

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018**



**A PROJECT REPORT
ON
MENTOR MENTEE MANAGEMENT SYSTEM
BY**

ANANYA P
4SF20CI010

KARTHIK SK
4SF20CI028

In the partial fulfillment of the requirement for V Sem. B. E.
Computer Science & Engineering (Artificial Intelligence and Machine Learning)

DBMS LABORATORY WITH MINI PROJECT

Under the guidance of

Ms. Vaishnavi V Rao

Asst. Professor, Dept. of Computer Science & Engineering
(Artificial Intelligence and Machine Learning)



**Department of Computer Science & Engineering
(Artificial Intelligence and Machine Learning)**

**SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution
Adyar, Mangaluru-575007
2022-23**

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
An Autonomous Institution
Adyar, Mangaluru – 575007

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

CERTIFICATE

This is to certify that the project entitled “Mentor Mentee Management System” is submitted in partial fulfillment for the requirement of V sem .B. E. Computer Science & Engineering(Artificial Intelligence and Machine Learning), “**DBMS LABORATORY WITH MINI PROJECT**” during the year 2022 – 23 is a result of bonafide work carried out by

ANANYA P
4SF20CI010

KARTHIK SK
4SF20CI028

.....
MS VAISHNAVI V RAO
Asst. Prof.
Dept. of CSE(AI&ML)
SCEM, Mangaluru

.....
DR. PUSHPALATHA K
Professor & Head
Dept. of CSE(AI&ML)
SCEM, Mangaluru

Signature of the Examiners

1.

2.

ABSTRACT

A database is an organized collection of data, generally stored and accessed electronically from a computer system. The database management system (DBMS) is a software that interacts with end user, applications and the database itself to capture and analyze the data and provides facilities to administer the database. Primary goal of database management system is to provide a way to store and retrieve database information that is convenient and efficient.

“Mentor Mentee Management system” is aimed at developing an application for mentor mentee management system and is an management system software project that serves the functionality of an mentor mentee relation. The system allows registered mentor, mentee login and are allowed to register too this system helps in the management of mentee details that are required through a java application. mentor needs to login at the initial phase, fill in their mentees profile details including usn, student name, semester, section and contact details. Mentor can modify student profile at any stage. All the data is logged in the database. Mentee too can login and check information to verify and also update to let the mentor know about their whereabouts.

This system can be utilized in any peer detail management system to have better coordination between mentor and students. The data of the students once added in the system can benefit in identifying achievement, growth of students. This is a generic system which can be applied to other peer detailmanagements with slight modifications and it reduces the redundancy and confusions about the data for monitoring mentor mentee relationship.

It is implemented using MySQL 8.0.30 and Java using the IDE Apache NetBeans 16.

ACKNOWLEDGEMENT

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on “**Mentor Mentee Management System**”. We have completed it as a part of the curriculum of Vishveshvaraya Technological University, Belagavi for the award of Bachelor of Engineering in Computer Science and Engineering (Artificial Intelligence and Machine Learning)

We are profoundly indebted to our guide, **Ms. Vaishnavi V Rao**, Assistant Professor, Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning) for innumerable acts of timely advice and encouragement. We sincerely express our gratitude.

We extend our warm gratitude to **Dr. Pushpalatha K.** Head of Department, , Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), for her constant support and advice that helped us to complete this project successfully.

We are extremely grateful to our beloved Principle, **Dr. Rajesha S** for encouraging us to come up with new ideas and express them in a systematic manner.

We also like to thank all Teaching and Non-teaching staff of Sahyadri College of Engineering and Management, Mangalore for their kind co-operation during the course of my work.

Finally, yet importantly, we express our heartfelt thanks to our family and friends for their wishes and encouragement throughout the work

ANANYA P
4SF20CI010
V Sem, B.E.,
CSE(AI&ML)
SCEM, Mangaluru

KARTHIK SK
4SF20CI028
V Sem, B.E.,
CSE(AI&ML)
SCEM, Mangaluru

INDEX

1. Introduction

1.1. Introduction to Database Management System.	1
1.2. Background of the Project.	1
1.3. Necessity of the Project.	2
1.4. Applications and Advantages.	2
1.5. Oracle.....	2
1.6. My SQL	3
1.7. JAVA.	3
1.8. Net Beans.....	4

2. Requirement Specifications

2.1. Hardware requirements.....	5
2.2. Software requirements.	5

3. Design

3.1. ER Diagram.	6
3.2. Relational Schema.	7
3.3. Schema Diagram.....	11

4. Normalization.....12

5. Implementation

5.1. Creation of tables	15
5.2. Description of tables	17
5.3. Jdbc driver for connecting Java and MySQL	20
5.4. Insert/update/delete used as.....	20
5.5. TRIGGERS AND STORING PROCEDURE	
5.5.1. Trigger Implementation in Java	23
5.5.2. Stored Procedure Implementation in Java	23

6. Results..... 24

7. Conclusion 31

8. Reference 32

List of figures:

Figure Number	Figure Name	Page Number
3.1.1	ER Syntax	6
3.1.2	ER Diagram of Mentor Mentee Management System	7
3.2.1	Mapping of regular entity type	8
3.2.2	Mapping of 1:N relationship types	9
3.2.3	Mapping of M:N relationship types	10
3.3.1	Schema Diagram	11
6.1	Mentor Mentee Login Page	24
6.2	Main page	24
6.3	Mentee Insert Page	25
6.4	Update Page	25
6.5	Delete Page	26
6.6	Mentor main Page	26
6.7	Mentee Main Page	27
6.8	Mentee attendance and marks update Page	27
6.9	Mentee achievement Page	28
6.10	Department Login Page	28
6.11	Mentee Regsitration Page	29
6.12	Mentor Registration Page	29
6.13	Subject Detail page	30
6.14	Trigger Successful Page	30

CHAPTER 1

1. INTRODUCTION

1.1. INTRODUCTION TO DBMS

DBMS Stands for "Database Management System." In short, a DBMS is a database program. Technically speaking, it is a software system that uses a standard method of cataloging, retrieving, and running queries on data. The DBMS manages incoming data, organizes it, and provides ways for the data to be modified or extracted by users or other programs.

A DBMS makes it possible for end users to create, read, update and delete data in a database. The DBMS essentially serves as an interface between the database and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

DBMS include change management, performance monitoring/tuning and backup and recovery. Many database management systems are also responsible for automated rollbacks, restarts and recovery as well as the logging and auditing of activity. The DBMS can offer both logical and physical data independence. That means it can protect users and applications from needing to know where data is stored or having to be concerned about changes to the physical structure of data (storage and hardware).

As long as programs use the application programming interface (API) for the database that is provided by the DBMS, developers won't have to modify programs just because changes have been made to the database. With relational DBMSs (RDBMSs), this API is SQL, a standard programming.

1.2. BACKGROUND OF THE PROJECT

The Mentor Mentee Management System has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and, in some cases, reduce the hardships faced by this existing system. Moreover, this system is designed for the particular need of the mentor to carry out operations in a smooth and effective manner.

1.3. NECESSITY OF THE PROJECT

Every mentor has more than one mentee to look after simultaneously and it is difficult to manage monitoring the data of each one. It is also important for mentors to recognize the growth of mentees to provide the right guidance, even though this is a tedious task we noticed that our college maintains records manually hence we felt the necessity to create this project to ease the struggles of mentors and data maintenance.

1.4. APPLICATION AND ADVANTAGES

Management System of data has different needs, so we designed mentor mentee management system. This is designed to assist in peer detail maintenance and it will help ensure that mentors are equipped with the right level of information and details of their mentees so that they can focus on providing right guidance and not on missing or spurious data. Mentors need not look for numerous records to find information of a mere student. This system will ultimately allow you to better manage resources and enhance mentor mentee relation over all.

1.5. ORACLE

Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is a multi-model database management system produced and marketed by Oracle Corporation. It is a database commonly used for running online transaction processing (OLTP), data warehousing (DW), etc. Larry Ellison and his two friends and former co-workers, Bob Miner and Ed Oates started a consultancy called Software Development and Laboratories (SDL) in 1977. SDL developed the original version of Oracle. Oracle Database is available by several service providers on-premises, on-cloud, or as hybrid cloud installation. It may be run on third party servers as well as on Oracle hardware.

1.6. MYSQL

MYSQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation. A relational database organizes data into one or more data tables in which data types may be related to each other and these relationships help structure the data. SQL is a language programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups. MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often, MySQL is used with other programs to implement applications that need relational database capability.

1.7. JAVA

Java is a popular programming language. It is owned by Oracle, and more devices run Java. Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.). It is open-source secure, fast, free and powerful. The syntax of Java is similar to C and C++, but has fewer low level facilities than either of them. Java was developed in the mid-1990s by James A. Gosling, a former computer scientist with Sun Microsystems. Java is high level, class based object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general purpose programming language intended to let programmers write once, run anywhere (WORA), meaning that compiled Java code can run on all platforms that support Java without the need to recompile. Java applications are typically compiled to bytecode that can run on Java Virtual Machine (JVM) regardless of the underlying computer architecture.

1.8. NETBEANS

NetBeans is an integrated development environment (IDE) for Java. NetBeans allows applications to be developed from a set of modular software components called modules. The NetBeans Platform is a framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications. The underlying NetBeans platform supports creation of new applications and further development of existing applications using modular software components. As an application running on the NetBeans Platform, the NetBeans IDE itself is extensible and can be extended to support new languages.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Hardware Requirements

Operating System: All Windows OS/Linux/MacOS

Memory: Minimum of 1GB of RAM, Minimum of 128GB hard disk space

2.2 Software Requirements

Backend: MySQL Workbench 8.0 CE

IDE: NetBeans IDE 16

Programming Language: Java

CHAPTER 3

3.1 ER-diagram

An entity–relationship model describes interrelated things of interest in a specific domain knowledge. The ER Diagram of our project is shown in the figure:3.1.1

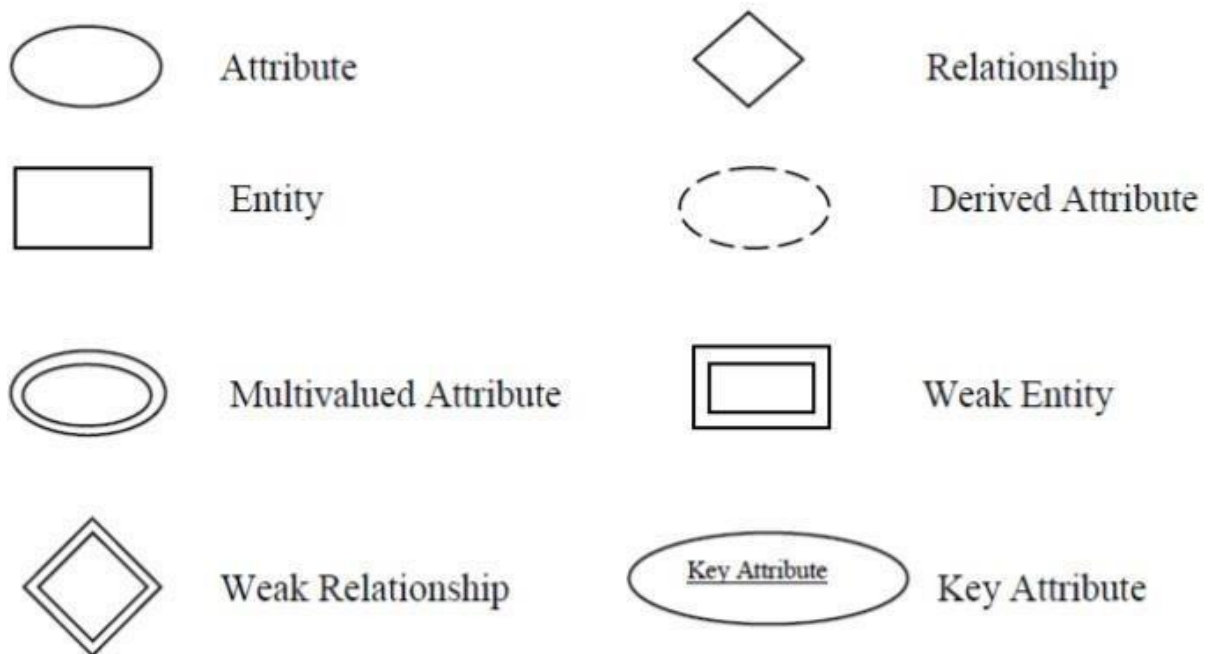


Figure 3.1.1 ER Notation

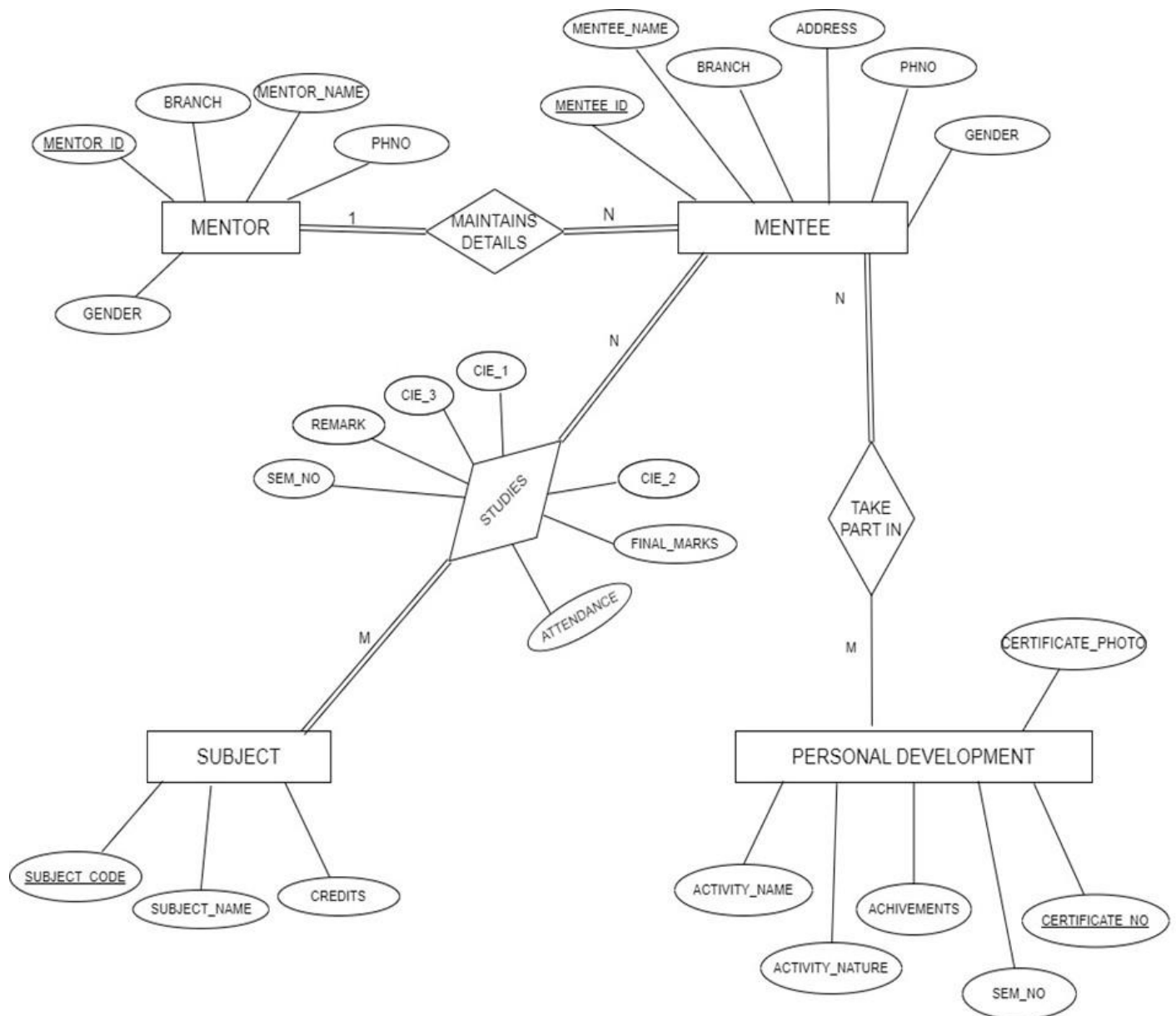


Figure 3.1.2 ER Diagram of Mentor Mentee Management System

3.2. RELATIONAL SCHEMA

Mapping from ER Diagram to Schema Diagram

1. Mapping of regular entities: -This step involves mapping all the regular entity types totabular format by identifying their primary keys.
2. Mapping of 1:1 Relation: -In this step foreign keys are assigned using foreign key approach.The primary key of the participating relation R or S is added as primary key to second entity types by looking at the participating constraints.
3. Mapping of 1: N Relation: -Foreign key approach is used to add one sided primary key to then sided entity at foreign key.
4. Mapping of M: N Relation:-Here we use the cross-reference approach where the relationshipis converted to a new relation within attributes on primary keys of both participating relation.
5. Mapping of Weak Entity: -When mapping weak entity types along with other attributes thepartial key and primary key of parent entity together will form their primary key of the new relation.
6. Mapping of N-ary Relation: -For mapping N array relationship we create a new relation witha relationship name in its attribute and primary keys of all participating entity types.
7. Mapping of Multivalued Relation: -For multivalued attributes a separate relation has to becreated along with primary key of parent relation. A relational schema for a database is an outline of how data is organized.

STEP 1: Mapping of regular entity types.

The regular entity types of our project are shown in figure

<u>Mentor_id</u>	Mentor_name	Branch	Gender	Phone No
MENTOR				

<u>Mentee_id</u>	Mentee_name	Mentor_id	Branch	Address	Phone number	Gender	Sem_no
MENTEE							

<u>Sub_code</u>	Subject_name	Credits
SUBJECT		

<u>Certificate_no</u>	Activity_name	Activity_nature	Achievements	Sem_no	Certificate_photo
PERSONAL DEVELOPMENT					

Figure 3.2.1 Mapping of regular entity type

STEP 2: Mapping of weak entity types.

The ERD of our project doesn't contain weak entity

STEP 3: Mapping of binary 1:1 relation types.

The ERD of our project does not contain any 1:1 relation type.

STEP 4: Mapping of 1: N relation types

For every 1: N relation types identify the entity which is in the N-side. Make primary key of entity which is participating in 1 side as foreign key of entity which is N-side. If there are any attributes for the relationship add to the N-side.

The ERD of our project 1: N relation type is shown below

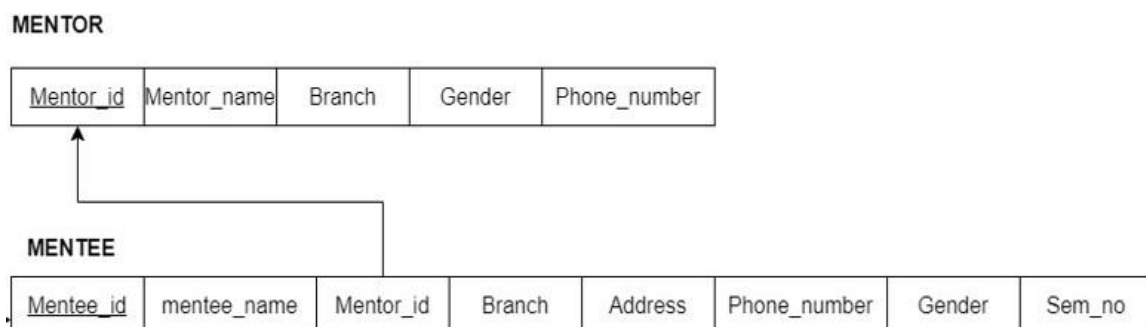


Figure 3.2.2 Mapping of 1:N relationship types

STEP 5: Mapping of M: N relation types

In every M: N relation types create a new relation and add the primary keys of both the entities participating in the relation as a foreign key relation.

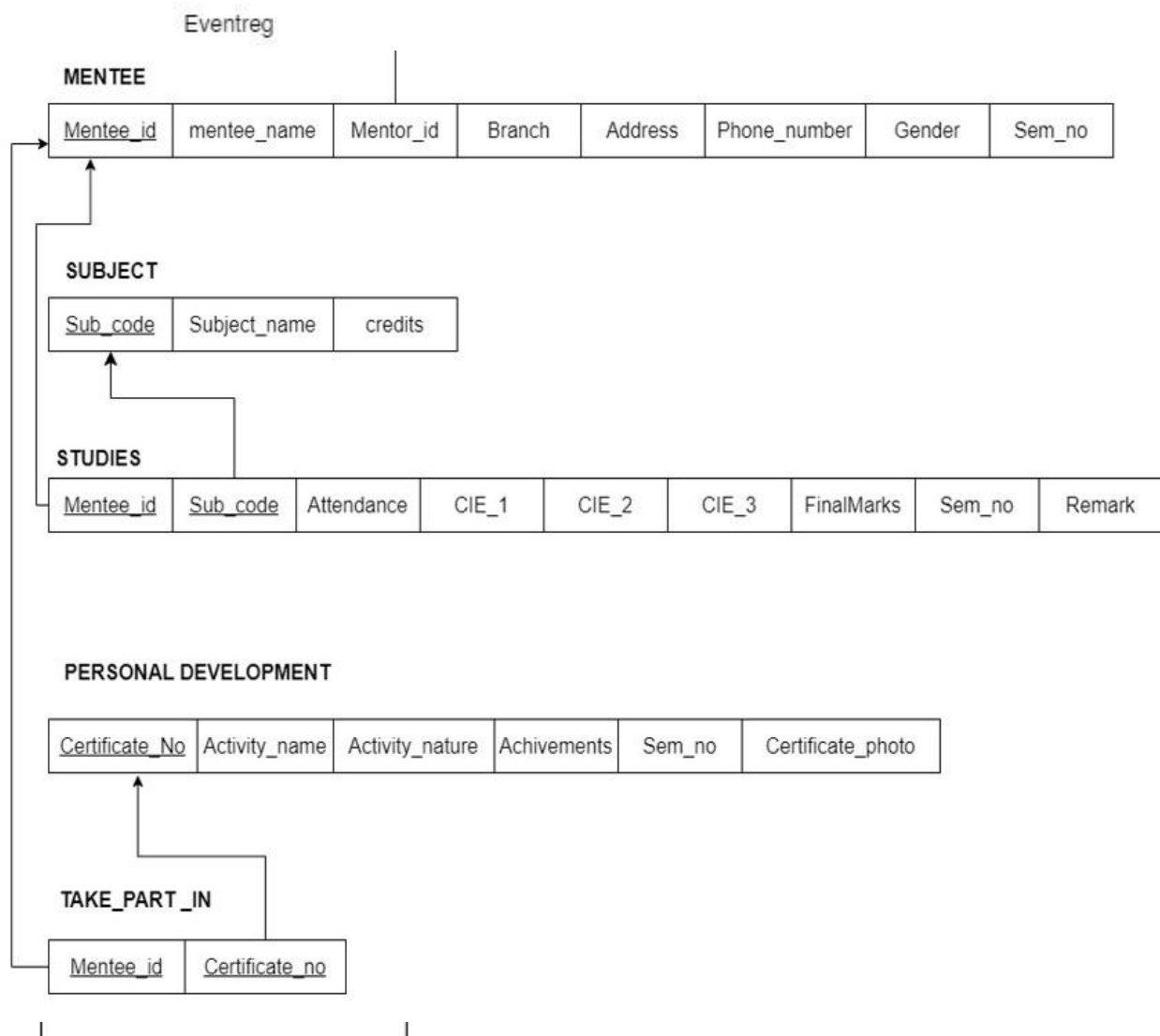


Figure 3.2.3 Mapping of M:N relationship types

STEP 6: Mapping of multivalued attributes

The ERD of our project does not contain any Multivalued Attributes.

STEP 7: Mapping of n-array relation types

The ERD of our project does not include any N- array relationship type.

3.3. SCHEMA DIAGRAM

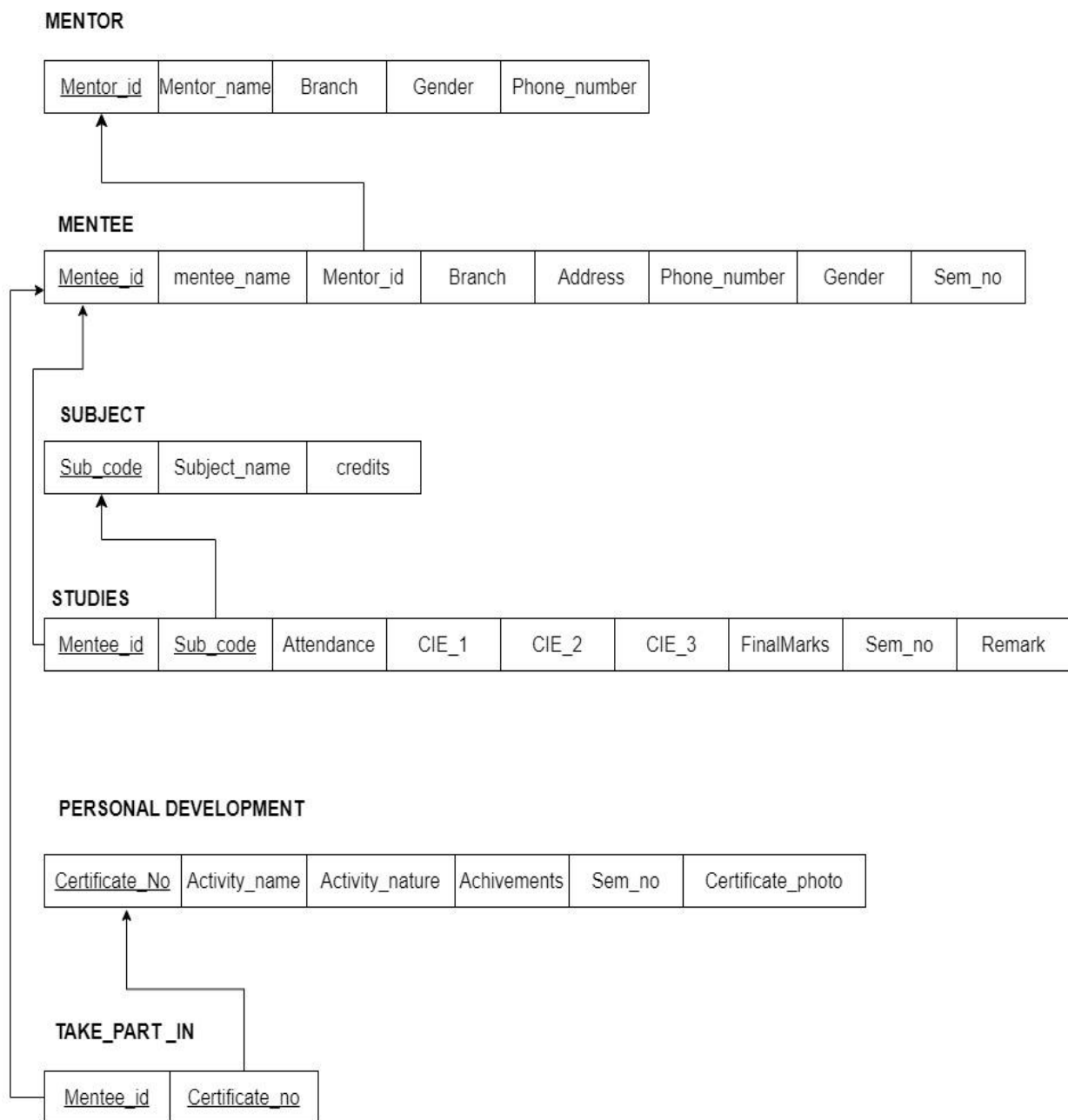
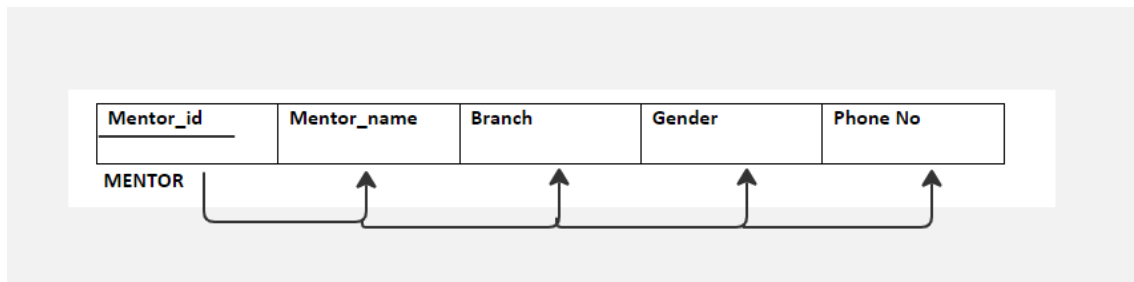


Figure 3.3.1 Schema Diagram

CHAPTER 4

4. NORMALIZATION

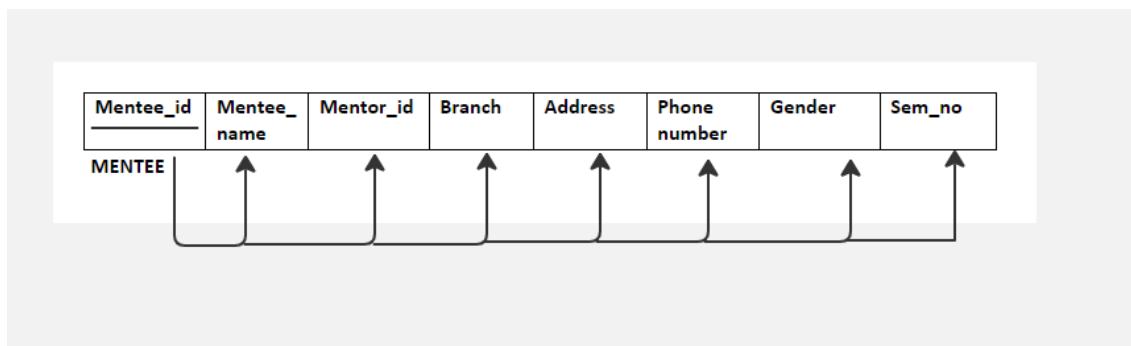


FD1: mentor_id -> mentor_name, branch, gender, phone_no (Mentor ID uniquely determines all other attributes)

1NF: It is 1NF because there are no multivalued attributes in the relational schema.

2NF: The relations are in 2NF since no non-key attributes are functionally dependent on part of the primary key (no partial dependencies).

3NF: The relations are in 3NF since no non-key attributes are functionally determined by another non-key attribute. The relation is in 1st, 2nd and 3rd Normal Form.

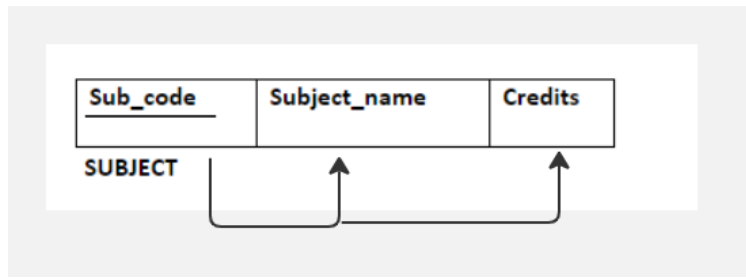


FD2: mentee_id -> mentee_name, mentor_id, address, phone_no, gender, sem_no (Mentee ID uniquely determines all other attributes)

1NF: It is 1NF because there are no multivalued attributes in the relational schema.

2NF: The relations are in 2NF since no non-key attributes are functionally dependent on part of the primary key (no partial dependencies).

3NF: The relations are in 3NF since no non-key attributes are functionally determined by another non-key attribute. The relation is in 1st, 2nd and 3rd Normal Form.

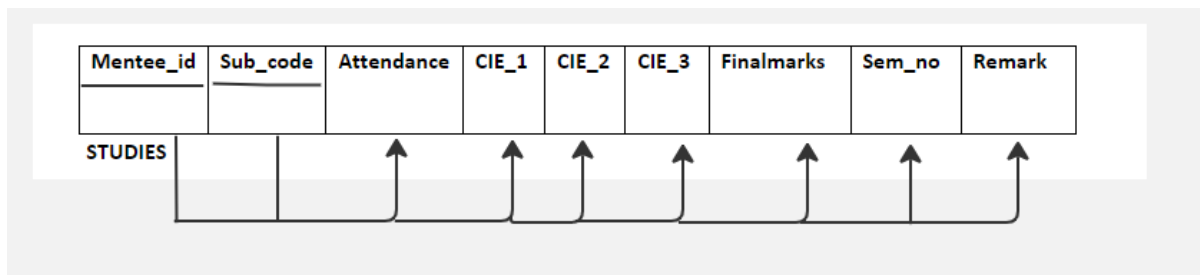


FD3: sub_code -> subject_name, credits (Subject code uniquely determines the subject name and credits)

1NF: It is 1NF because there is no multivalued attributes in the relational schema

2NF: The relations are in 2NF since no non-key attributes are functionally dependent on part of primary key (no partial dependencies).

3NF: The relations are in 3NF since no non-key attributes are functionally determined by another non-key attribute. The relation is in 1st, 2nd and 3rd Normal Form.

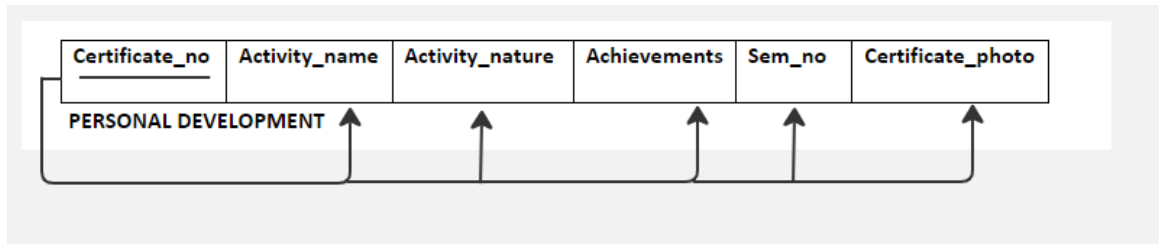


FD4: mentee_id, sub_code -> attendance, CIE_1, CIE_2, CIE_3, final_marks, sem_no, remarks (Combination of Mentee ID, Subject Code uniquely determines the other attributes)

1NF: It is 1NF because there is no multivalued attributes in the relational schema

2NF: The relations are in 2NF since no non-key attributes are functionally dependent on part of primary key (no partial dependencies).

3NF: The relations are in 3NF since no non-key attributes are functionally determined by another non-key attribute. The relation is in 1st, 2nd and 3rd Normal Form.

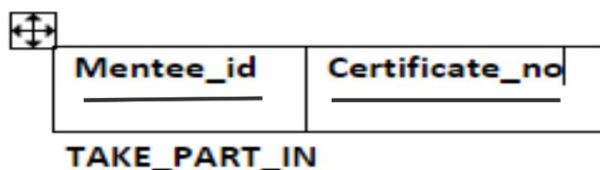


FD5: certificate_no -> activity_name, activity_nature, achievements, sem_no, certificate_photo
(Certificate number uniquely determines all other attributes)

1NF: It is 1NF because there is no multivalued attributes in the relational schema

2NF: The relations are in 2NF since no non-key attributes are functionally dependent on part of primary key (no partial dependencies).

3NF: The relations are in 3NF since no non-key attributes are functionally determined by another non-key attribute. The relation is in 1st, 2nd and 3rd Normal Form.



FD6: mentee_id, certificate_no -> { } (each attribute can serve as a unique identifier for tuples in the relation, and that there are no functional dependencies among the attributes.)

1NF: It is 1NF because there is no multivalued attributes in the relational schema

2NF: The relations are in 2NF since no non-key attributes are functionally dependent on part of primary key (no partial dependencies).

3NF: The relations are in 3NF since no non-key attributes are functionally determined by another non-key attribute. The relation is in 1st, 2nd and 3rd Normal Form.

CHAPTER 5

5. Implementation

5.1 Creation of tables:

```
CREATE TABLE mentor(  
    mentor_id varchar(15) NOT NULL,  
    mentor_name varchar(45) NOT NULL,  
    branch varchar(20) NOT NULL,  
    Gender varchar(10) NOT NULL,  
    phno bigint NOT NULL,  
    PRIMARY KEY ( mentor_id ) ,  
    phone_number_check2 CHECK ((`phno` between 6000000000 and 9999999999));
```

```
CREATE TABLE mentee (  
    mentee_id varchar(15) NOT NULL,  
    mentee_name varchar(30) NOT NULL,  
    mentor_id varchar(15) NOT NULL,  
    address varchar(30) NOT NULL,  
    gender varchar(10) NOT NULL,  
    phone_number bigint NOT NULL,  
    branch varchar(10) NOT NULL,  
    sem_no int NOT NULL,  
    PRIMARY KEY(mentee_id),  
    FOREIGN KEY (mentor_id) REFERENCES mentor (mentor_id) ON DELETE CASCADE,  
    phone_number_check CHECK ((`phone_number` between 6000000000 and 9999999999))
```

```

CREATE TABLE personal_development (
  sem_no int NOT NULL,
  activity_name varchar(45) NOT NULL,
  activity_nature varchar(45) NOT NULL,
  achivements varchar(20) NOT NULL,
  certificate_no varchar(15) NOT NULL,
  cerificate_photo blob NOT NULL, PRIMARY KEY ( certificate_no ));

```

```

CREATE TABLE studies (
  mentee_id varchar(15) NOT NULL,
  sub_code varchar(10) NOT NULL,
  attendance double DEFAULT NULL,
  CIE_1 int DEFAULT NULL,
  CIE_2 int DEFAULT NULL,
  CIE_3 int DEFAULT NULL,
  finalmarks int DEFAULT NULL,
  sem_no int NOT NULL,
  remark varchar(120) DEFAULT NULL,
  PRIMARY KEY ( mentee_id , sub_code ),
  FOREIGN KEY ( mentee_id ) REFERENCES mentee ( mentee_id ) ON DELETE CASCADE,
  FOREIGN KEY ( sub_code ) REFERENCES subject ( sub_code ) ON DELETE CASCADE,
  CHECK ((( CIE_1 >= 0) and ( CIE_1 <= 30))), CHECK ((( CIE_2 >= 0) and ( CIE_2 <=
  30))), CHECK ((( CIE_3 >= 0) and ( CIE_3 <= 30)))

```

```
CREATE TABLE subject (
  sub_code varchar(10) NOT NULL,
  subject_name varchar(50) NOT NULL,
  credits int NOT NULL,
  PRIMARY KEY ( sub_code ));
```

```
CREATE TABLE take_part_in (
  mentee_id varchar(15) NOT NULL,
  certificate_no varchar(15) NOT NULL,
  PRIMARY KEY ( mentee_id , certificate_no ),
  FOREIGN KEY( certificate_no ) REFERENCES
  personal_development( certificate_no )ON DELETE CASCADE,
  FOREIGN KEY ( mentee_id ) REFERENCES mentee ( mentee_id )ON DELETE CASCADE);
```

5.2 Description of tables:

```
mysql> desc mentor;
```

Field	Type	Null	Key	Default	Extra
mentor_id	varchar(15)	NO	PRI	NULL	
mentor_name	varchar(45)	NO		NULL	
branch	varchar(20)	NO		NULL	
gender	varchar(10)	NO		NULL	
phno	bigint	NO		NULL	

```
5 rows in set (0.03 sec)
```

```
mysql> desc mentee;
```

Field	Type	Null	Key	Default	Extra
mentee_id	varchar(15)	NO	PRI	NULL	
mentee_name	varchar(30)	NO		NULL	
mentor_id	varchar(15)	NO	MUL	NULL	
address	varchar(30)	NO		NULL	
gender	varchar(10)	NO		NULL	
phone_number	bigint	NO		NULL	
branch	varchar(10)	NO		NULL	
sem_no	int	NO		NULL	

8 rows in set (0.01 sec)

```
mysql> desc personal_development;
```

Field	Type	Null	Key	Default	Extra
sem_no	int	NO		NULL	
activity_name	varchar(45)	NO		NULL	
activity_nature	varchar(45)	NO		NULL	
achivements	varchar(20)	NO		NULL	
certificate_no	varchar(15)	NO	PRI	NULL	
cerificate_photo	blob	NO		NULL	

6 rows in set (0.01 sec)

```
mysql> desc take_part_in;
```

Field	Type	Null	Key	Default	Extra
mentee_id	varchar(15)	NO	PRI	NULL	
certificate_no	varchar(15)	NO	PRI	NULL	

2 rows in set (0.00 sec)

```
mysql> desc subject;
```

Field	Type	Null	Key	Default	Extra
sub_code	varchar(10)	NO	PRI	NULL	
subject_name	varchar(50)	NO		NULL	
credits	int	NO		NULL	

3 rows in set (0.00 sec)

```
mysql> desc studies;
```

Field	Type	Null	Key	Default	Extra
mentee_id	varchar(15)	NO	PRI	NULL	
sub_code	varchar(10)	NO	PRI	NULL	
attendance	double	YES		NULL	
CIE_1	int	YES		NULL	
CIE_2	int	YES		NULL	
CIE_3	int	YES		NULL	
finalmarks	int	YES		NULL	
sem_no	int	NO		NULL	
remark	varchar(120)	YES		NULL	

9 rows in set (0.01 sec)

5.3 JDBC USED TO CONNECT JAVA AND MYSQL

```
import java.sql.*;

public class ConnectionProvider {
    public static Connection getCon(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/mentor_mentee","root","Spartan
s420@");
            return con ;
        }
        catch(Exception e)
        {
            e.printStackTrace();
            return null;
        }
    }
}
```

5.4 Insert/update/delete operation has been implemented as follows

5.4.1 INSERT

```
try{
    Connection con=ConnectionProvider.getCon();
    String query="insert into mentor values(?,?,?,?)";
    PreparedStatement ps=con.prepareStatement(query);
    ps.setString(1, mentor_id.getText());
    ps.setString(2, mentor_name.getText());
    String value=branchCombo.getSelectedItem().toString();
    ps.setString(3, value);
    if(rbMale.isSelected())
    {
        ps.setString(4, rbMale.getText());
    }
}
```

```

    }

    else{

        ps.setString(4, rbFemale.getText());

    }

    ps.setLong(5, Long.parseLong(phno.getText()));

    ps.executeUpdate();

    JOptionPane.showMessageDialog(null, "insertion sucessfull");

}

catch( Exception e){

    e.printStackTrace();

    JOptionPane.showMessageDialog(null, e);

}

```

5.4.2 UPDATE

```

String mentor_id =jTextField1.getText();

String mentor_name =jTextField2.getText();

String branch =jTextField3.getText();

String gender=jTextField4.getText();

long phone_number=Long.parseLong(jTextField5.getText());

try{

    Connection con=ConnectionProvider.getCon();

    String query1="update mentor set mentor_name=?,branch=?,phno=? where mentor_id
    ="+mentor_id+"";

    java.sql.PreparedStatement pst = con.prepareStatement(query1);

    pst.setString(1, mentor_name);

    pst.setString(2, branch);

    pst.setLong(3, phone_number);

    pst.executeUpdate();

    JOptionPane.showMessageDialog(null, "details edited sucessful

}

catch(Exception e){

```

```
        JOptionPane.showMessageDialog(null, e);  
    }  
}
```

5.4.3 DELETE

```
String sub_code=jTextField3.getText();  
  
try{  
    Connection con=ConnectionProvider.getCon();  
    String Query3="delete from subject where sub_code= ?";  
    java.sql.PreparedStatement pst = con.prepareStatement(Query3);  
    pst.setString(1, sub_code);  
    pst.executeUpdate();  
    JOptionPane.showMessageDialog(null, "deleted the details");  
    showTale2();  
}  
catch(Exception e){  
    JOptionPane.showMessageDialog(null, e);  
}
```

5.5 TRIGGERS AND STORING PROCEDURE

5.5.1 Trigger Implementation in Java

A database trigger is a stored program which is automatically fired or executed when some events occur. A trigger can execute in response to INSERT,DELETE AND UPDATE:

The "avg_cie_insert" trigger calculates average of "CIE_1", "CIE_2", "CIE_3" before an insert on "studies" table and sets the result to "finalmarks". The "avg_cie_update" trigger does the same before an update on the "studies" table.

```

DDELIMITER $$
CREATE TRIGGER avg_cie_insert
BEFORE INSERT ON studies
FOR EACH ROW
BEGIN
    SET NEW.finalmarks = (NEW.CIE_1 +
NEW.CIE_2 + NEW.CIE_3) / 3;
END$$
DELIMITER ;

```

```

DELIMITER $$
CREATE TRIGGER avg_cie_update
BEFORE UPDATE ON studies
FOR EACH ROW
BEGIN
    SET NEW.finalmarks = (NEW.CIE_1 +
NEW.CIE_2 + NEW.CIE_3) / 3;
END$$
DELIMITER ;

```

5.5.2 Stored ProcedureImplementation in Java

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.

The stored procedure inserts a new row into the "subject" table with the values of the input parameters, "sub_code", "subject_name", and "credits", for the columns "sub_code", "subject_name", and "credits" respectively.

```

DELIMITER $$
CREATE PROCEDURE insert_subject (IN sub_code VARCHAR(10), IN subject_name VARCHAR(50), IN
credits INT)
BEGIN
    INSERT INTO subject (sub_code, subject_name, credits)
    VALUES (sub_code, subject_name, credits);
END$$
DELIMITER ;

```

CHAPTER 6

RESULTS

6.1 LOGIN DETAILS PAGE

This figure shows the login details page of mentor and mentee.

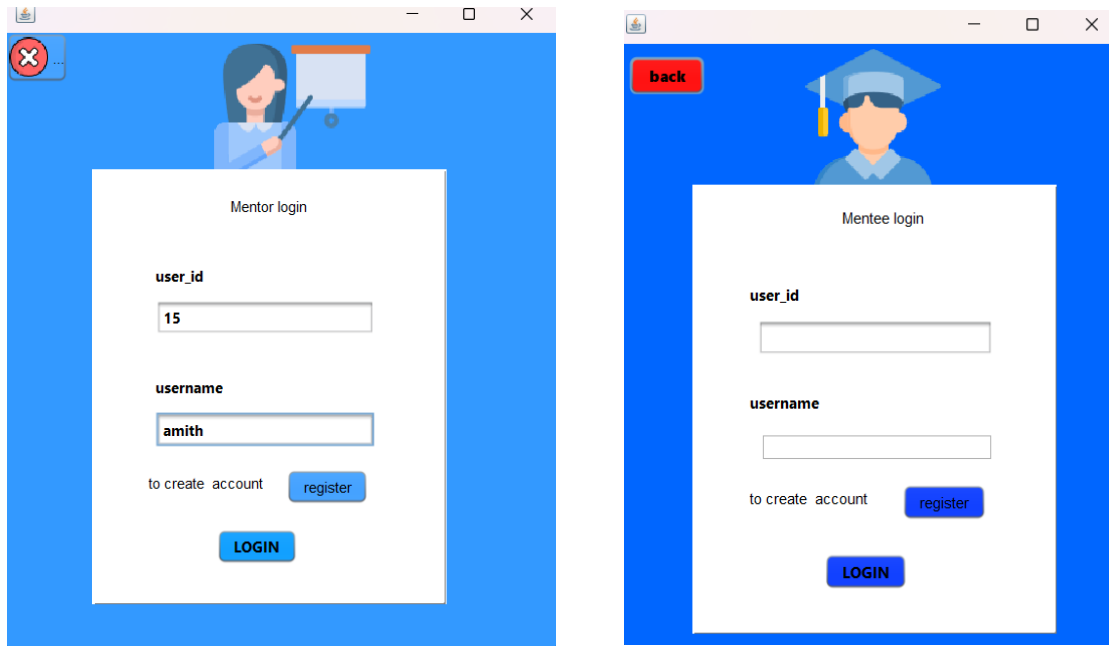


Fig 6.1 Login interface

6.2 MAIN PAGE

The following image shows the main front page of our project.



Fig 6.2 main page

6.3 MENTEE DATA INSERTION IMAGE

The following image shows successful insertion of data of mentee.

DATA ENTRY OF MENTEEES

mentee_id: 4sf20cio15

mentee_name: aster

mentor_id: 15

address: managalore

gender: Female

phnoNumber: 6744455679

sem, branch: 5, AIML

Buttons: INSERT, UPDATE, DELETE

mentee_id	mentee_name	mentor_id	address	gender	phone_number	branch	sem_no
4sf20cio14	anil	15	managalore	Male	6744455697	AIML	4
4sf20cio20	divij	15	kasargod	Male	9741112861	AIML	5
4sf20cio28	karthik	15	udupi	Male	8073744954	AIML	5
4sf20cio29	snehith	15	puttur	Male	8073744578	AIML	5
4sf20cio30	manhor	15	kundapur	Male	7944321145	AIML	5

Message: insertion successfull

Fig 6.3 Mentee details insertion

6.4 UPDATE MENTEE DATA PAGE

The image shows the successful updation of mentee details.

DATA ENTRY OF MENTEEES

mentee_id: 4sf20cio28

mentee_name: karthik

mentor_id: 15

address: mangalore

gender: Male

phnoNumber: 8073744954

sem, branch: 5, AIML

Buttons: INSERT, UPDATE, DELETE

mentee_id	mentee_name	mentor_id	address	gender	phone_number	branch	sem_no
4sf20cio14	anil	15	managalore	Male	6744455697	AIML	4
4sf20cio15	aster	15	managalore	Female	6744455679	AIML	5
4sf20cio20	divij	15	kasargod	Male	9741112861	AIML	5
4sf20cio28	karthik	15	udupi	Male	8073744954	AIML	5
4sf20cio29	snehith	15	puttur	Male	8073744578	AIML	5
4sf20cio30	manhor	15	kundapur	Male	7944321145	AIML	5

Message: updated successfull

Fig 6.4 Mentee details updating

6.5 DELETE MENTEE DATA IMAGE

The image shows the deleting of mentee details.

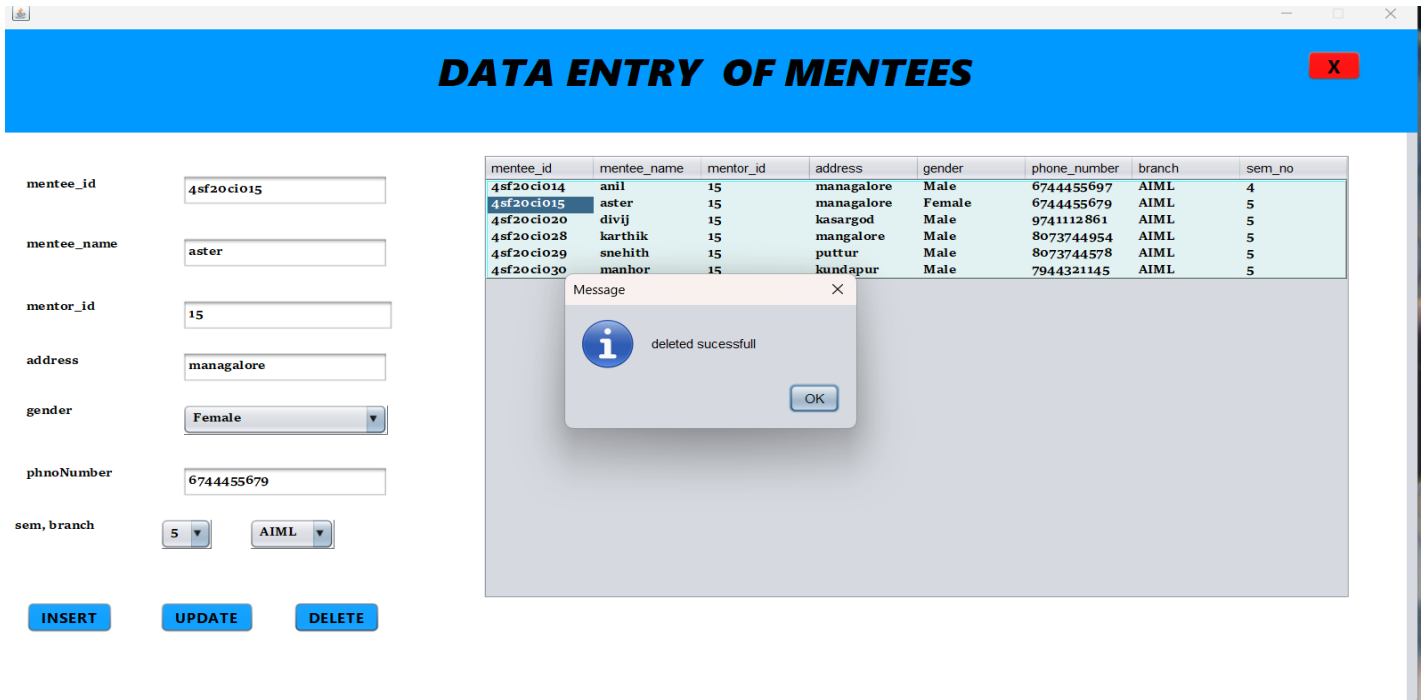


Fig 6.5 deleting the mentee details

6.6 MENTOR HOME PAGE IMAGE

The image shows the home page of mentor that gives overview of mentee details

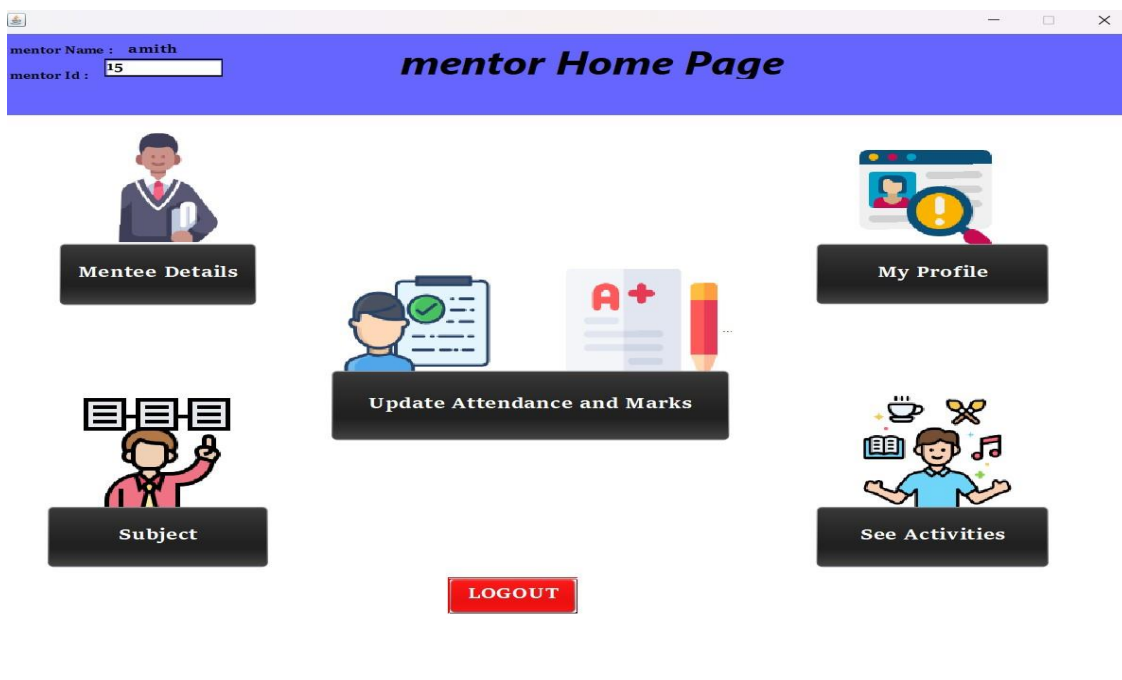


Fig 6.6 Mentor home page

6.7 MENTEE HOME PAGE IMAGE

The image shows the home page of mentee to access their details.

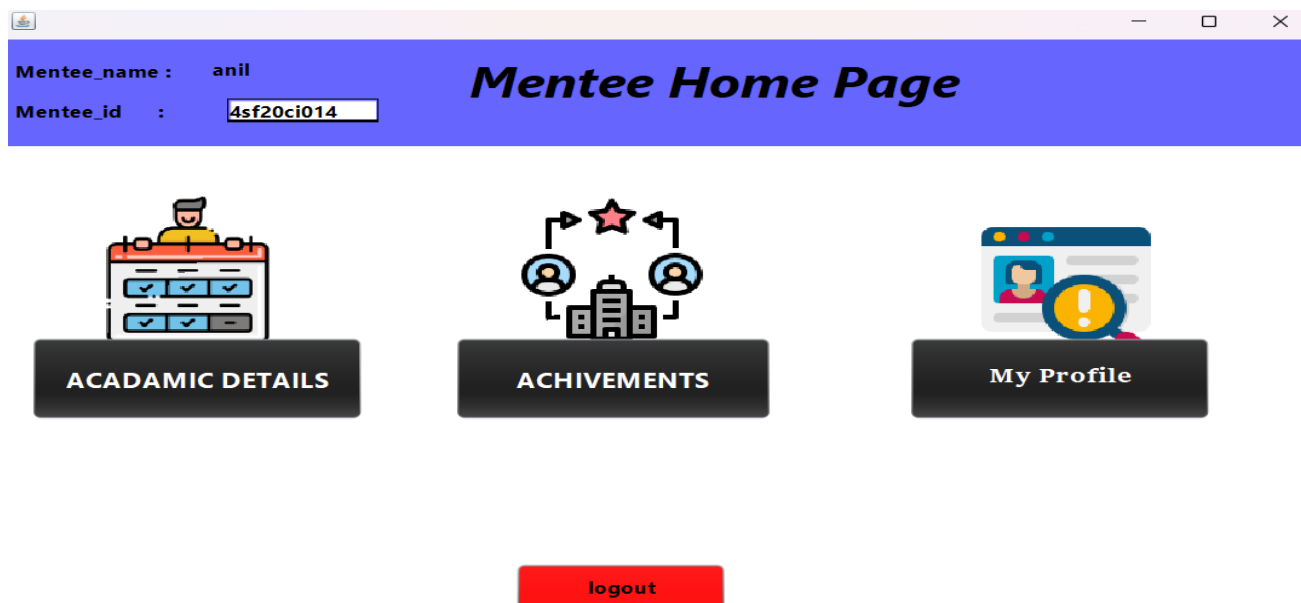


Fig 6.7 Mentee home page

6.8 ATTENDANCE AND MARKS UPDATING IMAGE

The image shows the attendance and marks updation of mentees

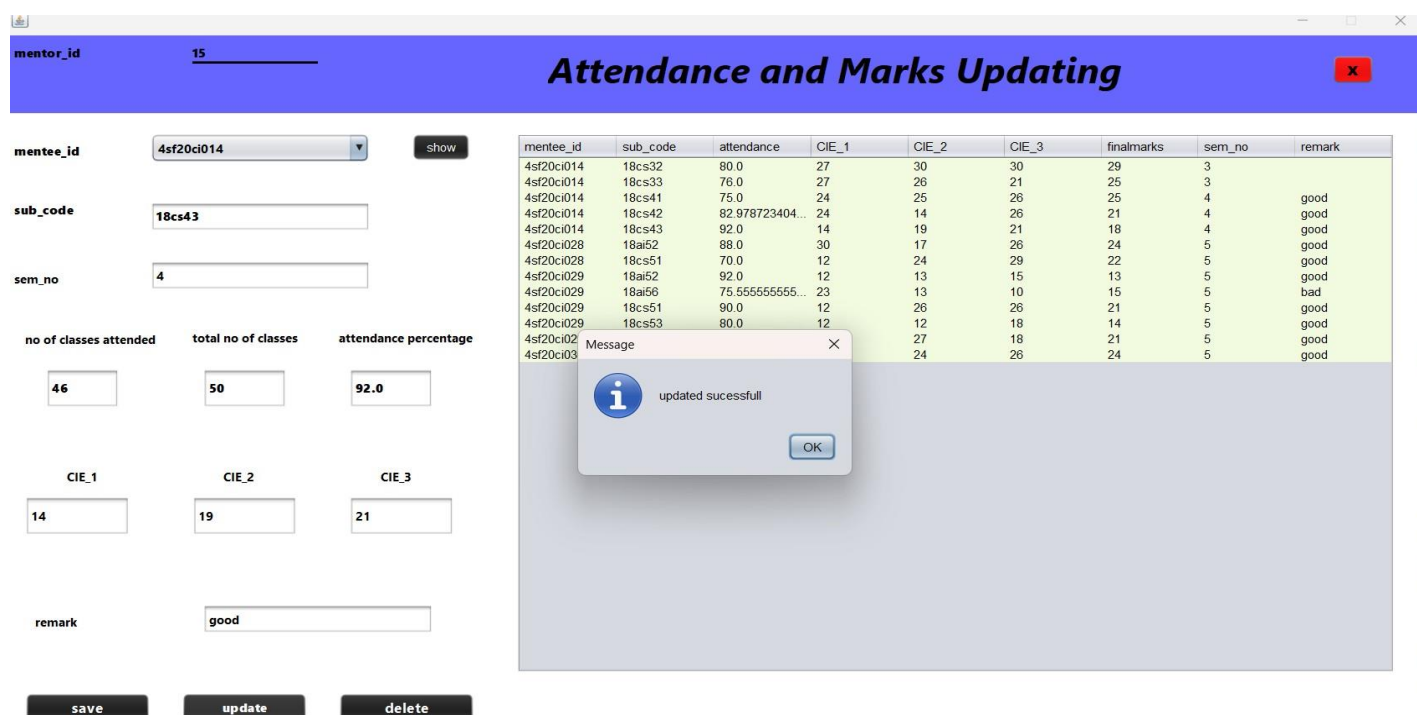


Fig 6.8 Attendance and marks page

6.9 MENTEE ACHIEVEMENT PAGE

The image shows the achievement page of mentee that comes under personal development

mentee_id	sem_no	activity_name	activity_nature	achivements	certificate_no	certificate_photo
4sf20ci014	3	hacknight	coding	1st place	13234vds	[B@abfe842
4sf20ci014	3	sinchana	dance	3rd place	23234vds	[B@7ad38831
4sf20ci014	3	cricket	sports	2nd place	232378	[B@7d444567
4sf20ci014	3	handball	sports	2nd place	2342342	[B@53be7c4
4sf20ci028	5	handball	sports	2nd place	123cd	[B@26e3e803
4sf20ci028	5	cricket	sports	2nd place	123vd	[B@40976eee
4sf20ci028	5	volleyball	sports	2nd	1324323	[B@1f2774e6
4sf20ci030	5	dbms project	coding	1st place	1223tfd	[B@13fe7501

Fig 6.9 Mentee achievement page

6.10 DEPARTMENT LOGIN

The image shows the department login page

Fig 6.10 department login

6.11 MENTEE REGISTRATION IMAGE

The image shows the registration page for mentees

Mentee registration

mentee_id
4sf20ci020

mentee_name
divij

mentor_id
15

address
kasargod

gender
☒ Male
 ☐ Female

phoneNumber
9741112861

sem ,section and scheme
 5
 AIML

register

Fig 6.11 Mentee registration page

6.12 MENTOR REGISTRATION IMAGE

The image shows the registration page for mentors

Mentor registration

mentor_id
3

mentor_name
tejas

branch
AIML

gender
☒ Male
 ☐ Female

phnoNumber
8073744954

register

Fig 6.12 Mentor registration page

6.13 SUBJECT DETAILS IMAGE

The following image shows the subject details of mentees

sub_code	subject_name	credits
18ai52	python	4
18ai55	artificial intelligence	4
18ai56	mml	4
18ai61	digital image proc	3
18civ57	enviroment	1
18cs32	data strutures	4
18cs33	ade	3
18cs34	co	3
18cs41	daa	4
18cs42	os	4
18cs43		3
18cs51		4
18cs53		5
18cs54		4

Fig 6.13 subject details page

6.14 TRIGGER SUCCESSFUL IMAGE (UPDATION)

The image shows the success of trigger that updates the attendance and final marks.

mentee_id	sub_code	attendance	CIE_1	CIE_2	CIE_3	finalmarks	sem_no	remark
4sf20ci014	18cs32	80.0	27	30	30	29	3	
4sf20ci014	18cs33	76.0	27	26	21	25	3	
4sf20ci014	18cs41	75.0	24	25	26	25	4	good
4sf20ci014	18cs42	82.978723404...	24	14	26	21	4	good
4sf20ci014	18cs43	95.238095238...	14	19	21	18	4	good
4sf20ci028	18ai52	88.0	30	17	26	24	5	good
4sf20ci028	18cs51	70.0	12	24	29	22	5	good
4sf20ci029	18ai52	92.0	12	13	15	13	5	good
4sf20ci029	18ai56	75.555555555...	23	13	10	15	5	bad
4sf20ci029	18cs51	90.0	12	26	26	21	5	good
4sf20ci029	18cs53	80.0	12	12	18	14	5	good
4sf20ci02				27	18	21	5	good
4sf20ci03				24	26	24	5	good

Fig 6.14 final marks trigger update image

CHAPTER 7

CONCLUSION

“Mentor Mentee Management System” software developed has been designed to achieve maximum efficiency and reduce the time taken to handle the peer detail and their activity. It is designed to replace an existing manual record system thereby reducing time taken for calculations and for storing data. The system uses Java (using the Netbeans framework) as front end and Microsoft MySQL as a backend for the database. The system is strong enough to withstand regressive daily operations under conditions where the database is maintained and cleared over a certain time of span. The implementation of the system in the organization will considerably reduce data entry, time and also provide readily calculated reports. This project has many future applications like it can be used in trainee and trainer details in corporate world. This project was build keeping in mind all the requirements of the mentoring system and can be implemented without much problems.

REFERENCES

- [1] **RESEARCH:** <https://www.geeksforgeeks.org/introduction-to-jdbc/>
- [2] **RESEARCH:** Sahyadri College Of Engineering and Management Mentor Mentee Book(Pink Book)
- [3] **TEXTBOOK:**Database System Concepts Seventh Edition Avi Silberschatz
HenryF.KorthS.SudarshanMcGraw-Hill
- [4] **DEMONSTRATION:** https://github.com/22skd/MENTOR_MENTEE_NEW