STUDENTS' FINGERPRINT ATTENDANCE SYSTEM FOR FACULTY OF ENGINEERING

By
MOHAMMAD KAMAL AL-DIN ABBAS
INDEX NO: 044061

Supervisor
Prof. Sami Sharif

A REPORT SUBMITTED TO
University of Khartoum
in partial fulfillment of the requirements for the degree of
B.Sc (HONS) Electrical and Electronic Engineering
(COMMUNICATION ENGINEERING)
Faculty of Engineering
Department of Electrical and Electronic Engineering
July 2009
DECLARATION OF ORIGINALITY

I declare that this report entitled “Students' Fingerprint Attendance System” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature: MOHA_KAMAL
Name: MOHAMMAD KAMAL AL-DIN ABBAS
Date: July 2, 2009
Acknowledgement

I would like to express my great gratitude towards my supervisor, Prof. Sami Sharif for his help and support. Also I would like to thank personnel and engineers in department of "Information Technology Network" for their full co-operation. I also give thanks to all others who have given me support for this project or in other aspects of my study at Faculty of Engineering.
Abstract:
A finger-print based attendance system has been established to track students' attendance reliably in various sessions through the year. It mainly consists of fingerprint reader connected via Ethernet LAN to PC containing interface through which attendance data received from reader are displayed as well as required system information is handled. Database was constructed to store all system information about students, departments, courses and lecture times. It has been constructed using SQL database management system while interface built using WAMP package and PHP scripts. Another Visual Basic interface has been built to connect reader and PHP interface. Random students have been assigned unique ids for each and their fingerprints caught and stored in the reader. They were also assigned to specific courses attendance system page through the interface. Random information about lecture times, courses and departments’ codes have been inserted into the database for testing. Attendance of previous students has been taken at different times for testing. At each time attendance of students has been obtained correctly through the required course attendance page in the interface.
# TABLE OF CONTENTS

TITLe i
DECLARATION OF ORIGINALITY ii
ACKNOWLEDGMENTS iii
ABSTRACT ix
TABLE OF CONTENTS x
LIST OF TABLES xi
LIST OF FIGURES xii
LIST OF ABBREVIATIONS xiii

CHAPTER 1: INTRODUCTION 1
  1.1: Problem Statement 1
  1.2: Background 1
    1.2.1: Fingerprint technology background 1
    1.2.2: Background about problem's solution 2
  1.3: Motivation 2
  1.4: Objectives 3
  1.5: Report Layout 3

CHAPTER 2: LITERATURE REVIEW AND DATA COLLECTION 5
  2.1: Criteria for device selection 5
    2.1.1: Fingerprint recognition algorithm 5
    2.1.2: Scanner 5
    2.1.3: Other preferable features about reader 6
  2.2: Previous work 7
    2.2.1: Faculty of medicine (U of K) 7
    2.2.2: Biometric identification system by research Company in Ploiesti 7
    2.2.3: Time and attendance system for labor management 10
  2.3: Database management system (DBMS) 13
  2.4: Site survey 14
  2.5: Networking 15

CHAPTER 3: METHODS 17
  3.1: General requirements for any attendance system 17
  3.2: Specified requirements for operation of project 18
  3.3: Attendance tracking method 20
3.4: Received reader data format 21
3.5: Database construction 21
3.6: Interface design 24

CHAPTER 4: IMPLEMENTATION AND RESULT 26
4.1: Required steps 26
4.2: Interfacing and results obtaining 29
4.3: Results discussion 38

CHAPTER 5: PROJECT REVIEW 40
5.1: Results and Objectives Compatibility 40
5.2: Problems encountered 40
5.3: Possible future development 41

Appendix A
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table Number</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 2.1</td>
<td>SQL limitations</td>
<td>14</td>
</tr>
<tr>
<td>Table 3.1</td>
<td>used fingerprint reader specifications</td>
<td>18</td>
</tr>
<tr>
<td>Table 3.2</td>
<td>functions of BioBridge tool</td>
<td>20</td>
</tr>
<tr>
<td>Table 3.3</td>
<td>database tables and their fields</td>
<td>24</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>students' assigned IDs</td>
<td>26</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>reader identification for various students</td>
<td>27</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Lecturetimes table</td>
<td>28</td>
</tr>
<tr>
<td>Table 4.4</td>
<td>Courses table</td>
<td>28</td>
</tr>
<tr>
<td>Table 4.5</td>
<td>departments table</td>
<td>29</td>
</tr>
<tr>
<td>Table 4.6</td>
<td>lecturetimes table after modification</td>
<td>35</td>
</tr>
<tr>
<td>Table 4.7</td>
<td>Lecturetimes table after modification (2)</td>
<td>37</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure Number</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Fingerprint bifurcation and termination</td>
<td>1</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>System architecture for a single location</td>
<td>8</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Architecture of a distributed system</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Module architecture for labour attendance system</td>
<td>11</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>Deployment diagram of labour attendance system</td>
<td>12</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Interaction among WAMP components in response to page request</td>
<td>19</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Attendance data transfer from reader to general table and then to courses tables</td>
<td>22</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Transfer of data to courses tables according to Lecture times table</td>
<td>23</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Modules of system interface</td>
<td>29</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Student's registration</td>
<td>30</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Registration error</td>
<td>30</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>View of registered students</td>
<td>31</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Selection of courses of 5\textsuperscript{th} level in electrical dept.</td>
<td>31</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Displaying courses of 5\textsuperscript{th} level in electrical dept.</td>
<td>32</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Students assigned to java course</td>
<td>32</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>List of identified students</td>
<td>33</td>
</tr>
<tr>
<td>Figure 4.9</td>
<td>General table contents</td>
<td>33</td>
</tr>
<tr>
<td>Figure 4.10</td>
<td>Artificial intelligence attendance page</td>
<td>34</td>
</tr>
<tr>
<td>Figure 4.11</td>
<td>Java attendance page</td>
<td>34</td>
</tr>
<tr>
<td>Figure 4.12</td>
<td>Security attendance page</td>
<td>35</td>
</tr>
<tr>
<td>Figure 4.13</td>
<td>Artificial intelligence attendance page (2)</td>
<td>35</td>
</tr>
<tr>
<td>Figure 4.14</td>
<td>Java attendance page (2)</td>
<td>36</td>
</tr>
<tr>
<td>Figure 4.15</td>
<td>Security attendance page (2)</td>
<td>36</td>
</tr>
<tr>
<td>Figure 4.16</td>
<td>Artificial intelligence attendance page (3)</td>
<td>37</td>
</tr>
<tr>
<td>Figure 4.17</td>
<td>Java attendance page (3)</td>
<td>37</td>
</tr>
<tr>
<td>Figure 4.18</td>
<td>Security attendance page (3)</td>
<td>38</td>
</tr>
</tbody>
</table>
**LIST OF ABBREVIATIONS**

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Identifier</td>
</tr>
<tr>
<td>CCD</td>
<td>Charge Couple Device</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency Identifier</td>
</tr>
<tr>
<td>RJ45</td>
<td>Registered Jack 45</td>
</tr>
<tr>
<td>RS232/485</td>
<td>Recommended Standard232/485</td>
</tr>
<tr>
<td>SQL</td>
<td>Structure Query Language</td>
</tr>
<tr>
<td>PHP</td>
<td>Hypertext Preprocessor</td>
</tr>
<tr>
<td>DBMS</td>
<td>DataBase Management System</td>
</tr>
<tr>
<td>MTTF</td>
<td>Mean Time To Failure</td>
</tr>
<tr>
<td>FAR</td>
<td>False Acceptance Ratio</td>
</tr>
<tr>
<td>WAMP</td>
<td>Windows Apache MySQL PHP</td>
</tr>
<tr>
<td>AES</td>
<td>American Encryption Standard</td>
</tr>
</tbody>
</table>
CHAPTER 1: INTRODUCTION

1.1: PROBLEM STATEMENT:
The target of this project is to establish a students’ fingerprint-based attendance system for the faculty of engineering that holds attendance of students in the various sessions throughout the year. The main concept of the system is built on dealing with fingerprint readers that are used to obtain the students' fingerprints. Dealing with fingerprint readers is to be through an interface. The interface to be designed is intended to display and provide handling means with all information that is necessary for the attendance system. This information relates to students' identities (including their IDs, departments and levels), courses and attendance state of students in each course throughout the year. All this information is to be included in a dedicated database for the attendance system which should be managed and modified throughout the interface. This is the software side of the problem. The other side is formulating a view about the specification of the network through which the various readers are connected to the server.

1.2: Background:
1.2.1: Fingerprint technology background:
This project is based mainly on fingerprint identification technology. Fingerprint belongs to a group that is called "Biometrics". Biometrics represents automated methods for person identification based on a physiological or behavioral characteristic. Among the features measured are: facial features, fingerprints, iris and retina features, hand veins, hand geometry, handwritten signature, keystroke dynamics and voiceprint. Fingerprints are represented by the pattern of ridges and furrows on the surface of a fingertip. The fingerprints are unique and the patterns remain unchanged throughout life. Fingerprints are so distinct that even the ones of identical twins are different. The prints of each finger of the same person are also different.

The fingerprint scanner captures an image of the fingerprint and uses complex fingerprint identification algorithms to either convert the image into a unique "map" of minutiae points or analyze the pattern. Minutiae points are local ridge characteristics that occur at either a ridge bifurcation (split) or a ridge ending (termination).

![Fingerprint bifurcation and termination](image)

Figure 1.1: fingerprint bifurcation and termination
This technique reads specific fingerprint 'ridge' characteristics and assigns an x/y co-ordinate. In most countries, a minimum of twelve of these points are legally required for positive identification in a criminal case, a typical biometric fingerprint reader can record in excess of 40 points\[1\]. Only the data containing the location of the points of minutiae is stored in the template, not the actual fingerprint image. This keeps the file size to a minimum and helps prevent fraud as a fingerprint cannot be recreated from the stored template.

Another approach, which uses image-based methods, tries to do matching based on the global features of a whole fingerprint image. Sub-areas of the ridge thickness, curves, or density are some of the finger’s features. The area around the minutia, with low curvature, or combination of ridges is taken from the fingerprint. The extracted area is then processed and compared with a registered template. It is an advanced and newly emerging method for fingerprint recognition and it is useful to solve some intractable problems of the first approach.

Fingerprint technology has been spread widely in various fields in the life. The main field historically was the criminal investigations in which, fingerprints of criminals are recorded in database in order to specify their identities later on. Nowadays the use of this technology is spreading in the civil fields which this project belongs to. For example, in some countries fingerprint recognition systems are used in determining eligibility for government benefits and for ensuring that persons do not exceed their lawful allocation of goods such as social services benefits and also in identities specification in voting operation. It also has been used widely in companies to recognize times of when personnel and workers arrive and leave.

1.2.2: Background about problem's solution:
1- Gathering information about fingerprint readers’ characteristics to obtain the suitable one for the system requirement.
2- Determination of the reader's output format through a specific tool or terminology.
3- Determination of information required for the attendance system and networking.
4- Constructing the database in which pre-determined information is included.
5- Determining operations to be performed through the interface.
6- Designing the interface and testing the system.

1.3: Motivation:
1- Fingerprint is the most convenient and most reliable way to identify someone.
2- Most interfaces that supplied with fingerprint readers are dedicated for personnel attendance in companies and establishments where they fingerprint just when they arrive and leave. Such those interfaces are not compatible with the aspects that should be considered when students' attendance tracking system with required constraints is to be built.
3- Easiness of networking within the faculty.
4- Unreliability and time wastage of current manual attendance system.
5- To deal with a new technology that has been.

1.4: Objectives:
This project's target mainly is to provide an efficient and reliable attendance system for students in various sessions throughout predetermined constrained attendance tracking method. It is intended to provide a graphical interface that makes it easy and convenient to handle and display all information related to the system.

1.5: Report Layout:
Following chapters in this report are as following:
1- Chapter 2 contains 5 subsections:
a- Subsection 2.1: provides proposed criteria containing some specifications for appropriate selection of fingerprint reader according to what is required in the system.
b- Subsection 2.2: provides literature review and previous work that has been done in field of implementation of fingerprint-based attendance systems.
c- Subsection 2.3: provides overview about database management system concept including overview about SQL and its limitations to be considered when used.
d- Subsection 2.4: provides site-survey about engineering faculty needed to determine number of fingerprint readers that would be needed as well as their required specifications for future implementation of the system in the whole faculty.
e- Subsection 2.5: provides concept about desired networking links means including various types of cables and their specifications.

2- Chapter 3 contains 6 subsections:
a- Subsection 3.1: provides concept about requirements that would be needed in an implementation of fingerprint attendance system generally.
b- Subsection 3.2: provides concept of hardware and software requirements that have been used in implementation of this project.
c- Subsection 3.3: provides concept about attendance tracking method that apply constraints required for the successful and reliability of the attendance system and that forms the design concept of the code.
d- Subsection 3.4: provides concept about format of data received from the fingerprint reader and means used for this.
e- Subsection 3.5: provides concept about construction of database to be used as system information storage. It also shows the mechanism of interaction between various tables that will be implemented throughout queries.
f- Subsection 3.6: provides concept about the design of interface, its modules and function of each module within it.

3- Chapter 4 contains 3 subsections:
a- Subsection 4.1: shows steps that have been required to be performed with the fingerprint reader and the database in order to get the required results.
b- Subsection 4.2: shows how interface was used to handle system information as well as results of attendance taken in different times and its displaying throughout the interface.
c- Subsection 4.3: includes discussion of the obtained results and provides interpretation for them.

4- Chapter 5 contains 3 subsections:
a- Subsection 5.1: provides evaluation of compatibility between the intended objectives at the beginning of the project and results obtained at the end.
b- Subsection 5.2: shows the problems that have been encountered throughout implementation of this project, their reasons and how they could be handled.
c- Subsection 5.3: provides some suggested possible developments that can be applied to the system in the future.
CHAPTER 2: LITERATURE REVIEW AND DATA COLLECTION

2.1: CRITERIA OF DEVICE SELECTION:
Fingerprint reader is the basic component in the system. It should be ensured that its characteristics are compatible and go with the objectives of the system. Following subsections provide some characteristics to be considered:

2.1.1: FINGERPRINT RECOGNITION ALGORITHM:
Generally fingerprint readers operate on one of two fingerprint recognition algorithms, which are, fingerprint identification or fingerprint verification:
1- Fingerprint identification is to specify one person’s identity by his fingerprint(s). Without knowledge of the person’s identity, the fingerprint identification system tries to match his fingerprint(s) with those in the whole fingerprint database.
2- Fingerprint verification is to verify the authenticity of one person by his fingerprint. The user provides his fingerprint together with his identity information like his ID number. The fingerprint verification system retrieves the fingerprint template according to the ID number and matches the template with the real-time acquired fingerprint from the user.

Fingerprint identification requires more processing since the applied fingerprint is to be compared with those in the whole fingerprint database. This process could take significant time, especially in the case of large databases such in case of faculty or university containing thousands of students. Fingerprint verification requires less processing since the applied fingerprint is to be compared with a single template in the database specified by the ID. So fingerprint verification algorithm is preferred in case of attendance system with large database on base of time consuming (but on the other hand fingerprint identification could be more convenient in the case of small database since student is to fingerprint only. Small databases can be applied if each device is dedicated for a group of students but this will be costly).

2.1.2: SCANNER:
The most important part in any fingerprint-based system is the scanner or sensor which is the ‘gate’ of the fingerprint into the system. Two types of scanners have been viewed:
1- The Optical Scanner:
Optical scanners use a CCD or charge couple device much like the ones used in camcorders and digital cameras. The CCD makes use of photo sites that are sensitive to light. Each photo site generates an electrical signal in the presence of photons produced by light emitting diodes. These photo sites are very small they cover the entire screen where a user will place his or her finger.
Since photo sites only emit electricity in response to the presence and the strength of light, the overall result will be an inverted image of the finger. This image is converted into digital form and checked for clarity and sharpness before being compared to the saved images of other prints in the database.

2- The Capacitance Scanner:
Capacitance scanners make use of electrical current to form a fingerprint image. The sensor part of the capacitance scanner is made up of tiny plates that act as conductors. These conductors are then overlaid by an insulating film. These form the cells that make up the sensor's semiconductor chip or chips.

An inverting operational amplifier is then used in conjunction with an electrical circuit and the sensor to determine the relative differences in capacitance and voltage value of the different areas of a finger. A scanner processor is then used to separate the valleys and the ridges in the resulting data. This results in a reverse image of a fingerprint.

From previous context it can be shown that capacitance scanners are more secure than optical ones since they have higher ability to detect fake fingerprints. This is due to that capacitance scanners depend on the resulting capacitance of the portions of the finger that are in contact with the sensor plus the air capacitance on the ridges of a finger instead of just relying on the amount of reflected light like what the optical scanners do. A picture of a fingerprint pressed onto the sensor, can trick an optical scanner that it is processing a real fingerprint. On the other hand optical scanners have advantages of:
1- Resistance to shock and extreme weather.
2- Lower maintenance cost.
3- Bigger sensing area leading to better resolution.

2.1.3: OTHER PREFERABLE FEATURES OF READER:
A point that should be considered in hardware devices associated with scanners that it is preferable for efficiency of system that database is included in the device itself rather than the server to reduce amount of time required for processing the acquired fingerprint. Also some devices are connected wirelessly to the server; this could limit allowable distance between them, hence reducing the system efficiency and convenience in addition to healthy effect that transmitted waves may result in. So devices with wiring capabilities could be better although they may involve increased installation and maintenance costs.

Another feature that is preferred to be available in the reader is the capability of extracting data to the server while it is offline, for example, by using USB port that enables transaction via USB flash disk. This feature could be important when electricity is gone off (and reader is operated using battery). Also the reader operation should be controlled by specified individuals, that is, it does not start processing of applied fingerprints unless one of the specified individuals fingerprints (in case of attendance system the individuals assumed to be the lecturers).
2.2: PREVIOUS WORK:

2.2.1: FACULTY OF MEDICINE (UNIVERSITY OF KHARTOUM):
Fingerprint attendance system readers applied in the faculty of medicine were interfaced using Visual basic. This interface operates on concept of timing of lectures. Through interface operator on server can determine the time and hall in which session takes place in addition to the subject and level of students. So the operator assumes that attendance taken at the specified time and hall belongs to the targeted students. So the operator has to compare the verified IDs with those of the targeted students. This operation is inconvenient for the operator and is time consuming.

The fingerprint attendance system applied in medicine faculty is no longer working because of some problems that took place during application such as skin sensitivity of some students to scanners, others’ fingerprints are not clear for the reader to capture minutiae from for further processing and the most important problem is hardware failure that takes place more than once in short period requiring inconvenience commitment of maintenance.

The faculty of medicine has replaced the previous interface with another one that deals with attendance taken manually. In this interface the operator has to determine the subject and level of students. Once determined, a list containing the targeted students is displayed with default state of (absent) or (attendant) for all students as specified by the operator. Then the operator can modify the state of each one according to the manual attendance. This algorithm can be applied to an interface that is dedicated for fingerprint-based system.

2.2.2: BIOMETRIC IDENTIFICATION SYSTEM BY RESEARCH COMPANY IN PLOIESTI:

A-PHYSICAL CONSTRUCTION AND CONNECTIONS OF THE SYSTEM:
This system has been implemented for the purpose of management employees' time in companies. The system is based on fingerprint analysis. The fingerprint template is not stored on a memory inside the device, but it is stored on a RFID card memory (one for each user). This allows unlimited number of personnel to use any reader since there is no storage capacity limitation. RFID – Radio Frequency IDentification – is an automated data collection technology, which uses radio communication for data transfer between two entities: a reader and a mobile entity that must be identified. The mobile entity is known as a RFID tag. This tag has two sections: one for radio communication and a memory used for storing data and custom fields. There are mainly two categories of RFID tags: active and passive tags. The active tags are writeable and readable and have an internal power source (a battery), so the lifetime of an active tag is limited. The passive tags obtain the energy from the magnetic field of the reader. These types of tags are smaller, cheaper and could be used an unlimited time so they are ones used in this system. The functionality of RFID passive tags is very simple as following:
When a tag is in the nearby of a reader, it detects the radio signals generated by the reader and starts to transmit the data stored in the memory of RFID cards. The radio signal generated by the
reader offers the power needed to function and the synchronization data for communication between the two entities.

When a person wants to use the system he must place the RFID card in the nearby of the reader, then the system reads the data on the card and asks the user to put the finger on the fingerprint sensor. The fingerprint reader scans the finger of the user and then a template is created in real time. After that the generated template is compared with the one stored in the card’s memory. If the templates match, the comparison score is better than a threshold value, then the person is identified and the time of attendance is obtained.

Each fingerprint access controller with integrated RFID reader has an internal memory where an event log is stored. This log contains every system event such as: successful user identification, failed user authentication, modifications of system’s configuration, user enrollment, user account removal. The event log could be downloaded via RS232/485 or Ethernet (RJ45) interface. The RS485 serial interface could connect up to 127 devices to create a local network. The Ethernet interface uses TCP/IP protocol, so there could be created a distributed time & attendance system with devices connected via Internet. The system architecture is presented in figures below (Figure 2.1 and Figure 2.2):

![Figure 2.1: System architecture for a single location](image)
**B- SOFTWARE CONSTRUCTION OF THE SYSTEM:**
The event log (containing user ID, date, time and successful or fail authentication) is imported to MySQL tables. The database has three tables as following:

1- **Employee_data**: stores the employee personal data and realize the logical connection between them and USER_ID from the event log.

2- **Time_attendance_table**: stores the data about the time the enrolled employee comes to or leaves the building. It is the table that contains the data necessary for computing the working time and eventually the salary for each employee. This table is archived at a preconfigured time interval, default 6 months.

3- **Sys_user_data**: stores the usernames, the passwords and the rights assigned for every person that uses the reporting software. The users may have different ranks such as: system administrators, operators or simple statistics viewers. The privileges are assigned corresponding to the company’s hierarchy.

The statistics software is a web application written in PHP. The application runs on any web server that is PHP enabled, and the statistics are available wherever in the world, where an Internet connection is present. The interface is user-friendly and the system can be used by any
inexperienced user. PHP interface has three modules: one for employee data management, another for system user data administration and one for statistics.

Employee’s data management module is used for creating, removing and modifying employee profiles. The system operator should introduce personal data, information about the workplace (including the hierarchy in the company) and the working time schedule for each employee. This module is used for issuing RFID cards for each employee. All the data is introduced using forms generated dynamically by PHP scripts.

System’s user data management module is used for creating, removing and modifying user accounts. Each user account has some privileges corresponding to the hierarchy of the company, so a department manager can view the statistics only for his subordinates. Using this module the system administrator could define the operating rights for each user and even can delegate administrative privileges.

The statistics module is user-friendly. It generates the presence reports in table or graphic format. The reports can be exported in PDF (Portable Document Format). The statistics are generated in real-time so the human resources personnel can view who is at work, who and when entered or left the building. The statistics module could generate reports about presence for current day, number of hours worked by each employee, lists with present and absent employees and list of employees that are on holiday.

2.2.3: TIME AND ATTENDANCE SYSTEM FOR LABOUR MANAGEMENT [2]:

The system presented here was implemented to cooperate with Microsoft SQL Server 2005 and it was quite natural to explore the capabilities of its business intelligence platform to produce sophisticated and useful analyses that go beyond mere late coming and attendance reports. Microsoft Business Intelligence solutions provide tools for data integration from many sources for data analysis and data reporting. It is possible to analyze data in Microsoft Business Intelligence platform using data warehouses or data mining or both. In SQL Server 2005 there were added more data mining algorithms and the scalability of the platform was extended.

System architecture and functions:
The Access Control and Time and Attendance System was implemented using Microsoft .NET environment with SQL Server 2005 as a database management system. It was assumed the system could be exploited using commercial editions of the SQL Server 2005 as well as the free SQL 2005 Express Edition. The limitations of the latter to 4 GB of the database size and the RAM size of 1 GB are not essential for medium and small organizations, so that the system can be deployed in a broad spectrum of companies and institutions. The main functions of the system are as following:
1- Tracking of late coming and absence including holidays, sick leave, training, business trips, changed/unscheduled shift work etc.
2- Defining individual calendars and schedules.
3- Calculation of time worked during working days and holidays.
4- Balancing holidays, work time and overtime.
5- Producing reports including employees’ absence, overtime or punctuality by personnel criteria such as department, position, day, month etc.
6- Integration and export of data to payroll, HR and other systems.

Figure 2.3 shows the module architecture of the system:

![Figure 2.3: Module architecture of labour attendance system](image)

Functions of applications depicted in the previous figure are as following:

1- Core: Contains libraries used by other applications, the libraries include among others converter, database access, and communication classes.

2- ACAdmin: Is the application devoted to access control system administration, and allows for configuration of ports and controllers, granting user rights, introducing employees, browsing event and alert history.

3- ACService: Performs the communication with online controllers and deals with all controller events.

4- TAAdmin: It is the application designed for the time and attendance system administration.
5- ACGuard: Reports where at a given moment each worker stays and how many persons there are in individual rooms and places.

6- WrkImport: Enables to import data of employees from other systems and common formats, simplifying therefore initialization of the system.

7- ACTester: Is used during deployment and enables to test quickly in field conditions all equipment: online controllers, card readers.

8- SysBackup: Enables the user to make database backup by pressing only one button.

Figure 2.4 shows the system deployment diagram:

The readers are connected to the controllers which communicate with the server through the RS 485 bus, and this connection is encoded by means the AES-128 protocol. Due to the computers are not equipped with the RS 485 input, various converters can be used e.g. RS 232/RS 485, USB/RS 485, and Ethernet/RS 485. All the controllers are managed by the server to which they are connected. The service managing the readers runs on the server and all events and alerts are stored in the SQL Server 2005 database. The database is used by client applications such as control and access, time and attendance and guard ones. The reader deploying cards for new employees registered in the system is connected directly to the computer with ACAdmin or TAAdmin application through the USB port. An employee when entering or leaving the controlled area approaches his card to a reader and his identifier is read and sent to the server. The server application checks whether the card holder is authorized to enter the area. If so the card punch event with its date and time, type and the worker’s identifier is
recorded in the database. The basis of the reports generated for the management purpose are aggregates calculated for individual employees and for each day.

2.3: DATABASE MANAGEMENT SYSTEM (DBMS):
It could be considered the most important part in any system that requires dynamic and flexible to deal with data of large number of human beings. Database management system is a set of software programs that controls the organization, storage, management, and retrieval of data in a database. DBMS are categorized according to their data structures or types. It is a set of prewritten programs that are used to store, update and retrieve a Database. The DBMS accepts requests for data from the application program and instructs the operating system to transfer the appropriate data. When a DBMS is used, information systems can be changed much more easily as the organization's information requirements change. New categories of data can be added to the database without disruption to the existing system.

Organizations may use one kind of DBMS for daily transaction processing and then move the detail onto another computer that uses another DBMS better suited for random inquiries and analysis. Overall systems design decisions are performed by data administrators and systems analysts. Detailed database design is performed by database administrators.
Database servers are computers that hold the actual databases and run only the DBMS and related software. Database servers are usually multiprocessor computers, with generous memory used for stable storage.

A DBMS includes four main parts: modeling language, data structure, database query language, and transaction mechanisms:

- Modeling language:
  A data modeling language to define the schema of each database hosted in the DBMS, according to the DBMS database model. The most important database model is relational model, where database can be viewed as set of relations connecting its components (tables) via common keys between tables and also each table represents a relation that connects its fields together. Relational model simplifies data modification and retrieval processes.

- Data structure:
  Data structures (fields, records, files and objects) optimized to deal with very large amounts of data stored on a permanent data storage device. A DBMS allows large amounts of data to be stored in the most efficient manner possible. The data is stored in lists of independent information. Additional data sets can be added at any time. Relationships between the data sets are defined in the DBMS, and can be altered without affecting the data.

- Database query language:
  A database query language and report writer allows users to interactively interrogate the database, analyze its data and update it according to the users' privileges on data.
d- Transaction mechanisms:
They are mechanisms provided in the order of data integrity. For example mechanism that prevents entering to students into the database with the same ID. DBMS should also provide controlled authorization access to database for security.

There are some factors that should be considered while deciding which DBMS to be used for a specific system. These factors include number of CPUs that can be supported by DBMS at any time. So some DBMSs can be installed on a server with multiple CPUs but it will use only one at a time. Another factor is maximum memory used by the DBMS for its data buffer. If the server provides more than this limit to the DBMS, it will not take advantage of the over-limit memory space. The most important factor is the maximum database size for a specific DBMS. This limit should be considered where DBMS to be selected for data storage of large organizations.

**SQL (Structured Query Language):** is the most important database computer language designed for the retrieval and management of data in relational database management systems (RDBMS). It is fast and efficient in processing queries. Also it is easy to be used. Following table shows some of SQL limitations for some characteristics that should be considered when it is used to store data of large organizations [8]:

<table>
<thead>
<tr>
<th>Characteristic</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Columns per base table</td>
<td>1024</td>
</tr>
<tr>
<td>Columns per select statement</td>
<td>4096</td>
</tr>
<tr>
<td>Columns per insert statement</td>
<td>1024</td>
</tr>
<tr>
<td>Database size</td>
<td>1,048,516 terabytes</td>
</tr>
<tr>
<td>Files per database</td>
<td>32767</td>
</tr>
<tr>
<td>Rows per table</td>
<td>Limited by available storage</td>
</tr>
<tr>
<td>Tables per database</td>
<td>Limited by number of objects in database</td>
</tr>
<tr>
<td>Tables per select statement</td>
<td>256</td>
</tr>
</tbody>
</table>

Table 2.1: SQL limitations

**2.4: SITE SURVEYING:**
The project targets to cover the whole attendance of the faculty of engineering. So it is required firstly to determine number of halls to be covered, the capacity of students for each and number of readers to be used in each site. It may also be required to determine whether readers in sites to be dedicated for specific groups or levels of students or each reader is used to store the fingerprints of all students. This point is determined by distribution of sessions of students' levels according to the timetable of the departments of faculty. Another point to be considered is the identity and location of server including the whole data of attendance.
The faculty of engineering can be divided into two parts, the northern tower and the southern one. Northern tower contains 8 halls (EN21-EN26 in addition to ETs halls). Two of the 8 halls (EN21 and EN26) are bigger than others containing two amphitheaters. These two halls may require two readers if we consider their capacities, but on the other hand they are least used for sessions (specified mainly for sessions gathering two or more departments). ETs halls are also not used widely for sessions. Readers will be used most in EN22, 23, 24 and EN25. So it is suggested that readers used in latter halls to be with higher fingerprint template capacities and transaction log capabilities. It is also preferred for them to be supplied with optical scanners because their maintenance costs are lower than those of capacitance ones (these readers are expected to have smaller mean time to failure MTTF). Since number of students study in this part is about 1000-1500, hence each reader used should have capacity of fingerprint templates of about 3000-5000 because it may needed more than one fingerprint for each student as backup templates. One reader for each hall could be enough leading to total of 8 readers in this part. Southern tower contains about four large halls occupied approximately equally for sessions. Numbers of students in this part is more than previous one, so readers should have larger fingerprint templates capacity and transaction log capabilities. If these requirements are not available, two readers may be located in each hall with specific groups of students use either of them.

2.5: NETWORKING:

Readers from different sites are to be connected to the server. The server includes the database in which attendance event logs are downloaded and stored. Possible physical links between readers and the server include RS232/RS485/RS422 (RS stands for recommended standards by Electronic Industry Association EIA).

RS232 \(^4\) specification allows for data transmission from one transmitter to one receiver at relatively slow data rates (up to 20 K bits/sec) and short distances (up to 50 feet at the maximum data rate). Independent channels are established for two-way (full-duplex) communications. RS422 \(^4\) was designed for greater distances and higher Baud rates than RS232. In its simplest form, a pair of converters from RS232 to RS422 (and back again) can be used to form an "RS232 extension cord." Data rates of up to 100K bits / second and distances up to 4000 Ft. can be accommodated with RS422. RS422 is also specified for multi-drop (party-line) applications where only one driver is connected to, and transmits on, a "bus" of up to 10 receivers.

RS232 and RS422 are not preferable to be operated in a truly multi-point network like that of the attendance system where the server has to receive data from readers and even transmit data (several types of readers are configured remotely with special software running on the server). RS458 could be considered the best. RS485 serial interface could connect up to 127 readers to create a local network. Also, RS485 \(^4\) drivers are able to withstand "data collisions" (bus contention) problems and bus fault conditions. The recommended arrangement of the wires is as a connected series of point-to-point (multi-dropped) nodes, a line or bus, not a star, ring, or multiply-connected network. Star and ring topologies are not recommended because of signal
reflections or excessively low or high termination impedance. Repeaters can be used if needed to extend the network.

Another possible interface between server and readers is the Ethernet. Ethernet networks generally use RJ45 (Registered Jack 45) cables. RJ-45s look similar to the RJ-11 modular telephone connectors. It is recommended that that star configuration to be adopted where readers are connected to the switch via RJ45 cable. The Ethernet interface uses TCP/IP protocol, so there could be created a distributed time & attendance system with devices connected via Internet. Ethernet has advantages that it is not expensive technology to implement and cables used within it are very well shielded, yielding to high noise immunity from other electrical sources.
CHAPTER 3: METHODS AND REQUIREMENTS:

3.1: GENERAL REQUIREMENTS FOR ANY ATTENDANCE SYSTEM:
Establishing a fingerprint-based attendance system consisting of multiple readers connected to a server should be started firstly by using a single reader connected to an ordinary PC then the applied method is extended for a network. As other systems including networking, the problem has two parts to be considered:

1- Hardware to be used which in this system include fingerprint readers, physical connections and interfaces linking the network and also determining whether the server to be used should have special requirements or any PC can be used as server. Selected readers should have the required characteristics that give an efficient and also inexpensive system. Main readers' characteristics to be considered include:

a- Type of scanner used: scanner is critical part in the system since it specifies the efficiency of fingerprint processing operation and also the expected degree of maintenance required and its cost.
b- Fingerprint template capacity: size of database containing students' fingerprint within the device is desired to be as large as possible to decrease number of devices needed and hence the cost of the system.
c- Transaction log capability: this characteristic specifies how much attendance log events the reader can hold at the same time before they are downloaded successfully to the server database. This characteristic determines the period that should not be exceeded to get the attendance from readers and to prevent system confusion to users. This period is determined using the expected usage of device per day or per week.
d- Power source: the readers are DC powered either using batteries or AC/DC adapters. For readers use batteries, determination of their lives is essential to prevent system cut-off. For readers operate with AC/DC adapters determination of acceptable AC level is essential. Using of AC/DC adapters could be more convenient (especially if it is capable to operate more than one reader) unless in cases of continuous power cut-off.
e- False Acceptation Rate (FAR): it is a percentage shows probability that unauthorized users may be recognized due to some error as authorized ones. Generally this percentage is about 0.05% and could be less for modern devices.

2- Second part to be considered is software. It is the mechanism that arranges and controls the operation of whole the system from fingerprinting till issuing reports showing the attendance. Basic software to be considered for operation includes:
a- Database (DB): that is the main part in the whole operation. It should include information about all students and readers involved in the system. Information about students includes their
IDs, names, departments and levels. Information about devices generally should describe their locations in the system. Essential point to be considered in selection of the appropriate database management system is limitations on database size. So the selected DBMS must be capable of containing the required information about students as well as about devices.

b- Interface: it is the window through which the operator of the system deals with the database. It provides convenient and flexible means for the operator to modify the database with no needing to be experienced with the way of operation of the database management system used to construct the database. The interface also should be designed to contain clear steps for the users through which they can show the attendance state. It is desirable that interface showing attendance can be loaded to web page so everyone can show at any time. Some parts of interface used for database modification should be secured and allow accessing only for the operator of the system by using, for example, username and password.

3.2: SPECIFIED REQUIREMENTS FOR OPERATION OF PROJECT:

1- FINGERPRINT READER:

Table 3.1 shows the most important specifications of the fingerprint reader that has been used:

<table>
<thead>
<tr>
<th>Specification</th>
<th>Capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fingerprint template capacity</td>
<td>3000</td>
</tr>
<tr>
<td>Transaction log</td>
<td>12000</td>
</tr>
<tr>
<td>Supported connections</td>
<td>TCP/IP, RS232, RS485, USB flash disk</td>
</tr>
<tr>
<td>Sensor</td>
<td>Optical sensor</td>
</tr>
<tr>
<td>Power source</td>
<td>AC/DC adapter</td>
</tr>
</tbody>
</table>

Table 3.1: used fingerprint reader specifications

2- WAMP PACKAGE:

WAMP[^6] is package of independently-created programs installed on computers that use a Microsoft Windows operating system. The interaction of these programs enables dynamic web pages to be served over a computer network, such as the internet or a private network. "WAMP" is an acronym formed from the initials of the operating system (Windows) and the package's principal components: Apache, MySQL and PHP. Other programs may also be included in a package, such as phpMyAdmin which provides a graphical user interface for the MySQL.

Apache[^7] acts as web server. Its main job is to parse any file requested by the browser and display the correct results according to the code within that file. PHP originally stands for "Hypertext Preprocessor". PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. PHP is a scripting language, as opposed to a programming language: PHP was designed to write Web scripts, not standalone applications. PHP scripts run only after an event occurs—for example, when a user submits a form or goes to a URL. PHP is server-side, that is, everything PHP does occurs on the server. A Web server application, like Apache, is required and all PHP scripts must be accessed.
through a URL (http://-something, for local host-http://localhost/-php file name). MySQL is the most common used DBMS as shown in previous chapter.

Interaction among the components of WAMP for a specific page request is shown in figure 3.1:

![Diagram of WAMP components interaction](image)

**Fig 3.1:** interaction among WAMP components in response to page request

When user requests page written in php, Apache server reads the php code that contains MySQL queries and then processes it according to its scripted directions. These queries cause data to be retrieved from MySQL database to be sent to the user through the server.

Generally, only a user with administrator privileges can install a WAMP package. This means that these packages cannot be installed to a hosted service but only to a computer to which the user has complete access. So, for the attendance system WAMP package should be installed in the server of the system, especially to avoid problems that could take place when host database to be linked to remote server.

3- **BioBridge tool:**

BioBridge tool is a software development kit supplied with reader used in project. Its main function is to bridge the attendance system to the reader, that is, determining method by which data can be extracted from the reader in readable and understood format, to be transferred to the database and displayed by the interface. In addition to this, this software tool allows operator of the system to prepare the configurations of readers remotely from the server and also gives capability of extracting and reading attendance log offline through USB flash disk.

BioBridge tool contains list of functions to perform the required bridging between reader and system. These functions should be included within environment of Visual Basic to perform as desired. So it is needed to construct a "communicating interface" using Visual Basic. Table 3.2 shows most important functions and what they do:
<table>
<thead>
<tr>
<th>BioBridge function</th>
<th>Parameters required</th>
<th>Performed function</th>
</tr>
</thead>
<tbody>
<tr>
<td>BioBridge.Connect_TCP/IP</td>
<td>Device IP address, Device NO, Port NO</td>
<td>Connect to a reader using TCP/IP</td>
</tr>
<tr>
<td>BioBridge.Disconnect</td>
<td>No parameters required</td>
<td>Disconnect a connected device</td>
</tr>
<tr>
<td>BioBridge.EnableDevice</td>
<td>No parameters required</td>
<td>Enable reader operation</td>
</tr>
<tr>
<td>BioBridge.DisableDevice</td>
<td>No parameters required</td>
<td>Disable reader operation</td>
</tr>
<tr>
<td>BioBridge.SetDeviceTimeLocal</td>
<td>No parameters required</td>
<td>Set device date and time using PC local time</td>
</tr>
<tr>
<td>BioBridge.ReadGeneralLog</td>
<td>No of general transaction records</td>
<td>Read all general transaction logs into memory buffer to be retrieved by GetGeneralLog() function</td>
</tr>
<tr>
<td>BioBridge.GetGeneralLog</td>
<td>No parameters required</td>
<td>Get a general transaction log record from memory buffer from ReadGeneralLog()</td>
</tr>
<tr>
<td>BioBridge.DeleteGeneralLog</td>
<td>No parameters required</td>
<td>Delete all attendance logs</td>
</tr>
</tbody>
</table>

Table 3.2: Functions of BioBridge tool

3.3: ATTENDANCE TRACKING METHOD:

Attendance tracking method represents the base on which the design of interface and queries of database management system are built on. Students' attendance tracking vary from attendance tracking of personnel in companies where the main issue is to determine time of entering and leaving from the company. Students' attendance tracking is to be more complicated and restricted to ensure efficiency and reliability of the system. Following constraints were but in consideration for attendance tracking:

1- Attendance tracking is to be "course centric", that is, attendance to be considered is attendance taken for a specific course to which pre-determined group of students are involved. In such way, attendance received from a specific reader and which is assumed to belong to a specific level or group of students, if it contains real-time successful attendance log that belong to other students not included within the specified group should not appear in the interface showing attendance of the specific course. This leads to increased system efficiency and reliability, removes confusion and to be more convenient for operator responsible for issuing reports of attendance for each course.

2- There must be time constraints for attendance. Attendance taken for students that fingerprint after the end of session or even lately during the session should not included in the interface even if it to be included in database dedicated for downloaded data from reader.

3- For more accurate attendance, half of sessions should be determined very well, meaning determination of reader from which course attendance should be taken for a given time.

Based on the previous attendance tracking method, database queries have been written to satisfy it.
3.4: RECEIVED READER DATA FORMAT:
In addition to attendance tracking method, it was needed to specify the format of data received from the reader in order to construct the system database. As mentioned earlier, BioBridge software tool functions are used for bridging the reader to the system. To use them, an appropriate visual basic interface to be constructed firstly. Functions to be used data downloading were (BioBridge.ReadGeneralLog) and (BioBridge.GetGeneralLog). The connection with the reader generally requires specifying an IP address for it. This IP address could be static or dynamic and is to fall within the range of IP addresses used by the local network (this range was known from network operator). TCP/IP connection was carried out using (BioBridge.Connect_TCP/IP) function.
The data format received from the reader after successful identification was found to be as following:

	<USER_ID> <DATE> <TIME>

-USER_ID is a numeric unique user identifier.
-DATE is the date in YYYY/MM/DD format. For purpose of transferring to the database it is broken down into year, month and day each as single integer unit.
-TIME is the time when the event happened in HH:MM:SS format. For purpose of transferring to database it is broken down into hour, minute and seconds each as single integer unit.

Previous data are supplied originally by the reader. Other data that can be added through Visual basic interface includes device id. Device id is applied as TCP/IP connection function parameter and it is number assigned to each device connected to the network to distinguish their locations since this would be easier than distinguishing through IP addresses. From previous format it was clear that setting reader time and date to those of PC was necessary to give the actual timing of attendance and prevent confusion. That could be carried out through function (BioBridge.SetDeviceTimeLocal). Through specific code previous data are sent from Visual Basic environment and received by pre-specified PHP scripts. Next step was constructing database to which previous data to be transferred from reader.

3.5: DATABASE CONSTRUCTION:
Database (named fingerteck) construction has been started from previous points of format of data transferred from reader as well as attendance tracking method which is "course centric". Since format of data does not show any required information about course and at the same time each attendance event from the reader should be included within the database, a table that can be called "General table" or "plate table" was built. This table has columns that correspond to each unit of received data from reader including student id, year, month, day, hour, minute, second and device id. After that, and according to specific queries, attendance information is transferred
from this table to various tables each of them is dedicated for a specific subject. This state can be expressed by figure 3.2.

Bases on which attendance is transferred from general table to courses table are time and the identity of course itself. From this view, another table called "Lecturetimes" was constructed containing information about course (Course number), starting time, ending time (in hours and minutes), year, month, day, device id for place determination. The course is indicated using foreign key called "course_number", by which course can be determined from another table called "courses". Attendance data is transferred from the general table to courses tables when it is compatible with information contained in table of Lecturetimes for each record of course (matching performed using special queries. The database view development at this stage is expressed by figure 3.3.

![Figure 3.2: attendance data transfer from the reader to general table and then to courses tables](image)

Attendance state for each course is shown through its table. Initially courses tables include only the IDs of students involved in each course. By this way more reliability is achieved, because even if attendance data in general table for a student matches information included in lecture times table, it will not included in the course attendance table unless he is involved in the course. Courses attendance tables are updated and a new column is to be added each time attendance data has information about date identical to that of the course session in Lecturetimes table. Once matched a new column is added using query of ("alter table"-table name-"add lecture" n). The added column is called lecture n, where n is the number of the lecture. Number of lecture could
be obtained using query that counts number of columns in the table (mysql_num_rows), hence \( n \) is the number retrieved by this query (since numbering starts from 0 and first column is dedicated for students' IDs). For example, if no lectures have been taken before, the query would retrieve 1, so added column is named lecture1 which is correct.

Figure 3.3: transfer of data to courses tables according to lecturetimes table

Once the new column of lectures was added, it was set by default to "BOOLEAN FALSE" which is represented in the database by 0, where it is translated to "Absent" in the interface. At this moment all students involved in the specified course were set to "Absent" in the column of the new lecture. If attendance data from reader to general table contains time information (hours and minutes) within limitation of session time as well as same device ID and ID of a student involved in the course, the column of the new lecture next to that student would be set to "Attendant" in the interface and 1 in the database.

The previous operation can be summarized as following:
1- Attendance data from reader containing student ID, time and data information was to be transferred to general table containing the whole attendance.
2- Attendance date and time was to be compared to that assigned for a course session in lecturetimes table. Once matched, a new column indicating new lecture was added to the table of the course with all students to be initially set as Absents.
3- If the attendance data from general table contains time less time limit of the session and ID of involved student, his attendance would be set to "Attendant".

Additional tables are included for flexibility of database and to make interface dealing easier, such as: "courses" table that include fields showing course no, department and level in which it is studied and its name. "departments" table include two fields for the department number and department name. Another table which is "names" table just includes one field for student id and another for student's name (since the name has not been mentioned at all in the previous attendance processing operation). Table (3.3) contains description of database used above:
3.6: INTERFACE DESIGN:

Generally interface could be a user-graphic friend means for dealing with database. It allows displaying tables, retrieving required data and also modifying database tables without needing to be experienced in the database management system used for constructing the database. The interface designed for this system has three basic modules:

<table>
<thead>
<tr>
<th>Table</th>
<th>Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeneralTable</td>
<td>id, yr, mth, day, hr, min, sec, Device_ID</td>
</tr>
<tr>
<td>LectureTimes</td>
<td>Device_ID, Start_Hour, Start_Minute,</td>
</tr>
<tr>
<td></td>
<td>End_Hour, End_minute, yr, mth, day, Course_</td>
</tr>
<tr>
<td></td>
<td>Number</td>
</tr>
<tr>
<td>Courses</td>
<td>Course_Number, Department_Code, Level,</td>
</tr>
<tr>
<td></td>
<td>Course_Name</td>
</tr>
<tr>
<td>Departments</td>
<td>Dno, Dname</td>
</tr>
<tr>
<td>Names</td>
<td>Name, ID</td>
</tr>
<tr>
<td>Deviceip</td>
<td>IPAddress, Device_ID</td>
</tr>
<tr>
<td>Courses tables (table for each course)</td>
<td>Student_id, Lecture X (X is number of lecture)</td>
</tr>
</tbody>
</table>

Table 3.3: Database tables and their fields

1- First module is for students' registration: it allows operator to enter name and id for each student involved in the system. Later on name is to be added next to the id in each course attendance page (table). For ensuring non-repeated registration for a student the id field has been set as primary key. So, no two students with the same id can be registered as well as attempt to register a student more than once would result in error message indicating that registration could not take place due to duplication. On the other hand this module allows deleting of students from the system using their IDs. In addition all registered students can be shown using this module. This module affects the table "names".

2- Second module of the interface allows showing attendance pages of different courses. Courses were determined through department and level which were provided through forms (select type forms). If server is connected to the internet, any student can access this page remotely for attendance show.

3- Third module of this interface is used to assign students to the courses of their level and department. Form requires selecting the department, the level and inserting id of the student. The student then has to be assigned to all attendance pages of courses belong to selected department and level.
For any interface that handles data retrieved from database or modifies contents of it, connection script that communicates between database queries and data applied through html forms (php-MySQL interaction) should be included within php code [code9 in App.A]. This script contains database access information including: database user, database host, database password and database name. Database can be accessed without password but it is preferable to set for it for security issues.
CHAPTER 4: IMPLEMENTATION AND RESULTS:

4.1: Required Steps:
That stage contained the following steps:

a- **Fingerprints storage and reader matching test:**
1- First step in implementation was powering the fingerprint reader. The reader was powered through an adapter that converts AC power 220v/50 Hz into DC power of 12 volts.
2- After powering the reader fingerprint templates of different users could be taken. Before storing a specific fingerprint template in the reader, a distinguishing id relating to the user had to be accessed. For successful storing of the template, the finger print should be placed correctly on the scanner for three successive times. According to reader the user could be assigned as administrator or normal user. The advantage of this classification that when the reader was powered, no action could be done with it unless one of the users that were assigned as administrators put his fingerprint and the reader identified it. This property can give full control over the reader operation especially in the attendance system where lecturers can be assigned as administrator and students as normal users. In addition it provides reliability that the reader storage capacity and transaction log capability are to be exploited efficiently and the attendance of students is taken within the lecture time and in presence of the lecturer.

At that point, some fingerprints that relate to different students were stored and each student has been assigned a unique ID. Following table shows theses IDs:

<table>
<thead>
<tr>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>44018</td>
</tr>
<tr>
<td>44061</td>
</tr>
<tr>
<td>44050</td>
</tr>
<tr>
<td>44064</td>
</tr>
<tr>
<td>44027</td>
</tr>
<tr>
<td>44066</td>
</tr>
<tr>
<td>44016</td>
</tr>
</tbody>
</table>

Table 4.1: Students' assigned IDs
Next step was to test the reader matching capability by ensuring that it would identify each of students whose fingerprints were stored and not identify other students. Table 4.2 shows identification state for different students:

<table>
<thead>
<tr>
<th>ID</th>
<th>Identification state</th>
</tr>
</thead>
<tbody>
<tr>
<td>44018</td>
<td>Identified</td>
</tr>
<tr>
<td>44061</td>
<td>Identified</td>
</tr>
<tr>
<td>44020</td>
<td>Not identified</td>
</tr>
<tr>
<td>44050</td>
<td>Identified</td>
</tr>
<tr>
<td>44064</td>
<td>Identified</td>
</tr>
<tr>
<td>44048</td>
<td>Not identified</td>
</tr>
<tr>
<td>4404</td>
<td>Not identified</td>
</tr>
<tr>
<td>44016</td>
<td>Identified</td>
</tr>
<tr>
<td>44066</td>
<td>Identified</td>
</tr>
<tr>
<td>44027</td>
<td>Identified</td>
</tr>
</tbody>
</table>

Table 4.2: Reader identification for various students

From previous two tables, it was shown that reader had performed matching totally right.

b- Communication connection:
Next step was to establish communication connection between the fingerprint reader and the PC that holds the database and the interface. The connection facility used was Ethernet LAN that is available in networking lab in electrical engineering department and information technology network department. The communication type was TCP/IP connection. At that point the reader had to be configured for communication by assigning an IP address to it that fall within a specific range of addresses that the network holds. IP address held by the fingerprint reader was dynamic, meaning that it could change if the reader disconnected from the network and reconnected later on. IP address for different trials had the form of (192.16.14.X), where X was almost 90 or 91. The reader should be restarted when its IP address had been changed unless the former one would not change. To ensure that fingerprint reader had been connected correctly to the network, an order called (ping XXX) on DOS was used (XXX is the reader IP address). This order shows how much packets transmitted by the reader to the PC, how much received and how
much lost. The PC was connected to the LAN without importance to determining its IP address, so it was connected to the reader through the LAN.

c- Inserting data into Tables:
After that, and as explained in previous chapter, attendance data is transferred from general table to various courses tables according to time of attendance and id of student. The time constraints which determine acceptable time for attendance (session time) is located in "Lecturetimes" table. The following values were inserted into lecturetimes table through SQL interface (table 4.3):

<table>
<thead>
<tr>
<th>Dev_ID</th>
<th>St_Hr</th>
<th>St_Min</th>
<th>End_Hr</th>
<th>End_Min</th>
<th>Yr</th>
<th>Mth</th>
<th>Day</th>
<th>Course_NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>2009</td>
<td>5</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>0</td>
<td>10</td>
<td>0</td>
<td>2009</td>
<td>5</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>2009</td>
<td>5</td>
<td>20</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 4.3: Lecturetimes table

Before using interface to show attendance data for each course, "courses" table was filled with the following values shown in table 4.4:

<table>
<thead>
<tr>
<th>Course_NO</th>
<th>Department_Code</th>
<th>Level</th>
<th>Course_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>Artificial_intelligence</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>5</td>
<td>Java</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>5</td>
<td>Security</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>5</td>
<td>Engineering_management</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>5</td>
<td>Simulation</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>4</td>
<td>Telecommunication</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>4</td>
<td>Digital_signal_processing</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>3</td>
<td>Database management</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1</td>
<td>Calculus</td>
</tr>
</tbody>
</table>

Table 4.4: Courses table

Next table was departments table that assign each department a unique number. Following values were inserted as shown in table 4.5:
Next table to be filled was "names" table in which the names of student as well as their IDs were stored.

4.2: INTERFACING AND RESULTS:
As explained in previous chapter, attendance data was transferred from general table to courses table showing attendance state for each student involved in these courses. So, it is needed firstly to assign students to their courses. That was done through interface. Interface mainly had three modules as in figure 4.3:

![Figure 4.1: modules of system interface](image)

Students' registration module allows entering student's name and ID, deleting a student and viewing all students that have been registered ordered by their IDs. An attempt to register a
Figure 4.2: students' registration

student with an ID that belongs to previously registered one resulted in error message indicating duplication as in figure 4.5:

Figure 4.3: Registration error

Students that fingerprinted had been registered and they can be viewed through "view registered students" selection:
Second module was courses assigning through which students are assigned to courses according to department and level. For example, previous students were assigned to courses of fifth level in electrical department. By return to "ATTENDANCE SHOW" module, it was found that intended students have been assigned to all attendance pages belong to courses of fifth level in electrical department as shown by following figures:
At that point the attendance page included just name and id of students since no attendance had been taken till that moment.

**Data reception and Results obtaining:**

After communication had been established, the PC had to be ready to receive data from the reader. As had been shown in previous chapter, data from reader could be determined throughout BioBridge tool functions that work in Visual Basic environment. At that point data from reader was shown as list of statements. Each statement included the identified student id, the date and the time of identification (attendance). Only data about successful identification is received, meaning for example, that time of unsuccessful identification was not listed with data. So, for previous group of students, downloaded data list was as shown in figure 4.1 (with some modification by visual basic since data sent by reader contains only numbers).

At that time, and using some code in Visual Basic, above data was sent to "general table" in the database. To ensure that, database had been accessed. Its contents have been reflected through the table 4.2 that was done through PHP script.
At that time, and using some code in Visual Basic (codes are appended in appendix A), above data was sent to "general table" in the database. To ensure that, database had been accessed. Its contents have been reflected through the table 4.2 that was done through PHP script.

As shown from previous figure, attendance data was sent to the general table as record of single integers showing student's id, year, month, day, hour, minute, second and device id which was not sent by reader but added through Visual Basic interface. At that moment students' attendance could be obtained throughout the interface in various courses pages. For attendance provided latter in general table, only attendance page of artificial intelligence course was updated as shown in figure 4.3. Pages of other two courses that had been given dates identical to that of artificial intelligence (java and security) were as figure 4.4 and figure 4.5 show.
<table>
<thead>
<tr>
<th>Student_ID</th>
<th>Student Name</th>
<th>lecture1</th>
</tr>
</thead>
<tbody>
<tr>
<td>44016</td>
<td>Amjad</td>
<td>Attendant</td>
</tr>
<tr>
<td>44018</td>
<td>Ahmed Alssafi</td>
<td>Attendant</td>
</tr>
<tr>
<td>44027</td>
<td>Hamid</td>
<td>Attendant</td>
</tr>
<tr>
<td>44050</td>
<td>Basheer</td>
<td>Attendant</td>
</tr>
<tr>
<td>44061</td>
<td>MOHAMMMEED KAMAL</td>
<td>Attendant</td>
</tr>
<tr>
<td>44064</td>
<td>Musaab</td>
<td>Attendant</td>
</tr>
<tr>
<td>44066</td>
<td>Ali</td>
<td>Attendant</td>
</tr>
</tbody>
</table>

Figure 4.10: Artificial intelligence attendance page

<table>
<thead>
<tr>
<th>student_id</th>
<th>Student Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>44016</td>
<td>Amjad</td>
</tr>
<tr>
<td>44027</td>
<td>Hamid</td>
</tr>
<tr>
<td>44050</td>
<td>Basheer</td>
</tr>
<tr>
<td>44061</td>
<td>MOHAMMMEED KAMAL</td>
</tr>
<tr>
<td>44066</td>
<td>Ali</td>
</tr>
<tr>
<td>44064</td>
<td>Musaab</td>
</tr>
<tr>
<td>44018</td>
<td>Ahmed Alssafi</td>
</tr>
</tbody>
</table>

Figure 4.11: Java attendance page
Next, fields of "lecturetimes" table were modified as shown in Table 4.6:

<table>
<thead>
<tr>
<th>Devic_ID</th>
<th>St_Hr</th>
<th>St_Min</th>
<th>End_Hr</th>
<th>End_Min</th>
<th>Yr</th>
<th>Mth</th>
<th>day</th>
<th>Course_NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>13</td>
<td>0</td>
<td>14</td>
<td>0</td>
<td>2009</td>
<td>5</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>14</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>2009</td>
<td>5</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>2009</td>
<td>5</td>
<td>20</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 4.6: lecturetimes table after modification

Another attendance had been taken for previous students except for students of ids 44064 and 44050 at about 14:30. Another tour around artificial intelligence, java and security attendance tables has been made. Results were as shown in following figures:

Figure 4.12: Security attendance page

Figure 4.13: Artificial intelligence attendance page (2)
<table>
<thead>
<tr>
<th>student_id</th>
<th>Student Name</th>
<th>lecture1</th>
</tr>
</thead>
<tbody>
<tr>
<td>44016</td>
<td>Amjad</td>
<td>Attendant</td>
</tr>
<tr>
<td>44027</td>
<td>Hamid</td>
<td>Attendant</td>
</tr>
<tr>
<td>44050</td>
<td>Basheer</td>
<td>Absent</td>
</tr>
<tr>
<td>44061</td>
<td>MOHAMMED KAMAL</td>
<td>Attendant</td>
</tr>
<tr>
<td>44066</td>
<td>Ali</td>
<td>Attendant</td>
</tr>
<tr>
<td>44064</td>
<td>Musaab</td>
<td>Absent</td>
</tr>
<tr>
<td>44018</td>
<td>Ahmed Alssafi</td>
<td>Attendant</td>
</tr>
</tbody>
</table>

Figure 4.14: java attendance page (2)

<table>
<thead>
<tr>
<th>student_id</th>
<th>Student Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>44016</td>
<td>Amjad</td>
</tr>
<tr>
<td>44027</td>
<td>Hamid</td>
</tr>
<tr>
<td>44050</td>
<td>Basheer</td>
</tr>
<tr>
<td>44061</td>
<td>MOHAMMED KAMAL</td>
</tr>
<tr>
<td>44066</td>
<td>Ali</td>
</tr>
<tr>
<td>44064</td>
<td>Musaab</td>
</tr>
<tr>
<td>44018</td>
<td>Ahmed Alssafi</td>
</tr>
</tbody>
</table>

Figure 4.15: security attendance page (2)
Next, "lecturetimes" table had been modified as shown in table 4.7:

<table>
<thead>
<tr>
<th>Devic_ID</th>
<th>St_Hr</th>
<th>St_Min</th>
<th>End_Hr</th>
<th>End_Min</th>
<th>Yr</th>
<th>Mth</th>
<th>day</th>
<th>Course_NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>14</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>2009</td>
<td>5</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>12</td>
<td>0</td>
<td>14</td>
<td>0</td>
<td>2009</td>
<td>5</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>2009</td>
<td>5</td>
<td>20</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 4.7: lecturetimes table after modification (2)

Attendance had been taken for the same previous group of students at about 14:45. Device ID was changed (configuration done with device) and device was restarted. Also device id was changed in Visual Basic interface to be 2. Another tour around the previous attendance pages gave results shown in following figures:

![Artificial intelligence attendance page](image1.png)

Figure 4.16: Artificial intelligence attendance page (3)

![java attendance page](image2.png)

Figure 4.17: java attendance page (3)
4.3: Results Discussion:

1- Results have shown that interface handles applied data to it and retrieves required data successfully. Applied information about students (names and ids) required for registration in the system or deleting, request of listing registered students, assigning them to specific courses and request of showing attendance state for particular course all were handled correctly by the interface. That indicated the validity of PHP scripts dedicated for various functions, validity of written queries and correctness of connection and interaction between MySQL database and PHP.

2- Table of artificial intelligence course has been updated alone when its time has been assigned between 13:00 and 15:00 because sent data time falls within its time limits as well as they were assigned a unique device id (compatibility in time and place). Others tables that sent data time fell out of their time limits were not modified, meaning no new lecture in the course has been taken yet.

3- Previous point is to be said also in the second case where java table was updated only. It could be noticed that involved students in java course that have not identified by the fingerprint reader were set to state "Absent", meaning that it was the default state for any added new lecture column.
4- In the third case, just table security has been altered although sent data time fell within time limits of both security and artificial intelligence. That was because security was assigned to receive attendance data from device 2, while artificial intelligence was assigned to device 1 and sent data marked to be sent by device 2 (marked through Visual Basic interface).

5- It can be shown finally through previous results that information included in "Lecturetimes" table controls flow of attendance data from general table to courses' attendance tables when there is "time and place compatibility" between specific course session information and sent data.
CHAPTER 5: PROJECT REVIEW:

5.1: Results and Objectives Compatibility:
Objectives that have been intended to be achieved throughout implementation of this project were summarized in providing a constrained and reliable students' attendance tracking system and a friendly graphic user interface allowing handling and retrieval of system's information in clear and convenient way. Throughout results obtained in previous chapter, it can be said that attendance tracking method that represents the base for code design of the system, allows for efficient and reliable attendance tracking, since it puts in consideration various possibilities that could be encountered throughout implementation of the system. Results were satisfactory and indicated both efficiency of applied attendance tracking method as well as efficiency of the used fingerprint reader and generally fingerprint matching technology. Provided interface has been used efficiently and successful results had been obtained through it and that could be ensured by referencing SQL database interface.

5.2: Problems encountered:
1- Successful connecting of the fingerprint reader to a network was always not performed immediately during implementation. It was needed to track IP addresses starting from specific one that falls in the network dedicated range of addresses until getting one through which communication with network, and hence with the installed attendance system, could take place. Once disconnected for some reason, a communication difficulty might be encountered for successful reconnection. This may take place because IP addresses are assigned dynamically to the devices connected to the network, that is, now device in the network was assigned unique IP address and it may unable to regain it if disconnected from the network. So, to ensure consistent attendance system it is preferable to assign fingerprint readers in the network statistic IP addresses. Another reason that may result in such a problem was not well-perfection of the network communication links.

2- Another problem encountered was transfer of attendance data from Visual Basic environment to MySQL-PHP environment to be displayed throughout the interface of the system. This problem was solved by splitting attendance data including student's ID, date and time and assigning them integer variables, then these variables could be directed to a specific PHP script by which data could be inserted into SQL environment and hence handled easily among PHP scripts and displayed by the interface.
5.3: Possible future development:
Several features over this system could be considered for future development:
1- "Lecturetimes" table in the database through which specific course session can be determined for attendance at any time, was constructed so that date information (year, month, day) is used for this determination. This means that each session date throughout the semester should be pre-inserted into "Lecturetimes table", which leads to sizable table as well as inconvenient required work. This can be developed so that determination is done through day of week sent along with attendance data from Visual Basic interface, leading to table of acceptable size. On the other hand this requires updating weekday information daily throughout Visual Basic interface.

2 – In the interface, there are several developments that can be done. In this project, some information that considered to be almost static throughout a semester or even throughout years such lecture times and courses and departments’ codes were inserted directly though SQL interface. It could be more convenient for inexperienced in SQL system operator to handle this information throughout the interface.

3- More security may be supplied to interface by ensuring that inserted information will be inserted through authorized operator. This can be done through password login.
APPENDIX A

1- Following code has been used to allow connection between fingerprint reader and Visual Basic interface:

Private Sub cmdConnect_Click()
If bConnected Then
    FingerTec.Disconnect
    bConnected = False
    cmdConnect.Caption = "Connect"
Else
    Dim devModel, i, ipAdd, portNo, comm

    devModel = "AC100"
    i = 1
    ipAdd = txtIP.Text
    portNo = 4370
    comm = 0
    If FingerTec.Connect_TCPIP(devModel, i, ipAdd, portNo, comm) = 0 Then
        MsgBox ("Connected")
        cmdConnect.Caption = "Disconnected"
        bConnected = True
    Else
        MsgBox ("Connection Failed")
    End If
End If
End Sub

2- Following code has been used to read and download data from reader to Visual Basic interface and later to PHP scripts:

Dim log As Long
If FingerTec.ReadGeneralLog(log) = 0 Then
Dim enrollNo As Long
Dim yr As Long
Dim mth As Long
Dim day As Long
Dim hr As Long
Dim min As Long
Dim sec As Long
Dim ver As Long
Dim io As Long
Dim work As Long
Dim PRM As String
Dim Query As String
Do While FingerTec.GetGeneralLog(enrollNo, yr, mth, day, hr, min, sec, ver, io, work) = 0
PRM = "&enrollno=" & enrollNo & "&yr=" & yr & "&mth=" & mth & "&day=" & day & 
"&hr=" & hr & "&min=" & min & "&sec=" & sec & "&did=1
Randomize
Query = "http://localhost/dbquery.php?rnd=" & Rnd(10) & PRM
Query1 = "http://localhost/process.php?rnd=" & Rnd(10) & PRM
Me.Caption = Query
Me1.Caption = Query1
"DownloadFile Query, "c:\allapi.txt
List1.AddItem ("No: " + Str(enrollNo) + " Date:" + Str(day) + "/" + Str(mth) + "/" + Str(yr)
((+ " Time: " + Str(hr) + ":" + Str(min) + ":" + Str(sec
Loop
Else
("MsgBox ("Failed
End If
End Sub

3- Following code has been used for data sent to be stored in the general table and displayed:

```php
<?php
echo $_GET['rnd'];
include("dbconnect.php");
$id=$_GET['enrollno'];
$yr=$_GET['yr'];
$mth=$_GET['mth'];
$day=$_GET['day'];
$hr=$_GET['hr'];
$min=$_GET['min'];
$sec=$_GET['sec'];
$did=$_GET['did'];

$line = "insert into table1 values ($id,$yr,$mth,$day,$hr,$min,$sec,$did)";
echo "query:".$line."n";
mysql_query($line) or die("error: ". mysql_error());
?>
```
4- Following code has been used to process attendance data and update courses' attendance tables accordingly:

```php
<?php
include( "dbconnect.php" );
echo $GET["rnd"];  
    include("dbconnect.php");
    $id=$_GET["enrollno"];  
    $yr=$_GET["yr"];  
    $mth=$_GET["mth"];  
    $day=$_GET["day"];  
    $hr=$_GET["hr"];  
    $min=$_GET["min"];  
    $sec=$_GET["sec"];  
    $did=$_GET["did"];  

$result = mysql_query( "SELECT course_number FROM LectureTimes WHERE yr = " . $yr . " and mth = " . $mth . " and day = " . $day . " and St_Hr*60 + St_Min <= " . ($hr*60 + $min) . " and End_Hr*60 + End_Min >= " . ($hr*60 + $min) . " and Device_ID =" . $did);    if ( !$result )
    {
        echo 'Invalid query: ' . mysql_error();
    }
    $row1 = mysql_fetch_row($result);
    $course_number = $row1[0];

$result2 = mysql_query( "SELECT Course_Name FROM Courses WHERE Course_Number = " . $course_number );
    if ( !$result2 )
    {
        echo 'Invalid query: ' . mysql_error();
    }
    $row2 = mysql_fetch_row( $result2 );
    $course_name = $row2[0];

$result3 = mysql_query( "show columns from " . $course_name );
    if ( !$result3 )
    {
        echo 'Invalid query: ' . mysql_error();
    }
```
$row3 = mysql_num_rows($result3);
$result4 = mysql_query("alter table ". $course_name . " add LECTURE" . $row3 . " BOOLEAN DEFAULT FALSE");
if ( !$result4 )
{
    echo 'Invalid query: ' . mysql_error();
}

$result5 = mysql_query("update ". $course_name . " set LECTURE" . $row3 . " = true where Student_ID in = " . $id);

$result6 = mysql_query("delete from table1");
if ( !$result6 )
{
    echo 'Invalid query: ' . mysql_error();
}
?>

5- Following code has been used for displaying interface page containing the basic three modules:

<html>
<head>
<title>UOFK ATTENDANCE SYSTEM</title>
</head>
<body>
<h1><p style="background-color: blue; color: white" align ="center">
UNIVERSITY OF KHARTOUM ATTENDANCE SYSTEM</p></h1>
<a href="register_students.php"><h3><p style="background-color: white; color: green" align ="left">
STUDENTS' REGISTRATION</p></h3></a>
<a href="form1.php"><h3><p style="background-color: white; color: green" align ="left">ATTENDANCE SHOW</p></h3></a>
<a href="sub_assigning.php"><h3><p style="background-color: white; color: green" align ="left">COURSES ASSAIGNING</p></h3></a>
</body>
</html>
<?php  include( "dbconnect.php" ); ?>
<?php
$page_title = 'Registration Checking';
include ('includes/header_students.html');
if (isset($_GET['submitted'])) {
    $errors = array(); // Initialize an error array.
    if (empty($_GET['name'])) {
        $errors[] = 'You forgot to enter name.';
    } else {
        $n = ($_GET['name']);
    }
    if (empty($_GET['id'])) {
        $errors[] = 'You forgot to enter ID.';
    } else {
        $i = ($_GET['id']);
    } if (empty($errors)) {
        // Connect to the db.
        // Make the query:
        $q = mysql_query("INSERT INTO names VALUES('$n', '$i')");
        if ($q) {
            // If it ran OK.
            // Print a message:
            echo "<p><h1>$n</h1>is now registered.</p>";
        } else {
            // If it did not run OK.
            // Public message:
            echo '<h2>REGISTRATION ERROR!</h2><p class="error">Registration could not take place due to duplication.Please check again.</p>";
        }
    } else {
        // Report the errors.
        echo '<h1>Error!</h1><p class="error">The following error(s) occurred:<br />
";}
foreach ($errors as $msg) {
    // Print each error.
    echo " - $msg<br />
    ";
}
    echo '<p>Please check again.</p>';
} // End of if (empty($errors)) IF.
} // End of the main Submit conditional.

<form action="register_students.php" method="GET">
    <fieldset>
        <p><b>NAME:</b> <input type="text" name="name" size="25" maxlength="40" value="<?php if (isset($_GET['name'])) echo $_GET['name']; ?>" /></p>
        <p><b>ID:</b> <input type="text" name="id" value="<?php if (isset($_GET['id'])) echo $_GET['id']; ?>" /></p>
    </fieldset>
    <p><input type="submit" name="submit" value="Register" /></p>
    <input type="hidden" name="submitted" value="TRUE" />
</form>
</html>

7- Following code has been used for deleting students:

<?php  include( "dbconnect.php" ); ?>
<?php
    $page_title = 'DELETE STUDENT';
    include ('includes/header_students.html');
    // This page lets administrator to update individuals'data
    // Check if the form has been submitted:
    if (isset($_POST['submitted'])) {
        // Connect to the db.
        if (empty($_POST['id'])) {
            echo "PLEASE INSERT ID";
        } else { /* Code for deleting student */
    } // End of if (isset($_POST['submitted']))
$i = ($_POST['id']);
// If everything’s OK.
// Check validity of ID
$q = mysql_query("SELECT id FROM names WHERE (id='$i')");
$num = mysql_num_rows ($q);
if ($num == 1){
$q = mysql_query ("DELETE FROM names WHERE id=$i");
// If it ran OK.
// Print a message.
echo '<h2>DELETE HAS BEEN PERFORMED SUCCESSFULLY!</h2>
<p>You may ensure by showing registered students list</p>);
} else {
// Invalid input ID.
echo '<h1>Error!</h1>
<p class="error">Inserted ID is not available.</p>';
}

$q1 = mysql_query ("select * from courses");
while ($row = mysql_fetch_row($q1))
{
$course_name = $row[3];
$q2 = mysql_query ("select * from" . $row[3]);
$q3 = mysql_query ("delete from" . $row[3] . "where student_ID = $i");
}

<h1><b>DELETE</b></h1>
<form action="delete_students.php" method="post">
<fieldset><legend>Enter ID to specify deleted individual:</legend>
<p><b>ID:</b> <input type="text" name="id" size="15" maxlength="20" value="<?php if (isset($_POST['id'])) echo $_POST['id']; ?>" /></p>
</fieldset>
<p><input type="submit" name="submit" value="DELETE" /></p>
<input type="hidden" name="submitted"} else {
}
8- Following code has been used to view registered students:

```
<?php include( "dbconnect.php" ); ?>

<?php
# Script 8.4 - view_registered_individuals.php
// This script retrieves all the records from the users table.
$page_title = 'View the Current registered individuals';
include ('includes/header_students.html');
// Page header:
// Make the query:
$q = mysql_query("SELECT (name) AS NAME, (id) AS ID FROM names ORDER BY id ASC");
if ($q) {
// If it ran OK, display the records.
// Print how many users there are:
// Table header.
echo '<table align="center" border="3"
cellspacing="3">
<tr><td align="left"><b>NAME</b></td><td align="left"><b>ID</b></td></tr>

// Fetch and print all the records:
while ($row = mysql_fetch_array($q)) {
    echo '<tr><td align="left">'. $row['NAME'] . '</td><td align="left">'. $row['ID'] . '</td></tr>';
}
echo '</table>'; // Close the table.
}
else {
// If it did not run OK.
// Public message:
echo '<p align="center" class="error">'.NO STUDENT IS REGISTERED CURRENTLY.'.</p>';
} // End of if ($r) IF.
?>
```

9- Following code has been used to assign students to specific courses:

```
<?php include ("dbconnect.php"); ?>

<html>
```
<?php
if (isset($_GET['submitted'])) {
    if (empty($_GET['id'])) {
        echo "PLEASE INSERT STUDENT'S ID";
    } else {
        $i = ($_GET['id']);
    }
    $l = ($_GET['level']);
    $d = ($_GET['department']);
    $result = mysql_query("SELECT course_name FROM COURSES WHERE department_code = "$d and level = "$l");
    if ( !$result ) {
        echo 'Invalid query: ' . mysql_error();
    } else {
        $course_name = $row[0];
        $result2 = mysql_query('insert into ' . $course_name . ' values ('.$i.')');
        if ( !$result2 ) {
            echo 'Invalid query: ' . mysql_error();
        } else {
            echo "COURSES ASSIGNMENT HAS BEEN PERFORMED SUCCESSFULLY. YOU CAN GO TO COURSES PAGES FOR ENSURING";
        }
    }
} else {
    echo '
<form action = "sub_assigning.php" method="GET">
<fieldset><legend>PLEASE insert ID and select DEPARTMENT and LEVEL:</legend>
<p><b>Department:</b><br>

</p></fieldset>
</form>

<?php
$result = mysql_query( "select * from departments" );
if ( !$result ) {
    echo 'Invalid query: ' . mysql_error();
}
// echo "<option value="0" SELECTED>Select Department";
while ( $row = mysql_fetch_row( $result ) ) {
    echo "<option value="" . $row[0] . "">" . $row[1] . "". $row[1];
}
?>
</select></p>
<p><b>Level:</b><br>
<select name="level">
<?php /* <option value="0" SELECTED>Select Level */ ?></select></p>
<p><b>ID:</b> <input type="text" name="id" value="<?php if (isset($_GET['id'])) echo $_GET['id']; ?>" /></p>
</fieldset>
<p><input align="center" type="submit" name="submit" value="assign courses of selected level and department" /></p>
<input type="hidden" name="submitted" value="TRUE" />
</form>
</body>
</html>

10- Following code has been used to select course to display its attendance page:
<?php include( "dbconnect.php" ); ?>
<html>
<head>
<title>Form1</title>
</head>
<title>Form1</title>
UNIVERSITY OF KHARTOUM ATTENDANCE SYSTEM

```
<?php
$result = mysql_query( "select * from departments" );
if ( !$result )
{
    echo 'Invalid query: ' . mysql_error();
}
// echo "<option value="0" SELECTED>Select Department";
while ( $row = mysql_fetch_row( $result ) )
{
    echo "<option value="" . $row[0] . "">" . $row[1];
}

/* <option value="0" SELECTED>Select Level */ ?>
<option value="1">Level 1
<option value="2">Level 2
<option value="3">Level 3
<option value="4">Level 4
<option value="5">Level 5
</select></p>
</fieldset>
<br>
<input align = "center" type="submit" value="SELECT SUBJECT">
</form>
</body>
</html>
```

11- Following code has been used to show attendance page of the course selected previously:

```
<html>

<body>

<h1><p style="background-color: blue; color: white" align ="center">

</p></h1>
<form action = "courses.php" method="GET">
<fieldset><legend>PLEASE SELECT REQUIRED INFORMATON BELOW:</legend>
<p><b>Department:</b><br>
<select name="department">
<?php
$result = mysql_query( "select * from departments" );
if ( !$result )
{
    echo 'Invalid query: ' . mysql_error();
}
// echo "<option value="0" SELECTED>Select Department";
while ( $row = mysql_fetch_row( $result ) )
{
    echo "<option value="" . $row[0] . "">" . $row[1];
}

/* <option value="0" SELECTED>Select Level */ ?>
<option value="1">Level 1
<option value="2">Level 2
<option value="3">Level 3
<option value="4">Level 4
<option value="5">Level 5
</select></p>
<br>
<input align = "center" type="submit" value="SELECT SUBJECT">
</fieldset>
</form>
</body>
</html>
```
<?php
foreach( $_GET as $key => $value)
{
    ${$key} = $value;
}
include ( "dbconnect.php" );
?>

<html>
<head>
<title>Attendance Sheet for <?php echo $course; ?></title>
</head>
<body>
<?php
$result = mysql_query( "show columns from $course " );
if ( !$result )
{
    echo 'Invalid query: ' . mysql_error();
}

echo "<table border = 5>";
echo "<tr bgcolor = yellow>";
$row = mysql_fetch_row( $result );
echo "<td><b>$row[0]</b></td>";
echo "<td><b>Student Name</b></td>";
while ( $row = mysql_fetch_row( $result ) )
{
    echo "<tr><b>$row[0]</b></td>";
}
}
</tr>
</table>
$result = mysql_query( "select * from $course" );
if ( !$result )
{
    echo 'Invalid query: ' . mysql_error();
}

while ( $row = mysql_fetch_row( $result ) )
{
    echo "<tr>";
```
12- Finally, following code has been used to connect MySQL database (named fingerteck) and PHP scripts:

```
REFERENCES:
[7]: Elizabeth Naramore, Jason Gerner, Jeremy stolz, . Beginning PHP5, Apache, MySQL Web Development, page No. 6