A Mini Project Report

on

MARKS AND EXAM MANAGEMENT WITH INTERACTIVE CHATBOT (MEMIC)

Submitted by

SHETTY PRAJNESH SHIVANATH	VIKYATH K. NAIGA
4SN15CS082	4SN15CS097

COMPUTER SCIENCE & ENGINEERING

(Visvesvaraya Technological University)

Under the Guidance of

Mrs. Rashmi M.

Associate Professor

Department of Computer Science & Engineering

SRINIVAS INSTITUTE OF TECHNOLOGY

MANGALURU-574143, KARNATAKA

2017 – 2018

SRINIVAS INSTITUTE OF TECHNOLOGY
MANGALURU -574143, KARNATAKA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled ***"MARKS AND EXAM MANAGEMENT WITH INTERACTIVE CHATBOT (MEMIC)",*** *is an authentic record of the work carried out by*

SHETTY PRAJNESH SHIVANATH	**VIKYATH K. NAIGA**
4SN15CS082	**4SN15CS097**

*as prescribed by Visvesvaraya Technological University, Belagavi, for V Semester B.E. in Computer Science & Engineering during the year 2017-2018***.**

Mrs. Rashmi M.	**Prof. Shivakumar G.S.**	**Dr. Shrinivasa Mayya D.**
Project Guide	**Head of the Department**	**Principal**

Name of the Examiners **Signature with Date**

1.

2.

Acknowledgement

The satisfaction that accompanies the successful completion of any work would be incomplete without thanking the persons who made it perfect with their constant guidance and encouragement.

We take this opportunity to express our profound gratitude and deep regards to our Project guide and mentor**, Mrs. Rashmi M.,** Associate Professor, Department of Computer Science and Engineering**,** for her support and guidance. Her vision and suggestions throughout the project period has been fundamental in the completion of the project.

We are highly grateful and would like to express our whilehearted thankfulness to our Project Co-ordinator **Ms. Anusha Anchan,** Assistant Professor, Department of Computer Science and Engineering, for her encouragement and timely advice provided to us during the project work.

We extend our warm gratitude to **Prof. Shivakumar G. S.,** Head of the Department, Computer Science and Engineering, for his constant support and advice that helped us to complete this project successfully.

We are extremely grateful to our beloved **Principal, Dr. Shrinivasa Mayya D.** for his encouragement to come up with new ideas and advice to express them in a systematic manner.

We also like to thank all Teaching & Non-teaching staff of Department of Computer Science and Engineering, for their kind co-operation during the course of our work.

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

Shetty Prajnesh Shivanath
Vikyath K. Naiga

ABSTRACT

MEMIC, acronym for Marks and Exam Management with Interactive Chatbot, is a software which deals with the academic details of the students of the college. It incorporates the MySQL database which is the primary aim of this mini-project . The purpose of this project includes the insertion, updation and deletion of marks and attendance of the students as well as the adding and removing of teachers and subjects. It also facilitates the scheduling of exams and avoiding conflicts between exams of the same semester and branch. It ensures fail-proof functionalities required by the user.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 What is MEMIC?

A Chatbot is a program that communicates with a user. It is a layer on top of or a gateway to service. It is designed to simulate conversations with human users in a real-world environment.

MEMIC(Marks and Exam Management with Interactive Chatbot) is a software designed to help the students know about their academic details entered by the teachers in the college.

The primary aim of the software are as follows-

- Teachers can CRUD(Create, Read, Update and Delete) marks, exam schedules and attendance percentages.
- Students can query for the same details.
- Queries are handled by a user-friendly chatbot in a normal-interaction based chat.
- Using this chatbot, students can also know about the details of the teachers teaching subjects.

With the help of MEMIC, one will be able to glance at all the academic details of the current year without having an in-depth knowledge of the database. All you have to do is to login and check for the details. It also facilitates the registering of teachers for multiple subjects and for different classes.

It also handles the scheduling of exams for all the semesters and helps in solving conflicts between any two exams which are scheduled at the same time accidentally.

The purpose of MEMIC is to demonstrate the usage of MySQL database. It incorporates all the features of MySQL database, thereby allowing the software to store and retrieve information with ease.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Functional Requirements

- The homepage provides an interface with three options – Student, Teacher and Admin.
- Students must provide an authenticate USN and password to login to chatbot.
- The chatbot deals with all the queries posed by the students regarding their academic details.
- Teachers after logging in with their TID and password are confronted with two options - Marks and Attendance. A teacher can CRUD marks of the students in their respective subjects. Any attempt to CRUD marks for subjects not handled by the teacher is discouraged. Attendance details are dealt in the same way as the marks.
- Admin has the following choices – Teacher, Subject And Exam Schedules.
- In the option teacher, Admin can register new teachers, update their details and even remove them. Next is the subject where the admin has the authority can add or remove the subjects for a particular teacher.
- In the exam schedule options, the admin can enter the exam schedules, provided there is no conflict between the newly entered data and the previously scheduled exams.

2.2 Non-Functional Requirements

The non-functional requirements mainly comprise of the performance and the maintainability of the software. The chatbot (henceforth called as MEMIC) must be able to answer the queries sufficiently.

- Performance – The software must perform the functions smoothly and on time.
- Maintainability – All the functionalities of the software must be well-maintained.
- Security – No breaching of the login details.
- Reliability – The software must be the foolproof way of handling all the functionalities.
- Capacity – Can be able to handle the functionalities even on a large scale.

2.3 Hardware Requirements

- 2 GB RAM and above for smoother running.
- 2 GB Hard-disk space.
- i3 processor or higher versions.

2.4 Software Requirements

- Java (JDK version 8 and greater).
- Netbeans 8.2 IDE.
- MySQL database.
- AnimationClass.jar.

The project requires the library file of MYSQL JDBC Driver for ensuring the connection between the application and the database. For running the program, any basic PC running Ubuntu or windows XP (or higher) with compatible version of JDK is sufficient.

CHAPTER 3

SYSTEM DESIGN

3.1 ER Diagram

An Entity-Relationship Diagram is a data modelling technique that graphically illustrates an information system’s entities and the relationship between those entities. An ER diagram is a conceptual and representational model of data ussed to represent the entity framework infrastrucutre.

Figure 3.1 refers to the ER Diagram of the MEMIC project.

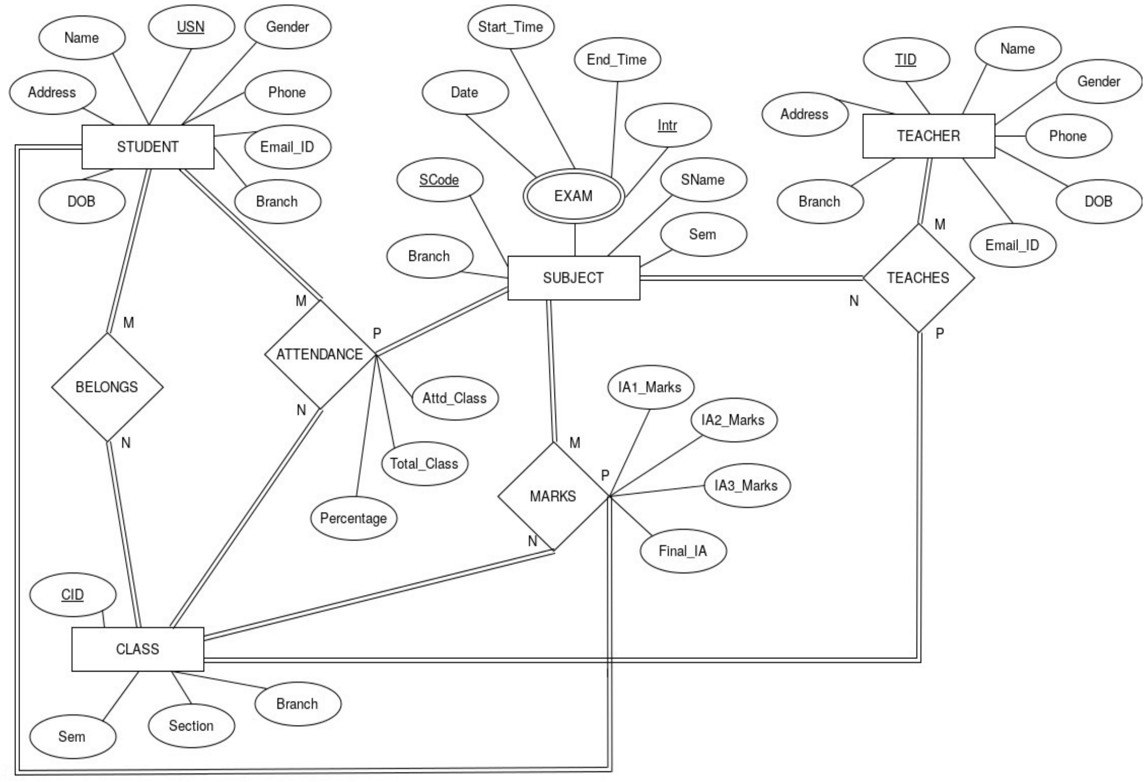


Figure 3.1 ER Diagram

3.2 ER to Relational Mapping

Step 1 : Mapping of Regular Entity Types

For each regular entity type, create a relation R that contains all the simple attributes of E.

With reference to Figure 3.1, the regular relations TEACHER, STUDENT, SUBJECT and CLASS are mapped.

TEACHER

TID	Name	Gender	Address	DOB	Phone	Email_ID	Branch

STUDENT

USN	Name	Gender	Address	DOB	Phone	Email_ID	Branch

SUBJECT

Scode	SName	Sem	Branch

CLASS

CID	Sem	Section	Branch

Step 2 : Mapping of Weak Entity Types

For each weak entity type, create a relation R that contains all the simple attributes of E in the ER Diagram

There is no weak entity in Figure 3.1

Step 3 : Mapping of Binary 1:1 Relationship Types

For each binary 1:1 relationship type, identify relation that correspond to entity types participating in R.

Possible approaches are

- Foreign Key approach
- Merged Relationship Approach
- Cross reference or relationship relation approach

There is no binary 1:1 relationship in Figure 3.1

Step 4 : Mapping of Binary 1:N Relationship Types

For each binary 1:N relationship type, identify relation that represents participating entity type at N-side of relationship type, include primary key of other entity type as foreign key in S, include simple attributes of 1:N relationship type as attributes of S to entity types participating in R.

Possible approaches are

- Foreign Key approach
- Merged Relationship Approach
- Cross reference or relationship relation approach

There is no binary 1:N relationship in Figure 3.1

Step 5 : Mapping of Binary M:N Relationship Types

For each binary M:N relationship type, create a new relation S, include primary key of participating entity types as foreign key attributes in S, include any simple attributes of M:N attributes.

In Figure 3.1, the relationships BELONGS and TEACHES are converted into relations.

BELONGS

USN	CID

TEACHES

TID	SCode	CID

Step 6 : Mapping of Multivalued Attributes

For each multivalued attributes, create a new relation, primary key of R is the combination of A and K, if the multivalued attributes is composite, include its simple components.

In Figure 3.1, the multivalued attribute EXAM is converted into a relation.

EXAM

Intr	SCode	Sem	Start_Time	End_Time	Date	Branch

Step 7 : Mapping of N-ary Relationship

For each n-ary relationship type R, create a new relation S to represent R, include primary keys of participating entity types as foreign keys, include any simple attributes as attributes.

In Figure 3.1, the n-ary relationships MARKS and ATTENDANCE are mapped into relations

MARKS

USN	SCode	CID	IA1_Marks	IA2_Marks	IA3_Marks	Final_IA

ATTENDANCE

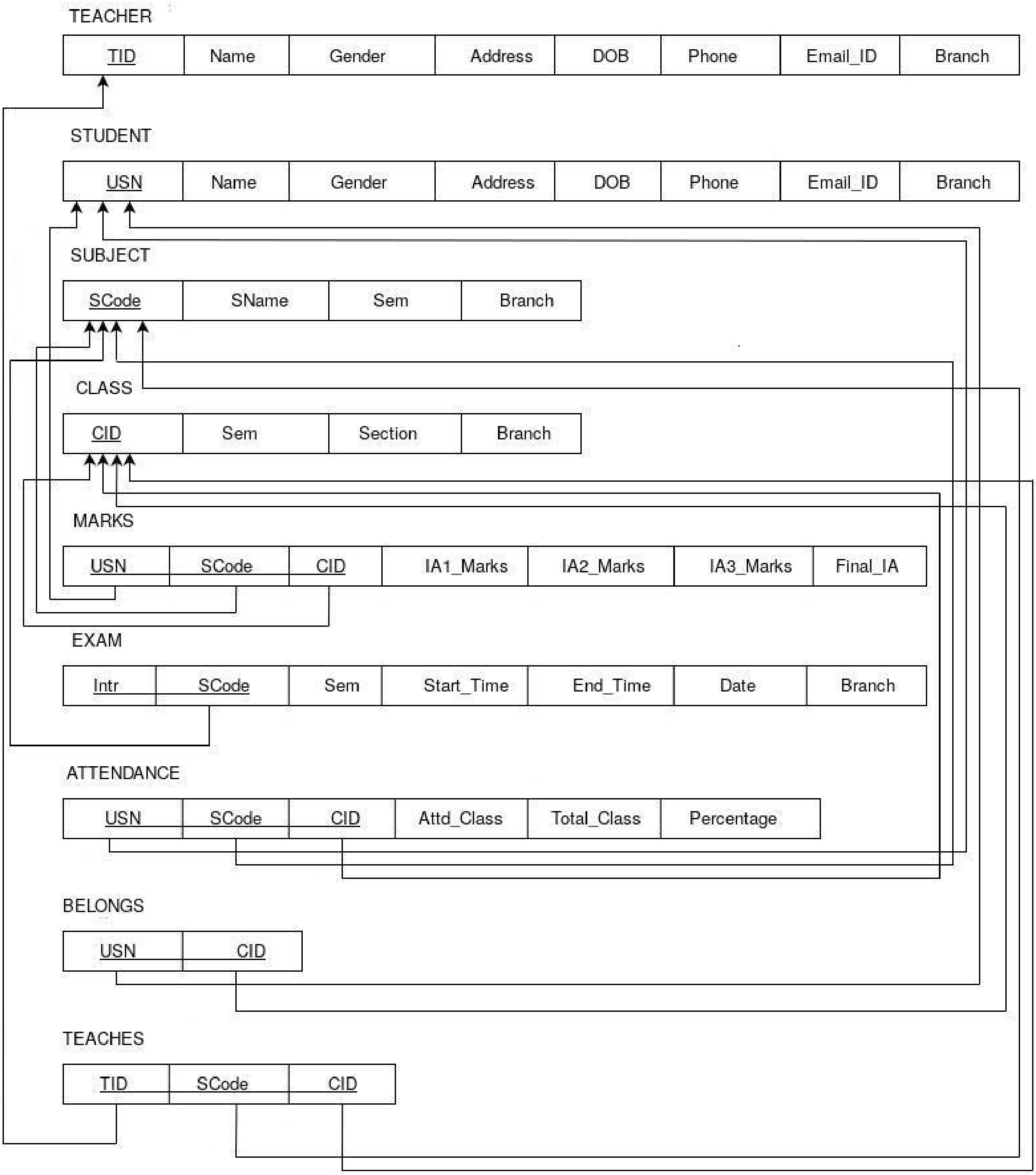
USN	SCode	CID	Attd_Class	Total_Class	Percentage

3.3 Relational Schema

A relational schema refers to the organization of data as a blueprint of how the database is constructed (divided into relation tables). In relational schema theory, a relation, as originally defined by E.F.Codd, is a set of tuples ($d_{1,}d_{2,}$……..$d_{n)},$ where each elements d_j is a member of $D_{j,}$ a data domain.

Codd’s original definition notwithstanding, and contrary to the usual definition in mathematics, there is no ordering to the elements of the tuples in a relation. Instead each element is termed an attribute value.

Figure 3.2 refers to the relational schema of the MEMIC project.



TEACHER

TID	Name	Gender	Address	DOB	Phone	Email_ID	Branch

STUDENT

USN	Name	Gender	Address	DOB	Phone	Email_ID	Branch

SUBJECT

SCode	SName	Sem	Branch

CLASS

CID	Sem	Section	Branch

MARKS

USN	SCode	CID	IA1_Marks	IA2_Marks	IA3_Marks	Final_IA

EXAM

Intr	SCode	Sem	Start_Time	End_Time	Date	Branch

ATTENDANCE

USN	SCode	CID	Attd_Class	Total_Class	Percentage

BELONGS

USN	CID

TEACHES

TID	SCode	CID

Figure 3.2 Relational Schema

3.4 Use Case Model

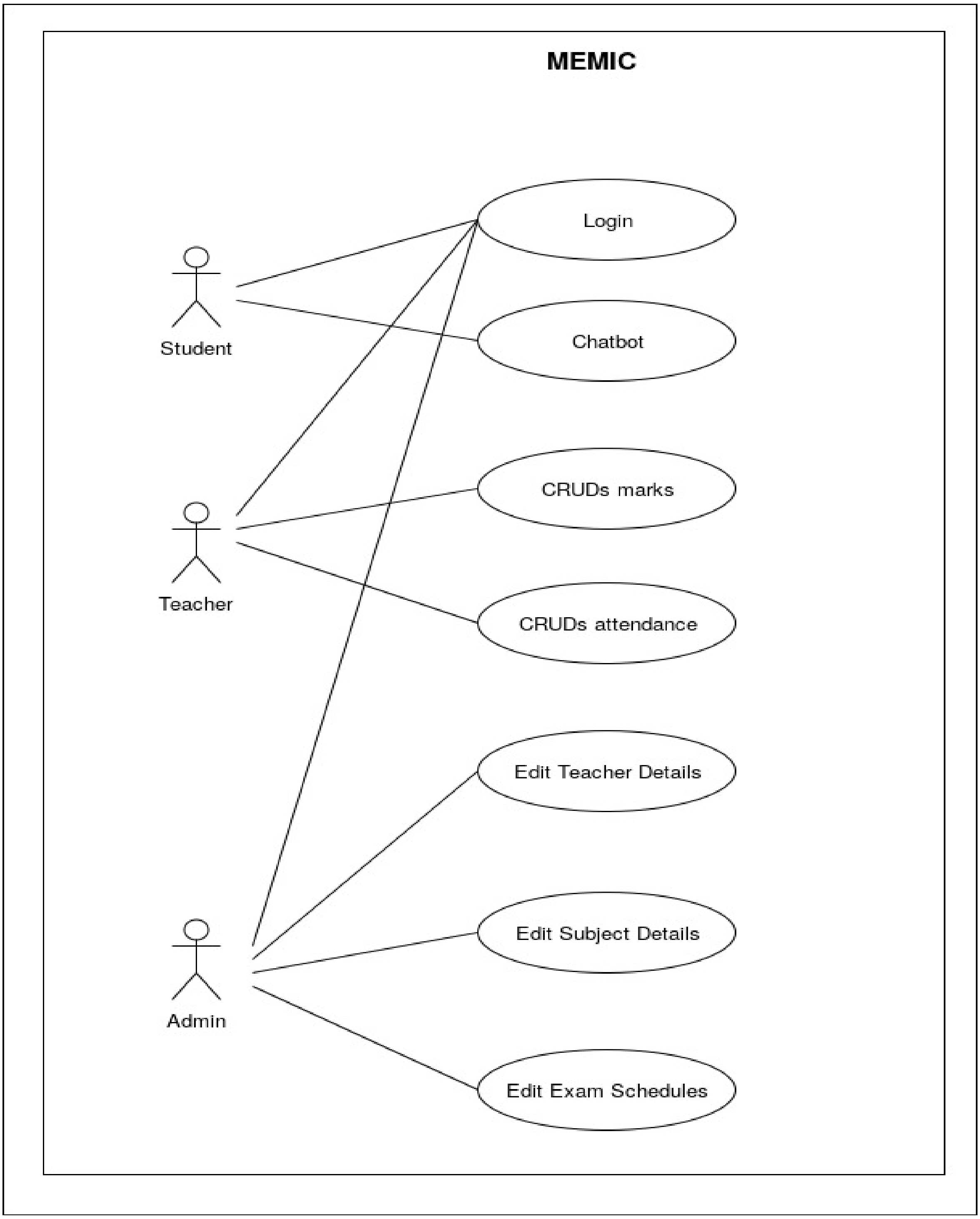


Figure 3.3 Use Case Model Diagram

Use Case – Login.

Summary – To login into the respective homepages.

Actors – Student, Teacher and Admin.

Description – The actor has to login into their respective homepages. For Student, the USN and Password is required, for Teacher, TID and Password is required and for the Admin, only the Password is required.

Exception – If the entered details are invalid, then entry to the homepage is restricted and a message is displayed.

Use Case – Chatbot.

Summary – To ask queries to the chatbot and receive answers to the queries.

Actors – Student.

Pre-Condition – The actor has to login with an authenticate USN and Password.

Description – The actor has to ask queries in a well-constructed language. The chatbot will interpret the queries, refer to the database and give specific and satisfactory answers to the actor.

Exception – If the queries are not in a well-defined langauge, then the chatbot won’t be able to handle the queries.

Use Case – CRUDs marks

Summary – To edit the marks details of the subjects taught by the Teacher.

Actors – Teacher.

Pre-Condition – The actor has to login with an authenticate TID and Password.

Description – The actor can enter, update or delete the marks details of the students of the subject which are handled by the actor.

Exception – If the actor tries to enter the marks of the subject which are not handled by him/her, then an appropriate message is displayed.

Use Case – CRUDs attendance

Summary – To edit the attendance details of the subjects taught by the Teacher.

Actors – Teacher.

Pre-Condition – The actor has to login with an authenticate TID and Password.

Description – The actor can enter, update or delete the attendance details of the students of the subject which are handled by the actor.

Exception – If the actor tries to enter the attendance of the subject which are not handled by him/her, then an appropriate message is displayed.

Use Case – Edit Teacher details

Summary – To edit the teacher details.

Actors – Admin.

Pre-Condition – The actor has to login with an authenticate Password.

Description – The actor can enter, update, display or delete the teacher details of the students of the subject which are handled by the actor.

Use Case – Edit Subject details

Summary – To edit the subject details of a teacher.

Actors – Admin.

Pre-Condition – The actor has to login with an authenticate Password.

Description – The actor can enter, update, display or delete the details of the the subject which are handled by a particular teacher.

Use Case – Edit Exam Schedules

Summary – To edit the exam schedules of all the subjects.

Actors – Admin.

Pre-Condition – The actor has to login with an authenticate Password.

Description – The actor can enter, update, display or delete the exam schedules of the students of all the subjects.

Exception – If the actor tries to enter the exam schedule which conflicts with an exam of the same semester, branch, date and time, then the entry is restricted and an appropriate message is shown.

CHAPTER 4

IMPLEMENTATION

4.1 Description of Relations

Table 4.1 TEACHER Table

Field	Type	Null	Key	Default
TID	varchar(15)	NO	PRI	
Name	varchar(40)	YES		NULL
Gender	varchar(2)	YES		NULL
Address	varchar(50)	YES		NULL
DOB	date	YES		NULL
Phone	mediumtext	YES		NULL
Email_ID	varchar(40)	YES		NULL
Branch	varchar(10)	YES		NULL

Table 4.1 refers to the relation table of TEACHER with TID as its primary key.

Table 4.2 STUDENT Table

Field	Type	Null	Key	Default
USN	varchar(15)	NO	PRI	
Name	varchar(40)	YES		NULL
Gender	varchar(2)	YES		NULL
Address	varchar(50)	YES		NULL
DOB	date	YES		NULL
Phone	mediumtext	YES		NULL
Email_ID	varchar(40)	YES		NULL
Branch	varchar(10)	YES		NULL

Table 4.2 refers to the relation table of STUDENT with USN as its primary key.

Table 4.3 SUBJECT Table

Field	**Type**	**Null**	**Key**	**Default**
SCode	varchar(15)	NO	PRI	
SName	varchar(40)	YES		NULL
Sem	int(11)	YES		NULL
Branch	varchar(10)	YES		NULL

Table 4.3 refers to the relation table of SUBJECT with SCode as its primary key.

Table 4.4 CLASS Table

Field	**Type**	**Null**	**Key**	**Default**
CID	varchar(10)	NO	PRI	
Sem	int(11)	YES		NULL
Section	varchar(2)	YES		NULL
Branch	varchar(10)	YES		NULL

Table 4.4 refers to the relation table of CLASS with CID as its primary key.

Table 4.5 MARKS Table

Field	**Type**	**Null**	**Key**	**Default**
USN	varchar(15)	NO	PRI	
SCode	varchar(15)	NO	PRI	
CID	varchar(10)	NO	PRI	
IA1_Marks	int(11)	YES		NULL
IA2_Marks	int(11)	YES		NULL
IA3_Marks	int(11)	YES		NULL
Final_IA	float	YES		NULL

Table 4.5 refers to the relation table of MARKS with {USN,SCode,CID} as its primary key. The attribute USN refers to the USN of STUDENT, SCode refers to the SCode of SUBJECT and CID refers to the CID of CLASS.

Table 4.6 EXAM Table

Field	**Type**	**Null**	**Key**	**Default**
Intr	int(11)	NO	PRI	
SCode	varchar(15)	NO	PRI	
Sem	int(11)	YES		NULL
Start_Time	varchar(15)	YES		NULL
End_Time	varchar(15)	YES		NULL
Date	date	YES		NULL
Branch	varchar(10)	YES		NULL

Table 4.6 refers to the relation table of EXAM with {Intr,SCode} as its primary key. The attribute SCode refers to the SCode of SUBJECT.

Table 4.7 ATTENDANCE Table

Field	**Type**	**Null**	**Key**	**Default**
USN	varchar(15)	NO	PRI	
SCode	varchar(15)	NO	PRI	
CID	varchar(10)	NO	PRI	
Attd_Class	int(11)	YES		NULL
Total_Class	int(11)	YES		NULL
Percentage	float	YES		NULL

Table 4.7 refers to the relation table of ATTENDANCE with {USN,SCode,CID} as its primary key. The attribute USN refers to the USN of STUDENT, SCode refers to the SCode of SUBJECT and CID refers to the CID of CLASS.

Table 4.8 BELONGS Table

Field	**Type**	**Null**	**Key**	**Default**
USN	varchar(15)	NO	PRI	
CID	varchar(10)	NO	PRI	

Table 4.8 refers to the relation table of BELONGS with {USN,CID} as its primary key. The attribute USN refers to the USN of STUDENT and CID refers to the CID of CLASS.

Table 4.9 TEACHES Table

Field	Type	Null	Key	Default
TID	varchar(15)	NO	PRI	
SCode	varchar(15)	NO	PRI	
CID	varchar(10)	NO	PRI	

Table 4.9 refers to the the relation table of TEACHES with {TID,SCode,CID} as its primary key. The attribute TID refers to the TID of TEACHER, SCode refers to the SCode of SUBJECT and CID refers to the CID of CLASS.

4.2 Stored Procedures

A stored procedure is a subroutine available to applications that access a relational database management system. Such procedures are stored in the database dictionary.

```
DELIMITER //
CREATE PROCEDURE AverageM()
BEGIN
UPDATE EXAMS
SET
Final_IA=GREATEST(IA1_Marks+IA2_Marks,IA3_Marks+IA2_Marks,IA1_Marks+IA3_
Marks)/2;
END//  DELIMITER ;
```

The above stored procedure AverageM() calculates the average of best of two of the three internal tests in EXAM.

```
DELIMITER //
CREATE PROCEDURE PercA()
BEGIN
UPDATE ATTENDANCE SET Percentage=(Attd_Class/Total_Class)*100;
END//
```

```
DELIMITER ;
```

The above stored procedure PercA() calculates the percentage of the classes attended by a student in ATTENDANCE.

4.3 Trigger

A trigger is a special kind of stored procedure that automatically executes when an event occurs in the database server. Triggers are executed when a user tries to modify data through a data manipulation language (DML) event. The events are INSERT , UPDATE or DELETE statements on a table or view.

```
DELIMITER //
CREATE TRIGGER Trial
BEFORE INSERT
ON EXAM
FOR EACH ROW BEGIN
DECLARE c INT;
SELECT COUNT(*) INTO c FROM EXAM WHERE Intr=NEW.Intr AND Sem=NEW.Sem
AND Start_Time=NEW.Start_Time AND Date=NEW.Date AND Branch=NEW.Branch;
IF c=1 THEN
SET NEW.Intr='0';
END IF;
END //
DELIMITER ;
```

TRIGGER Trail gets executed when the user tries to enter an exam schedule which gets conflicted with the previously scheduled exam. It sets the Intr as ‘0’ if there arises a conflict.

4.4 JDBC Connection

Java Database Connectivity (JDBC) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is Java based data access technology and used for Java database connectivity. It is part of the Java

Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database, and is oriented towards relational databases.

```
package miniproject;
import java.sql.*;
public class Conn11 {
        public static void DriverTest() {
                 try {
                         Class.forName("com.mysql.jdbc.Driver");
                         System.out.println("Driver tested!");
                 }
                catch (java.lang.ClassNotFoundException e) {
                         System.out.println("Class Not Found Exception!");
                 }
        }
}
```

```
package miniproject;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class Conn12 {
        public static void main(String[] args) {
                if (Conn12.getConnection().equals(null))
                        System.out.println("Failed to connect !!");
                else
                        System.out.println("Successfully connected !!");
        }
        public static Connection getConnection() {
        Connection con = null;
        try {
```

```
                    Conn11.DriverTest();
                    con=DriverManager.getConnection("jdbc:mysql://127.0.0.1:3306/database-
                    name,user,password);
                     System.out.println("Connection established");
          }
          catch (SQLException ex) {
                    System.out.println("Connection failed "+ex);
                    con = null;
          }
          return (con);
          }
}
```

Connection Statements

```
          Statement stmt;
          Connection conn;
          conn = Conn12.getConnection();
          try {
                    stmt = conn.createStatement();
                    stmt.executeUpdate(SQL QUERY) ;
                    conn.close();
          }
          catch (SQLException sqlEx) {
          }
   }
```

4.5 System Architecture

Student Module

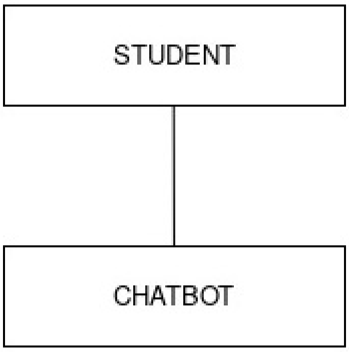


Figure 4.1 Student Module Diagram

Student Module has only one sub-module – Chatbot as shown in Figure 4.1. The Chatbot accept the queries entered by the student and answers the queries.

Teacher Module

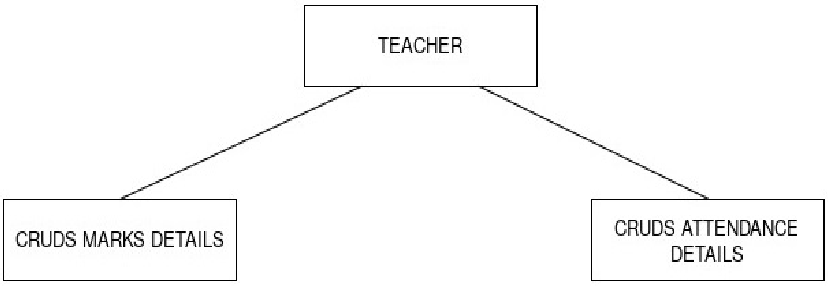


Figure 4.2 Teacher Module Diagram

Teacher Module has two sub-modules – Marks and Attendance, as shown in Figure 4.2. In Marks, the teacher can edit the marks details and in Attendance, the teacher can edit the attendance details.

Admin Module

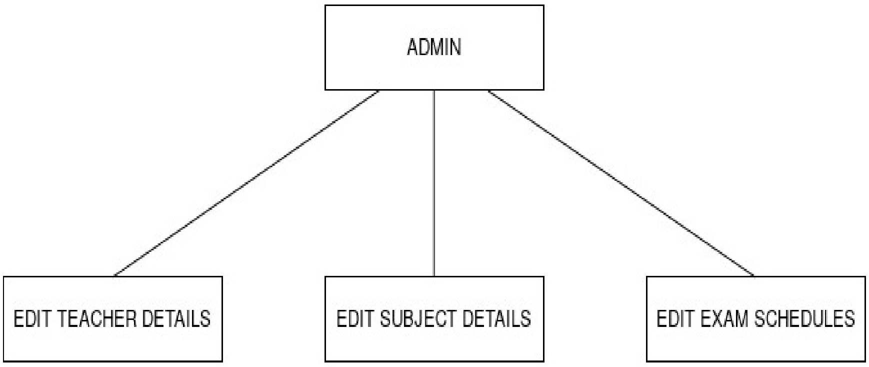


Figure 4.3 Admin Module Diagram

Admin Module has three sub-modules – Teacher, Subject and Exam Schedules as shown in Figure 4.3. In Teacher, the admin can edit the teacher details, in Subject, the admin adds the subjects to be handled and in Exam Schedules, the edition, deletion and updation of exam Schedules is managed.

4.6 Data Flow Model

A Data Flow Model is a diagrammatic representation of the flow and exchange of information within a system. Data flow model are used to graphically represent the flow of data in an information system by describing the processes involved in transferring data from input to file storage and reports generation.

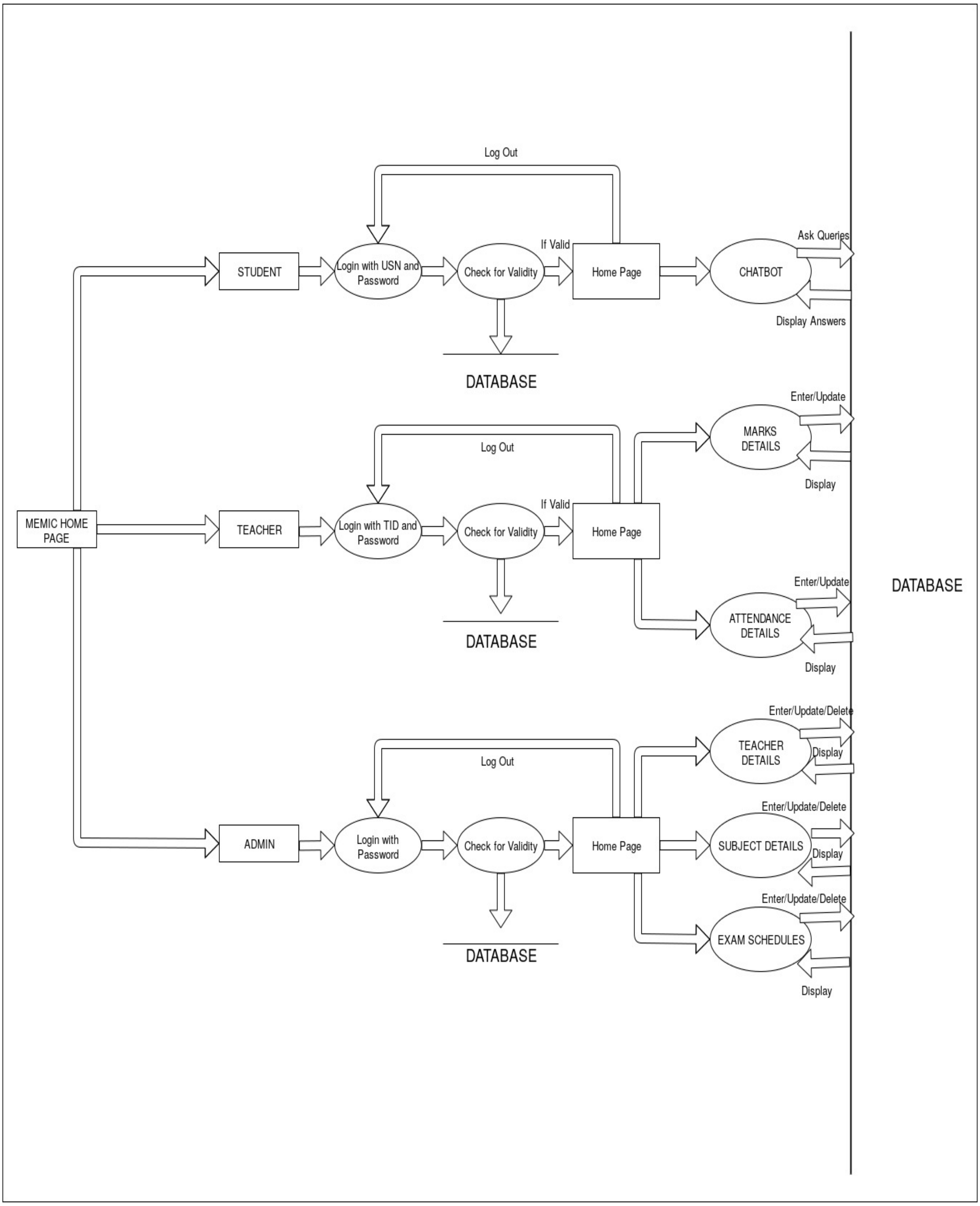


Figure 4.4 Data Flow Model

Figure 4.4 shows the data flow model of the MEMIC project.

Normal interactions with the database as well as interactions with different modules are shown.

CHAPTER 5

SCREEN SHOTS

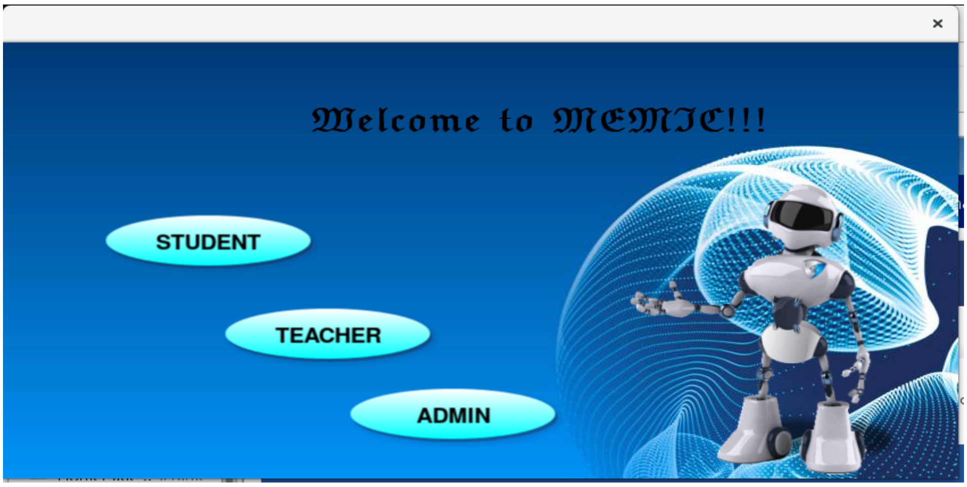
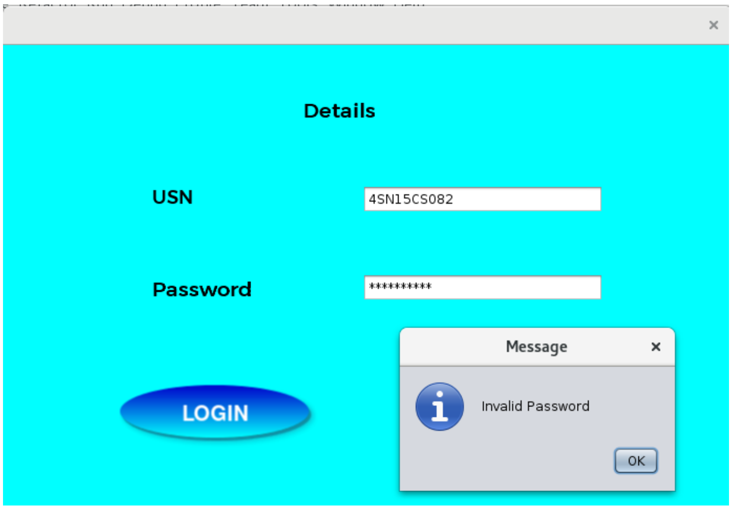


Figure 5.1 Home Page Screen

Figure 5.1 shows the home page of MEMIC. It provides an interface with three options – Student, Teacher and Admin.



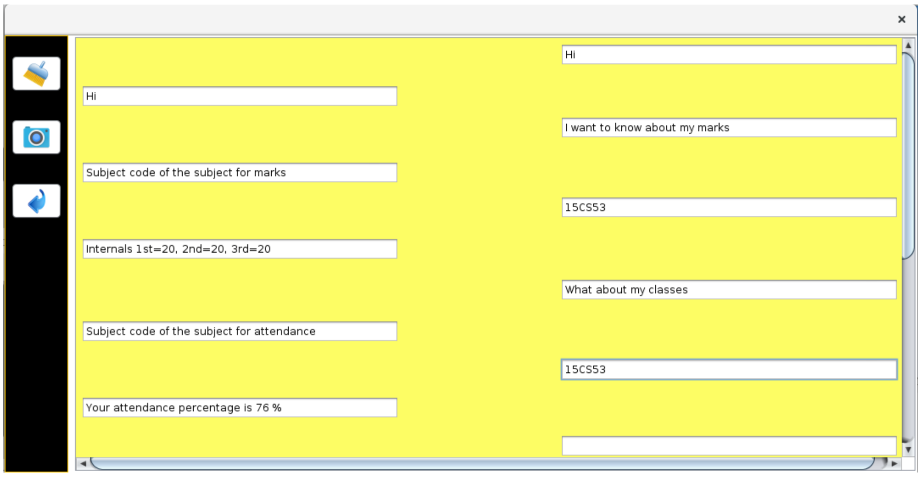
USN 4SN15CS082

Password **********

Invalid Password

Figure 5.2 Login Page For Student

Figure 5.2 shows the screen of Login Page for Student. The student has to enter the authenticate USN and Password to enter to the next screen. If the password is not authenticate then a message is displayed. Similarly, the teacher can login with TID and Password and Admin with Password.



Hi

Hi

I want to know about my marks

Subject code of the subject for marks

15CS53

Internals 1st=20, 2nd=20, 3rd=20

What about my classes

Subject code of the subject for attendance

15CS53

Your attendance percentage is 76 %

Figure 5.3: MEMIC Chatbot Screen

Figure 5.3 is the representation of the interface for MEMIC Chatbot where the student must ask for queries to the chatbot. The student can ask for marks, attendance percentage, exam schedules and lecturers, provided the questions are well constructed and easy to understand. In addition to the answer for the queries, the interface also gives the option for screenshot.

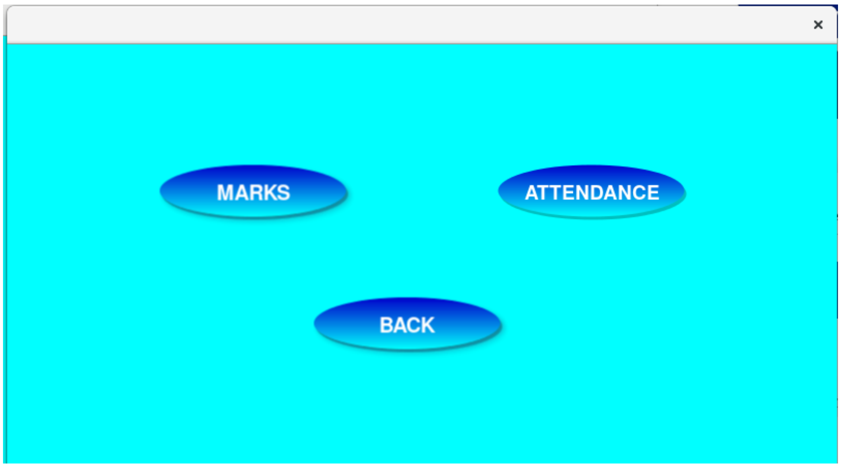
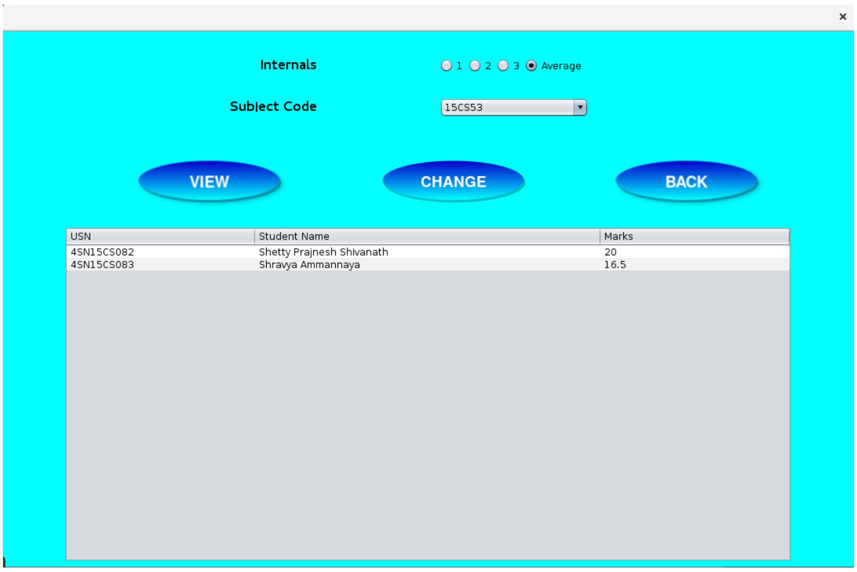


Figure 5.4 MEMIC Options Interface for Teacher

Figure 5.4 refers to the home page of Teacher which provides two options for Teacher – Marks and Attendance.



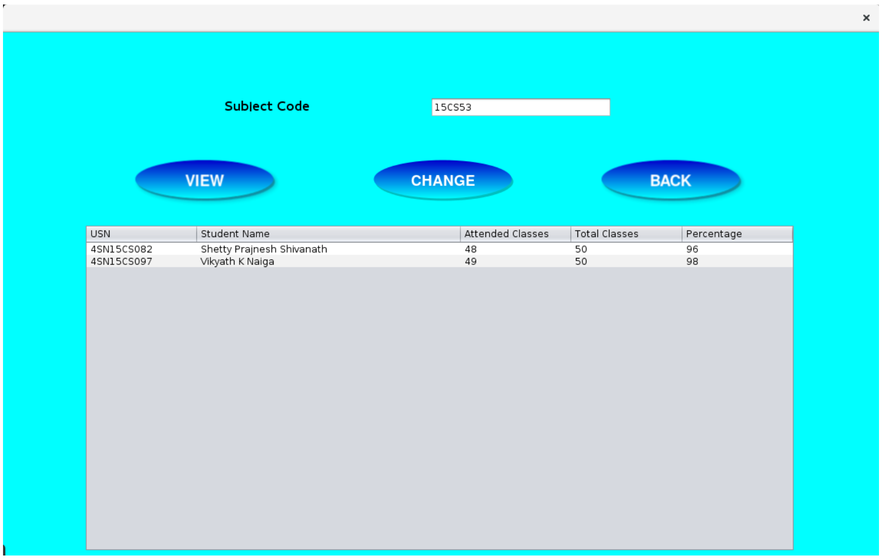
Internals ○ 1 ○ 2 ○ 3 ● Average

Subject Code 15CS53

USN	Student Name	Marks
4SN15CS082	Shetty Prajnesh Shivanath	20
4SN15CS083	Shravya Ammannaya	16.5

Figure 5.5 Marks Table Screen

Figure 5.5 refers to the screen where the teacher can enter, update or delete marks for each student. The teacher can enter the marks only for the subjects handled.

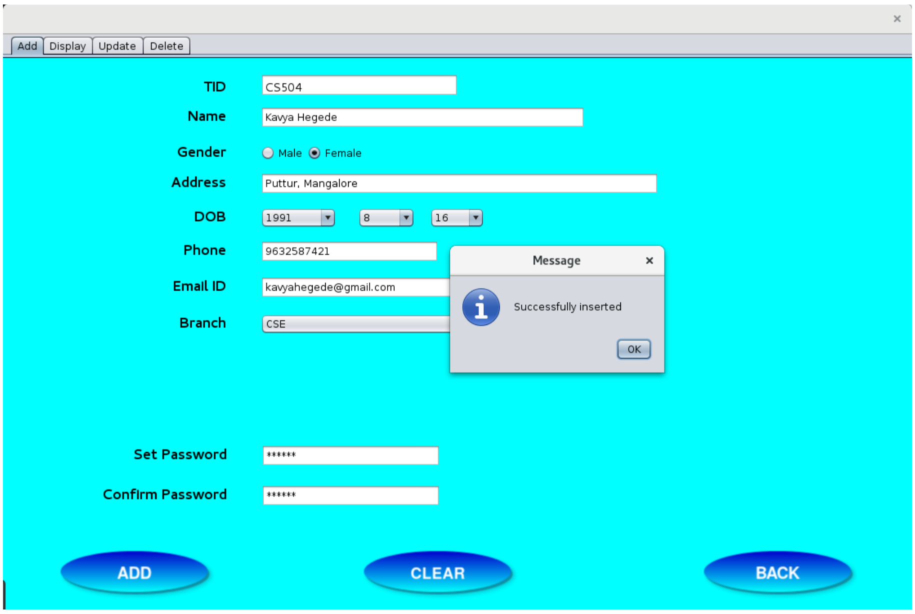


Subject Code 15CS53

USN	Student Name	Attended Classes	Total Classes	Percentage
4SN15CS082	Shetty Prajnesh Shivanath	48	50	96
4SN15CS097	Vikyath K Naiga	49	50	98

Figure 5.6 Attendance Table Screen

Figure 5.6 refers to the screen where the teacher can enter, update or delete attendance for each student. The teacher can enter the attendance only for the subjects handled.



TID	CS504
Name	Kavya Hegede
Gender	○ Male ● Female
Address	Puttur, Mangalore
DOB	1991 8 16
Phone	9632587421
Email ID	kavyahegede@gmail.com
Branch	CSE
Set Password	******
Confirm Password	******

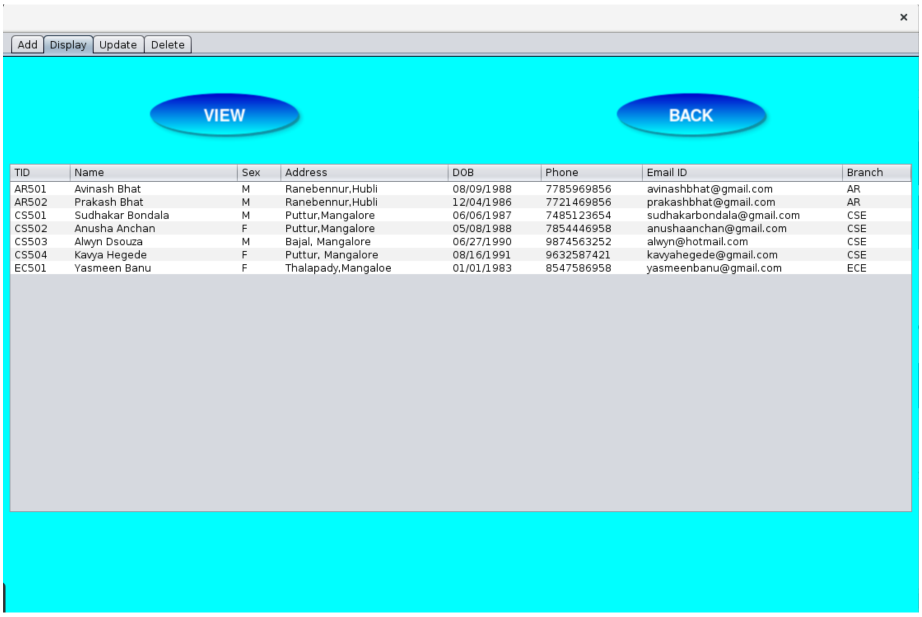
Message

Successfully inserted

OK

Figure 5.7 Adding of Teachers

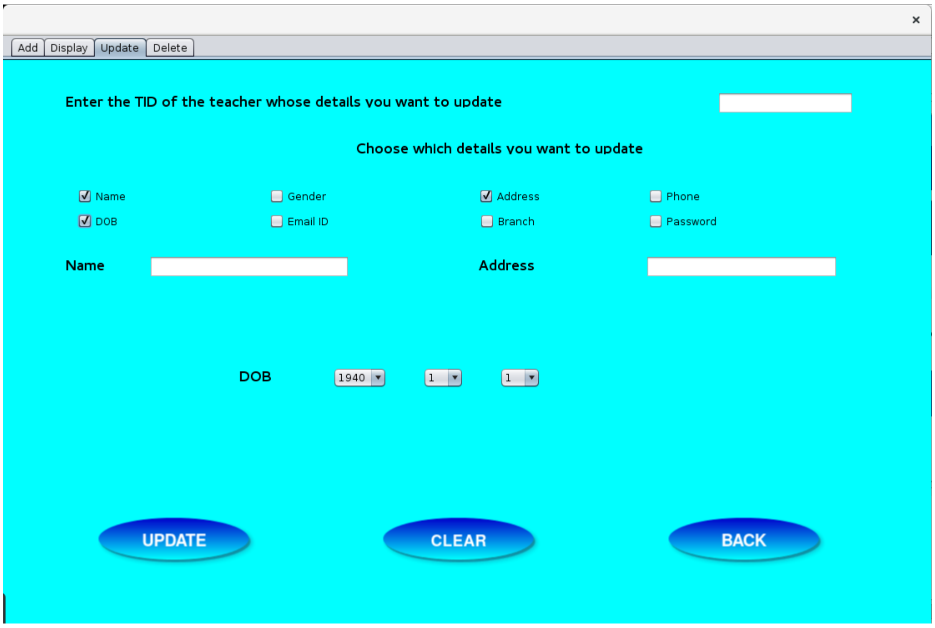
Figure 5.7 shows the screen where the Admin has the option to new register teachers into the database.



TID	Name	Sex	Address	DOB	Phone	Email ID	Branch
AR501	Avinash Bhat	M	Ranebennur,Hubli	08/09/1988	7785969856	avinashbhat@gmail.com	AR
AR502	Prakash Bhat	M	Ranebennur,Hubli	12/04/1986	7721469856	prakashbhat@gmail.com	AR
CS501	Sudhakar Bondala	M	Puttur,Mangalore	06/06/1987	7485123654	sudhakarbondala@gmail.com	CSE
CS502	Anusha Anchan	F	Puttur,Mangalore	05/08/1988	7854446958	anushaanchan@gmail.com	CSE
CS503	Alwyn Dsouza	M	Bajal, Mangalore	06/27/1990	9874563252	alwyn@hotmail.com	CSE
CS504	Kavya Hegede	F	Puttur, Mangalore	08/16/1991	9632587421	kavyahegede@gmail.com	CSE
EC501	Yasmeen Banu	F	Thalapady,Mangaloe	01/01/1983	8547586958	yasmeenbanu@gmail.com	ECE

Figure 5.8 Display of Teachers' List

Figure 5.8 is the screen for display of list of the teachers.



Enter the TID of the teacher whose details you want to update

Choose which details you want to update

☑ Name ☐ Gender ☑ Address ☐ Phone
☑ DOB ☐ Email ID ☐ Branch ☐ Password

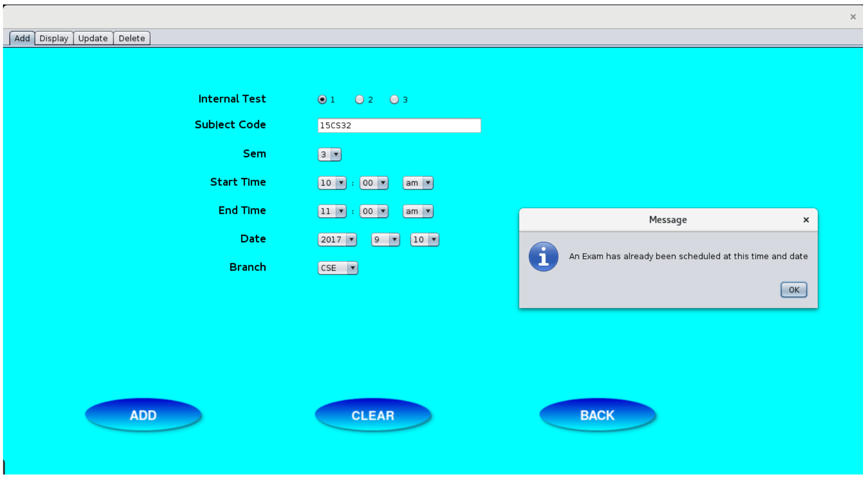
Name

Address

DOB 1940 1 1

Figure 5.9 Updation of Teacher’s Screen

Figure 5.9 is the screen where the updation of the details of the teachers take place. Similarly the updation of exam schedules and subjects are done.



Internal Test ● 1 ○ 2 ○ 3
Subject Code 15CS32
Sem 3
Start Time 10 : 00 am
End Time 11 : 00 am
Date 2017 9 10
Branch CSE

Message
An Exam has already been scheduled at this time and date

Figure 5.10 Adding Exam Schedules

Figure 5.10 is the screen where the admin enters the exam schedules. If an exam schedule is entered whose Internal, Semester, Start Time, Date and Branch coincides with the exam entered in the database, then it shows a Message Box regarding the conflict.

CONCLUSION AND SCOPE FOR FUTURE WORK

MEMIC plays a great role in accessing and storing of academic details without any interruption or difficulty. It is produced in such a way that it can handle multiple users simultaneously. The primary scope of this software, the MEMIC chatbot covers all the important aspects of the information a student needs to know.

Gone are the days where the user had to memorize every command for retrieving data. MEMIC stands as a proof of the demand for an advanced, progressive user interface. The addition of few more functionalities such as a defined scheduling of the exams and increasing the scope of the chatbot can lead to a more definitive and refined software with excellent performance and stability and with minimum glitches.

The progression of the 21st century calls for the upgrades and advancement of MEMIC. With proper implementation and bringing in Web Implementation, MEMIC can handle all the functionalities with increasing and productive output.

Bringing in advance features like making the chatbot understand natural language and answer the queries in the same manner, can open new vistas for the future of MEMIC.

MEMIC stands as a platform for the college for successful interpretation of the database system and its usage.

BIBLIOGRAPHY

[1] Database Systems Models, Languages, Design And Application Programming, Ramez Elmasri and Shamkant B. Navathe, 6th Edition, Pearson

[2] Netbeans: The Definitive Guide, 1st Edition, Tim Boudreau, Jesse Glick, Simeon Lewis

[3] https://stackoverfow.com

[4] Teknik Industries (for AnimationClass – www.teknikindustries.com)

[5] MySQL Tutorials by Nancy Steggart