CERTIFICATION OF APPROVAL

Web-Based Logbook System

by

Azat Shamsiev (1157)

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION SYSTEMS)

Approved by,

(MICHELLE BEH)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
JUN 2004

CERTIFICATION OF ORIGINALITY

This is to certify that author is responsible for the work submitted in this project, that the original work is his own except as specified in the references and the acknowledgments, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Azat Shamsiev

ABSTRACT

One of the major visions of Universiti Teknologi Petronas is to be a leader in technology and education and also to be a center for creativity and innovation. In any organization however there are always a few stumbling blocks along the way and in this case we find that there are still a few processes that are still being done manually. The new proposed system is going to reduce the paperwork. The system provides the submission of the Logbook or weekly reports on web-based approach. The system built on the Web application platform and should be easily launched on the Internet.

The title of this project is the "Web-based Logbook." This dissertation provides an overview of work that had been done in order for author to develop the final year project. There will be an in depth discussion of all the findings and also the problems that author has been facing with the development of this project. The most important sections for this report are the Literature Review and the Results and Discussions, since they will contain the bulk of the findings. Initially it was decided that author will concentrate only on the web application development and then develop the project based on that findings, however it has now been decided that he will research the various database mechanisms and thereafter build the application based on the theoretical knowledge that author has gained from the research but the implementation will be entirely up to the own discretion. In the sections that follow author will provide all of the aspects that he has been able to gather up to so far, and also allow the readers to be able to get a better understanding of the how web application and database works together as a unit.

ACKNOWLEDGEMENT

I would like to firstly thank ALLAH for giving me the opportunity and the strength to be able to fulfill all of the requirements of this project. I also would like to thank my family for giving me a strength and motivation to complete the project. I dedicate this project to my beloved mother. She was always there for me when I needed help. Special thanks to Petronas Sdn.Bhd for providing me with scholarship which is allowed me to study in such prestigious university like Universiti Teknologi Petronas. I also want to pay tribute to my supervisor Michele Beh for helping me in developing my system. There were also many lectures to which I would like to say thanks for supporting and helping me in the development of this project. Special thanks to the Fadzil Hassan, my much respected lecture, who gave me the hint and idea, how to develop the system. I also want to pay my tribute to my fellow students who were studying with me and provide with necessary information for the project.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Background of study

During the industrial training the author was mainly involved with the development of various web based applications and thus his interest began in the various web applications that we see on the Internet. Author decided that in order for him to gain much knowledge on the various web applications, he needs to build an application or research a topic which will combine all of the major aspects of web applications and thus he looked into the area of web applications. In order for author to have a more defined scope he decided that he should research the aspects of the web applications and the database mechanisms that formed the basis of the application. Many people know very well how to use a web application but author felt that for this project he wants to research the main elements that make up a web application and in this case it is the database mechanism. The application that he will be developing for this project is a web site that will display the functionality of the database mechanism that he has researched. It will be a repository for the student's Logbooks that have been carried out during the Industrial training and thus later on students can use this application to search for information here at UTP.

1.2 Problem Statement

In many of the web sites that we have on the World Wide Web today such as Yahoo and Google, just to name a few, we find that it is very simple to use or search for any type of information by simply stating the information that we want and thereafter clicking on the button. Many people however do not know what is really going on in the background and how all of the information that they require is found so quickly and so easily. After discussions with my supervisor the author has decided to base my research for this final year project on the database mechanisms that are used for the various types of web applications so that the question of what exactly happens in the background once the user clicks on the button can be answered. Therefore author's problem statement, which is in actual fact more of an opportunity, for this project will be to basically provide an answer to the question of how exactly a web application works together with a database. He will provide an application as well as documentation to support his findings. In terms of the significance of the project he is certain that the research that author will provide in terms of the database mechanisms that are used in the various web applications, will help the readers of his research as well as himself to gain a much better understanding of how exactly a web application works with regards to the database mechanism that it uses. This project will also help students later on who would like to further their research on web applications. Another significance of this project is that the application that will be developed to demonstrate findings can be used later by the UTP students as a Web application that can support the submission of the Logbook during the industrial training that they have undergone here at UTP.

1.3 Objectives and Scope of study

The objectives have been identified as follows and are listed below:

- To research the different types of Web applications that is available on the Internet today.
- To research a particular type of database mechanism that is used currently to develop Web-based application.
- To develop a Web application that displays the functionality of the database mechanism that has been researched.
- To develop the above mentioned application using the web as a platform.
- To give people a better understanding of how exactly a Web application works together with a database as one unit.
- To develop an application that will help the UTP students to submit the Logbook using Internet.

- To develop a repository of all the Logbooks those have been created by students and use it as a reference later.

These are the objectives that have been defined and the author has been able to meet many of them in my project.

The scope of study will be based on one particular database mechanism that is used in any of the web applications that are currently on the World Wide Web. It refers mainly to the research aspect since for the implementation aspect author was using his own methodology that was based on meeting the objectives for the application. As it was mentioned previously the author will provide an in depth research of a particular database mechanism, and the main aim of this will be to provide a better understanding of how web application and database actually work together as a unit. All of the aspects of the scope remained the same as in preliminary report. However, the one significant change to note is that in terms of the application that will be developed the author will be using his own methodology to develop the application. This is mainly due to the fact that the author simply wants to show the functionality of the database mechanism in this project but author does not want to duplicate the mechanism and thus his application will be developed according to his own discretion.

The author thinks that this project is very relevant especially in the day and age that we are living in where the Internet forms a very significant part of people's lives. Main objective is to give people a better understanding of how exactly a web application works. This project is very relevant especially here at UTP since it can form a platform for future research into this particular field.

In terms of the feasibility of the project, it had already narrowed down the scope considerably and now author has coded the application based mainly on his own knowledge with assistance from other resources, instead of coding the application by duplicating the code. The main objective with regard to this aspect is to ensure that the entire project can be completed within the timeframe that has been allocated for this project.

CHAPTER 2
LITERATURE REVIEW AND/OR THEORY

According to the problem statement that was given in the previous section, the problem which is in actual fact an opportunity in this case is to provide the users with a Web-based application to submit the Logbook

The number of students which are undergoing the industrial training is increasing rapidly. Most of the students are located at various companies due to their different undergraduate programs. This makes difficult of getting the student's logbook to the university. The only simply way of obtaining the student's logbook is by introducing a Web-based application for the logbook, which saves the cost and time significantly.

According to Mr. Richard K. Belew (2000) who states: "An organization can be viewed as a system of activities that co-ordinates by a flow of information. The information system is the underlying system concerned with maintaining this information flow".

Formally communicated information is represented as data. An organization's database is the set of data that is recorded within the organization at any one time. Database usually refers specifically to the data stored in computers.

The conventional file-oriented approach is to build computer systems support specific activities, or applications. This approach is unstable because of changing nature of activities, the need to share data, and the dependence of programs on the structure of the files.

The database approach is to treat database as an entity in its own right, and to design it to model the organization. A special program called a Database Management System (DBMS) is used to manage and provide access to the database. Selective logical views are defined for each application. The database administrator (DBA) is responsible for managing the database.

The advantage of database approach is independence of applications from changes in the representation of the database, or its evolution. Also, central control of data provides better data security and integrity.

Its data model defines the ways in which data can be structured and manipulated by a database. Most databases implement the relational data model, which allows data to be stored within sets of tables. There is also an emerging generation, of object and object-relational systems. These are base upon object-oriented features that allow complex structures and complex behaviors to be implemented, for example, to store multimedia information.

Two important milestones in the evolution of computer systems are the migration of computers out of the computer room to the workplace, and the use of computers networks to allow computers to communicate, within and between organizations. The Internet now takes these developments a large step further by linking together computer networks the world over, using TCP/IP protocols. Intranets and Extranets respectively use Internet technology to create private networks for communications within and between organizations.

"The Web provides a simple uniform interface to the Internet and a facility for storing and retrieving inter-linked hypermedia documents. Hypermedia documents are coded using a mark-up language called HTML, or a derivation of it, and are addressed by URLs. Programs called gateways on the Web server computers provide access to other environments. Gateways are accessed using the Common Gateway Interface (CGI). Users access Web resources from client computers using browser programs. Links between clients and serves are established using a protocol called HTTP. Web applications include publishing e-commerce, support for group activities, and embedded systems." Barry Eaglestone and Mick Ridley. *Web Database Systems.* McGraw – Hill Book Company. (2001).

Web database systems are systems in which both Web and database technologies are used. These provide wider access to database systems, ways of distributing systems and more services through integration of systems.

The overall architecture of web applications is similar to client-server systems in many aspects, but there is a key difference. In traditional client-server systems, the respective roles of the clients and servers and their interactions are predefined and static. In web applications, however, client side programs and contents may be generated dynamically. For example, a server may return a dynamically generated HTML file that contains dynamically generated Javascripts, links and contents. This means that the subsequent interactions between the client and server depend on the previous inputs.

"For traditional programs, correctness and efficiency are usually the most important quality factors. For web applications, other quality features can often be more important and yet there are few techniques for supporting them. For example, compatibility and interoperability are urgent and cause problems that are more serious than with traditional programs. Traditional programs are usually developed for a certain predefined, well understood environment, with very few conflicts and changes. Web applications often are affected by factors that may cause incompatibility and interoperability issues. For example, server components can be distributed to different operating systems, such as UNIX, Linux, Windows, MacOS, and AIX, each of which has multiple versions, and run with different web server packages, including IIS from Microsoft, Apache, WebLogic from IBM and others. The situation is even more complex on the client side, with different versions of web browsers running under a variety of operating systems. Clients may also use different connection approaches, such as dial-up modems, direct internet access or wireless, and may also use different ISP providers. All of this heterogeneity makes it harder to produce web application components that are compatible with one another and that inter-operate easily and correctly." Barry Eaglestone and Mick Ridley. *Web Database Systems*. McGraw – Hill Book Company. (2001).

Another difference between web applications and other types of programs is the variance in the control of execution of the application. For traditional programs, the control flow is fully managed by the program, so the user cannot affect it. When executing web application, users can break the normal control flow without alerting the program controller. For example, users can press the back or refresh button in the web browser, which totally changes the execution context, causing unexpected results. Furthermore, changes in the client-side configuration may affect the behavior of web applications. For example, users can turn off cookies, which can cause subsequent operations to malfunction.

Web applications also have much faster maintenance requirements than most traditional software. Web technologies evolve more rapidly than traditional software technologies, and the changes in web application requirements are also more dramatic. Therefore maintenance not only needs to be done more frequently, but needs to be done more efficiently due to the peculiar time-to-market pressure for web applications. These include session control, cookies, the stateless aspect of HTTP, and new security issues. Therefore, new solutions are necessary for these special features.

Web applications also have features that are not present in client-server and distributed systems. These include session control, cookies, the stateless aspect of HTTP, and new security issues. Therefore, new solutions are necessary for these special features.

Usability is another increasingly considered a key software quality attribute, particularly crucial in web-based applications where competition is literally one click away. To effectively design usable web applications, usability issues should be addressed early in the development process. To this end, high user involvement is crucial, and so is support for rapid iterative improvements based on user feedback. In addition, as web applications are commonly distributed, concurrent and multi-user, usability assessment should not be restricted to single user observation but should also consider how usability is affected by concurrent users collaborating (or interfering) with one another. In recent years a number of techniques for usability assessment have been developed. These techniques are mainly based on observing users interactions with the systems and as such require a working implementation of the system. This means that usability assessment can only be carried out late in the development process. At this stage it is very expensive to go back and make major architectural changes to the design. It also makes an agile iterative improvement of the application's usability difficult to achieve. Behaviour models are precise, abstract descriptions of the intended behaviour of a concurrent and possibly distributed system. We use a behaviour model to capture the interaction between users and the different components that form the target application. This model is simulated to allow end-users of the system to interactively explore the specified behaviour for the purposes of both validation – is this required behaviour, and usability – is this the best way of performing the interaction/task.

CHAPTER 3
METHODOLOGY/PROJECT WORK

3.1 Procedure Identification

As mentioned previously the model that author has chosen to use for this project is the Prototyping model. Author will base the development of the application for this project on this model. The main reason that he has chosen to use this model is because the author can use the previous applications that has been developed to help him to design this application. Since author has been previously involved with some web application development it will be able to base the development of this project on some of those applications. There are also many examples of web applications such as current project that author has found on the Internet and by using this model he will be able to take full advantage of those resources. As mentioned previously in the preliminary report it will customize this model to suite the needs of this project. The methodology still remains the same however it will now give more detail with respect to the phases that author has been using for this project. The phases are listed below:

3.1.1 Conduct research on the various web applications

This phase of the project consisted of researching the various types of web application that are available on the market today and find out exactly how they work in terms of their functionality. Author has already completed this phase of the project. He will put all of the findings with regard to this aspect in the findings section of this report. The main objective for this phase of the project was to gather as much information as possible with respect to how web application and database work together. Author also paid some attention to the aesthetics of the web sites that was visited by him so that he can implement some of the general concepts in own application. The main deliverable for this phase was to gather all of the database mechanisms that are used so that author could base the research on one particular database mechanism. Author managed to meet the objectives for this phase of the project and was able to successfully move on to the next phase of the project, however there were some problems that he did encounter and these issues will be discussed in the results and findings section of this report.

3.1.2 Choose one database mechanism to research

This phase of the project included choosing one of the database mechanisms that was found from the previous phase and then carrying out a research on that particular database mechanism. There was a problem that arose in this phase and author will also discuss this aspect in the results and findings section of this report. The main objective from this phase was to be able to have one particular mechanism that could be used to implement this project; it was decided that author should base the implementation on his own idea of how a web application engine works together with database based on the research that have been gathered. This was mainly due to the fact that the resources that author had been looking into did not give much information as to how exactly the database mechanism works. Thus the main objective for this phase was to simply research the various database mechanisms and provide documentation on those mechanisms and also to try and include the general concepts of the database mechanisms into the application that author will be developing. Author has also completed this phase of the project, and it will be included all the findings in the appropriate sections of this report.

3.1.3 Implementation

This is the last phase of the project and also the longest phase. It will consist of the actual development of the application from start to finish which will include both the system analysis and the system design. In terms of the implementation it was decided to use the past semester final year projects to implement the application. Author will provide samples of the interface and other aspects. The main idea that author has come up with here so far is to provide the users with a simple interface including a login page in which the user can enter the user name and password. The application will also be having a database that will be developed using Microsoft SQL. The front end of the application will be developed using Microsoft FrontPage. The application will be running on a web server running Internet Information Server 5.0.

3.2 Tools (e.g. Equipment, hardware, etc) Required

Hardware

- Pentium 4 CPU, 600 MHz.
- Web server running IIS 5.0.
- Database Server running Microsoft SQL database.

Please note that these are the specifications for the web server, the database server, and the Pentium CPU's.

- 256 MB Ram
- 10.2 GB Hard disk space
- 15" monitor
- Speakers
- Mouse
- Keyboard
- Network card
- Network cable

Software

- Windows 2000 operating system
- Microsoft FrontPage 2000
- Microsoft SQL
- Internet Information Server 5.0

3.3 System Architecture

Figure 3.3.1 on page 13, shows the general system architecture of systems that have an interaction between web server and a client machine. The diagram shows all of the operations that will take place from the time the student uploads the reports to the time that the reports had been downloaded by the lecture or supervisor on his or her machine. All of the web pages that had been developed were placed on a web server and author has also created a database which is holding all of the information on the reports and detailed information on the students and supervisors.

Once the student uploads the report, the uploaded report will be available for the supervisor to download form the server. The student will also have an option to view the reports in case any correction should be made. Basically report will be in word-processing format, in order for them to be downloaded the supervisor must have word-processing software. After supervisor have graded the report or possibly use the option to write comments, the feedback could be sent back to the server for the student revision.

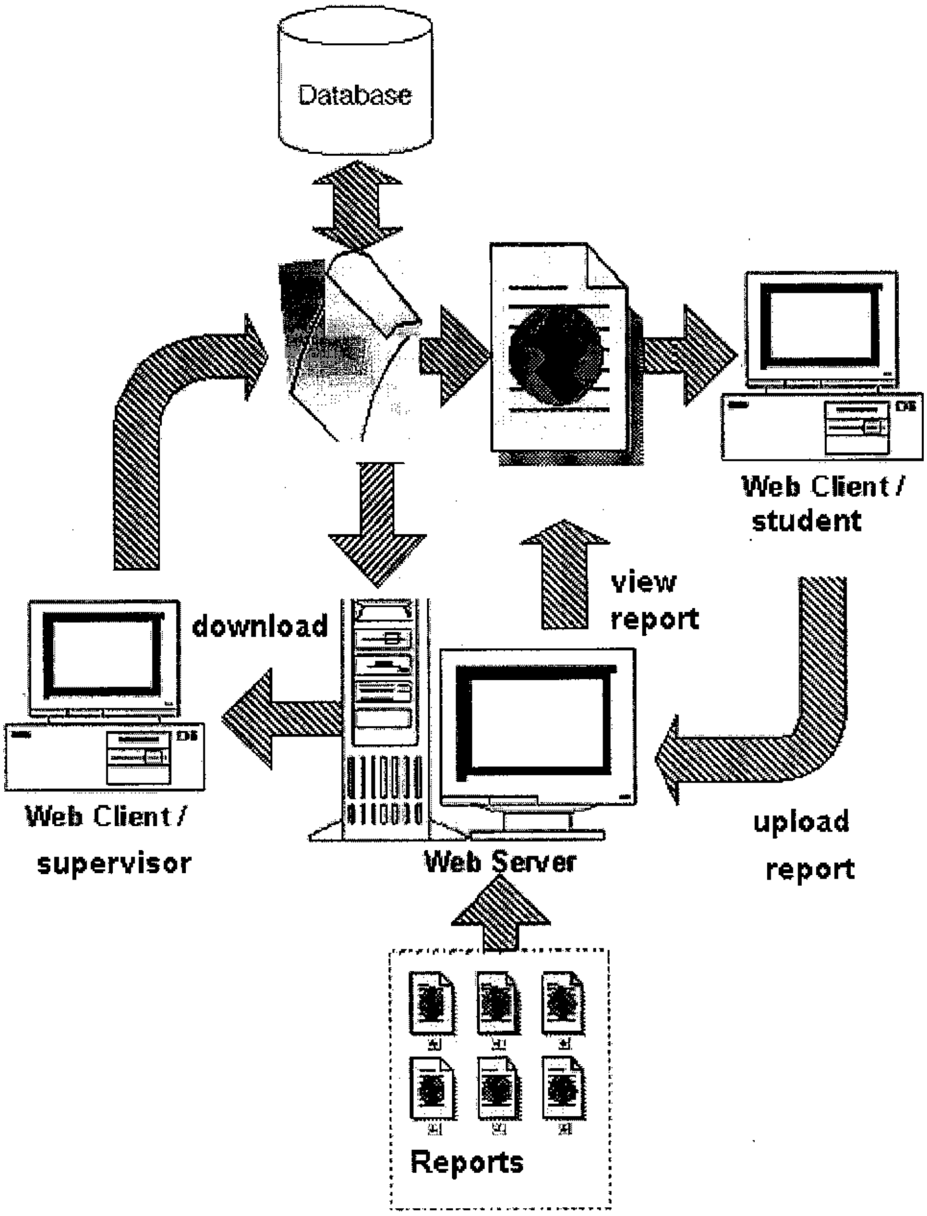


Figure 3.3.1 General System Architecture Diagram

3.4 Process Models

Figure 3.4.1 on page 15, shows the flow of information within my system. The diagram represents all of the actions that occur from the time the student uploads the report till the time the user exits. But it is not necessarily could be done concurrently. Once the user accesses the application the login page will be displayed. The user then simply enters the user name and password in the text box provided. The login is required for both students and supervisors.

The student uploads the report into the server. All the reports will be stored in the server's database together with the user's detailed information. Student can review the report in the system by downloading it in case of any correction should be made. Supervisor must check the report availability, if the report is not uploaded on time, the supervisor can exit the application.

If the report is available for the supervisor to mark, the supervisor can pass the comments and grade the report in the provided option box and text box. The supervisor could send all the feedback to the server for the student review. After all the marking had been done, the supervisor can successfully exit the application.

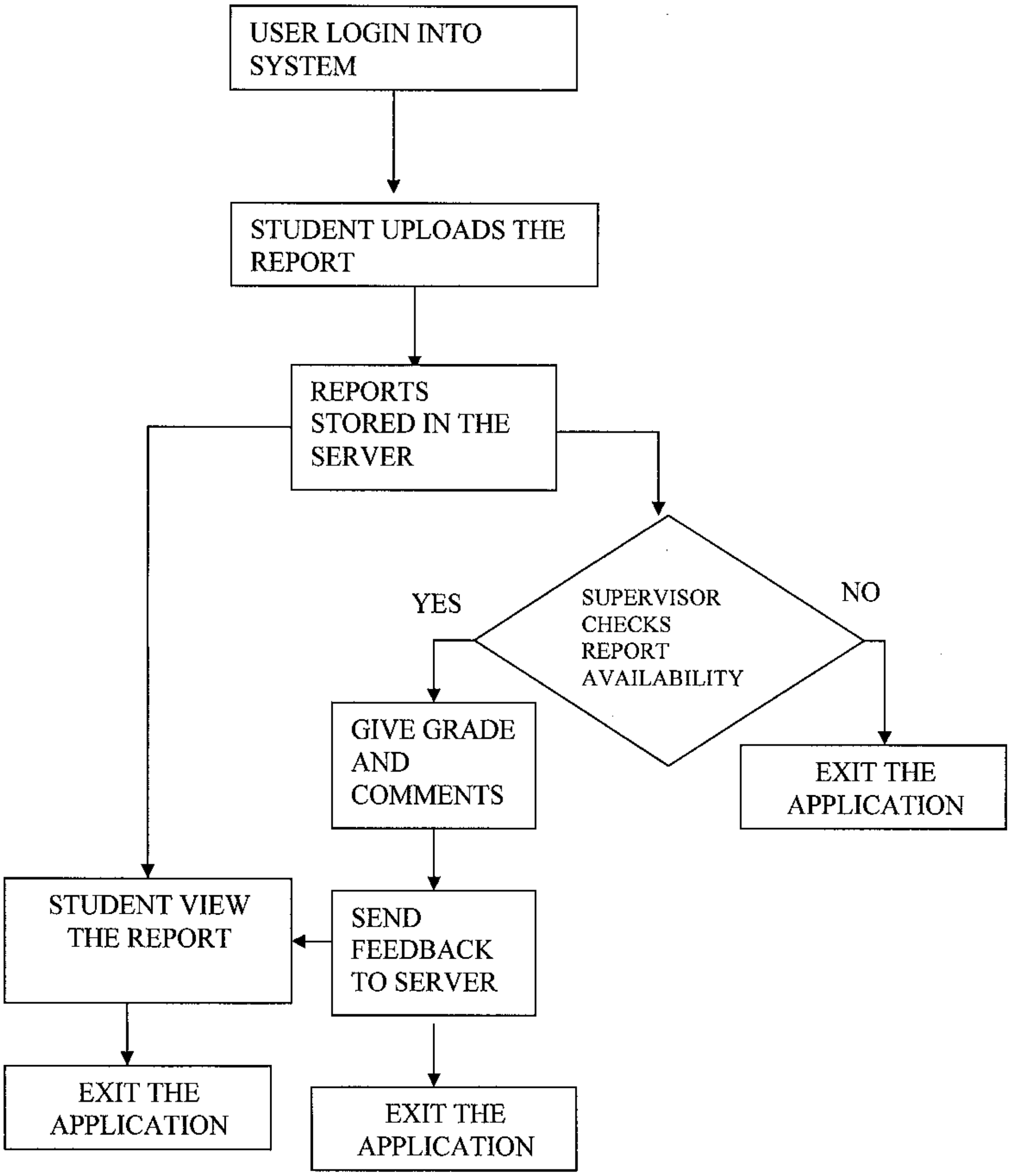


Figure 3.4.1 Process flow diagram

Figure 3.4.2 and 3.4.3 are representatives of the entity relationship diagram. Figure 3.4.2 shows two tables, which are Supervisor and Students. They are related to each other by the relationship "one-to-many". The primary key for the Supervisor table is SupervisorID, whereas the primary key for the Students table is StudentID. Since the relationship is one-to-many, Students table consist of foreign key SupervisorID, which is from Supervisor table.

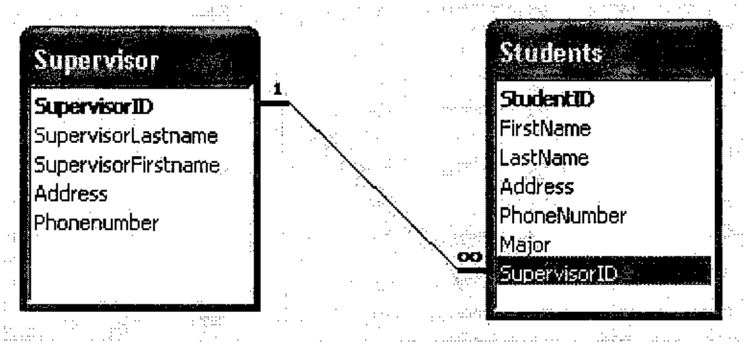


Figure 3.4.2 Entity Relationship Diagram

Figure 3.4.3 is a representative of the entity relationship diagram. The following tables are used for the login purpose in the system. They don't have any relationship between each other. The use of two different tables for the login purpose will set the priorities for each of the user whether it is a student or supervisor.

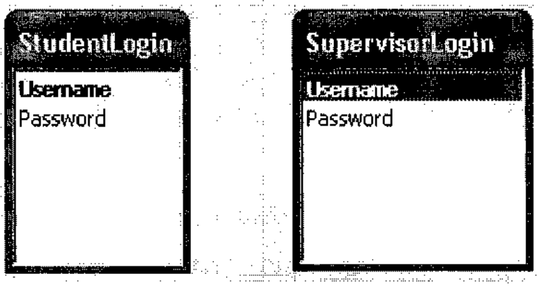


Figure 3.4.3 Entity Relationship Diagram

Figure 3.4.4 shows all of the fields that are contained in the tables and it also show the various data types of the different fields.

Students

Name	Data type
StudentID	Number
FirstName	Varchar2
LastName	Varchar2
Address	Varchar2
PhoneNumber	Number
Major	Varchar2
SupervisorID	Number

Supervisor

Name	Data type
SupervisorID	Number
FirstName	Varchar2
LastName	Varchar2
Address	Varchar2
PhoneNumber	Number

StudentLogin

Name	Data type
UserName	Varchar2
Password	Varchar2

SupervisorLogin

Name	Data type
UserName	Varchar2
Password	Varchar2

Figure 3.4.4 Data Dictionary

CHAPTER 4
RESULTS AND DISCUSSION

The author has come up with the following results and findings with respect to the various steps that he has outlined for this project. The following section will contain a summary of all information that author has collected for this project to date. In terms of the first phase of this project, research carried out on the various web applications that are available today.

The use of both database and Web technology within a single system is advantageous in the following situation:

- Wider access to a database - by connecting the database system to the Web will make possible worldwide access to database.
- System distribution – the Web allows a database and its applications to be distributed. Data can be stored where it is used and/or applications can be located where the activities they support take place. An advantage is the ability to configure computers for specific tasks.
- Better querying, manipulation and administration of Web data - DBMS provide sophisticated facilities for data querying, manipulation and administration. Web applications that must utilize large volumes of data will therefore benefit from using a DBMS to store their data.

4.1 Database

There are many sites such as Alta Vista and Yahoo that provide the web users information by retrieving data from their databases. The web users fill out a form on the web page and the search engines return the results.

There are a variety of approaches for a web page to access data in the database.

- JDBC (Java Database Connectivity) is made up of a set of Java interfaces that specify the API (Application Programming Interface), and several drivers supplied by database vendors that let Java programs connect to a database.

The JDBC API, whose specification was defined by JavaSoft, is a standard data access interface developed by Sun. The JDBC API is a Java application programming interface which is not vendor specific, since JDBC can be obtained from different vendors

- SQLJ (Structured Query Language Java) is another JDBC-based approach that uses Java with embedded SQL. SQLJ is an emerging technology sponsored by Oracle, IBM, Sybase, Tandem and Javasoft for embedding precompiled static SQL statements in Java programs. A SQLJ translator transforms the SQLJ code into standard Java code that accesses the database through a call-level interface.

- CGI (Common Gateway Interface) is a standard interface used to write applications that remote users can fill out an HTML Form on Web browser to communicate with programs on an HTTP server.

The most prominent and mature approach for accessing relational DBMSs from Java appears to be JDBC. With JDBC, Java can be used as the host language for writing database applications. JDBC is a standard Java classes library that queries and modifies data in a database. The JDBC driver acts as a translator. The driver receives the client applications request in Java methods, translates it into a format that the database can understand, and then presents the request to the database using the database native protocol. The response is received by the JDBC driver, translated back into Java data format, and presented to the client application. JDBC defines a set of interfaces and classes to be used for communicating with a database. This set of interfaces and classes are all contained in the java.sql package. The entire java.sql package is included in the Java core just as the java.awt or java.lang packages are.

All databases speak SQL. The goal of JDBC API is to give the Java programmer the ability to write applets and applications to access any SQL database servers. The JDBC API consists of a set of classes and interfaces written in the Java programming language that provides a standard API for database developers. JDBC API defines the common ground between the database and application. For example, it defines what commands can be executed, how to execute them, and how data will be formatted. The following JDBC file shows how to list all the employee names from the EMP table.

4.2 Web application

The first step toward differentiating web applications from traditional content-centric websites is to focus on the "application" part of the equation. According to the American Heritage Dictionary, an application is (among other things), "...a computer program designed for a specific task or use." That last phrase, "specific task," is perhaps the most important.

The fundamental purpose of all web applications is to facilitate the completion of one or more tasks. Unlike visitors to traditional, content-centric websites, users of web applications invariably arrive with specific goals, tasks, and expectations in mind. Of course, that's not to say that visitors to content-based websites don't also arrive with certain goals and expectations, but rather that the motivations for using a web application are almost always explicit and precise.

One of the most important implications of this task-based orientation is the degree to which the application should call attention to itself. Compared to content-centric websites, video games, and various forms of entertainment media, application design succeeds not when it draws attention to itself but when it recedes into the background. This requires the designer to find solutions fundamentally natural to both the user and the medium, allowing the application itself to become transparent. The paradox of application design is that the perfect solution is invariably the one that goes unnoticed.

A second implication of their task-based orientation is that web applications have to provide users with various milestones informing them when tasks are complete. In other words, web applications have to support an end-state in a way that content-based sites typically don't.

In addition to the challenges resulting from their focus on task completion, the manner in which web applications function and connect with users highlights other issues affecting web application design.

Without being overly concerned about semantics or classification, it is important to establish an objective means of differentiating between a web application and a traditional website. To wit, in contrast to content-based websites, a web application possesses both of the following observable properties:

- **One-to-one relationship** – Web applications establish a unique session and relationship with each and every visitor. Although this behavior is fundamental to Web applications it is not present in either content-based websites or desktop applications.
- **Ability to permanently change data** – Web applications allow users to create, manipulate, and permanently store data. Such data can take the form of completed sales transactions, human resources records, or email messages to name but a few. This contrasts with web services like Google that allow users to submit information but do not allow them to permanently store or alter information.

Although these two characteristics alone result in a fairly broad definition of web applications, websites that possess both of them necessarily contain a degree of application behavior, logic, and state lacking in traditional content-based sites. In addition, they require a significantly more sophisticated level of user interactivity and interaction design than what is associated with content sites.

This distinction between websites and web applications is most obvious in situations where a given site is almost exclusively composed of either content OR functionality. Newsweek.com (a website) and Ofoto (a web application) are two such cases. However, even popular web destinations such as Amazon, and myYahoo!, sites that combine both content AND functionality, should be considered web applications because they meet these two criteria and therefore exhibit the interactive complexities and behaviors associated with applications.

"In the case of Amazon, this takes the obvious form of personalized content and complex transactions, as well as a variety of other functions including the creation and storage of, the uploading and ordering of digital photographs, the editing and tracking of orders, and many others. That's not to say that all online stores qualify as web applications; in fact most don't. But Amazon and other stores of similar sophistication have the same characteristics and design considerations as more traditional applications such as email and contact management." Barry Eaglestone and Mick Ridley. *Web Database Systems*. McGraw – Hill Book Company. (2001).

Whether any particular application has sufficient complexity to require a highly skilled interaction designer is a question that can only be answered on a case-by-case basis. The point remains, however, that if a web property establishes a one-to-one relationship with its users and allows those users to edit, manipulate, and permanently store data, then it possess certain capabilities and complexities that distinguish it from traditional content-centric websites.

One of the most significant challenges and benefits results from the one-to-one relationship web applications form with their users.

Because a web application requires each user to uniquely identify themselves to the system, typically through a username and password pair, the application can be dynamically altered from one user to the next. This can take both the obvious form of personalized content and the more subtle and complex form of personalized functionality based on roles and privileges. This type of dynamic behavior allows a complex corporate accounting application, for example, to provide different functionality to account managers, regional directors, corporate executives, etc.

Although this type of capability has been a mainstay of enterprise applications for some time, many less sophisticated or expensive applications now employ this behavior. For example, consumer-based online services can add and remove features or advertising based on whether or not a particular user has paid a subscription fee.

More than any other interactive medium, a web application has the ability to adapt itself to each user, providing them with a personalized and unique experience. Accommodating the full range of permutations afforded by this capability is a unique and significant design challenge. Because various functions, interface controls, and data can dynamically come and go from the interface, designers are forced to think in terms of modular components that are simultaneously harmonious and autonomous.

In the same way that it is practically impossible for a visual designer to fully anticipate how a given web page will look in every situation, the designers of large-scale applications also struggle to fully document and consider every possible permutation of functionality and data.

Another unique design challenge associated with web applications results from their ability to allow users to make permanent changes to stored data. Because web applications are fundamentally database applications - that is, they store and present information contained in a defined database structure - the user's information almost always has to fit within a predetermined format. Certain fields are present; some fields are required, others are not; some require a specific type of value; and still others require a value within a precise range. The result of all this is a level of data validation and error recovery unseen in either content-based websites or most desktop applications.

Accommodating this behavior requires the designer to carefully consider the task flow from one operation to the next, the full scope of potential errors, the most expedient way to recover from errors, the ideal solution for avoiding errors altogether.

All this points to one critical conclusion: web applications are a new form of interactive media, distinct from both content-based websites and traditional desktop applications. Therefore, the creation of truly useful and usable web applications requires the designer to understand, appreciate, and exploit the unique capabilities, limitations, and conventions of this new medium rather than simply approaching the design problem from the perspective of more established interactive mediums.

CHAPTER 5
CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In any project, it is very important to review the goals that one has initially set and to compare them with the final output that is displayed. After conducting this process one can measure the success of his or her project. Author is very pleased to say that he has been able to satisfy all of the objectives of the project and the final product displays the initial goals.

The objectives that were initially set out are contained in the previous sections. With respect to all of the objectives, author has included the findings with regard to each aspect in the relevant sections and he will urge the readers of this dissertation to carefully go through it to ensure that they find all of the relevant information.

This has been a wonderful learning experience for author from the start right up to the finish of this project. He has been able to learn a great deal of information with respect to web application and also with respect to the database mechanisms that are used as the platform for them. Moreover author has also been able to learn a great deal with respect to time management and also with respect to working under pressure which he thinks is one of the objectives that were outlined in terms of what we should gain from the carrying out of this project. There were a numerous amount of problems that author had to overcome in order to produce the final product but due to perseverance and also a lot of help from the supervisor he was able to accomplish the objectives within schedule.

One of the most important aspects that author has learnt from this project is that although there are thousands of web applications available on the market today, the need for this type of application will never run out due to the fact that information is always needed to flow across the internal organization and its external users. The application that author has developed is providing this kind of service and he is very optimistic that there will be similar applications that will be developed in the future that provide an even faster and much more efficient way to transfer the information between the entities of the organization.

5.2 Recommendations

With respect to this project there are a few recommendations that author would like to make for expansion and continuation of this project.

The first suggestion that author would like to make is that this project or idea be implemented into the structure of UTP. The reason for this suggestion is that UTP currently uses a manual system to acquire the Logbooks from the students which are undergoing the industrial training. The developed system will allow its users to submit the Logbooks virtually, using web-based approach. This recommendation will go along way in saving students as well as other concerned parties a lot of time.

Another suggestion that author would like to make for future expansion is to expand the scope of the developed Web-based Logbook system. Currently this application is designed for only students who are undergoing the industrial training. The suggestion is that if the user is required to submit the weekly reports about the current progress should be incorporated into this system so that this tool can be available for all students in this university. For example, if you are required to submit the weekly reports during the Final Year Project, so the system could also be used for that. The web application should also be expanded for other projects that are done in the university so that the submission procedure could be done faster and the students will have more time to use the resources available to students when they are carrying out projects.

Another suggestion is to develop an online help resource for students that deal with academic issues and also other aspects that affect UTP students. The proposed online system could be incorporated into the developed web-based system to deal with the academic issues. The various links could be added to the Web-based Logbook system for connection with the help or any other related systems.

With regard to this project author has not incorporated all of the aspects of web application due to some of the feasibility factors. Thus the last suggestion would be to expand on this project to try and incorporate more of the aspects of web application and database mechanisms so that the application can provide greater functionality to the students.

REFERENCES

Whitten, L.J and Bentley, D.L and Dittman, C.K. 2001. *Systems Analysis and Design Methods*, (5), New York, McGraw-Hill.

Kirk, C. 1998. *Internet and Web Answers!*, Berkeley, Osborne McGraw-Hill.

Gilster, P. 1996. *Finding It On The Internet*, (2), New York, John Wiley & Sons, Inc.

Belew, K.R. 2000. *Finding Out About*, New York, Cambridge University Press.

R. Lee. A Quick Guide to Web ServerAcceleration.http://www.novell.com/bordermanager/accel.html

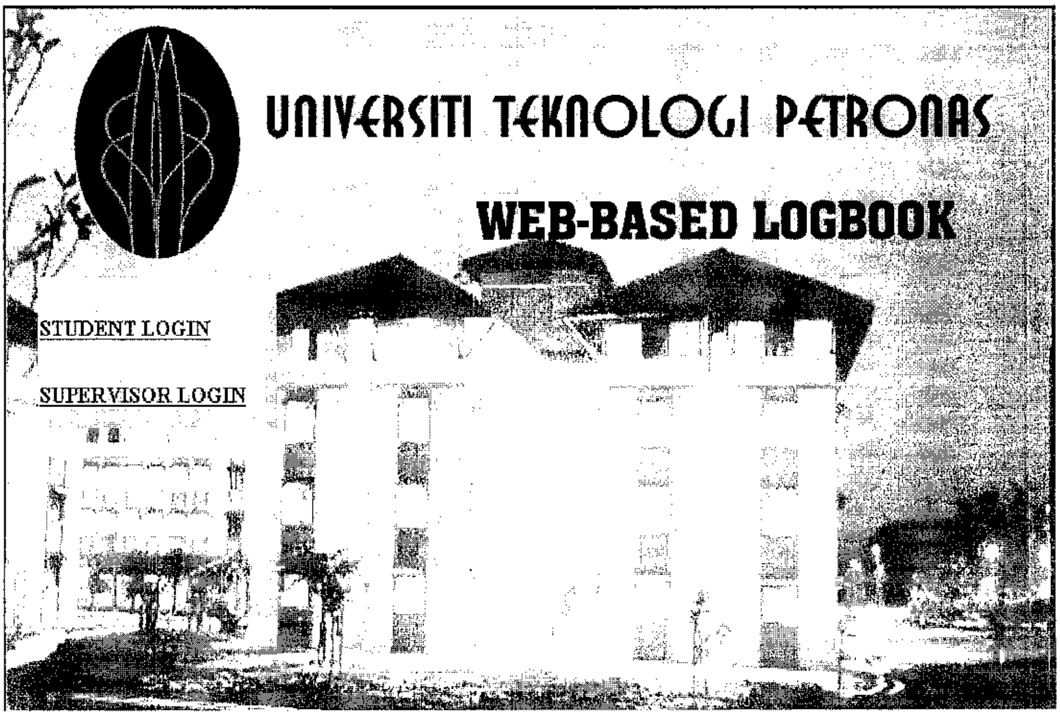
J. Challenger, P. Dantzig, and A. Iyengar A Scalable and Highly Available System for Serving Dynamic Data at Frequently Accessed Web Sites.
In *Proceedings of ACM/IEEE SC98*, November 1998.

A. Chankhunthod et al. A Hierarchical Internet Object Cache. In *Proceedings of the 1996 USENIX Technical Conference*, pages 153-163, January 1996.

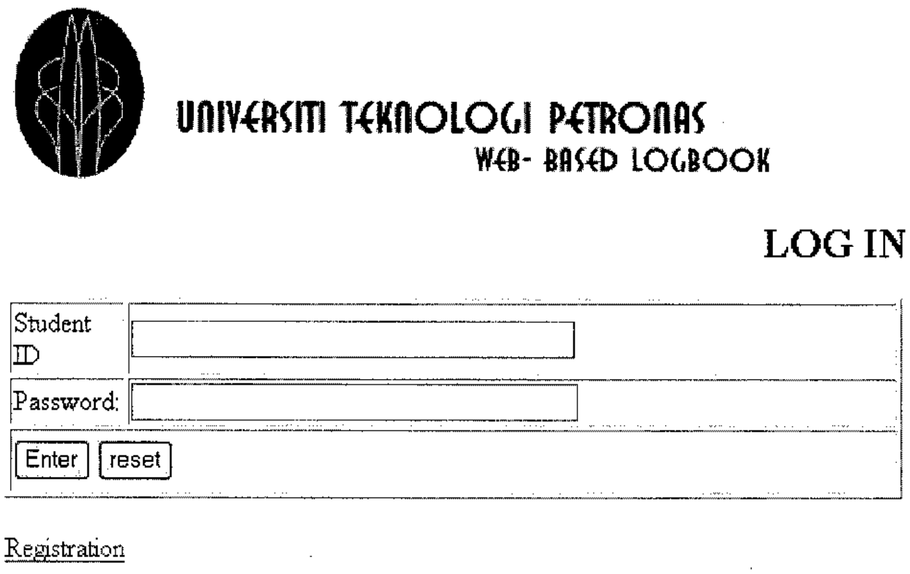
Resonate, Inc Central Dispatch - Data Sheets.
http://www.resonate.com/products/central_dispatch/data_sheets.html.

APPENDICES

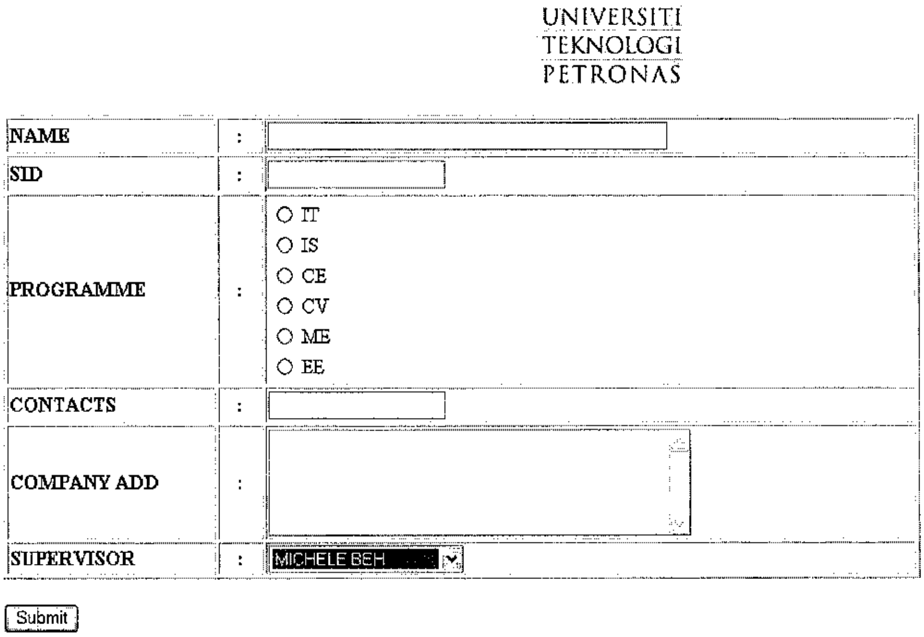
Login page



Student login

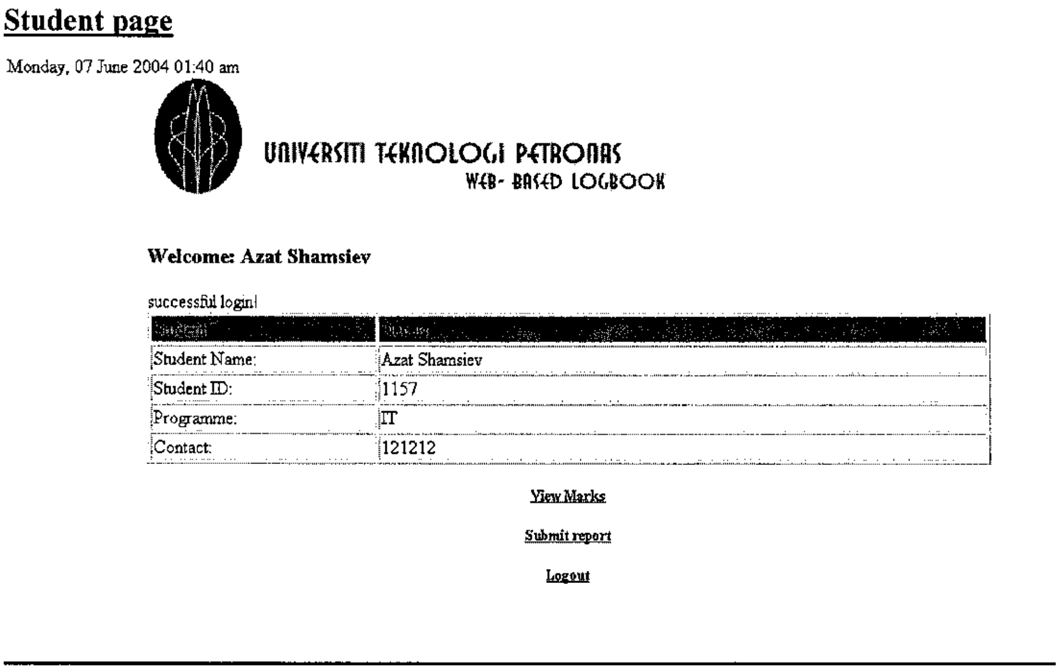


Student registration



UNIVERSITI
TEKNOLOGI
PETRONAS

NAME	:	
SID	:	
PROGRAMME	:	○ IT ○ IS ○ CE ○ CV ○ ME ○ EE
CONTACTS	:	
COMPANY ADD	:	
SUPERVISOR	:	MICHELE BEH



Student page

Monday, 07 June 2004 01:40 am

Welcome: Azat Shamsiev

successful login!

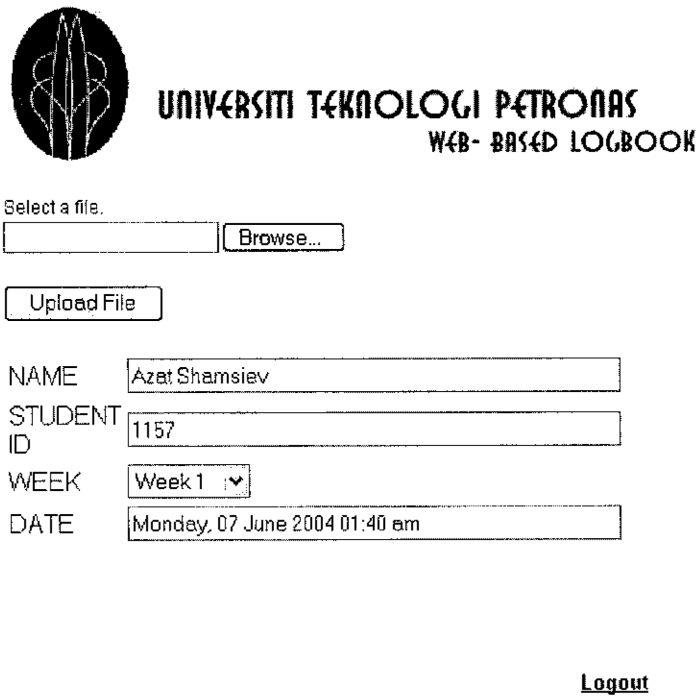
[illegible]	[illegible]
Student Name:	Azat Shamsiev
Student ID:	1157
Programme:	IT
Contact:	121212

View Marks

Submit report

Logout

Student page (upload the report)

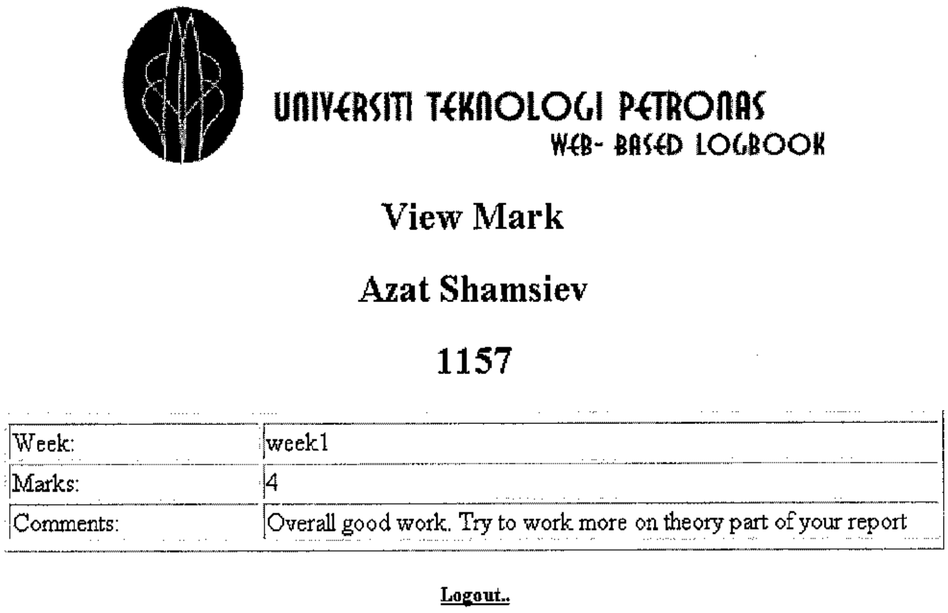


UNIVERSITI TEKNOLOGI PETRONAS
WEB- BASED LOGBOOK

Select a file.

NAME	Azat Shamsiev
STUDENT ID	1157
WEEK	Week 1
DATE	Monday, 07 June 2004 01:40 am

Student page (view the marks)



UNIVERSITI TEKNOLOGI PETRONAS
WEB- BASED LOGBOOK

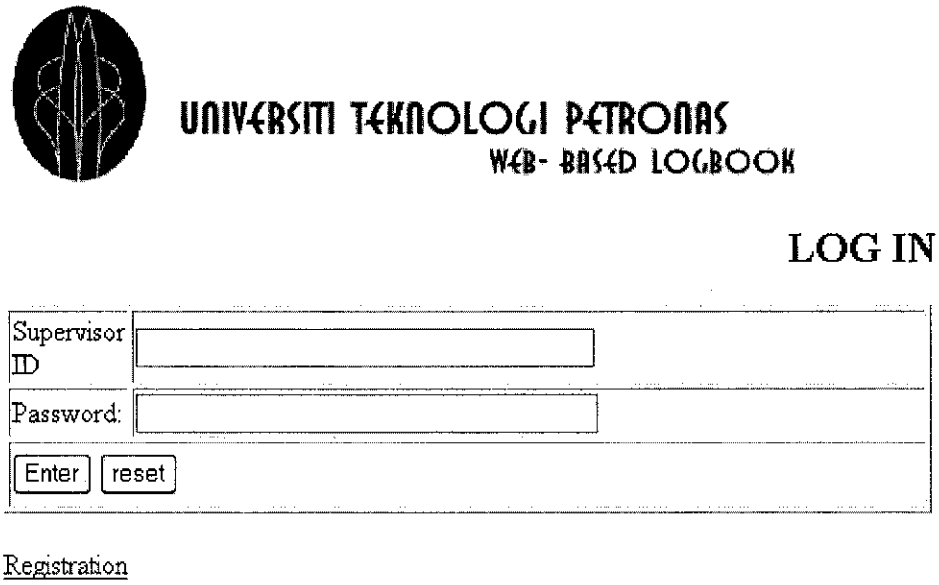
View Mark

Azat Shamsiev

1157

Week:	week1
Marks:	4
Comments:	Overall good work. Try to work more on theory part of your report

Supervisor Login



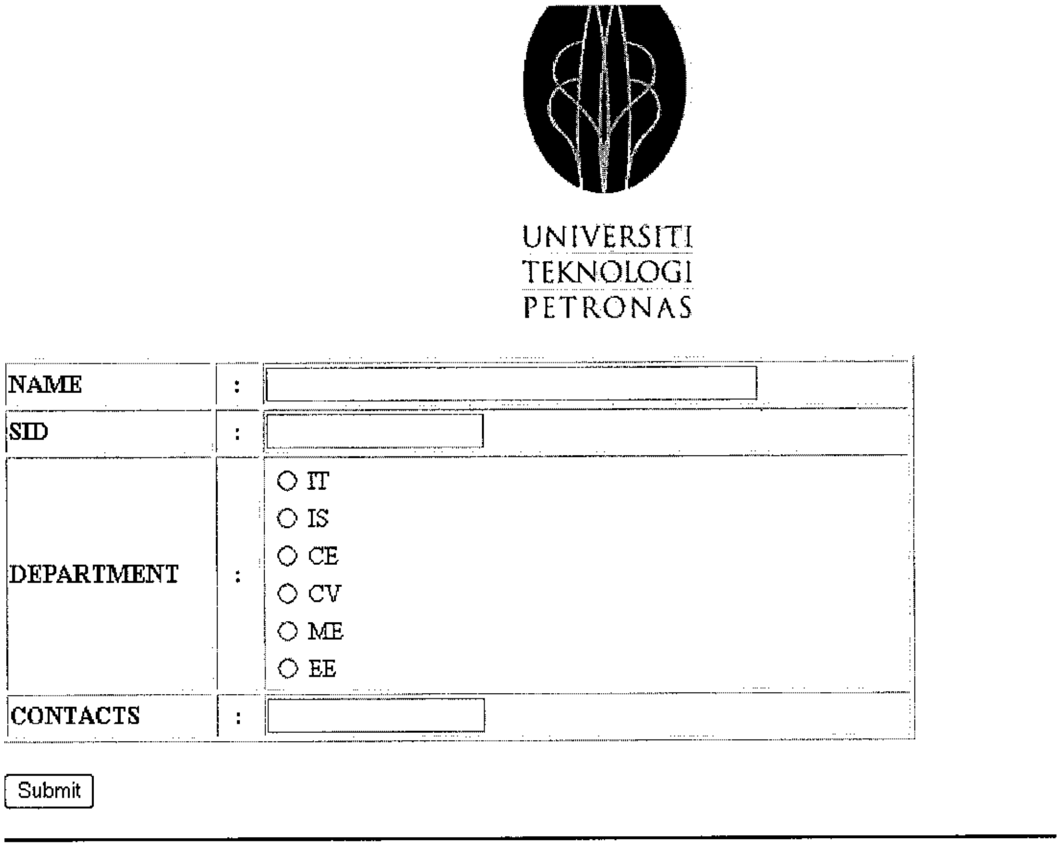
UNIVERSITI TEKNOLOGI PETRONAS

WEB- BASED LOGBOOK

LOG IN

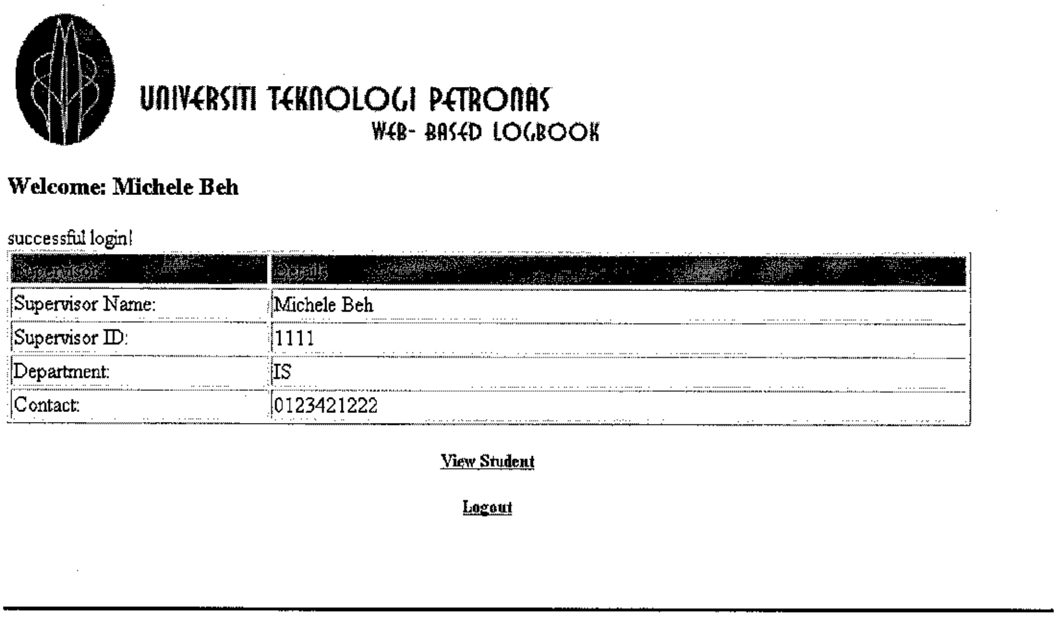
Supervisor ID	
Password:	

Supervisor registration



NAME	:	
SID	:	
DEPARTMENT	:	○ IT ○ IS ○ CE ○ CV ○ ME ○ EE
CONTACTS	:	

Supervisor page



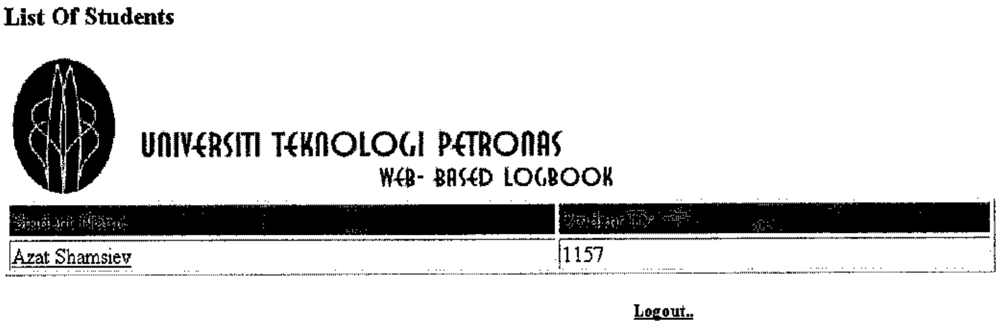
UNIVERSITI TEKNOLOGI PETRONAS
WEB- BASED LOGBOOK

Welcome: Michele Beh

successful login!

Supervisor	Details
Supervisor Name:	Michele Beh
Supervisor ID:	1111
Department:	IS
Contact:	0123421222

Supervisor (list od the students)



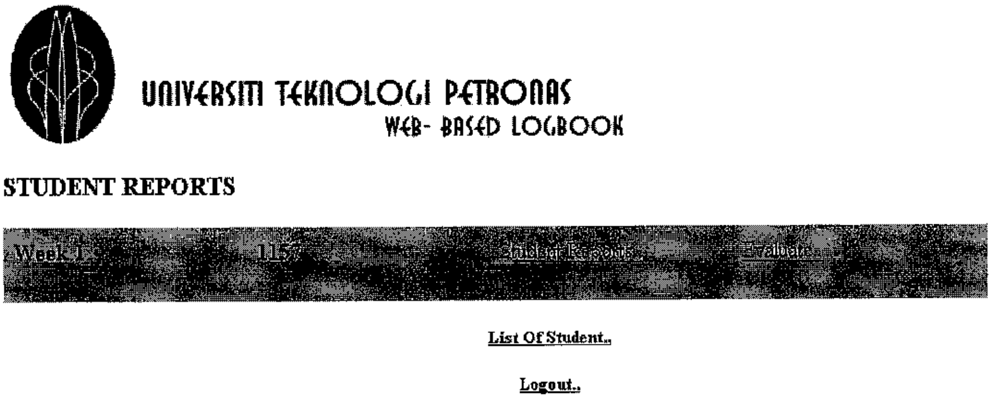
List Of Students

UNIVERSITI TEKNOLOGI PETRONAS
WEB- BASED LOGBOOK

Student Name	Student ID
Azat Shamsiev	1157

Logout..

Supervisor (view the report)



UNIVERSITI TEKNOLOGI PETRONAS
WEB- BASED LOGBOOK

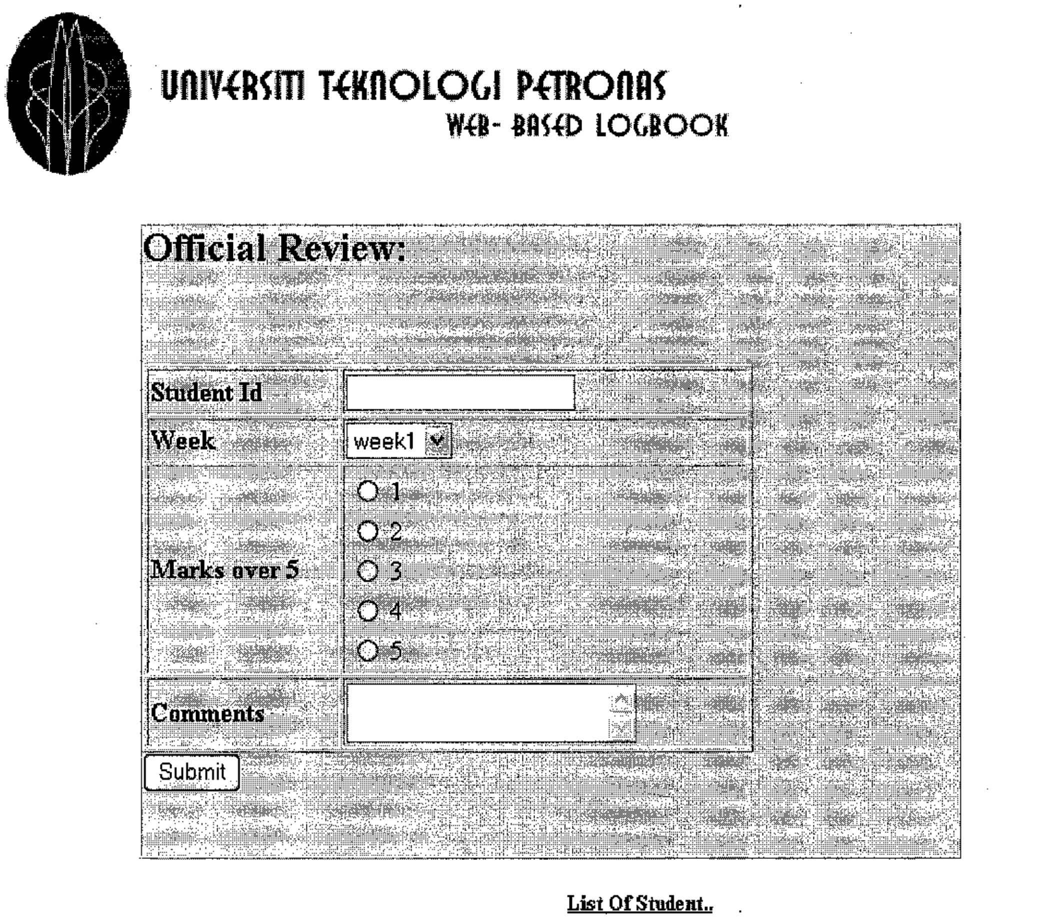
STUDENT REPORTS

Week 1	1157	Student Reports	Evaluate

List Of Student..

Logout..

Supervisor (evaluate the report)



UNIVERSITI TEKNOLOGI PETRONAS
WEB- BASED LOGBOOK

Official Review:

Student Id	
Week	week1
Marks over 5	○ 1 ○ 2 ○ 3 ○ 4 ○ 5
Comments	