

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY
BELAGAVI-590018**



**A PROJECT REPORT
ON
NURSERY MANAGEMENT SYSTEM**

BY

ASHWIN SHETTY
4SF17CS035

RETHESH HV
4SF19CS412

SHREYAS
4SF19CS414

In the partial fulfillment of the requirement for V Sem. B. E. (CSE)

DBMS LABORATORY WITH MINI PROJECT

Under the guidance of

POOJA N.S
Asst. Professor, Dept. of CSE



**Department of Computer Science & Engineering
SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
Adyar, Mangaluru-575007
2020-21**

SAHYADRI
COLLEGE OF ENGINEERING & MANAGEMENT
(Affiliated to Visvesvaraya Technological University, BELAGAVI)
Adyar, Mangaluru – 07

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled “**ACTING SCHOOL MANAGEMENT SYSTEM**” is submitted in partial fulfillment for the requirement of V sem. B. E. (Computer Science & Engineering), “**DBMS LABORATORY WITH MINI PROJECT**” during the year 2020 – 21 is a result of bonfire work carried out by

ASHWIN SHETTY	4SF17CS035
RETHESH HV	4SF19CS412
SHREYAS	4SF19CS414

.....
Mrs. Pooja N S
Asst. Prof. Dept. of CS&E
SCEM, Mangalore

.....
Dr. Pushpalatha K
HOD, Dept. of CS&E
SCEM, Mangalore

Signature of the Examiners

1.
2.

ABSTRACT

This Acting school Management System realized transmission and control of Acting school system, so as to

Facilitate the management and decision of sales, and reduce a bid burden for nursery business and for their managers.it also helps in improving the efficiency of nursery. Its requirement are to provide the basic information maintain function of employers, managers and customers so that managers can use the functions add update and delete the basic information of products and their employees details. Acting school management system is very convenient input output and searching for available products so as to make work of this market so efficient. In aspect of software, the nursery management system uses java language and MySQL as the background database. Various configuration in computer including input and output internal memory and external memory capacities are meet the requirement of nursery managers.

ACKNOWLEDGEMENT

It is with the great satisfaction and emporia that we are submitting the Mini Project report on “**ACTING SCHOOL MANAGEMENT SYSTEM**”. We have completed the part of 5th semester **DBMS Laboratory with Project** of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi.

We take this opportunity to express our sincere thanks and indebtedness to our Project guide and mentor, **Mrs. Pooja N S Professor, Department of CSE**, for her support and guidance. Her vision and suggestions throughout the project period have been fundamental in the completion of project.

We extend our warm gratitude to Dr. Pushpalatha K, Head of the Department, Department of CSE, for her constant support and advice that helped us to complete this project successfully.

We are extremely grateful to our beloved Principal, Dr. Rajesh S, Principal, Sahyadri College of engineering and management for encouraging us to come up with new ideas and to express them in a systematic manner.

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

Finally, we are thankful to our family and friends who helped us in our work and made the project a successful one.

Ashwin Shetty
(4SF17CS035)

Rethesh Hv
(4SF19CS412)

Shreyas
(4SF19CS414)

PAGE INDEX

Chapter No.	Topic	Page No.
1.	Introduction	6-7
1.1	MYSQL	7
1.2	Java	7-8
1.3	NetBeans	9-10
2.	Design	11
2.1	E-R diagram	12-13
2.2	Relational schema (ER to relational schema)	14-17
2.3	Schema diagram	18-21
2.4	Normalization	22
3.	Implementation	16
3.1	Results	24-28
3.2	Conclusion	29
3.3	References	30

CHAPTER : 1

INTRODUCTION

Database is a collection of data and Management System is a set of programs to store and retrieve those data. Based on this we can define DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.

Database systems are basically developed for large amount of data. When dealing with huge amount of data, there are two things that require optimization: Storage of data and retrieval of data. 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.

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.

1.1 INTRODUCTION TO MYSQL

MySQL is an open-source, fast reliable, and flexible relational database management system, typically used with PHP. It is a database system used for developing web-based software applications.

It was developed by Michael Widenius and David Axmark in 1994. It is presently developed, distributed, and supported by Oracle Corporation. It was written in C, C++. It is fully multithreaded by using kernel threads. It can handle multiple CPUs if they are available. It provides transactional and non-transactional storage engines.

1.2 JAVA

Java is a popular programming language. It is owned by Oracle, and more than 3 billion devices run Java. Java works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.). It is open-source, secure, fast, free, and powerful. As Java is close to C++ and C#, it makes it easy for programmers to switch to Java.

1.3 INTRODUCTION TO NET BEANS

NetBeans IDE is a free, open source, integrated development environment (IDE) that enables you to develop desktop, mobile, and web applications. The IDE supports application development in various languages, including Java, HTML5, PHP, and C++.

The IDE provides integrated support for the complete development cycle, from project creation through debugging, profiling, and deployment. The IDE runs on Windows, Linux, Mac OS X, and other UNIX-based systems.

1.4 NECESSITY OF PROJECT

This project “**ACTING SCHOOL MANAGEMENT SYSTEM**” will help employees to quickly view what items are available in their industry, and they can add the sold details to the data base, and these records can track by managers.

CHAPTER :2

DESIGN

1.1 ER DIAGRAM

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

ER Diagram:

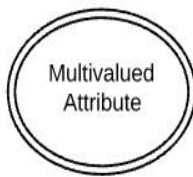


Fig 2.1.1

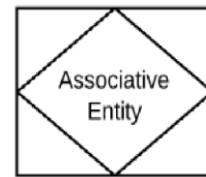


Fig 2.1.2

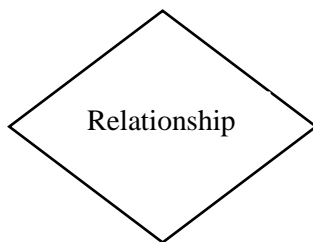


Fig 2.1.3

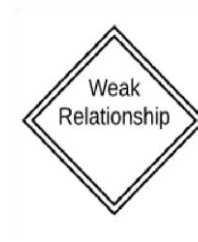


Fig 2.1.4

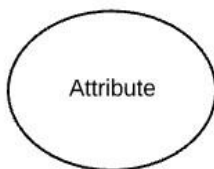


Fig 2.1.5

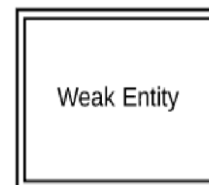
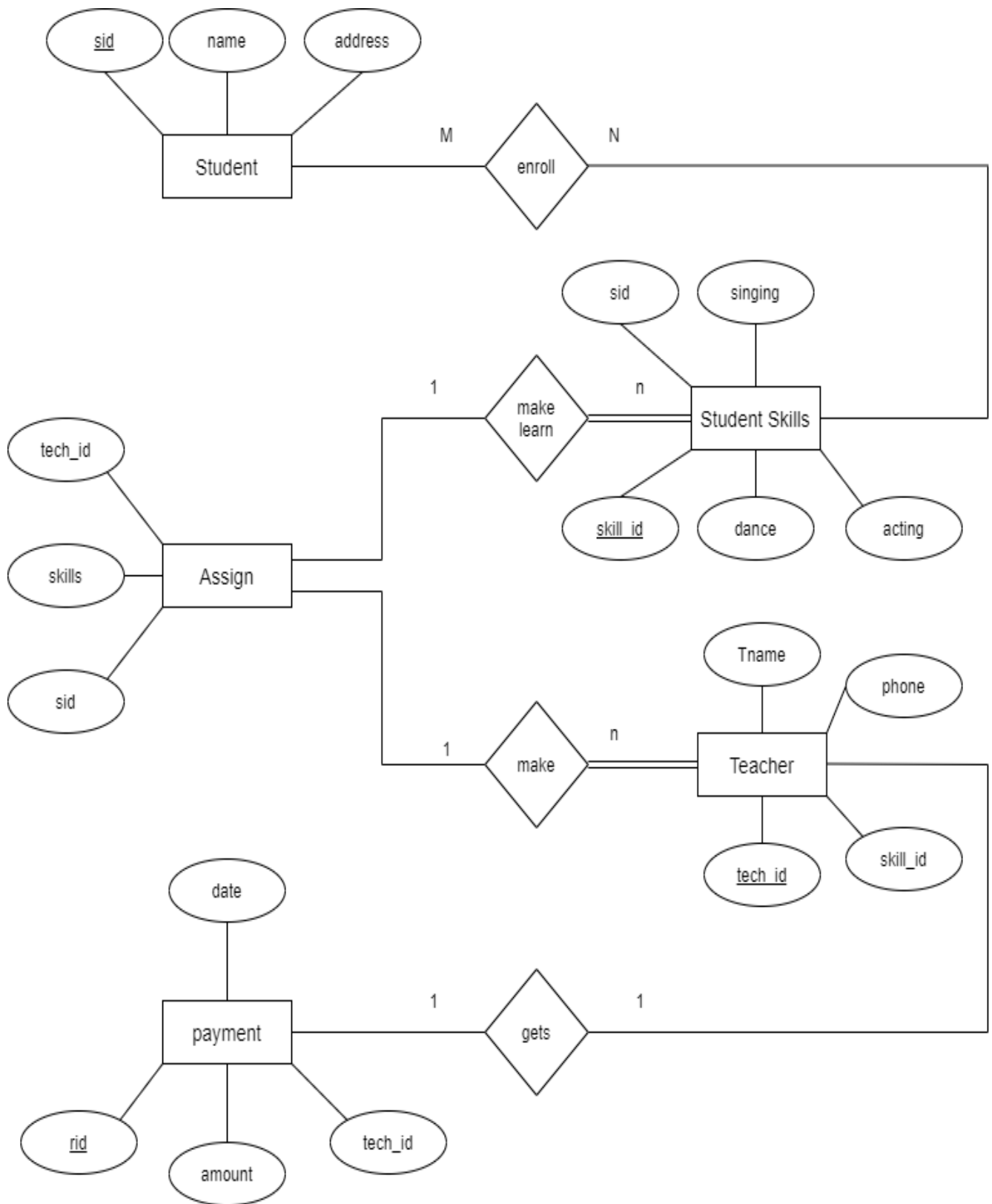


Fig 2.1.6

2.1 ER DIAGRAM



2.1 Relational Schema

Step 1: Mapping of strong entity

Student

SID	NAME	ADDRESS
-----	------	---------

Student Skills

SKILL_ID	SID	DANCE	ACTING	SINGING
----------	-----	-------	--------	---------

Teacher

TECH_ID	TNAME	PHONE	SKILL_ID
---------	-------	-------	----------

Assign

SID	TECH_ID	SKILL_ID
-----	---------	----------

Payment

RID	AMMOUNT	TECH_ID	DATE
-----	---------	---------	------

Strong entity is a relation

Entity is said to be a strong entity when it contains primary key.

Primary key has an attribute with unique values.

1. In the first relation Sid is a primary key.
2. In the second relation Skill_id is a primary key.
3. In the third relation Tech_id is a primary key.
4. In the fourth relation Rid is a primary key.

STEP 2: Mapping of weak entity Types

There is no weak entity hence this step is ignored

STEP 3: Mapping of 1:1 relation type

There is no 1.1 relation hence this step is ignored

STEP 4: Mapping of 1: N relation types

For every 1: N relation types identify the entity the N side. Make primary key entity

Which is participation in 1 side as foreign key of entity which is N side. If there are any attributes for relationships add to n side.

Student Skills

SKILL_ID	SID	DANCE	ACTING	SINGING
----------	-----	-------	--------	---------

Teacher

TECH_ID	TNAME	PHONE	SKILL_ID
---------	-------	-------	----------

Assign

SID	TECH_ID	SKILL_ID
-----	---------	----------

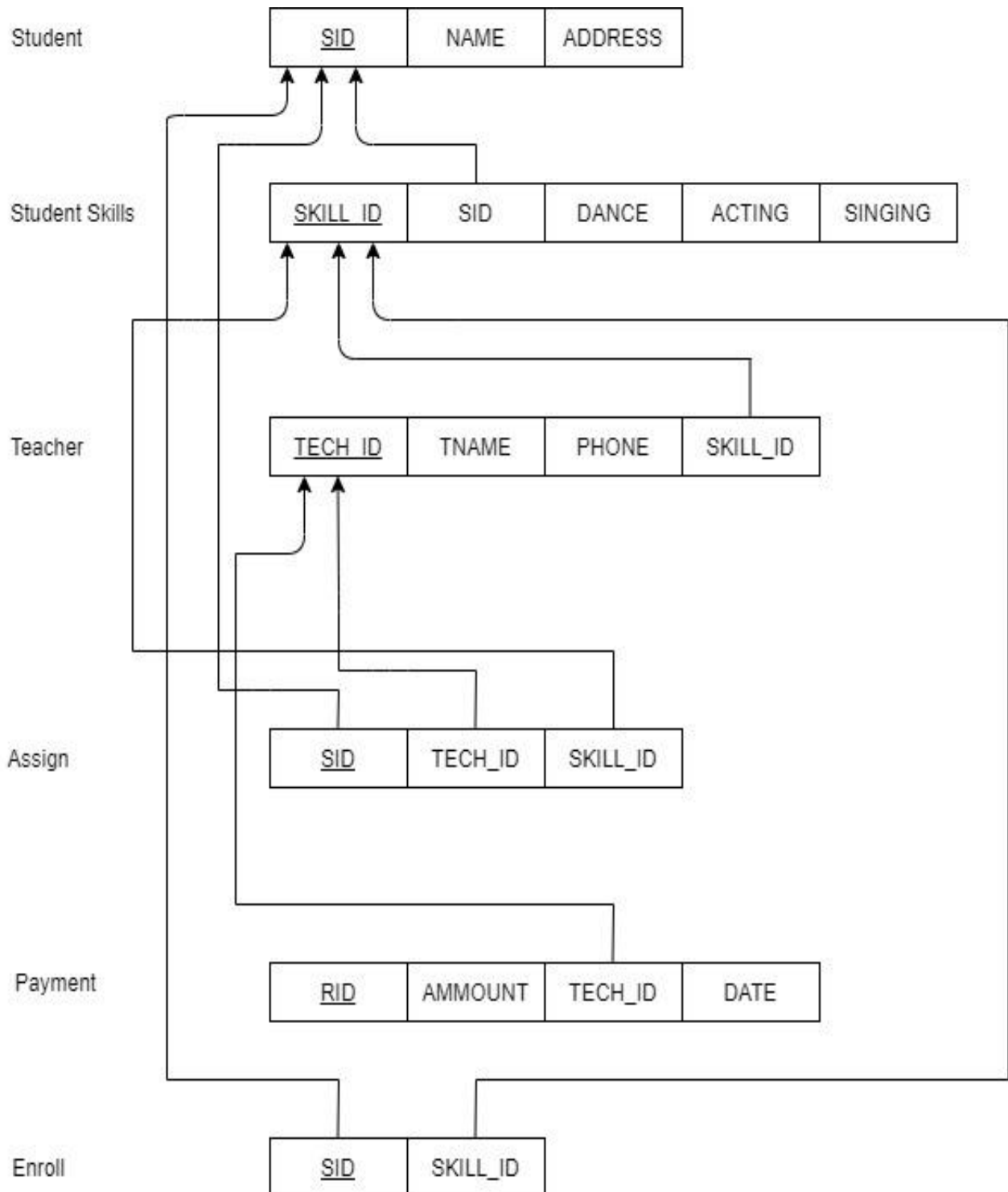
STEP 5: Mapping of M: N relation types

There is no M: N relation hence this step is ignored.

STEP 6: Mapping of multivalued attributes

There is multivalued attributes relation hence this step is ignored.

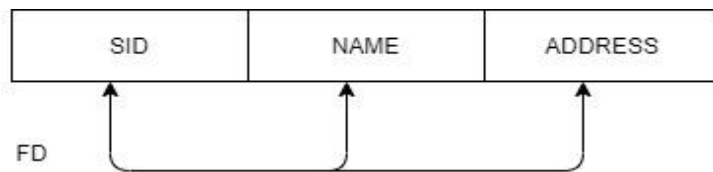
2.2 SCHEMA DIAGRAM



CHAPTER 3

NORMALIZATION

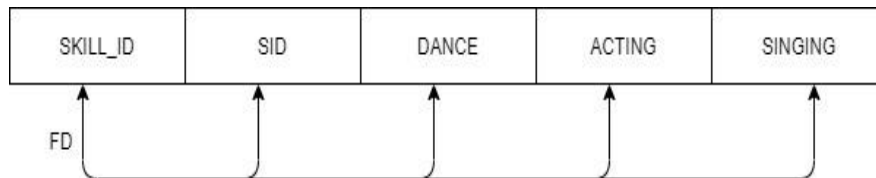
Student



$SID \rightarrow \{NAME, ADDRESS\}$

- The above relation is in 1NF because there are no multivalued attributes in the relational Schema.

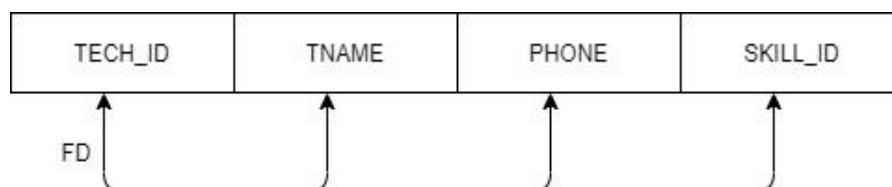
Student Skill



$SKILL_ID \rightarrow \{SID, DANCE, ACTING, SINGING\}$

- The above relation is in 1NF because there are no multivalued attributes in the relational Schema.
- The above relation is 2NF because all the attributes in the relational are fully functionally dependent on primary key.
- The above relation is 3NF because there is no transitive dependency on primary key.

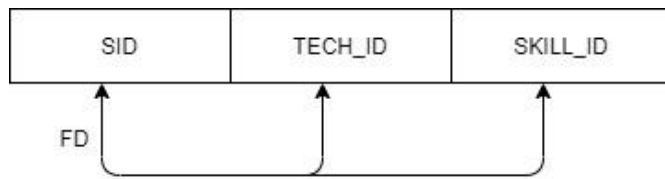
Teacher



$TECH_ID \rightarrow \{TNAME,PHONE,SKILL_ID\}$

- The above relation is in 1NF because there are no multivalued attributes in the relational Schema.
- The above relation is 2NF because all the attributes in the relational are fully functionally dependent on primary key.
- The above relation is 3NF because there is no transitive dependency on primary key.

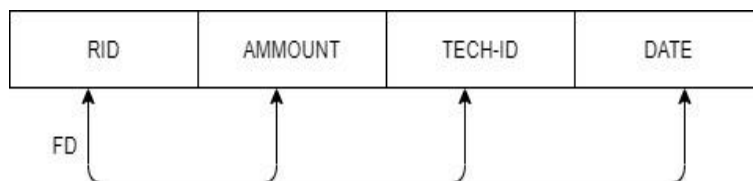
Assign



$SID \rightarrow \{TECH_ID,SKILL_ID\}$

- The above relation is in 1NF because there are no multivalued attributes in the relational Schema.
- The above relation is 2NF because all the attributes in the relational are fully functionally dependent on primary key.
- The above relation is 3NF because there is no transitive dependency on primary key.

Payment



$RID \rightarrow \{AMOUNT,TECH_ID,DATE\}$

- The above relation is in 1NF because there are no multivalued attributes in the relational Schema.
- The above relation is 2NF because all the attributes in the relational are fully functionally dependent on primary key.
- The above relation is 3NF because there is no transitive dependency on primary key.

1st Normal Form (1NF)

The relations are in 1NF since there is no multivalued attributes or nested relation.

2nd Normal Form (2NF)

The relation is in 2NF since every nonprime attribute are fully functionally dependent on part of the primary key.

3rd Normal Form (3NF)

The relations are in 3NF since no nonprime attributes are transitively dependent on primary key.

CHAPTER 4

IMPLEMENTATION

4.1 System Specification

Operating System: Windows 10

Memory: Minimum of 1GB of RAM, Minimum of 2GB

hard disk space. Backend: MySQL Edition

Frontend: NetBeans IDE 8.2

Table Structure

4.2 Student

```
create table student(
sid int(10),
name varchar(30) not null,
address varchar(30) not null,
constraint con_act_pk primary key(sid));
constraint con_stu_fk23 foreign key(stu_id) references student();
```

ATTRIBUTES	TYPE
SID	INT(10)
NAME	VARCHAR(30)
ADDRESS	VARCHAR(30)

4.3 Student Skill

```
create table student skills(
skill_id int(10),
sid int(10),
dance varchar(30) not null,
singing varchar(30) not null,
acting varchar(30) not null,
constraint con_mal_pk primary key(skill_id));
```

ATTRIBUTES	TYPE
SKILL_ID	INT(10)
SID	INT(10)
DANCE	VARCHAR(30)
SINGING	VARCHAR(30)
ACTING	VARCHAR(30)

4.4 Teacher

```
create table teacher(
tid int(10),
skill_id int(10),
phone int(15),
tname varchar(30) not null,constraint con_ten_pk primary key(tid),
constraint con_ten_fk foreign key(skill_id) references skills(skill_id));
```

ATTRIBUTES	TYPE
TID	INT(10)
SKILL_ID	INT(10)
PHONE	INT(10)
NAME	VARCHAR(30)

4.5 Assign

```
create table assign(
tid int(10),skill_id int(10),
sid int(10),
constraint con_act_fk122 foreign key(tid) references teacher(tid),
constraint con_act_fk124 foreign key(skill_id) references skills(skill_id),
constraint con_act_fk125 foreign key(sid) references student(sid));
```

ATTRIBUTES	TYPE
TID	INT(10)
SID	INT(10)
SKILLS_ID	INT(10)

4.5 Payment

```
create table payment(
rid int(10) not null,
tid int(10) not null,
amount int(15) not null,pay_date date,
constraint con_ten_pk1 primary key(rid),
constraint con_ten_fk1 foreign key(tid) references teacher(tid));
```

ATTRIBUTES	TYPE
RID	INT(10)
TID	INT(10)
AMMOUNT	VARCHAR(30)
DATE	DATE

4.2 FUNCTIONALTIES

4.2.1 Connecting to Database

```

Class.forName ("com.mysql.jdbc.Driver");

Connection Conn;

Conn=DriverManager.getConnection("jdbc:mysql://localhost:3306/login","admin"
,"password");

```

4.2.2 Insert

```

String sql;

sql = "insert into skills values(?, ?, ?, ?)";

PreparedStatement pstmt = conn.prepareStatement(sql);

pstmt.setInt(1,Integer.parseInt(jTextField1.getText()));

pstmt.setString(2,jTextField2.getText());

pstmt.setString(3,jTextField3.getText());

```

```

pstmt.setString(4,jTextField4.getText());
pstmt.executeUpdate();
JOptionPane.showMessageDialog(null,"Saved successfully");
conn.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(null,e);
}
}

```

4.2.3 Delete

```

String sql = "delete from skills where skill_id = ?";
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setInt(1,Integer.parseInt(jTextField1.getText()));
pstmt.executeUpdate();
JOptionPane.showMessageDialog(null,"Record Deleted Successfully");
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
conn.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(null,e);
}
}

```

4.2.4 Update

```
String sql = "update skills set dance=?,classic_act=?,sing=? where skill_id=?";
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setString(1,jTextField2.getText());
pstmt.setString(2,jTextField3.getText());
pstmt.setString(3,jTextField4.getText());
pstmt.setInt(4,Integer.parseInt(jTextField1.getText()));

pstmt.executeUpdate();
JOptionPane.showMessageDialog(null,"Record Updated Successfully");
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
conn.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(null,e);
}
}
```

4.2.5 Search

```
String sql = "select * from skills where skill_id=?";
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setInt(1,Integer.parseInt(jTextField1.getText()));

ResultSet rs = pstmt.executeQuery();
if(rs.next())
{
jTextField2.setText(rs.getString("Dance"));
jTextField3.setText(rs.getString("classic_act"));
jTextField4.setText(rs.getString("sing"));
}
```

```

    }
else
{
JOptionPane.showMessageDialog(null,"Record not found");
}

conn.close();
}
catch(Exception e)
{
JOptionPane.showMessageDialog(null,e);
}
}

```

4.2.6 Tigger

```

DELIMITER //
create trigger lg_password
before insert on login
for each row
begin
if length (new.password)<4
then
signal sqlstate '45000'
set message_text = 'Password field is too short';
end if;
end;
//
DELIMITER;

```

4.2.7 Stored Procedure

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `addstudent`(  
IN STUDENTID int(30),  
IN NAME varchar(30),  
IN ADDRESS varchar(30))  
BEGIN  
INSERT INTO student(STUDENTID,NAME,ADDRESS)  
VALUES(STUDENTID,NAME,ADDRESS);  
END
```

CHAPTER 5

RESULT

4.6 Snapshots

4.6.1 Login page

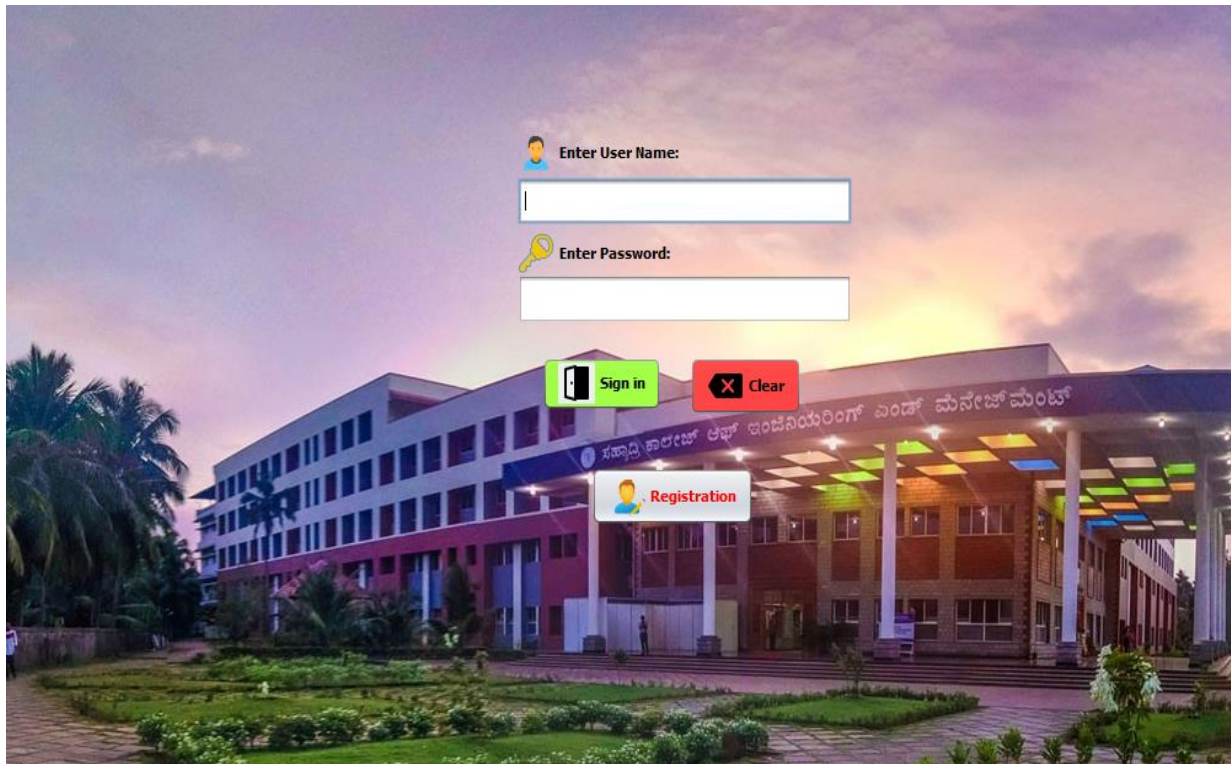


Fig 5.1

This page shows the login of the student using his credentials after which they can manipulate the data beign stored in database.

4.6.2 Registration page

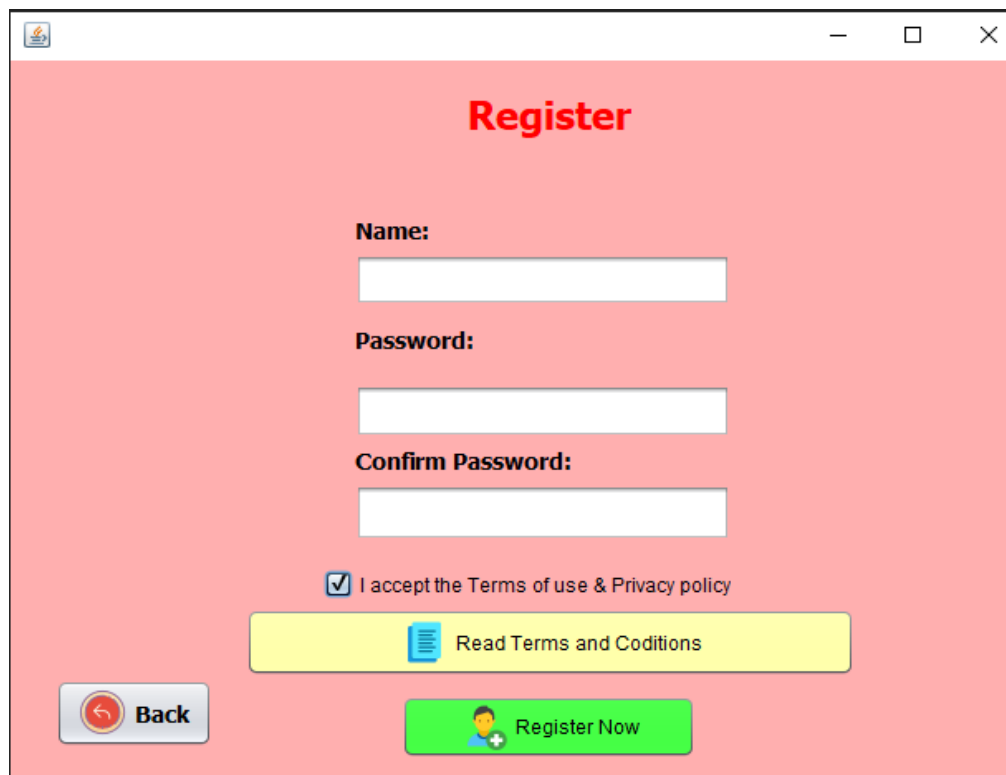


Fig 5.2

This page where student can register .Actor can manipulate the details who register.

4.6.3 Home page



Fig 5.3

This page shows the over view of the acting school, Actor can manipulate the details.

4.6.4 Student page

Student

student id

name

address

Save Delete Update

Display Clear

Back

Fig 5.4

This table shows the student s details. student details are storeed in database.

4.5.5 Student Skills

Student SKILLS

Student Skill Id

Dance

Acting

Singing

Student Id

SAVE DELETE UPDATE

Display Clear Search

Back

Fig 5.4

This page shows the details of student skills, student can alter the skill details.

4.6.5 Teacher

TEACHER

TechID

Skill Id

Phone

Teacher Name

SAVE Update Clear

Display

Back

Fig 5.5

This page shows the teacher details ,stored in the database.

4.6.6 Assign

Assign

Teacher Id

Skill Id

student id

Save Update Delete

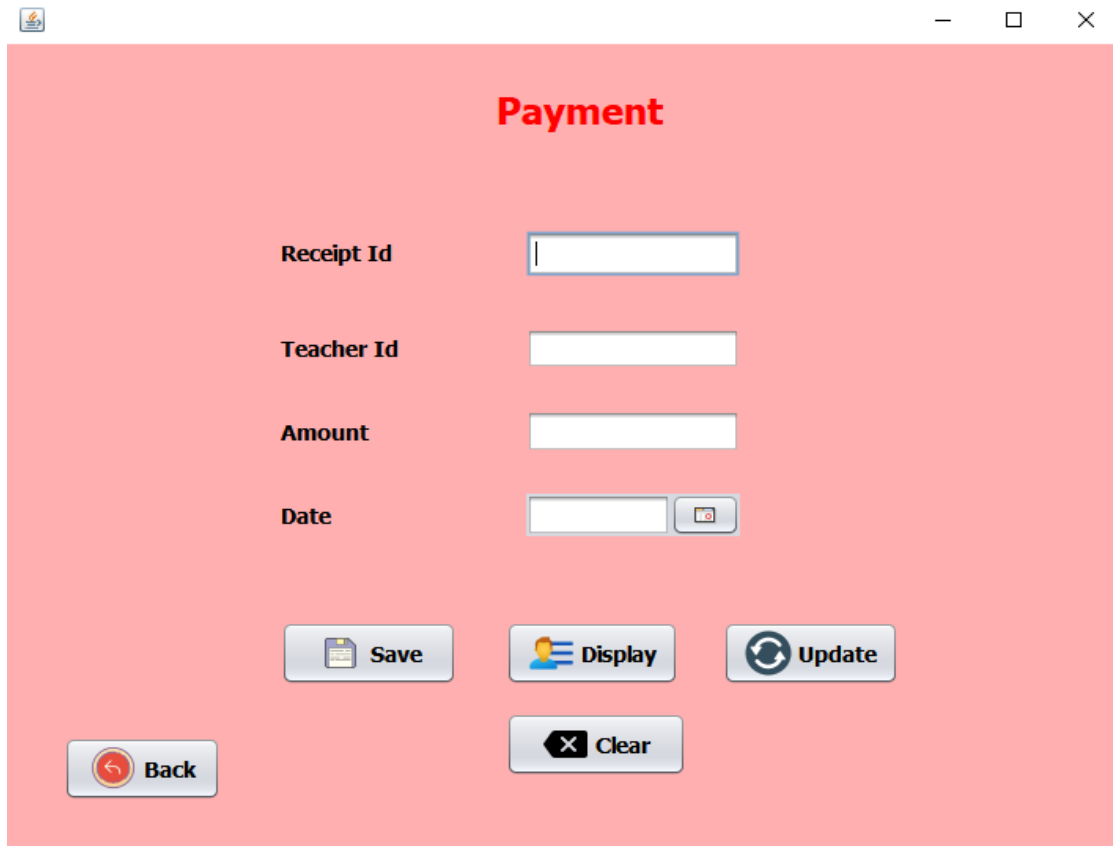
Display Clear

Back

Fig 5.6

This table shows the details of teacher who assign the work to student and can keep track of the teacher.

4.6.7 Payment



Payment

Receipt Id

Teacher Id

Amount

Date

Fig 5.7

This table shows the details of teacher who get the payment and actor can keep track of the teacher.

CONCLUSION

Acting school management system is an application which can be implemented easily to learn and teach the skill for the students. The main objective of the Acting school management system is to manage the teaching, students and skills. The application is user-friendly and anyone can be trained in a short span of time to master the usage of this application. Acting school Management System has been developed and automate the over-all processing of any skills details. The project offers user to enter the data through simple and interactive manner. Data storage and retrieval will become easier and faster to maintain. This project is a very flexible software.

REFERENCES

- www.youtube.com
- www.wikipedia.com
- <https://www.mysqltutorial.org>
- [Dbms textbook](#)
- [Sql study textbook](#)