies on sufficient detail being available to allow the allocation of appropriate codes. Where the description given on the report form is inadequate, coders may submit that nature with an explanation to the online database.

5.1.3.c. Bodily Location of Injury or Disease

Description: The bodily location of injury/disease is intended to identify the part of the body affected by the most serious injury or disease.

Purpose: To enable analysis of employment injuries affecting specific bodily locations to assist in the development of programs to counteract such injuries (for example, eye injuries via an eye protection program). To enable more detailed analysis of the nature of the employment injury.

Coding: Numeric Format, variable integer.

Length: 3.

Comment: The classification has been structured hierarchically.

The classification relies on sufficient detail being available to allow the allocation of appropriate codes. Where the description given on the report form is inadequate, coders may submit that nature with an explanation to the online database.

5.1.3.d. Mechanism of Injury or Disease

Description: The mechanism of injury/disease is intended to identify the action, exposure or event which was the direct cause of the most serious injury or disease.

Purpose: To facilitate analysis of the circumstances and events leading to the employment injury enabling the identification of hazardous work practices or environments and development of suitable prevention strategies.

Coding: Format numeric, variable integer.

Length: 2.

Comment: The classification has been structured hierarchically.

The classification relies on sufficient detail being available to allow the allocation of appropriate codes. Where the description given on the report form is inadequate, coders may submit that nature with an explanation to the online database.

5.1.3.e. Agency of Injury or Disease

Description: The agency of injury/disease refers to the object, substance or circumstance directly involved in inflicting the most serious injury or disease.

Purpose: To facilitate analysis of the circumstances and events leading to the employment injury enabling the identification of hazardous work practices or environments and, ultimately, the development of suitable prevention strategies.

Coding: Format numeric, variable integer.

Length: 3

Comment: The classification has been structured hierarchically.

The classification relies on sufficient detail being available to allow the allocation of appropriate codes. Where the description given on the report form is inadequate, coders may submit that nature with an explanation to the online database.

5.1.3.f. Breakdown Agency of Injury or Disease

Description: The breakdown agency is intended to identify the object, substance or circumstance that was principally involved in, or most closely associated with, the point at which things started to go wrong and which ultimately led to the most serious injury or disease.

Purpose: To facilitate analysis of the circumstances and events leading to the employment injury enabling the identification of hazardous work practices or environments and, ultimately, the development of suitable prevention strategies.

Coding: Format Numeric, variable integer.

Length: 3

Comment: The classification has been structured hierarchically.

The classification relies on sufficient detail being available to allow the allocation of appropriate codes. Where the description given on the report form is inadequate, coders may submit that nature with an explanation to the online database.

5.1.4.a. Time Lost due to Injury

Description: The number of hours and minutes lost for which compensation was paid by any party (for example, employer, insurer, work cover authority).

Purpose: To provide an indication of the severity of injury/disease occurrences and to enable data to be analysed on a more comparable basis.

Coding: Format numeric, specified as HHHHHMM and zero filled. The number of hours and minutes lost for which compensation was paid. If number of hours lost are reported in whole hours and parts thereof then the decimal place should be converted to minutes (for example, 37.5 hours equates to 37 hours and 30 minutes). If minutes are not known the MM columns should be zero filled.

Length: 7.

Comment: Collection of the number of hours and minutes lost for which compensation was paid will enable data to be presented on a more comparable basis.

Cases not finalized at the end of the reporting period will not require an estimate of future time lost to be included. However, time lost should be added when the case is finalized.

5.1.4.b. Severity Indicator

Description: The severity indicator categorises cases as either fatal, total or partial permanent incapacity or temporary incapacity.

Purpose: To provide indicators of the severity of work-related injury and disease.

Coding: Format numeric, variable integer. To be coded according to the following classification:

- | | |
|---|---------------------------------------|
| 1 | Fatal |
| 2 | Total or partial permanent incapacity |
| 3 | Temporary incapacity |

Length: 1.

Comment: Chapter 2.9. includes definitions related with this parameter.

5.1.4.c. Payments Made

Definition: All payments made in respect of the injury or disease. This includes payments below any non-compensable payments thresholds operated by workers' compensation authorities.

Purpose: To provide an indicator of the economic cost of employment injuries and provide a basis for evaluation of planned preventive strategies.

Coding: To be collected in TL. Format numeric, variable integer.

Comment: Cases not finalized at the end of the reporting period will not require an estimate of future payments to be included. However, payments should be updated each when the case is finalized.

Payments, as they are individually recorded by workers' compensation authorities, are to be reported as:

- compensation or sustenance payments to worker or worker's family;
- non-compensation payments.

Payments made is calculated by adding non-compensation and compensation payments

5.1.4.d. Compensation Payments:

Definition: Any compensable payments thresholds operated by workers' compensation authorities.

Purpose: To provide an indicator of the economic cost of employment injuries and provide a basis for evaluation of planned preventive strategies.

Coding: To be collected in TL. Format numeric, variable integer.

Comment: Cases includes;

- Death
- Lump sums for permanent injury
- Lump sums for pain and suffering
- Lump sums for redemption
- Partially incapacitated compensated as fully incapacitated
- Weekly payments for total incapacity

- Weekly payments for partial incapacity

Length: 8.

5.1.4.e. Non-compensation Payments:

Definition: Any non-compensable payments thresholds operated by workers' compensation authorities.

Purpose: To provide an indicator of the economic cost of employment injuries and provide a basis for evaluation of planned preventive strategies.

Coding: To be collected in TL. Format numeric, variable integer.

Comment: Cases includes;

- Transport and maintenance
- Investigation expenses
- Damages
- Ambulance services
- Accommodation expenses
- Funeral expenses
- Medical treatment
- Hospital treatment
- Rehabilitation services
- Physiotherapy services

Length: 8.

5.1.4.f. DIFR:

Definition: Calculated from related fields.

Coding: Format numeric, variable integer.

Length: 4.

5.1.4.g. FIFR:

Definition: Calculated from related fields.

Coding: Format numeric, variable integer.

Length: 8.

5.1.4.h. AFR:

Definition: Calculated from related fields.

Coding: Format numeric, variable integer.

Length: 2.

5.1.4.i. ASR:

Definition: Calculated from related fields.

Coding: Format numeric, variable integer.

Length: 2.

APPENDIX B

Mechanism of Injury/Disease Details

5.2.1. Injury And Poisoning

Code Mechanism

- 010 Fractures
- 020 Fracture of vertebral column with or without mention of spinal cord lesion
- 030 Dislocation
- 040 Sprains and strains of joints and adjacent muscles
- 050 Intracranial injury, including concussion
- 060 Internal injury of chest, abdomen and pelvis
- 070 Traumatic amputation including enucleation of eye (loss of eyeball)
- 080 Open wound not involving traumatic amputation
- 090 Superficial injury
- 100 Contusion with intact skin surface and crushing injury excluding those with fracture
- 110 Foreign body on external eye, in ear or nose or in respiratory, digestive or reproductive systems
- 120 Burns
- 130 Injuries to nerves and spinal cord without evidence of spinal bone injury
- 140 Poisoning and toxic effects of substances
- 150 Effects of weather, exposure, air pressure and other external causes, not elsewhere classified
- 160 Multiple injuries

This code should only be used where no principal injury can be identified.

- 170 Damage to artificial aid(s)
- 180 Other and unspecified injuries

5.2.2. Diseases

Code Mechanism

- 250 Deafness
- 260 Disorders of muscle, tendons and other soft tissues
- 270 Asbestosis, Pneumoconiosis due to other silica or silicates
- 280 Chronic bronchitis, emphysema and allied conditions
- 290 Mental disorders
- 300 Other diseases

APPENDIX C

Bodily Location of Injury/Disease Classification Details

5.3.1. Major Groups

1. Head
2. Neck
3. Trunk
4. Upper Limbs
5. Lower Limbs
6. Multiple Locations
7. Systemic Locations
8. Non-Physical Locations
9. Unspecified Locations

5.3.2. Subgroups:

Group 1: Head

11. Skull
12. Eye
13. Ear
14. Mouth
15. Nose
16. Face, not elsewhere classified
18. Head - multiple locations

19. Head - unspecified locations

Group 2: Neck

21 Neck

Group 3: Trunk

31. Back - upper or lower

32. Chest (thorax)

33. Abdomen

34. Pelvic region

35. Trunk - multiple locations

36. Trunk - unspecified locations

Group 4: Upper Limbs

41. Shoulder

42. Upper arm

43. Elbow

44. Forearm

45. Wrist

46. Hand, fingers and thumb

47. Upper limb - multiple locations

48. Upper limb - unspecified locations

Group 5: Lower Limbs

51. Hip

52. Upper leg

53. Knee

- 54. Lower leg
- 55. Ankle
- 56. Foot and toes
- 57. Lower limb - multiple locations
- 58. Lower limb - unspecified locations

Group 6 : Multiple Locations

Multiple locations can be selected from database. Connection with “and logic” is used in such a case.

Group 7: Systemic Locations

- 71. Circulatory system
- 72. Respiratory system
- 73. Digestive system
- 74. Genitourinary system
- 75. Nervous system
- 76. Other and multiple systemic conditions
- 77. Unspecified systemic conditions

Group 8: Non-Physical Locations

- 81. Psychological system

Group 9: Unspecified Locations

- 91. Unspecified locations

APPENDIX D

Mechanism of Injury/Disease Classification Details

5.4.1. Major Groups

0. Falls, trips and slips of a person
1. Hitting objects with a part of the body
2. Being hit by moving objects
3. Sound and pressure
4. Heat and Electricity
5. Other and unspecified mechanisms of injury

5.4.2. Subgroups

Group 0: Falls, Trips And Slips Of A Person

When the injuries are a direct result of the action of falling, or in trying to recover from a fall, this group is used. It includes intentional jumps as well as unintentional falls.

- 01 Falls from a height
- 02 Falls on the same level
- 03 Stepping, kneeling or sitting on objects

Group 1: Hitting Objects With A Part Of The Body

This group is used when the injuries result from the motion of the person when hitting, grasping or otherwise striking objects.

Include as objects: machines, vehicles, appliances, equipment, tools, materials, substances, animals and other people

Include as hitting: brushing, bumping, colliding, cutting, grabbing, grasping, leaning on, piercing, rubbing, slicing, stabbing, caught, striking and touching

- 11 Hitting stationary objects
- 12 Hitting moving objects
- 13 Rubbing and chafing

Group 2: Being Hit By Moving Objects

Use this group when the injury is a result of the action of an object hitting the person.

- 21 Being hit by falling objects
- 22 Being hit by a person accidentally
 - Includes: all accidental pushes, knocks, bumps
- 23 Being trapped by moving machinery or equipment
 - Includes: being caught in, or by, equipment in operation
- 24 Being trapped between stationary and moving objects
 - Includes: being caught or trapped between moving and stationary objects where the objects are not part of the same machine or equipment
- 25 Exposure to mechanical vibration
 - Includes:
 - mechanical vibration from chainsaws, jackhammers
 - vibration transmitted through the seat or other parts of vehicles
- 26 Being hit by moving objects
 - Includes:
 - moving vehicles
 - moving parts of operating equipment

- 27 Being assaulted by a person or persons
Includes: deliberate kicks, bites, punches, pushes etc
Excludes: all accidental knocks by a person

Group 3: Sound And Pressure

Use this group when injuries or disorders are a result of sound or changes in pressure.

- 31 Exposure to single, sudden sound
Includes: deafness as a result of an explosion or implosion
- 32 Long term exposure to sounds

Group 4: Heat and Electricity

- 40 Contact with electricity
Includes:
 - all forms of electric current
 - struck by lightning
 - all forms of static electricity
- 41 Exposure to environmental heat
Includes: mechanisms producing sunstroke, heat exhaustion, dehydration
- 42 Exposure to environmental cold
Includes: mechanisms producing hypothermia

Group 5: Other And Unspecified Mechanisms Of Injury

Use this group to code mechanisms not classifiable to other groups.

- 51 Slide or cave-in
Includes:
 - earth, rock, stone or grain fall

- mining cave-ins
- mud slides, avalanches
- trench collapse

52 Vehicle accident

A vehicle being a means of transport by rail, road, water or air. Examples are locomotives, trolleys, trucks, shovels etc. This code is to be used for any accident or incident involving a vehicle

Includes:

- any accident or incident on a private road, farm, mine site or footpath involving a vehicle where the most serious injury is sustained as a result of that accident or incident
- vehicle catching on fire after accident
- any accident or incident in a factory, mine or carpark involving a fall from a moving vehicle

53 Other and multiple mechanisms of injury

Includes:

- occurrences where more than one mechanism of injury code is indicated, and the most serious personal damage code cannot be identified
- occurrences where the mechanism is not classifiable to other codes
- constant pressure from kneeling etc
- swallowing or choking on objects

54 Unspecified mechanisms of injury

Includes:

- occurrences where the mechanism of injury cannot be identified

APPENDIX E

Agency Classification Details

5.5.1. Major Groups

- 1 Machinery and Fixed Plant
- 2 Mobile Plant and Transport
- 3 Powered Equipment, Tools and Appliances
- 4 Non-Powered Handtools, Appliances and Equipment
- 5 Chemicals and Chemical Products
- 6 Materials and Substances
- 7 Environmental Agencies
- 8 Human and Agencies
- 9 Other and Unspecified Agencies

5.5.2. Subgroups:

Group 1: Machinery and (Mainly) Fixed Plant

- 11 Cutting, slicing machinery
- 12 Crushing, pressing, rolling machinery
- 13 Conveyors and lifting plant
- 14 Electrical installation
- 15 Other plant and machinery

Group 2: Mobile Plant and Transport

- 21 Self-propelled plant
- 22 Semi-portable plant
- 23 Other mobile plant
- 24 Road transport
- 25 Rail transport
- 26 Air transport
- 27 Other transport

Group 3: Powered Equipment, Tools and Appliances

- 31 Workshop and worksite tools and equipment
- 32 Other powered equipment, tools and appliances

Group 4: Non-Powered Handtools, Appliances and Equipment

- 41 Handtools, non-powered, edged
- 42 Ladders, mobile ramps and stairways, and scaffolding
- 49 Other non-powered equipment

Group 5: Chemicals and Chemical Products

- 51 Nominated chemicals
- 52 Other basic chemicals
- 53 Chemical products

Group 6: Materials and Substances

- 61 Non-metallic minerals and substances
- 62 Other materials and objects
- 63 Other substances

Group 7: Environmental Agencies

- 71 Roof of tunnel/mine
- 72 Face, walls of tunnel/mine
- 73 Wet, oily or icy floor of tunnel/mine
- 74 Hazardous objects on floor of tunnel/mine
- 75 Floor of tunnel/mine other
- 76 Shaft
- 77 Underground fire
- 78 Underground water
- 79 Supporting structures
- 80 Other and multiple underground environment

Group 8: Human Agencies

- 81 Human agencies

Group 9: Other and Unspecified Agencies

- 91 Non-physical agencies
- 92 Other and unspecified agencies

APPENDIX F

WRITTEN PHP CODES

Database Administration PHP Code:

```
<?php

include ("DB.php");
session_start();
include ("LOGIN.php");
if ($admin_boolean == 'F')
{
    print "Unauthorized User!<br>";
    exit;
}
else
{
    if ($record_agency)
    {
        print "<br><b><u>Currently registered agencies:</u></b><br>";
        $query = "select definition from AC_agency";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            print "$fields[0]<br>";
        }
        mssql_free_result($result);
        print "
            <form method=post action=\"ADMIN.php\">
            New Agency: <input type=text
name=\"form_new_agency\"><br>
            <input type=submit name=\"record_agency_submit\" value=\"
Record  \">>
            </form>
        ";
    }

    elseif ($record_agency_submit)
    {
        $query = "insert into AC_agency values('$form_new_agency')";
```

```

$result = mssql_query($query,$db);
print "
    <br>Record Complete.<br>
    New Agency: $form_new_agency<p>
    <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
    ";
}

elseif ($record_subagency)
{
    print "<form method=post action=\"ADMIN.php\">
        <b><u>Select Agency:</u></b><br>
        <select name=\"form_agency_id\">";
    $query = "select agent_id,definition from AC_agency";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "<option value=\"".$fields[0].\">".$fields[1].\"n\"";
    }
    mssql_free_result($result);
    print "
        </select>
        <input type=submit name=\"record_subagency_submit\"
value=\" Next \">
        </form>
    ";
}

elseif ($record_subagency_submit)
{
    print "<br><b><u>Currently registered subagencies:</u></b><br>";
    $query = "select definition from AC_subagency where
agent_id=$form_agency_id";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "$fields[0]<br>";
    }
    mssql_free_result($result);
    print "
        <form method=post action=\"ADMIN.php\">
        <input type=hidden name=\"form_agency_id\"
value=\"$form_agency_id\">
        New Subagency: <input type=text
name=\"form_new_subagency\"><br>
        <input type=submit name=\"record_subagency_submit_last\"
value=\" Record \">
        </form>
    ";
}

```

```

        ";
    }

    elseif ($record_subagency_submit_last)
    {
        $query = "insert into AC_subagency
values('$form_agency_id','$form_new_subagency')";
        $result = mssql_query($query,$db);
        print "
        <br>Record Complete.<br>
        New Subagency: $form_new_subagency<p>
        <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
        ";
    }

    if ($record_mechanism)
    {
        print "<br><b><u>Currently registered mechanisms:</u></b><br>";
        $query = "select definition from AC_mechanism";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            print "$fields[0]<br>";
        }
        mssql_free_result($result);
        print "
        <form method=post action=\"ADMIN.php\">
        New Mechanism: <input type=text
name=\"form_new_mechanism\"><br>
        <input type=submit name=\"record_mechanism_submit\"
value=\" Record \"/>
        </form>
        ";
    }

    elseif ($record_mechanism_submit)
    {
        $query = "insert into AC_mechanism
values('$form_new_mechanism')";
        $result = mssql_query($query,$db);
        print "
        <br>Record Complete.<br>
        New Mechanism: $form_new_mechanism<p>
        <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
        ";
    }
}

```

```

elseif ($record_submechanism)
{
    print "<form method=post action=\"ADMIN.php\">
        <b><u>Select mechanism:</u></b><br>
        <select name=\"form_mechanism_id\">";
    $query = "select mech_id,definition from AC_mechanism";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "<option value=\"\$fields[0]\"> \$fields[1]\n";
    }
    mssql_free_result($result);
    print "
        </select>
        <input type=submit name=\"record_submechanism_submit\"
value=\" Next \">>
        </form>
    ";
}

elseif ($record_submechanism_submit)
{
    print "<br><b><u>Currently registered
submechanisms:</u></b><br>";
    $query = "select description from AC_submechanism where
mech_id=$form_mechanism_id";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "$fields[0]<br>";
    }
    mssql_free_result($result);
    print "
        <form method=post action=\"ADMIN.php\">
        <input type=hidden name=\"form_mechanism_id\"
value=\"\$form_mechanism_id\">
        New Submechanism: <input type=text
name=\"form_new_submechanism\"><br>
        <input type=submit
name=\"record_submechanism_submit_last\" value=\" Record \">>
        </form>
    ";
}

elseif ($record_submechanism_submit_last)
{
    $query = "insert into AC_submechanism
values('$form_mechanism_id','$form_new_submechanism')";
    $result = mssql_query($query,$db);
}

```

```

        print "
            <br>Record Complete.<br>
            New Submechanism: $form_new_submechanism<p>
            <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
        ";
    }

    if ($record_body_location)
    {
        print "<br><b><u>Currently registered body
locations:</u></b><br>";
        $query = "select description from AC_body_location";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            print "$fields[0]<br>";
        }
        mssql_free_result($result);
        print "
            <form method=post action=\"ADMIN.php\">
            New body location: <input type=text
name=\"form_new_body_location\"><br>
            <input type=submit name=\"record_body_location_submit\"
value=\"    Record    \">
            </form>
        ";
    }

    elseif ($record_body_location_submit)
    {
        $query = "insert into AC_body_location
values('$form_new_body_location')";
        $result = mssql_query($query,$db);
        print "
            <br>Record Complete.<br>
            New body location: $form_new_body_location<p>
            <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
        ";
    }

    elseif ($record_body_sublocation)
    {
        print "<form method=post action=\"ADMIN.php\">
            <b><u>Select body location:</u></b><br>
            <select name=\"form_body_location_id\">";
        $query = "select body_location_id,description from
AC_body_location";
    }

```

```

$result = mssql_query($query,$db);
while ($fields = mssql_fetch_row($result))
{
    print "<option value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
print "
    </select>
    <input type=submit
name=\"record_body_sublocation_submit\" value=\"  Next  \">
    </form>
";
}

elseif ($record_body_sublocation_submit)
{
    print "<br><b><u>Currently registered body
sublocations:</u></b><br>";
    $query = "select description from AC_body_sublocation where
body_location_id=$form_body_location_id";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "\$fields[0]<br>";
    }
    mssql_free_result($result);
    print "
        <form method=post action=\"ADMIN.php\">
        <input type=hidden name=\"form_body_location_id\"
value=\"\$form_body_location_id\">
        New body sublocation: <input type=text
name=\"form_new_body_sublocation\"><br>
        <input type=submit
name=\"record_body_sublocation_submit_last\" value=\"  Record  \">
        </form>
";
}

elseif ($record_body_sublocation_submit_last)
{
    $query = "insert into AC_body_sublocation
values('$form_body_location_id','$form_new_body_sublocation)";
    $result = mssql_query($query,$db);
    print "
        <br>Record Complete.<br>
        New body sublocation: \$form_new_body_sublocation<p>
        <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
";
}

```

```

}

if ($record_results)
{
    print "<br><b><u>Currently registered results:</u></b><br>";
    $query = "select definition from AC_results";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "$fields[0]<br>";
    }
    mssql_free_result($result);
    print "
        <form method=post action=\"ADMIN.php\">
        New result: <input type=text
name=\"form_new_results\"><br>
        <input type=submit name=\"record_results_submit\" value=\"
Record  \">>
        </form>
    ";
}

elseif ($record_results_submit)
{
    $query = "insert into AC_results values('$form_new_results')";
    $result = mssql_query($query,$db);
    print "
        <br>Record Complete.<br>
        New result: $form_new_results<p>
        <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
    ";
}

elseif ($record_subresults)
{
    print "<form method=post action=\"ADMIN.php\">
        <b><u>Select result:</u></b><br>
        <select name=\"form_results_id\">";
    $query = "select results_id,definition from AC_results";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "<option value=\"\$fields[0]\"> \$fields[1]\n";
    }
    mssql_free_result($result);
    print "
        </select>

```

```

        <input type=submit name=\"record_subresults_submit\"
value=\" Next \">>
        </form>
    ";
}

elseif ($record_subresults_submit)
{
    print "<br><b><u>Currently registered subresults:</u></b><br>";
    $query = "select description from AC_subresults where
results_id=$form_results_id";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "$fields[0]<br>";
    }
    mssql_free_result($result);
    print "
        <form method=post action=\"ADMIN.php\">
        <input type=hidden name=\"form_results_id\"
value=\"$form_results_id\">
        New Subresult: <input type=text
name=\"form_new_subresults\"><br>
        <input type=submit name=\"record_subresults_submit_last\"
value=\" Record \">>
        </form>
    ";
}

elseif ($record_subresults_submit_last)
{
    $query = "insert into AC_subresults
values('$form_results_id','$form_new_subresults)";
    $result = mssql_query($query,$db);
    print "
        <br>Record Complete.<br>
        New Subresult: $form_new_subresults<p>
        <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
    ";
}

elseif ($record_company)
{
    print "<br><b><u>Currently registered companies:</u></b><br>";
    $query = "select company_name from AC_company";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {

```

```

        print "$fields[0]<br>";
    }
    mssql_free_result($result);
    print "
        <form method=post action=\"ADMIN.php\">
        <b></u>New company:</u></b><br>
        Company Name: <input type=text
name=\"form_new_company_name\"><br>
        Company Address: <input type=text
name=\"form_new_company_address\"><br>
        Number of Underground Workers: <input type=text
name=\"form_new_company_underground\"><br>
        Number of Open Pit Workers: <input type=text
name=\"form_new_company_openpit\"><br>
        <input type=submit name=\"record_company_submit\"
value=\" Record \">
        </form>
    ";
}

elseif ($record_company_submit)
{
    $query = "insert into AC_company
values('$form_new_company_name','$form_new_company_address','$form_new_co
mpany_underground','$form_new_company_openpit')";
    $result = mssql_query($query,$db);
    print "
        <br>Record Complete.<br>
        New Company:<br>
        Company Name: $form_new_company_name<br>
        Company Address: $form_new_company_address<br>
        Number of Underground Workers:
$form_new_company_underground<br>
        Number of Open Pit Workers:
$form_new_company_openpit<p>
        <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
    ";
}

elseif ($record_mine)
{
    print "<form method=post action=\"ADMIN.php\">
        <b></u>Select Company:</u></b><br>
        <select name=\"form_company_id\">";
    $query = "select company_id,company_name from AC_company";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {

```

```

        print "<option value=\"\$fields[0]\">\$fields[1]\n";
    }
    mssql_free_result($result);
    print "
        </select>
        <input type=submit name=\"record_mine_submit\" value=\"
Next  \">
        </form>
    ";
}

elseif ($record_mine_submit)
{
    print "<br><b><u>Currently registered mines:</u></b><br>";
    $query = "select mine_name from AC_mines where
company_id=$form_company_id";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "$fields[0]<br>";
    }
    mssql_free_result($result);
    print "
        <form method=post action=\"ADMIN.php\">
        <input type=hidden name=\"form_company_id\"
value=\"\$form_company_id\">
        <b></u>New mine:</u></b><br>
        Mine Name: <input type=text
name=\"form_new_mine_name\"><br>
        Number of Shifts: <input type=text
name=\"form_new_mine_shifts\"><br>
        Shift Duration: <input type=text
name=\"form_new_mine_sduration\"><br>
        <input type=submit name=\"record_mine_submit_last\"
value=\" Record  \">
        </form>
    ";
}

elseif ($record_mine_submit_last)
{
    $query = "insert into AC_mines
values('$form_company_id','$form_new_mine_name','$form_new_mine_shifts','$for
m_new_mine_sduration)";
    $result = mssql_query($query,$db);
    print "
        <br>Record Complete.<br>
        New Mine:<br>
        Mine Name: $form_new_mine_name<br>

```

```

        Number of Shifts: $form_new_mine_shifts<br>
        Shift Duration: $form_new_mine_sduration<p>
        <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
        ";
    }

elseif ($record_user)
{
    print "<form method=post action=\"ADMIN.php\">
        <b><u>Select Company:</u></b><br>
        <select name=\"form_company_id\">";
    $query = "select company_id,company_name from AC_company";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "<option value=\"\$fields[0]\"> \$fields[1]\n";
    }
    mssql_free_result($result);
    print "
        </select>
        <input type=submit name=\"record_user_submit\" value=\"
Next \">>
        </form>
        ";
    }

elseif ($record_user_submit)
{
    print "<form method=post action=\"ADMIN.php\">
        <b><u>Select Mine:</u></b><br>
        <select name=\"form_mine_id\">";
    $query = "select mine_id,mine_name from AC_mines where
company_id=$form_company_id";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "<option value=\"\$fields[0]\"> \$fields[1]\n";
    }
    mssql_free_result($result);
    print "
        </select><br>
        Username: <input type=text name=\"form_username\"
maxlength=16>(max. 16 chars)<br>
        Password: <input type=text name=\"form_password\"
maxlength=16>(max. 16 chars)<br>
        <input type=submit name=\"record_user_submit_last\"
value=\" Record \">>
        </form>

```

```

        ";
    }

    elseif ($record_user_submit_last)
    {
        $query = "insert into AC_users
values('$form_username','$form_password','$form_mine_id','F')";
        $result = mssql_query($query,$db);
        print "
        <br>Record Complete.<br>
        New User:<br>
        User Name: $form_username<br>
        Password: $form_password<p>
        <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
        ";
    }

    elseif ($record_personnel)
    {
        print "<form method=post action=\"ADMIN.php\">
        <b><u>Select Company:</u></b><br>
        <select name=\"form_company_id\">";
        $query = "select company_id,company_name from AC_company";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            print "<option value=\"".$fields[0].">".$fields[1]."\n";
        }
        mssql_free_result($result);
        print "
        </select>
        <input type=submit name=\"record_personnel_submit\"
value=\" Next \">
        </form>
        ";
    }

    elseif ($record_personnel_submit)
    {
        print "<form method=post action=\"ADMIN.php\">
        <b><u>Select Mine:</u></b><br>
        <select name=\"form_mine_id\">";
        $query = "select mine_id,mine_name from AC_mines where
company_id=$form_company_id";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            print "<option value=\"".$fields[0].">".$fields[1]."\n";

```

```

    }
    mssql_free_result($result);
    print "
        </select><br>
        Name: <input type=text name=\"form_name\"><br>
        Gender: <select name=\"form_gender\"><option
value=\"M\">Male<option value=\"F\">Female</select><br>
        Birth Date: <input type=text
name=\"form_birthdate\">(yyyymmdd)<br>
        Insurance No: <input type=text
name=\"form_insuranceno\"><br>
        Job Title: <input type=text name=\"form_job_title\"><br>
        <input type=submit name=\"record_personnel_submit_last\"
value=\" Record \">
        </form>
    ";
}

elseif ($record_personnel_submit_last)
{
    $query = "insert into AC_personnel
values('$form_name','$form_gender','$form_birthdate','$form_insuranceno','$form_j
ob_title','$form_mine_id')";
    $result = mssql_query($query,$db);
    print "
        <br>Record Complete.<br>
        New Personnel:<br>
        Name: $form_name<br>
        Gender: $form_gender<br>
        Birth Date: $form_birthdate<br>
        Insurance No: $form_insuranceno<br>
        Job Title: $form_job_title<p>
        <a href=\"ADMIN.php\">Return to Admin main
page</a><br>
    ";
}

else
{
    print "
        <form method=\"post\" action=\"ADMIN.php\">
        <input type=\"submit\" name=\"record_agency\" value=\"New
Agency Record\">
        </form>

        <form method=\"post\" action=\"ADMIN.php\">
        <input type=\"submit\" name=\"record_subagency\"
value=\"New Subagency Record\">
        </form>
    ";
}

```

```

        <form method="post" action="ADMIN.php">
        <input type="submit" name="record_mechanism"
value="New Mechanism Record">
        </form>

        <form method="post" action="ADMIN.php">
        <input type="submit" name="record_submechanism"
value="New Submechanism Record">
        </form>

        <form method="post" action="ADMIN.php">
        <input type="submit" name="record_body_location"
value="New Body Location Record">
        </form>

        <form method="post" action="ADMIN.php">
        <input type="submit" name="record_body_sublocation"
value="New Body Sublocation Record">
        </form>

        <form method="post" action="ADMIN.php">
        <input type="submit" name="record_results" value="New
Result Record">
        </form>

        <form method="post" action="ADMIN.php">
        <input type="submit" name="record_subresults"
value="New Subresult Record">
        </form>

        <form method="post" action="ADMIN.php">
        <input type="submit" name="record_company"
value="New Company Record">
        </form>

        <form method="post" action="ADMIN.php">
        <input type="submit" name="record_mine" value="New
Mine Record">
        </form>

        <form method="post" action="ADMIN.php">
        <input type="submit" name="record_user" value="New
User Record">
        </form>

        <form method="post" action="ADMIN.php">
        <input type="submit" name="record_personnel"
value="New Personnel Record">

```

```
}  
?>  
}  
};  
</form>
```

Login PHP Code:

```
<?php

if ($authorized==1)
{
    print "<b>Current User:</b> $user_name<br>";
    print "<b>Company Name:</b>
$company_name<br>";
    print "<b>Mine Name:</b> $mine_name<br>";
}
else
{
    if ($submit_user)
    {
        if ($form_user_name && $form_password)
        {
            $query = "select
u.user_id,u.user_name,u.mine_id,u.admin_boolean,m.company_id,c.company_name,
m.mine_name from AC_users u, AC_mines m, AC_company c where
(u.user_name='$form_user_name' and u.password='$form_password' and
u.mine_id=m.mine_id and m.company_id=c.company_id)";
            $result = mssql_query($query,$db);
            $numrows=mssql_num_rows($result);
            if (!$numrows) die(print "Unauthorized
User!");
            else
            {
                session_register("user_id");
                session_register("user_name");
                session_register("mine_id");

                session_register("admin_boolean");
                session_register("company_id");

                session_register("company_name");
                session_register("mine_name");
                while ($fields =
mssql_fetch_row($result))
                {
                    $user_id = $fields[0];
                    $user_name = $fields[1];
                    $mine_id = $fields[2];
                    $admin_boolean =
$fields[3];
                    $company_id =
$fields[4];
                    $company_name =
$fields[5];
```

```

$mine_name =
$fields[6];
}
mssql_free_result($result);

session_register("authorized");$authorized=1;
print "<b>Current User:</b>"

$user_name<br>";
print "<b>Company Name:</b>"

$company_name<br>";
print "<b>Mine Name:</b>"

$mine_name<br>";
}
}
else
{
    if ($authorized != 1)
    {
        print "
        <form method=\"post\"
        User Name: <input type=text
        Password: <input
        <input type=\"submit\"
        name=\"submit_user\" value=\" Login  \">>
        <p>
        </form>
        ";
        session_unregister("user_id");

        session_unregister("user_name");
        session_unregister("mine_id");

        session_unregister("admin_boolean");

        session_unregister("company_id");

        session_unregister("company_name");

        session_unregister("authorized");
        session_destroy();
        exit;
    }
}
else

```

```

        {
            if ($authorized != 1)
            {
                print "
                <form method=\"post\"
                User Name: <input type=text
                Password: <input type=password
                <input type=\"submit\"
                name=\"submit_user\" value=\" Login \"/>
                <p>
                </form>
                ";
                session_unregister("user_id");
                session_unregister("user_name");
                session_unregister("mine_id");
                session_unregister("admin_boolean");
                session_unregister("company_id");
                session_unregister("company_name");
                session_unregister("authorized");
                session_destroy();
                exit;
            }
        }
    }
?>

```

Statistical Calculations PHP Code:

```
<?php
include ("DB.php");

if ($calculate_DIFR)
{
    print "<form method=post action=\"CALC.php\">
        <b><u>Select Company:</u></b><br>
        <select name=\"form_company_id\">";
    $query = "select company_id,company_name from
AC_company";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "<option
value=\"\$fields[0]\">\$fields[1]\n";
    }
    mssql_free_result($result);
    print "
        </select>
        <input type=submit
name=\"calculate_DIFR_submit\" value=\" Next \">
        </form>
";
}

elseif ($calculate_DIFR_submit)
{
    print "<form method=post action=\"CALC.php\">
        <b><u>Select Mine:</u></b><br>
        <select name=\"form_mine_id\">";
    $query = "select mine_id,mine_name from
AC_mines where company_id=\$form_company_id";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "<option
value=\"\$fields[0]\">\$fields[1]\n";
    }
    mssql_free_result($result);
    print "
        </select><br>
        <input type=submit
name=\"calculate_DIFR_submit_last\" value=\" Calculate \">
        </form>
";
}
}
```

```

elseif ($calculate_DIFR_submit_last)
{
    $query = "select count(*) from AC_accident where
(accident_result_id!='5' and mine_id='$form_mine_id')";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $number_of_disabling_inj = $fields[0];
    }
    mssql_free_result($result);

    $query = "select shift_duration from AC_mines
where (mine_id='$form_mine_id')";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $shift_duration = $fields[0];
    }
    mssql_free_result($result);

    $query = "select count(*) from AC_personnel
where (mine_id='$form_mine_id')";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $personnel_number = $fields[0];
    }
    mssql_free_result($result);
    $DIFR = ($number_of_disabling_inj * 1000000) /
($shift_duration * $personnel_number * 300);
    print "
        DIFR = ($number_of_disabling_inj * 1000000)
/ ($shift_duration * $personnel_number * 300)<br>
        DIFR = $DIFR (According to hours of
exposure)
";
}

elseif ($calculate_FIFR)
{
    print "<form method=post action=\"CALC.php\">
        <b><u>Select Company:</u></b><br>
        <select name=\"form_company_id\">";
    $query = "select company_id,company_name from
AC_company";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {

```

```

        print "<option
value=\"\$fields[0]\"> \$fields[1]\n";
    }
    mssql_free_result($result);
    print "
        </select>
        <input type=submit
name=\"calculate_FIFR_submit\" value=\"  Next  \"/>
        </form>
    ";
}

elseif ($calculate_FIFR_submit)
{
    print "<form method=post action=\"CALC.php\">
        <b><u>Select Mine:</u></b><br>
        <select name=\"form_mine_id\">";
    $query = "select mine_id,mine_name from
AC_mines where company_id=$form_company_id";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "<option
value=\"\$fields[0]\"> \$fields[1]\n";
    }
    mssql_free_result($result);
    print "
        </select><br>
        <input type=submit
name=\"calculate_FIFR_submit_last\" value=\"  Calculate  \"/>
        </form>
    ";
}

elseif ($calculate_FIFR_submit_last)
{
    $query = "select count(*) from AC_accident where
(accident_result_id='5' and mine_id='$form_mine_id)";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $number_of_disabling_inj = $fields[0];
    }
    mssql_free_result($result);

    $query = "select shift_duration from AC_mines
where (mine_id='$form_mine_id)";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))

```

```

        {
            $shift_duration = $fields[0];
        }
        mssql_free_result($result);

        $query = "select count(*) from AC_personnel
where (mine_id='$form_mine_id')";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            $personnel_number = $fields[0];
        }
        mssql_free_result($result);
        $FIFR = ($number_of_disabling_inj * 1000000) /
($shift_duration * $personnel_number * 300);
        print "
        FIFR = ($number_of_disabling_inj * 1000000)
/ ($shift_duration * $personnel_number * 300)<br>
        FIFR = $FIFR (According to hours of
exposure)
";
    }

elseif ($calculate_AFR)
{
        print "<form method=post action=\"CALC.php\">
        <b><u>Select Company:</u></b><br>
        <select name=\"form_company_id\">";
        $query = "select company_id,company_name from
AC_company";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            print "<option
value=\"\$fields[0]\">\$fields[1]\n";
        }
        mssql_free_result($result);
        print "
        </select>
        <input type=submit
name=\"calculate_AFR_submit\" value=\" Next \"/>
        </form>
";
    }

elseif ($calculate_AFR_submit)
{
        print "<form method=post action=\"CALC.php\">
        <b><u>Select Mine:</u></b><br>

```

```

                <select name=\`form_mine_id\`">";
                $query = "select mine_id,mine_name from
AC_mines where company_id=$form_company_id";
                $result = mssql_query($query,$db);
                while ($fields = mssql_fetch_row($result))
                {
                    print "<option
value=\`$fields[0]\`">$fields[1]\n";
                }
                mssql_free_result($result);
                print "
                </select><br>
                <input type=submit
name=\`calculate_AFR_submit_last\`" value=\`    Calculate    \`">
                </form>
                ";
            }

elseif ($calculate_AFR_submit_last)
{
                $query = "select count(*) from AC_accident where
(mine_id='\`$form_mine_id'\`");
                $result = mssql_query($query,$db);
                while ($fields = mssql_fetch_row($result))
                {
                    $number_of_disabling_inj = $fields[0];
                }
                mssql_free_result($result);

                $query = "select shift_duration from AC_mines
where (mine_id='\`$form_mine_id'\`");
                $result = mssql_query($query,$db);
                while ($fields = mssql_fetch_row($result))
                {
                    $shift_duration = $fields[0];
                }
                mssql_free_result($result);

                $query = "select count(*) from AC_personnel
where (mine_id='\`$form_mine_id'\`");
                $result = mssql_query($query,$db);
                while ($fields = mssql_fetch_row($result))
                {
                    $personnel_number = $fields[0];
                }
                mssql_free_result($result);
                $AFR = ($number_of_disabling_inj * 1000000) /
($shift_duration * $personnel_number * 300);
                print "

```

```

                                AFR = ($number_of_disabling_inj * 1000000)
/ ($shift_duration * $personnel_number * 300)<br>
                                AFR = $AFR (According to hours of exposure)
                                ";
}

elseif ($calculate_ASR)
{
                                print "<form method=post action=\"CALC.php\">
                                <b><u>Select Company:</u></b><br>
                                <select name=\"form_company_id\">";
                                $query = "select company_id,company_name from
AC_company";
                                $result = mssql_query($query,$db);
                                while ($fields = mssql_fetch_row($result))
                                {
                                print "<option
value=\"${fields[0]}\">${fields[1]}\n";
                                }
                                mssql_free_result($result);
                                print "
                                </select>
                                <input type=submit
name=\"calculate_ASR_submit\" value=\"  Next  \">>
                                </form>
                                ";
}

elseif ($calculate_ASR_submit)
{
                                print "<form method=post action=\"CALC.php\">
                                <b><u>Select Mine:</u></b><br>
                                <select name=\"form_mine_id\">";
                                $query = "select mine_id,mine_name from
AC_mines where company_id=${form_company_id}";
                                $result = mssql_query($query,$db);
                                while ($fields = mssql_fetch_row($result))
                                {
                                print "<option
value=\"${fields[0]}\">${fields[1]}\n";
                                }
                                mssql_free_result($result);
                                print "
                                </select><br>
                                <input type=submit
name=\"calculate_ASR_submit_last\" value=\"  Calculate  \">>
                                </form>
                                ";
}

```

```

elseif ($calculate_ASR_submit_last)
{
    $query = "select sum(period_unattended_days)
from AC_accident where (mine_id='$form_mine_id')";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $period_unattended_days = $fields[0];
    }
    mssql_free_result($result);

    $query = "select shift_duration from AC_mines
where (mine_id='$form_mine_id')";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $shift_duration = $fields[0];
    }
    mssql_free_result($result);

    $query = "select count(*) from AC_personnel
where (mine_id='$form_mine_id')";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $personnel_number = $fields[0];
    }
    mssql_free_result($result);
    $ASR = ($period_unattended_days * 1000) /
($shift_duration * $personnel_number * 300);
    print "
    ASR = ($period_unattended_days * 1000) /
($shift_duration * $personnel_number * 300)<br>
    ASR = $ASR (According to hours of exposure)
";
}

else
{
    print "
    <form method='post' action='\"CALC.php\"'>
    <input type='submit'
name='\"calculate_DIFR\"' value='\"Calculate DIFR\"'>
    </form>

    <form method='post' action='\"CALC.php\"'>
    <input type='submit'
name='\"calculate_FIFR\"' value='\"Calculate FIFR\"'>

```

```
        </form>

        <form method="post" action="CALC.php">
        <input type="submit"
name="calculate_AFR" value="Calculate AFR">
        </form>

        <form method="post" action="CALC.php">
        <input type="submit"
name="calculate_ASR" value="Calculate ASR">
        </form>
        ";
    }
?>
```

Accident Record PHP Code:

```
<?php
include ("DB.php");

session_start();

if ($submit_agency)
{
    session_register("agency_id");$agency_id=$form_a
agency_id;

    session_register("adefinition");
    $query = "select definition from AC_Agency where
(agent_id=$agency_id)";

    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $adefinition = $fields[0];
    }
    mssql_free_result($result);

    print "Agency: $adefinition<br>";
?>

<form method="post" action="LIST.php">
Subagency: <select name="form_subagency_id">
<option value="ALL">-- All Subagencies --
<?
    $query = "select sub_agent_id,definition from
AC_subagency where (agent_id=$agency_id)";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "<option
value=\"\$fields[0]\">\$fields[1]\n";
    }
    mssql_free_result($result);
?>

</select>
<p>
<input type="submit" name="submit_subagency"
value="  Next  ">

</form>
<?
}

elseif($submit_subagency)
{
    if ($form_subagency_id == 'ALL')
```

```

        {
            $sdefinition = 'All Subagencies';
        }
    else
    {
        $query = "select definition from AC_subagency
where (sub_agent_id=$form_subagency_id)";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            $sdefinition = $fields[0];
        }
        mssql_free_result($result);
    }
    print "Agency: $sdefinition<br>";
    print "Subagency: $sdefinition<br>";

    if ($form_subagency_id == 'ALL')
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_name,
            compensation_payments,
            noncompensation_payments
        from AC_accident where
        (agency_id=$agency_id) order by acc_id";
    }
    else
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_name,
            compensation_payments,
            noncompensation_payments
        from AC_accident where
        (agency_id=$agency_id and subagency_id=$form_subagency_id) order by acc_id";
    }
    $result = mssql_query($query,$db);
    $numrows=mssql_num_rows($result);
    print "Found $numrows records";
    print "<table border=1><tr>
        <td>Period Unattended Days</td>
        <td>Replacement</td>
        <td>Mine Name</td>
        <td>Compensation Payments</td>
        <td>Non-Compensation Payments</td></tr>";
    while ($fields = mssql_fetch_row($result))

```

```

        {
            print "<tr><td>$fields[0]</td>
                <td>$fields[1]</td>
                <td>$fields[2]</td>
                <td>$fields[3]</td>
                <td>$fields[4]</td></tr>";
        }
        print "</table>";
        mssql_free_result($result);
        session_unregister("agency_id");
        session_unregister("adefinition");
        print "<p><a href=\"LIST.php\">Back to Search
Page</a>";
    }

elseif ($submit_mechanism)
{
    session_register("mechanism_id");$mechanism_id=
$form_mechanism_id;
    session_register("mdefinition");
    $query = "select definition from AC_mechanism
where (mech_id=$mechanism_id)";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $mdefinition = $fields[0];
    }
    mssql_free_result($result);

    print "Mechanism: $mdefinition<br>";
?>
    <form method="post" action="LIST.php">
    Submechanism: <select
name="form_submechanism_id">
    <option value="ALL">-- All Submechanisms --
<?
        $query = "select sub_mech_id,description from
AC_submechanism where (mech_id=$mechanism_id)";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            print "<option
value=\"\$fields[0]\">$fields[1]\n";
        }
        mssql_free_result($result);
?>

    </select>
<p>

```

```

        <input type="submit"
name="submit_submechanism" value="  Next  ">
        </form>
<?
}

elseif($submit_submechanism)
{
    if ($form_submechanism_id == 'ALL')
    {
        $smdefinition = 'All Submechanisms';
    }
    else
    {
        $query = "select description from
AC_submechanism where (sub_mech_id=$form_submechanism_id)";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            $smdefinition = $fields[0];
        }
        mssql_free_result($result);
    }
    print "Mechanism: $mdefinition<br>";
    print "Submechanism: $smdefinition<br>";

    if ($form_submechanism_id == 'ALL')
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_name,
            compensation_payments,
            noncompensation_payments
            from AC_accident where
(mechanism_id=$mechanism_id) order by acc_id";
    }
    else
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_name,
            compensation_payments,
            noncompensation_payments
            from AC_accident where
(mechanism_id=$mechanism_id and submechanism_id=$form_submechanism_id)
order by acc_id";
    }
}

```

```

$result = mssql_query($query,$db);
$numrows=mssql_num_rows($result);
print "Found $numrows records";
print "<table border=1><tr>
    <td>Period Unattended Days</td>
    <td>Replacement</td>
    <td>Mine Name</td>
    <td>Compensation Payments</td>
    <td>Non-Compensation Payments</td></tr>";
while ($fields = mssql_fetch_row($result))
{
    print "<tr><td>$fields[0]</td>
        <td>$fields[1]</td>
        <td>$fields[2]</td>
        <td>$fields[3]</td>
        <td>$fields[4]</td></tr>";
}
print "</table>";
mssql_free_result($result);
session_unregister("mechanism_id");
session_unregister("mdefinition");
print "<p><a href=\"LIST.php\">Back to Search
Page</a>";
}

elseif ($submit_body_location)
{
    session_register("body_location_id");$body_locatio
n_id=$form_body_location_id;
    session_register("bdefinition");
    $query = "select description from
AC_body_location where (body_location_id=$body_location_id)";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $bdefinition = $fields[0];
    }
    mssql_free_result($result);

    print "Body Location: $bdefinition<br>";
?>
    <form method="post" action="LIST.php">
    Body Sublocation: <select
name="form_body_sublocation_id">
    <option value="ALL">-- All Body Sublocations --
<?
    $query = "select body_sub_location_id,description
from AC_body_sublocation where (body_location_id=$body_location_id)";
    $result = mssql_query($query,$db);

```

```

while ($fields = mssql_fetch_row($result))
{
    print "<option
value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
?>

</select>
<p>
<input type="submit"
name="submit_body_sublocation" value="  Next  ">
</form>
<?
}

elseif($submit_body_sublocation)
{
    if ($form_body_sublocation_id == 'ALL')
    {
        $sbdefinition = 'All Body Sublocations';
    }
    else
    {
        $query = "select description from
AC_body_sublocation where (body_sub_location_id=$form_body_sublocation_id)";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            $sbdefinition = $fields[0];
        }
        mssql_free_result($result);
    }
    print "Body Location: $bdefinition<br>";
    print "Body Sublocation: $sbdefinition<br>";

    if ($form_body_sublocation_id == 'ALL')
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_name,
            compensation_payments,
            noncompensation_payments
            from AC_accident where
(body_location_id=$body_location_id) order by acc_id";
    }
    else
    {
        $query = "select

```

```

        period_unattended_days,
        replacement,
        mine_name,
        compensation_payments,
        noncompensation_payments
    from AC_accident where
(body_location_id=$body_location_id and
body_sublocation_id=$form_body_sublocation_id) order by acc_id";
    }
    $result = mssql_query($query,$db);
    $numrows=mssql_num_rows($result);
    print "Found $numrows records";
    print "<table border=1><tr>
        <td>Period Unattended Days</td>
        <td>Replacement</td>
        <td>Mine Name</td>
        <td>Compensation Payments</td>
        <td>Non-Compensation Payments</td></tr>";
    while ($fields = mssql_fetch_row($result))
    {
        print "<tr><td>$fields[0]</td>
            <td>$fields[1]</td>
            <td>$fields[2]</td>
            <td>$fields[3]</td>
            <td>$fields[4]</td></tr>";
    }
    print "</table>";
    mssql_free_result($result);
    session_unregister("body_location_id");
    session_unregister("bdefinition");
    print "<p><a href=\"LIST.php\">Back to Search
Page</a>";
}

elseif ($submit_results)
{
    session_register("results_id");$results_id=$form_re
sults_id;

    session_register("rdefinition");
    $query = "select definition from AC_results where
(results_id=$results_id)";

    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $rdefinition = $fields[0];
    }
    mssql_free_result($result);

    print "Result: $rdefinition<br>";
}

```

```

?>
                                <form method="post" action="LIST.php">
                                Subresult: <select name="form_subresults_id">
                                <option value="ALL">-- All Subresults --
<?
                                $query = "select sub_results_id,description from
AC_subresults where (results_id=$results_id)";
                                $result = mssql_query($query,$db);
                                while ($fields = mssql_fetch_row($result))
                                {
                                print "<option
value=\"\$fields[0]\">\$fields[1]\n";
                                }
                                mssql_free_result($result);
?>
                                </select>
                                <p>
                                <input type="submit" name="submit_subresults"
value="  Next  ">
                                </form>
<?
}
elseif($submit_subresults)
{
                                if ($form_subresults_id == 'ALL')
                                {
                                $srdefinition = 'All Subresults';
                                }
                                else
                                {
                                $query = "select description from
AC_subresults where (sub_results_id=$form_subresults_id)";
                                $result = mssql_query($query,$db);
                                while ($fields = mssql_fetch_row($result))
                                {
                                $smdefinition = $fields[0];
                                }
                                mssql_free_result($result);
                                }
                                print "Result: $rdefinition<br>";
                                print "Subresult: $srdefinition<br>";

                                if ($form_subresults_id == 'ALL')
                                {
                                $query = "select
                                period_unattended_days,
                                replacement,
                                mine_name,

```

```

        compensation_payments,
        noncompensation_payments
        from AC_accident where
(accident_result_id=$results_id) order by acc_id";
    }
    else
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_name,
            compensation_payments,
            noncompensation_payments
            from AC_accident where
(accident_result_id=$results_id and accident_subresult_id=$form_subresults_id)
order by acc_id";
    }
    $result = mssql_query($query,$db);
    $numrows=mssql_num_rows($result);
    print "Found $numrows records";
    print "<table border=1><tr>
        <td>Period Unattended Days</td>
        <td>Replacement</td>
        <td>Mine Name</td>
        <td>Compensation Payments</td>
        <td>Non-Compensation Payments</td></tr>";
    while ($fields = mssql_fetch_row($result))
    {
        print "<tr><td>$fields[0]</td>
            <td>$fields[1]</td>
            <td>$fields[2]</td>
            <td>$fields[3]</td>
            <td>$fields[4]</td></tr>";
    }
    print "</table>";
    mssql_free_result($result);
    session_unregister("results_id");
    session_unregister("rdefinition");
    print "<p><a href='\"LIST.php\"'>Back to Search
Page</a>";
}

else
{
?>

<form method="post" action="LIST.php">
Agency: <select name="form_agency_id">
<?

```

```

AC_agency";
$query = "select agent_id,definition from
$result = mssql_query($query,$db);
while ($fields = mssql_fetch_row($result))
{
    print "<option
value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
?>
</select>
<input type="submit" name="submit_agency"
value=" Next ">
<p>
Mechanism: <select name="form_mechanism_id">
<?
AC_mechanism";
$query = "select mech_id,definition from
$result = mssql_query($query,$db);
while ($fields = mssql_fetch_row($result))
{
    print "<option
value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
?>
</select>
<input type="submit" name="submit_mechanism"
value=" Next ">
<p>
Body Location: <select
name="form_body_location_id">
<?
AC_body_location";
$query = "select body_location_id,description from
$result = mssql_query($query,$db);
while ($fields = mssql_fetch_row($result))
{
    print "<option
value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
?>
</select>
<input type="submit"
name="submit_body_location" value=" Next ">
<p>
Result: <select name="form_results_id">
<?

```

```

AC_results";

$query = "select results_id,definition from
$result = mssql_query($query,$db);
$numrows = mssql_num_rows($result);
if (!$numrows) die(print "No results found!");
while ($fields = mssql_fetch_row($result))
{
    print "<option
value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
?>

</select>
<input type="submit" name="submit_results"
value="  Next  ">

<p>
</form>

<?php
}
?>

```

Accident List PHP Code:

```
<?php
include ("DB.php");

session_start();

if ($submit_agency)
{
    session_register("agency_id");$agency_id=$form_a
agency_id;

    session_register("adefinition");
    $query = "select definition from AC_Agency where
(agent_id=$agency_id)";

    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $adefinition = $fields[0];
    }
    mssql_free_result($result);

    print "Agency: $adefinition<br>";
?>

<form method="post" action="LIST2.php">
Subagency: <select name="form_subagency_id">
<option value="ALL">-- All Subagencies --
<?

    $query = "select sub_agent_id,definition from
AC_subagency where (agent_id=$agency_id)";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        print "<option
value=\"\$fields[0]\"> \$fields[1]\n";
    }
    mssql_free_result($result);
?>

</select>
<p>
<input type="submit" name="submit_subagency"
value="  Next  ">

</form>
<?
}

elseif($submit_subagency)
{
    if ($form_subagency_id == 'ALL')
```

```

        {
            $sdefinition = 'All Subagencies';
        }
    else
    {
        $query = "select definition from AC_subagency
where (sub_agent_id=$form_subagency_id)";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            $sdefinition = $fields[0];
        }
        mssql_free_result($result);
    }
    print "Agency: $sdefinition<br>";
    print "Subagency: $sdefinition<br>";

    if ($form_subagency_id == 'ALL')
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_id,
            compensation_payments,
            noncompensation_payments
            from AC_accident where
(agency_id=$agency_id) order by acc_id";
    }
    else
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_id,
            compensation_payments,
            noncompensation_payments
            from AC_accident where
(agency_id=$agency_id and subagency_id=$form_subagency_id) order by acc_id";
    }
    $result = mssql_query($query,$db);
    $numrows=mssql_num_rows($result);
    print "Found $numrows records";
    print "<table border=1><tr>
        <td>Period Unattended Days</td>
        <td>Replacement</td>
        <td>Mine Name</td>
        <td>Compensation Payments</td>
        <td>Non-Compensation Payments</td></tr>";
    while ($fields = mssql_fetch_row($result))

```

```

        {
            print "<tr><td>$fields[0]</td>
                <td>$fields[1]</td>
                <td>$fields[2]</td>
                <td>$fields[3]</td>
                <td>$fields[4]</td></tr>";
        }
        print "</table>";
        mssql_free_result($result);
        session_unregister("agency_id");
        session_unregister("adefinition");
        print "<p><a href='\"LIST2.php\"'>Back to Search
Page</a>";
    }

elseif ($submit_mechanism)
{
    session_register("mechanism_id");$mechanism_id=
$form_mechanism_id;
    session_register("mdefinition");
    $query = "select definition from AC_mechanism
where (mech_id=$mechanism_id)";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $mdefinition = $fields[0];
    }
    mssql_free_result($result);

    print "Mechanism: $mdefinition<br>";

?>
    <form method="post" action="LIST2.php">
    Submechanism: <select
name="form_submechanism_id">
    <option value="ALL">-- All Submechanisms --
<?
        $query = "select sub_mech_id,description from
AC_submechanism where (mech_id=$mechanism_id)";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            print "<option
value='\"$fields[0]\"'>$fields[1]\n";
        }
        mssql_free_result($result);
?>

</select>
<p>

```

```

        <input type="submit"
name="submit_submechanism" value="  Next  ">
        </form>
<?
}

elseif($submit_submechanism)
{
    if ($form_submechanism_id == 'ALL')
    {
        $smdefinition = 'All Submechanisms';
    }
    else
    {
        $query = "select description from
AC_submechanism where (sub_mech_id=$form_submechanism_id)";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            $smdefinition = $fields[0];
        }
        mssql_free_result($result);
    }
    print "Mechanism: $mdefinition<br>";
    print "Submechanism: $smdefinition<br>";

    if ($form_submechanism_id == 'ALL')
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_id,
            compensation_payments,
            noncompensation_payments
            from AC_accident where
(mechanism_id=$mechanism_id) order by acc_id";
    }
    else
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_id,
            compensation_payments,
            noncompensation_payments
            from AC_accident where
(mechanism_id=$mechanism_id and submechanism_id=$form_submechanism_id)
order by acc_id";
    }
}

```

```

$result = mssql_query($query,$db);
$numrows=mssql_num_rows($result);
print "Found $numrows records";
print "<table border=1><tr>
    <td>Period Unattended Days</td>
    <td>Replacement</td>
    <td>Mine Name</td>
    <td>Compensation Payments</td>
    <td>Non-Compensation Payments</td></tr>";
while ($fields = mssql_fetch_row($result))
{
    print "<tr><td>$fields[0]</td>
        <td>$fields[1]</td>
        <td>$fields[2]</td>
        <td>$fields[3]</td>
        <td>$fields[4]</td></tr>";
}
print "</table>";
mssql_free_result($result);
session_unregister("mechanism_id");
session_unregister("mdefinition");
print "<p><a href=\"LIST2.php\">Back to Search
Page</a>";
}

elseif ($submit_body_location)
{
    session_register("body_location_id");$body_locatio
n_id=$form_body_location_id;
    session_register("bdefinition");
    $query = "select description from
AC_body_location where (body_location_id=$body_location_id)";
    $result = mssql_query($query,$db);
    while ($fields = mssql_fetch_row($result))
    {
        $bdefinition = $fields[0];
    }
    mssql_free_result($result);

    print "Body Location: $bdefinition<br>";
?>
    <form method="post" action="LIST2.php">
    Body Sublocation: <select
name="form_body_sublocation_id">
    <option value="ALL">-- All Body Sublocations --
<?
        $query = "select body_sub_location_id,description
from AC_body_sublocation where (body_location_id=$body_location_id)";
        $result = mssql_query($query,$db);

```

```

while ($fields = mssql_fetch_row($result))
{
    print "<option
value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
?>

</select>
<p>
<input type="submit"
name="submit_body_sublocation" value="  Next  ">
</form>
<?
}

elseif($submit_body_sublocation)
{
    if ($form_body_sublocation_id == 'ALL')
    {
        $sbdefinition = 'All Body Sublocations';
    }
    else
    {
        $query = "select description from
AC_body_sublocation where (body_sub_location_id=$form_body_sublocation_id)";
        $result = mssql_query($query,$db);
        while ($fields = mssql_fetch_row($result))
        {
            $sbdefinition = $fields[0];
        }
        mssql_free_result($result);
    }
    print "Body Location: $bdefinition<br>";
    print "Body Sublocation: $sbdefinition<br>";

    if ($form_body_sublocation_id == 'ALL')
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_id,
            compensation_payments,
            noncompensation_payments
            from AC_accident where
(body_location_id=$body_location_id) order by acc_id";
    }
    else
    {
        $query = "select

```

```

period_unattended_days,
replacement,
mine_id,
compensation_payments,
noncompensation_payments
from AC_accident where
(body_location_id=$body_location_id and
body_sublocation_id=$form_body_sublocation_id) order by acc_id";
}
$result = mssql_query($query,$db);
$numrows=mssql_num_rows($result);
print "Found $numrows records";
print "<table border=1><tr>
<td>Period Unattended Days</td>
<td>Replacement</td>
<td>Mine Name</td>
<td>Compensation Payments</td>
<td>Non-Compensation Payments</td></tr>";
while ($fields = mssql_fetch_row($result))
{
print "<tr><td>$fields[0]</td>
<td>$fields[1]</td>
<td>$fields[2]</td>
<td>$fields[3]</td>
<td>$fields[4]</td></tr>";
}
print "</table>";
mssql_free_result($result);
session_unregister("body_location_id");
session_unregister("bdefinition");
print "<p><a href=\"LIST2.php\">Back to Search
Page</a>";
}

elseif ($submit_results)
{
session_register("results_id");$results_id=$form_re
sults_id;

session_register("rdefinition");
$query = "select definition from AC_results where
(results_id=$results_id)";

$result = mssql_query($query,$db);
while ($fields = mssql_fetch_row($result))
{
$rdefinition = $fields[0];
}
mssql_free_result($result);

print "Result: $rdefinition<br>";

```

```

?>
                                <form method="post" action="LIST2.php">
                                Subresult: <select name="form_subresults_id">
                                <option value="ALL">-- All Subresults --
<?
                                $query = "select sub_results_id,description from
AC_subresults where (results_id=$results_id)";
                                $result = mssql_query($query,$db);
                                while ($fields = mssql_fetch_row($result))
                                {
                                print "<option
value=\"\$fields[0]\">\$fields[1]\n";
                                }
                                mssql_free_result($result);
?>
                                </select>
                                <p>
                                <input type="submit" name="submit_subresults"
value="  Next  ">
                                </form>
<?
}
elseif($submit_subresults)
{
                                if ($form_subresults_id == 'ALL')
                                {
                                $srdefinition = 'All Subresults';
                                }
                                else
                                {
                                $query = "select description from
AC_subresults where (sub_results_id=$form_subresults_id)";
                                $result = mssql_query($query,$db);
                                while ($fields = mssql_fetch_row($result))
                                {
                                $smdefinition = $fields[0];
                                }
                                mssql_free_result($result);
                                }
                                print "Result: $rdefinition<br>";
                                print "Subresult: $srdefinition<br>";

                                if ($form_subresults_id == 'ALL')
                                {
                                $query = "select
                                period_unattended_days,
                                replacement,
                                mine_id,

```

```

        compensation_payments,
        noncompensation_payments
        from AC_accident where
(accident_result_id=$results_id) order by acc_id";
    }
    else
    {
        $query = "select
            period_unattended_days,
            replacement,
            mine_id,
            compensation_payments,
            noncompensation_payments
            from AC_accident where
(accident_result_id=$results_id and accident_subresult_id=$form_subresults_id)
order by acc_id";
    }
    $result = mssql_query($query,$db);
    $numrows=mssql_num_rows($result);
    print "Found $numrows records";
    print "<table border=1><tr>
        <td>Period Unattended Days</td>
        <td>Replacement</td>
        <td>Mine Name</td>
        <td>Compensation Payments</td>
        <td>Non-Compensation Payments</td></tr>";
    while ($fields = mssql_fetch_row($result))
    {
        print "<tr><td>$fields[0]</td>
            <td>$fields[1]</td>
            <td>$fields[2]</td>
            <td>$fields[3]</td>
            <td>$fields[4]</td></tr>";
    }
    print "</table>";
    mssql_free_result($result);
    session_unregister("results_id");
    session_unregister("rdefinition");
    print "<p><a href='\"LIST2.php\"'>Back to Search
Page</a>";
}

else
{
?>

<?
<form method="post" action="LIST2.php">
Agency: <select name="form_agency_id">

```

```

AC_agency";
$query = "select agent_id,definition from
$result = mssql_query($query,$db);
while ($fields = mssql_fetch_row($result))
{
    print "<option
value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
?>
</select>
<input type="submit" name="submit_agency"
value=" Next ">
<p>
Mechanism: <select name="form_mechanism_id">
<?
AC_mechanism";
$query = "select mech_id,definition from
$result = mssql_query($query,$db);
while ($fields = mssql_fetch_row($result))
{
    print "<option
value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
?>
</select>
<input type="submit" name="submit_mechanism"
value=" Next ">
<p>
Body Location: <select
name="form_body_location_id">
<?
AC_body_location";
$query = "select body_location_id,description from
$result = mssql_query($query,$db);
while ($fields = mssql_fetch_row($result))
{
    print "<option
value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
?>
</select>
<input type="submit"
name="submit_body_location" value=" Next ">
<p>
Result: <select name="form_results_id">
<?

```

```

AC_results";

$query = "select results_id,definition from
$result = mssql_query($query,$db);
$numrows = mssql_num_rows($result);
if (!$numrows) die(print "No results found!");
while ($fields = mssql_fetch_row($result))
{
    print "<option
value=\"\$fields[0]\">\$fields[1]\n";
}
mssql_free_result($result);
?>

</select>
<input type="submit" name="submit_results"
value="  Next  ">

<p>
</form>

<?php
}
?>

```

APPENDIX G

APACHE SERVER CONFIGURATION FILE

Section 1: Global Environment

ServerType standalone

ServerRoot "C:/Program Files/Apache Group/Apache"

PidFile logs/httpd.pid

ScoreBoardFile logs/apache_runtime_status

#ResourceConfig conf/srm.conf

#AccessConfig conf/access.conf

Timeout: The number of seconds before receives and sends time out.

Timeout 300

KeepAlive: Whether or not to allow persistent connections (more than
one request per connection). Set to "Off" to deactivate.

KeepAlive On

MaxKeepAliveRequests: The maximum number of requests to allow
during a persistent connection. Set to 0 to allow an unlimited amount.
We recommend you leave this number high, for maximum performance.

MaxKeepAliveRequests 100

KeepAliveTimeout: Number of seconds to wait for the next request from the
same client on the same connection.

KeepAliveTimeout 15

Apache on Win32 always creates one child process to handle requests. If it
dies, another child process is created automatically. Within the child
process multiple threads handle incoming requests. The next two

directives control the behaviour of the threads and processes.

MaxRequestsPerChild 0

Number of concurrent threads (i.e., requests) the server will allow.
Set this value according to the responsiveness of the server (more
requests active at once means they're all handled more slowly) and
the amount of system resources you'll allow the server to consume.

ThreadsPerChild 50

Section 2: 'Main' server configuration

The directives in this section set up the values used by the 'main'
server, which responds to any requests that aren't handled by a
<VirtualHost> definition.

Port 80

ServerAdmin: Your address, where problems with the server should be
e-mailed. This address appears on some server-generated pages, such
as error documents.

ServerAdmin alperg@metu.edu.tr

127.0.0.1 is the TCP/IP local loop-back address, often named localhost. Your
machine always knows itself by this address. If you use Apache strictly for
local testing and development, you may use 127.0.0.1 as the server name.
#

ServerName www.diamond.mine.metu.edu.tr

DocumentRoot: The directory out of which you will serve your
documents. By default, all requests are taken from this directory, but
symbolic links and aliases may be used to point to other locations.
#

DocumentRoot "C:/Program Files/Apache Group/Apache/htdocs"

Each directory to which Apache has access, can be configured with respect
to which services and features are allowed and/or disabled in that
directory (and its subdirectories).
#

First, we configure the "default" to be a very restrictive set of
permissions.
#

<Directory />
Options FollowSymLinks
AllowOverride None

```

</Directory>
# This should be changed to whatever you set DocumentRoot to.
#
<Directory "C:/Program Files/Apache Group/Apache/htdocs">

#
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.

Options Indexes FollowSymLinks MultiViews

# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options", "FileInfo",
# "AuthConfig", and "Limit"

    AllowOverride None

# Controls who can get stuff from this server.

    Order allow,deny
    Allow from all
</Directory>

# UserDir: The name of the directory which is appended onto a user's home
# directory if a ~user request is received.
#
# Under Win32, we do not currently try to determine the home directory of
# a Windows login, so a format such as that below needs to be used. See
# the UserDir documentation for details.
#

<IfModule mod_userdir.c>
    UserDir "C:/Program Files/Apache Group/Apache/users/"
</IfModule>

<IfModule mod_dir.c>
    DirectoryIndex index.html, index.php
</IfModule>

AccessFileName .htaccess

<Files ~ "\.ht">
    Order allow,deny
    Deny from all
</Files>

```

```

UseCanonicalName On

<IfModule mod_mime.c>
    TypesConfig conf/mime.types
</IfModule>

DefaultType text/plain

<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>

HostnameLookups Off

ErrorLog logs/error.log

LogLevel warn

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

CustomLog logs/access.log common

ServerSignature On

<IfModule mod_alias.c>

    Alias /icons/ "C:/Program Files/Apache Group/Apache/icons/"

    <Directory "C:/Program Files/Apache Group/Apache/icons">
        Options Indexes MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>

    ScriptAlias /cgi-bin/ "C:/Program Files/Apache Group/Apache/cgi-bin/"

ScriptAliased
<Directory "C:/Program Files/Apache Group/Apache/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

```

```

</IfModule>
<IfModule mod_autoindex.c>

    IndexOptions FancyIndexing

    AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
    AddIconByType (TXT,/icons/text.gif) text/*
    AddIconByType (IMG,/icons/image2.gif) image/*
    AddIconByType (SND,/icons/sound2.gif) audio/*
    AddIconByType (VID,/icons/movie.gif) video/*

    AddIcon /icons/binary.gif .bin .exe
    AddIcon /icons/binhex.gif .hqx
    AddIcon /icons/tar.gif .tar
    AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
    AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
    AddIcon /icons/a.gif .ps .ai .eps
    AddIcon /icons/layout.gif .html .shtml .htm .pdf
    AddIcon /icons/text.gif .txt
    AddIcon /icons/c.gif .c
    AddIcon /icons/p.gif .pl .py
    AddIcon /icons/f.gif .for
    AddIcon /icons/dvi.gif .dvi
    AddIcon /icons/uuencoded.gif .uu
    AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
    AddIcon /icons/tex.gif .tex
    AddIcon /icons/bomb.gif core

    AddIcon /icons/back.gif ..
    AddIcon /icons/hand.right.gif README
    AddIcon /icons/folder.gif ^^DIRECTORY^^
    AddIcon /icons/blank.gif ^^BLANKICON^^

    DefaultIcon /icons/unknown.gif

    ReadmeName README
    HeaderName HEADER

    #
    # IndexIgnore is a set of filenames which directory indexing should ignore
    # and not include in the listing. Shell-style wildcarding is permitted.
    #
    IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

</IfModule>

<IfModule mod_mime.c>

    AddEncoding x-compress Z

```

```
AddEncoding x-gzip gz tgz
```

```
AddLanguage da .dk
AddLanguage nl .nl
AddLanguage en .en
AddLanguage et .ee
AddLanguage fr .fr
AddLanguage de .de
AddLanguage el .el
AddLanguage he .he
AddCharset ISO-8859-8 .iso8859-8
AddLanguage it .it
AddLanguage ja .ja
AddCharset ISO-2022-JP .jis
AddLanguage kr .kr
AddCharset ISO-2022-KR .iso-kr
AddLanguage no .no
AddLanguage pl .po
AddCharset ISO-8859-2 .iso-pl
AddLanguage pt .pt
AddLanguage pt-br .pt-br
AddLanguage ltz .lu
AddLanguage ca .ca
AddLanguage es .es
AddLanguage sv .se
AddLanguage cz .cz
AddLanguage ru .ru
AddLanguage tw .tw
AddCharset Big5 .Big5 .big5
AddCharset WINDOWS-1251 .cp-1251
AddCharset CP866 .cp866
AddCharset ISO-8859-5 .iso-ru
AddCharset KOI8-R .koi8-r
AddCharset UCS-2 .ucs2
AddCharset UCS-4 .ucs4
AddCharset UTF-8 .utf8
```

```
<IfModule mod_negotiation.c>
```

```
    LanguagePriority en da nl et fr de el it ja kr no pl pt pt-br ru ltz ca es sv tw
</IfModule>
```

```
ScriptAlias /php/ "c:/php/"
```

```
AddType application/x-httpd-php .php
```

```
Action application/x-httpd-php "/php/php.exe"
```

```
LoadModule php4_module c:/php/sapi/php4apache.dll
```

```
AddModule mod_php4.c
```

```
AddType application/x-httpd-php .php
```

AddType application/x-tar .tgz

<IfModule mod_setenvif.c>

BrowserMatch "Mozilla/2" nokeepalive

BrowserMatch "MSIE 4.0b2;" nokeepalive downgrade-1.0 force-response-1.0

BrowserMatch "RealPlayer 4\0" force-response-1.0

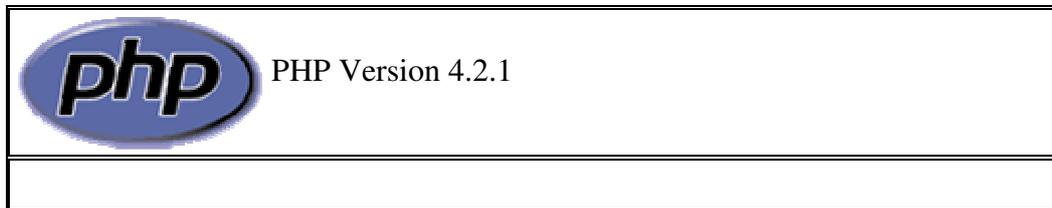
BrowserMatch "Java/1\0" force-response-1.0

BrowserMatch "JDK/1\0" force-response-1.0

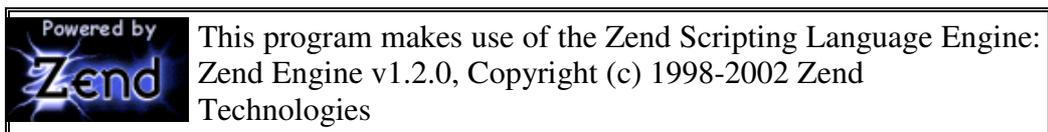
</IfModule>

APPENDIX H

PHP SERVER CONFIGURATION FILE



System	Windows NT 5.0 build 2195
Build Date	May 12 2002 23:51:56
Server API	Apache
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINNT\php.ini
Debug Build	no
Thread Safety	enabled



[PHP 4 Credits](#)

Configuration PHP Core

Directive	Local Value	Master Value
allow_call_time_pass_reference	On	On
allow_url_fopen	1	1
always_populate_raw_post_data	0	0
arg_separator.input	&	&

arg_separator.output	&	&
asp_tags	Off	Off
auto_append_file	<i>no value</i>	<i>no value</i>
auto_prepend_file	<i>no value</i>	<i>no value</i>
browscap	<i>no value</i>	<i>no value</i>
default_charset	<i>no value</i>	<i>no value</i>
default_mimetype	text/html	text/html
define_syslog_variables	Off	Off
disable_functions	<i>no value</i>	<i>no value</i>
display_errors	On	On
display_startup_errors	Off	Off
doc_root	<i>no value</i>	<i>no value</i>
enable_dl	On	On
error_append_string	<i>no value</i>	<i>no value</i>
error_log	<i>no value</i>	<i>no value</i>
error_prepend_string	<i>no value</i>	<i>no value</i>
error_reporting	2039	2039
expose_php	On	On
extension_dir	./	./
file_uploads	1	1
gpc_order	GPC	GPC
highlight.bg		
highlight.comment	#FF9900	#FF9900
highlight.default	#0000CC	#0000CC
highlight.html	#000000	#000000
highlight.keyword	#006600	#006600
highlight.string	#CC0000	#CC0000
html_errors	On	On
ignore_user_abort	Off	Off
implicit_flush	Off	Off
include_path	.;c:\php4\pear	.;c:\php4\pear
log_errors	Off	Off
magic_quotes_gpc	On	On
magic_quotes_runtime	Off	Off

magic_quotes_sybase	Off	Off
max_execution_time	30	30
open_basedir	<i>no value</i>	<i>no value</i>
output_buffering	<i>no value</i>	<i>no value</i>
output_handler	<i>no value</i>	<i>no value</i>
post_max_size	8M	8M
precision	12	12
register_argc_argv	On	On
register_globals	On	On
safe_mode	Off	Off
safe_mode_exec_dir	<i>no value</i>	<i>no value</i>
safe_mode_gid	Off	Off
safe_mode_include_dir	<i>no value</i>	<i>no value</i>
sendmail_from	me@localhost.com	me@localhost.com
sendmail_path	<i>no value</i>	<i>no value</i>
short_open_tag	On	On
SMTP	localhost	localhost
sql.safe_mode	Off	Off
track_errors	Off	Off
unserialize_callback_func	<i>no value</i>	<i>no value</i>
upload_max_filesize	2M	2M
upload_tmp_dir	<i>no value</i>	<i>no value</i>
user_dir	<i>no value</i>	<i>no value</i>
variables_order	EGPCS	EGPCS
xmlrpc_error_number	0	0
xmlrpc_errors	Off	Off
y2k_compliance	Off	Off

standard

Regex Library	Bundled library enabled
Dynamic Library Support	enabled
Internal Sendmail Support for Windows	enabled

Directive	Local Value	Master Value
assert.active	1	1

assert.bail	0	0
assert.callback	<i>no value</i>	<i>no value</i>
assert.quiet_eval	0	0
assert.warning	1	1
safe_mode_allowed_env_vars	PHP_	PHP_
safe_mode_protected_env_vars	LD_LIBRARY_PATH	LD_LIBRARY_PATH
url_rewriter.tags	a=href,area=href,frame=src,input=src,form=fakeentry	a=href,area=href,frame=src,input=src,form=fakeentry

bcmath

BCMath support	enabled
-----------------------	---------

calendar

Calendar support	enabled
-------------------------	---------

com

Directive	Local Value	Master Value
com.allow_dcom	Off	Off
com.autoregister_casesensitive	On	On
com.autoregister_typelib	Off	Off
com.autoregister_verbose	Off	Off
com.typelib_file	<i>no value</i>	<i>no value</i>

ftp

FTP support	enabled
--------------------	---------

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	3.23.39

Directive	Local Value	Master Value
mysql.allow_persistent	On	On

mysql.default_host	<i>no value</i>	<i>no value</i>
mysql.default_password	<i>no value</i>	<i>no value</i>
mysql.default_port	<i>no value</i>	<i>no value</i>
mysql.default_socket	<i>no value</i>	<i>no value</i>
mysql.default_user	<i>no value</i>	<i>no value</i>
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited

odbc

ODBC Support	enabled
Active Persistent Links	0
Active Links	0
ODBC library	Win32

Directive	Local Value	Master Value
odbc.allow_persistent	On	On
odbc.check_persistent	On	On
odbc.default_db	<i>no value</i>	<i>no value</i>
odbc.default_pw	<i>no value</i>	<i>no value</i>
odbc.default_user	<i>no value</i>	<i>no value</i>
odbc.defaultbinmode	return as is	return as is
odbc.defaultlrl	return up to 4096 bytes	return up to 4096 bytes
odbc.max_links	Unlimited	Unlimited
odbc.max_persistent	Unlimited	Unlimited

pcre

PCRE (Perl Compatible Regular Expressions) Support	enabled
PCRE Library Version	3.4 22-Aug-2000

session

Session Support	enabled
------------------------	---------

Directive	Local Value	Master Value
session.auto_start	Off	Off
session.cache_expire	180	180
session.cache_limiter	nocache	nocache

session.cookie_domain	<i>no value</i>	<i>no value</i>
session.cookie_lifetime	0	0
session.cookie_path	/	/
session.cookie_secure	Off	Off
session.entropy_file	<i>no value</i>	<i>no value</i>
session.entropy_length	0	0
session.gc_maxlifetime	1440	1440
session.gc_probability	1	1
session.name	PHPSESSID	PHPSESSID
session.referer_check	<i>no value</i>	<i>no value</i>
session.save_handler	files	files
session.save_path	/tmp	/tmp
session.serialize_handler	php	php
session.use_cookies	On	On
session.use_trans_sid	1	1

xml

XML Support	active
XML Namespace Support	active
EXPAT Version	1.95.2

wddx

WDDX Support	enabled
---------------------	---------

mssql

MSSQL Support	enabled
Active Persistent Links	0
Active Links	0
Library version	7.0

Directive	Local Value	Master Value
mssql.allow_persistent	On	On
mssql.batchsize	0	0
mssql.compatability_mode	Off	Off
mssql.connect_timeout	5	5
mssql.datetimeconvert	On	On

mssql.max_links	Unlimited	Unlimited
mssql.max_persistent	Unlimited	Unlimited
mssql.min_error_severity	10	10
mssql.min_message_severity	10	10
mssql.textlimit	Server default	Server default
mssql.textsize	Server default	Server default
mssql.timeout	60	60

apache

Apache for Windows 95/NT

Apache Version	Apache/1.3.24
Apache Release	10324100
Apache API Version	19990320
Hostname:Port	www.diamond.mine.metu.edu.tr:80
Timeouts	Connection: 300 - Keep-Alive: 15

Directive	Local Value	Master Value
child_terminate	0	0
engine	1	1
last_modified	0	0
xbithack	0	0

Apache Environment

Variable	Value
COMSPEC	C:\WINNT\system32\cmd.exe
DOCUMENT_ROOT	c:/program files/apache group/apache/htdocs
HTTP_ACCEPT	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */*
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_ACCEPT_LANGUAGE	tr
HTTP_CONNECTION	Keep-Alive

HTTP_COOKIE	phpbb2mysql_data=s%3A0%3A%22%22%3B
HTTP_HOST	144.122.239.73
HTTP_REFERER	http://144.122.239.73/accident/
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
PATH	C:\WINNT\system32;C:\WINNT;C:\WINNT\system32\WBEM
REMOTE_ADDR	144.122.239.73
REMOTE_PORT	1068
SCRIPT_FILENAME	c:/program files/apache group/apache/htdocs/accident/info.php
SERVER_ADDR	144.122.239.73
SERVER_ADMIN	alperg@metu.edu.tr
SERVER_NAME	www.diamond.mine.metu.edu.tr
SERVER_PORT	80
SERVER_SIGNATURE	<ADDRESS>Apache/1.3.17 Server at www.diamond.mine.metu.edu.tr Port 80</ADDRESS>
SERVER_SOFTWARE	Apache/1.3.17 (Win32) PHP/4.2.1
SystemRoot	C:\WINNT
WINDIR	C:\WINNT
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET
QUERY_STRING	
REQUEST_URI	/accident/info.php
SCRIPT_NAME	/accident/info.php

HTTP Headers Information

HTTP Request Headers	
HTTP Request	GET /accident/info.php HTTP/1.1
Accept	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */*
Accept-	gzip, deflate

Encoding	
Accept-Language	tr
Connection	Keep-Alive
Cookie	phpbb2mysql_data=s%3A0%3A%22%22%3B
Host	144.122.239.73
Referer	http://144.122.239.73/accident/
User-Agent	Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
HTTP Response Headers	
X-Powered-By	PHP/4.2.1
Keep-Alive	timeout=15, max=93
Connection	Keep-Alive
Transfer-Encoding	chunked
Content-Type	text/html

Additional Modules

Environment

Variable	Value
ALLUSERSPROFILE	C:\Documents and Settings\All Users
CLASSPATH	F:\Oracle\Ora81\orb\classes\yoj.jar;F:\Oracle\Ora81\orb\classes\share.zip
CommonProgramFiles	C:\Program Files\Common Files
COMPUTERNAME	ALPERIRA
ComSpec	C:\WINNT\system32\cmd.exe
NUMBER_OF_PROCESSES	1
OS	Windows_NT
Os2LibPath	C:\WINNT\system32\os2\dll;
Path	C:\WINNT\system32;C:\WINNT;C:\WINNT\system32\WBEM
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH

PROCESSOR_ARCHITECTURE	x86
PROCESSOR_IDENTIFIER	x86 Family 6 Model 6 Stepping 5, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	0605
ProgramFiles	C:\Program Files
SystemDrive	C:
SystemRoot	C:\WINNT
TEMP	C:\WINNT\TEMP
TMP	C:\WINNT\TEMP
USERPROFILE	C:\Documents and Settings\Default User
windir	C:\WINNT

PHP Variables

Variable	Value
PHP_SELF	/accident/info.php
_COOKIE["phpbb2mysql_data"]	s:0:"";
_SERVER["COMSPEC"]	C:\\WINNT\\system32\\cmd.exe
_SERVER["DOCUMENT_ROOT"]	c:/program files/apache group/apache/htdocs
_SERVER["HTTP_ACCEPT"]	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */*
_SERVER["HTTP_ACCEPT_ENCODING"]	gzip, deflate
_SERVER["HTTP_ACCEPT_LANGUAGE"]	tr
_SERVER["HTTP_CONNECTION"]	Keep-Alive
_SERVER["HTTP_COOKIE"]	phpbb2mysql_data=s%3A0%3A%22%22%3B
_SERVER["HTTP_HOST"]	144.122.239.73
_SERVER["HTTP_REFERER"]	http://144.122.239.73/accident/
_SERVER["HTTP_USER_AGENT"]	Mozilla/4.0 (compatible; MSIE 5.01;

NT"]	Windows NT 5.0)
_SERVER["PATH"]	C:\\WINNT\\system32;C:\\WINNT;C:\\WINNT\\system32\\WBEM
_SERVER["REMOTE_ADDR"]	144.122.239.73
_SERVER["REMOTE_PORT"]	1068
_SERVER["SCRIPT_FILENAME"]	c:/program files/apache group/apache/htdocs/accident/info.php
_SERVER["SERVER_ADDR"]	144.122.239.73
_SERVER["SERVER_ADMIN"]	alperg@metu.edu.tr
_SERVER["SERVER_NAME"]	www.diamond.mine.metu.edu.tr
_SERVER["SERVER_PORT"]	80
_SERVER["SERVER_SIGNATURE"]	<ADDRESS>Apache/1.3.17 Server at www.diamond.mine.metu.edu.tr Port 80</ADDRESS>
_SERVER["SERVER_SOFTWARE"]	Apache/1.3.17 (Win32) PHP/4.2.1
_SERVER["SystemRoot"]	C:\\WINNT
_SERVER["WINDIR"]	C:\\WINNT
_SERVER["GATEWAY_INTERFACE"]	CGI/1.1
_SERVER["SERVER_PROTOCOL"]	HTTP/1.1
_SERVER["REQUEST_METHOD"]	GET
_SERVER["QUERY_STRING"]	
_SERVER["REQUEST_URI"]	/accident/info.php
_SERVER["SCRIPT_NAME"]	/accident/info.php
_SERVER["PATH_TRANSLATED"]	c:/program files/apache group/apache/htdocs/accident/info.php
_SERVER["PHP_SELF"]	/accident/info.php
_SERVER["argv"]	Array ()
_SERVER["argc"]	0
_ENV["ALLUSERSPROFILE"]	C:\\Documents and Settings\\All Users
_ENV["CLASSPATH"]	F:\\Oracle\\Ora81\\orb\\classes\\yoj.jar;F:

	\\Oracle\Ora81\orb\classes\share.zip
_ENV["CommonProgramFiles"]	C:\Program Files\Common Files
_ENV["COMPUTERNAME"]	ALPERIRA
_ENV["ComSpec"]	C:\WINNT\system32\cmd.exe
_ENV["NUMBER_OF_PROCESSORS"]	1
_ENV["OS"]	Windows_NT
_ENV["Os2LibPath"]	C:\WINNT\system32\os2\dll;
_ENV["Path"]	C:\WINNT\system32;C:\WINNT;C:\WINNT\system32\WBEM
_ENV["PATHEXT"]	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
_ENV["PROCESSOR_ARCHITECTURE"]	x86
_ENV["PROCESSOR_IDENTIFIER"]	x86 Family 6 Model 6 Stepping 5, GenuineIntel
_ENV["PROCESSOR_LEVEL"]	6
_ENV["PROCESSOR_REVISION"]	0605
_ENV["ProgramFiles"]	C:\Program Files
_ENV["SystemDrive"]	C:
_ENV["SystemRoot"]	C:\WINNT
_ENV["TEMP"]	C:\WINNT\TEMP
_ENV["TMP"]	C:\WINNT\TEMP
_ENV["USERPROFILE"]	C:\Documents and Settings\Default User
_ENV["windir"]	C:\WINNT

PHP License

This program is free software; you can redistribute it and/or modify it under the terms of the PHP License as published by the PHP Group and included in the distribution in the file: LICENSE

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

If you did not receive a copy of the PHP license, or have any questions about PHP licensing, please contact license@php.net.

APPENDIX I

PHP MODULES

standard

Regex Library	Bundled library enabled
Dynamic Library Support	enabled
Internal Sendmail Support for Windows	enabled

Directive	Local Value	Master Value
assert.active	1	1
assert.bail	0	0
assert.callback	<i>no value</i>	<i>no value</i>
assert.quiet_eval	0	0
assert.warning	1	1
safe_mode_allowed_env_vars	PHP_	PHP_
safe_mode_protected_env_vars	LD_LIBRARY_PATH	LD_LIBRARY_PATH
url_rewriter.tags	a=href,area=href,frame=src,input=src,form=fakeentry	a=href,area=href,frame=src,input=src,form=fakeentry

bcmath

BCMath support	enabled
-----------------------	---------

calendar

Calendar support	enabled
-------------------------	---------

com

Directive	Local Value	Master Value
com.allow_dcom	Off	Off

com.autoregister_casesensitive	On	On
com.autoregister_typelib	Off	Off
com.autoregister_verbos	Off	Off
com.typelib_file	<i>no value</i>	<i>no value</i>

ftp

FTP support	enabled
--------------------	---------

mysql

MySQL Support	enabled
Active Persistent Links	0
Active Links	0
Client API version	3.23.39

Directive	Local Value	Master Value
mysql.allow_persistent	On	On
mysql.default_host	<i>no value</i>	<i>no value</i>
mysql.default_password	<i>no value</i>	<i>no value</i>
mysql.default_port	<i>no value</i>	<i>no value</i>
mysql.default_socket	<i>no value</i>	<i>no value</i>
mysql.default_user	<i>no value</i>	<i>no value</i>
mysql.max_links	Unlimited	Unlimited
mysql.max_persistent	Unlimited	Unlimited

odbc

ODBC Support	enabled
Active Persistent Links	0
Active Links	0
ODBC library	Win32

Directive	Local Value	Master Value
odbc.allow_persistent	On	On
odbc.check_persistent	On	On
odbc.default_db	<i>no value</i>	<i>no value</i>
odbc.default_pw	<i>no value</i>	<i>no value</i>
odbc.default_user	<i>no value</i>	<i>no value</i>

odbc.defaultbinmode	return as is	return as is
odbc.defaultlrl	return up to 4096 bytes	return up to 4096 bytes
odbc.max_links	Unlimited	Unlimited
odbc.max_persistent	Unlimited	Unlimited

pcre

PCRE (Perl Compatible Regular Expressions) Support	enabled
PCRE Library Version	3.4 22-Aug-2000

session

Session Support	enabled
------------------------	---------

Directive	Local Value	Master Value
session.auto_start	Off	Off
session.cache_expire	180	180
session.cache_limiter	nocache	nocache
session.cookie_domain	<i>no value</i>	<i>no value</i>
session.cookie_lifetime	0	0
session.cookie_path	/	/
session.cookie_secure	Off	Off
session.entropy_file	<i>no value</i>	<i>no value</i>
session.entropy_length	0	0
session.gc_maxlifetime	1440	1440
session.gc_probability	1	1
session.name	PHPSESSID	PHPSESSID
session.referer_check	<i>no value</i>	<i>no value</i>
session.save_handler	files	files
session.save_path	/tmp	/tmp
session.serialize_handler	php	php
session.use_cookies	On	On
session.use_trans_sid	1	1

xml

XML Support	active
XML Namespace Support	active
EXPAT Version	1.95.2

wddx

WDDX Support	enabled
---------------------	---------

mssql

MSSQL Support	enabled
Active Persistent Links	0
Active Links	0
Library version	7.0

Directive	Local Value	Master Value
mssql.allow_persistent	On	On
mssql.batchsize	0	0
mssql.compatibility_mode	Off	Off
mssql.connect_timeout	5	5
mssql.datetimeconvert	On	On
mssql.max_links	Unlimited	Unlimited
mssql.max_persistent	Unlimited	Unlimited
mssql.min_error_severity	10	10
mssql.min_message_severity	10	10
mssql.textlimit	Server default	Server default
mssql.textsize	Server default	Server default
mssql.timeout	60	60

apache

Apache for Windows 95/NT

Apache Version	Apache/1.3.24
Apache Release	10324100
Apache API Version	19990320
Hostname:Port	www.diamond.mine.metu.edu.tr:80
Timeouts	Connection: 300 - Keep-Alive: 15

Directive	Local Value	Master Value
child_terminate	0	0
engine	1	1

last_modified	0	0
xbithack	0	0

Apache Environment

Variable	Value
COMSPEC	C:\WINNT\system32\cmd.exe
DOCUMENT_ROOT	c:/program files/apache group/apache/htdocs
HTTP_ACCEPT	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */*
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_ACCEPT_LANGUAGE	tr
HTTP_CONNECTION	Keep-Alive
HTTP_COOKIE	phpbb2mysql_data=s%3A0%3A%22%22%3B
HTTP_HOST	144.122.239.73
HTTP_REFERER	http://144.122.239.73/accident/
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
PATH	C:\WINNT\system32;C:\WINNT;C:\WINNT\system32\WBEM
REMOTE_ADDR	144.122.239.73
REMOTE_PORT	1080
SCRIPT_FILENAME	c:/program files/apache group/apache/htdocs/accident/modules.php
SERVER_ADDR	144.122.239.73
SERVER_ADMIN	alperg@metu.edu.tr
SERVER_NAME	www.diamond.mine.metu.edu.tr
SERVER_PORT	80
SERVER_SIGNATURE	<ADDRESS>Apache/1.3.17 Server at www.diamond.mine.metu.edu.tr Port 80</ADDRESS>
SERVER_SOFTWARE	Apache/1.3.17 (Win32) PHP/4.2.1
SystemRoot	C:\WINNT
WINDIR	C:\WINNT

GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET
QUERY_STRING	
REQUEST_URI	/accident/modules.php
SCRIPT_NAME	/accident/modules.php

HTTP Headers Information

HTTP Request Headers	
HTTP Request	GET /accident/modules.php HTTP/1.1
Accept	image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/vnd.ms-powerpoint, application/vnd.ms-excel, application/msword, */*
Accept-Encoding	gzip, deflate
Accept-Language	tr
Connection	Keep-Alive
Cookie	phpbb2mysql_data=s%3A0%3A%22%22%3B
Host	144.122.239.73
Referer	http://144.122.239.73/accident/
User-Agent	Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)
HTTP Response Headers	
X-Powered-By	PHP/4.2.1
Keep-Alive	timeout=15, max=100
Connection	Keep-Alive
Transfer-Encoding	chunked
Content-Type	text/html

VITA

Alper Güngör was born in Zonguldak on August 21, 1972. He received his B. S. degree in Mining Engineering from the Middle East Technical University in June, 1994. He received his M.Sc. degree in Mining Engineering from the Middle East Technical University in February 1998. He has been a graduate assistant in the Department of Mining Engineering in Middle East Technical University during 1998-2003. Since March 2003, he has been working in Şişecam Mersin Soda Co. as a Limestone and Salt production plant engineer. Besides cavern modeling and simulation and other production related duties, he is also responsible of safety issues on working site as a safety engineer. His main areas of interest are occupational health and safety, mine ventilation, artificial intelligence, simulation, database programming, multimedia applications, Internet and networking. His hobbies are martial arts, swimming, scuba, basketball, and cinema.