

ent Student Database Management System

dream thE betteR futurE

Student Database Management System Technical Reference

Project By,
Abdullah Bin Rubiyath
032 036 044

Hossain M Reza Khan
032 034 040

M Iftekher Chowdhury
032 263 040

TABLE OF CONTENTS

1. Description Of Present System.....	03
2. Wok Flow Diagram Of Present System	05
3. Data Requirements	06
4. Technical Overview	07
4.1 Security	07
4.2 Scalability	07
4.3 Development Paradigm	07
5. Conventions	08
5.1 File Name Selection	08
5.2 Variable Name Selection	08
6. Design.....	09
6.1 Entity Set Designation	09
6.2 Relationship Set Designation.....	11
6.3 Entity Relationship Diagram (ERD).....	12
6.4 Reduced Table from ERD	13
6.5 Relational Schema	15
6.6 Schema Diagram	17
7. Appendix A: Database Schema	18
8. Appendix B: Limitations	24
9. Appendix C: Next Version Feature	25
10. Appendix D: Programs List	26

1. DESCRIPTION OF PRESENT SYSTEM

When a student enrolls/admits in North South University, his/her personal information including previous academic background is entered into the database. When a semester starts, advisor advises courses and at the time of advising, the courses that the student is going to take are inputted directly into the database. During the semester, a student's academic information (midterm marks, quiz marks, etc) is maintained in a file system. It is the job of the Teaching Assistant to maintain these marks and for this purpose spreadsheet software like Ms Excel is used. At the end of the semester, when all the grades are posted and finalized by the faculty, the administrator enters them into the database.

The academic and personal information of a student is stored in the database. A student does not have accessibility to the main Database. Students are therefore unable to view their personal information. However, students can check their CGPA, grades. In the case, where students needs to change their phone, address they have to submit an application to the university stating their new information. A student may be enrolled in several club activities but information regarding their club activities is not stored in the database.

The information regarding a faculty is also stored in the database. Faculties do have accessibility to the main Database. But, their accessibility is restricted to the view level and they are not allowed to update any information. It is only during the advising session, they can add or drop courses for a particular student.

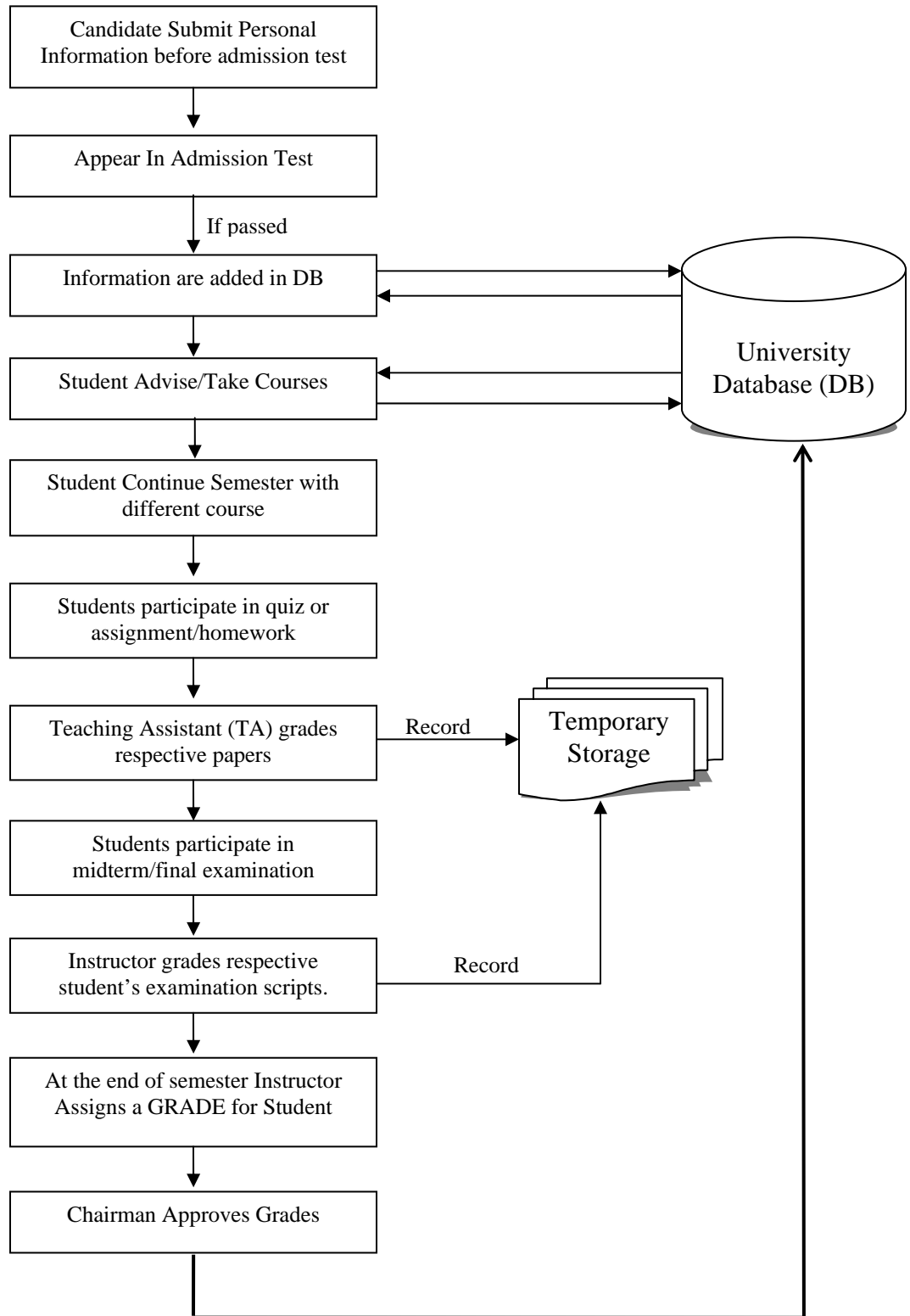
The Database does not contain any information about the Teaching Assistants. The information of the Teaching assistants is handled using file management system and on a semester-by-semester basis. Meaning, the information of the currently enrolled Teaching Assistants is stored in the database and upon start of a new semester all the previous record of the Teaching Assistants are deleted.

There is no web connectivity with the main Database. The only way to access the database is from inside the NSU premises.

Problems with the existing system:

The current system does not provide any access for the students to the main database. So, if the database contains any wrong information about a student then the student may not be aware of it. Also, since quiz and mid-term marks of a semester are maintained in a file, it is not possible for the students to get information directly. In most cases, it is often seen the total marks is only posted just before the final exam. As a result, if there is any error in calculation of marks, then the student has to go through certain hassle to correct his marks. If there are any new announcements (like class cancellation, change in mid-term date) then it is not possible for the students to know beforehand. The involvement of a student in club activities is a very important part, but it is not stored in the database.

2. WORK FLOW DIAGRAM OF PRESENT SYSTEM



3. DATA REQUIREMENTS

- **Students** are identified by their unique *student-id*. Student logs into the system via web or any client pc connected with the main database and can view his personal information and also can request for update if there is any change of address or phone number of that respective student. The student can check his/her current semester's course that he/she is running. A student can also check his/her quiz/ assignment/ homework/ midterm exam/final exam marks for a specific course of current semester. He/she can also read important notices posted by their Instructor/Teaching Assistant (TA). A student can be involved in different clubs, so he/she can see different activities of club and in addition if the student is the president of that club, then he/she will be able to manage members and post/upgrade different notice.
- An **Instructor** is identified by his/her unique *initial*. After logging into the system, he/she can view his/her personal information and also can request for update if there is any change of address or phone number. He/she can view the list of involved courses of current semester. For a specific course the instructor can view the enrolled student-id and as well as their progress (different marks) in that course. An instructor can post any special notice to the students via the system. After each midterm/final exams an instructor can post/update the marks of the students through the system. At the end of the semester an instructor can assign the grades of the students.
- A **Teaching Assistant(TA)** is identified by his *name* and *email* . A teaching assistant can also view his/her personal information and can request for update if there is any change of address or phone number. A teaching assistant can view the list courses of current semester where he/she is acting as TA. For a specific course the instructor can view the enrolled student-id and as well as their progress (different marks) in that course. TA can post any special notice to the students via the system. After each quiz/assignment/home work checking TA can post/update the marks of the students through the system.

4. TECHNICAL OVERVIEW

4.1 Security

The software is very secured in terms of out side threat or inside threat such as viruses. As long as it is managed within a secured container, it is safe. Being a web application it uses HTTP port to access the system. For the user all it is needed is a web browser.

4.2 Scalability

This system can be highly scalable depending on the server specific configuration. The software is anyway managed by web container with access to a database.

4.3 Development Paradigm

The system uses de facto standards in terms of development language and persistence. It uses JAVA and SQL the whole way.

HTML and Applets are used to make the system more managed in terms of consistency of the outlook and user interface. It is possible to change the entire outlook by just changing a single style sheet file at any moment.

5. CONVENTIONS USED

In order to keep the integrity and to make the maintainability process easier, this system uses few conventions in terms of file name selection and variable name selection in programs and database schema.

5.1 File Name Selection

Usually file names are selected considering real life entity. For example

LoginScreen.java

StudentScreen.java

AdminScreen.java

Login.JSP

Status.JSP

Message.JSP

The file's name structure is constructed as [operation][entity][operation-description].[JSP/java]. This convention helps to locate any file easily if the operation it is performing and the entity it is related to is known.

5.2 Variable Name Selection

This part is very important in order to make the codes maintainable. All the codes should be readable to any other developer/programmer who would be maintaining the codes or would be working for further extension

The system keeps consistency about the variable's names within database and JAVA and JSP codes. It increases readability of the program. Obviously it uses proper naming for the entity attributes of each of the entities within the database.

6. DESIGN

The description of each of the functions will be provided here. For that we will go through the Entity set designation, Relationship set designation, Entity relationship diagram, Reduced tables, Relational schema, Schema diagram, System structure diagram, Flow diagram of the proposed system.

6.1 Entity Set Designation

We have found out five strong entity sets and one weak entity set. Strong entity sets are

- a. Student
- b. Instructor
- c. Teaching Assistant
- d. Subject
- e. Club

Weak entity set is

- a. Section

Descriptions of the entity sets are given bellow:

a. **Student :**

Student is a strong entity set with the following attributes:

- (i) Name.
- (ii) Student ID.
- (iii) Password.
- (iv) Date of Birth.
- (v) Age.
- (vi) Sex.
- (vii) Father's name.
- (viii) Mother's name.
- (ix) Address.
- (x) Phone.
- (xi) Email address.
- (xii) Credits completed.
- (xiii) TGPA.
- (xiv) CGPA.
- (xv) Extra curriculum activities.
- (xvi) Program.

b. **Instructor**

Instructor is a strong entity set with the following attributes:

- (i) Name.
- (ii) Initial.
- (iii) Department.
- (iv) Designation.
- (v) Date of birth.
- (vi) Age.
- (vii) Sex
- (viii) Phone.
- (ix) Address.
- (x) Email.
- (xi) Date of join.
- (xii) Academic qualifications.
- (xiii) Category (part time / full time).

c. **Teaching Assistant.**

Teaching Assistant is a strong entity set with the following attributes:

- (i) Name.
- (ii) Department.
- (iii) Date of birth.
- (iv) Age.
- (v) Phone.
- (vi) Address.
- (vii) Email.
- (viii) Date of join.
- (ix) Academic qualifications.
- (x) Category (part time / full time).

d. **Course.**

Subject is a strong entity set with the following attributes:

- (i) Name.
- (ii) Title.
- (iii) Credit.
- (iv) Pre-requisite.
- (v) Description.

e. **Club.**

Club is a strong entity set with the following attributes:

- (i) Name.
- (ii) Advisor.
- (iii) Chairman.
- (iv) Administrative body.
- (v) Starting date.
- (vi) Description.
- (vii) Current events.
- (viii) Number of members.

f. **Course-Section.**

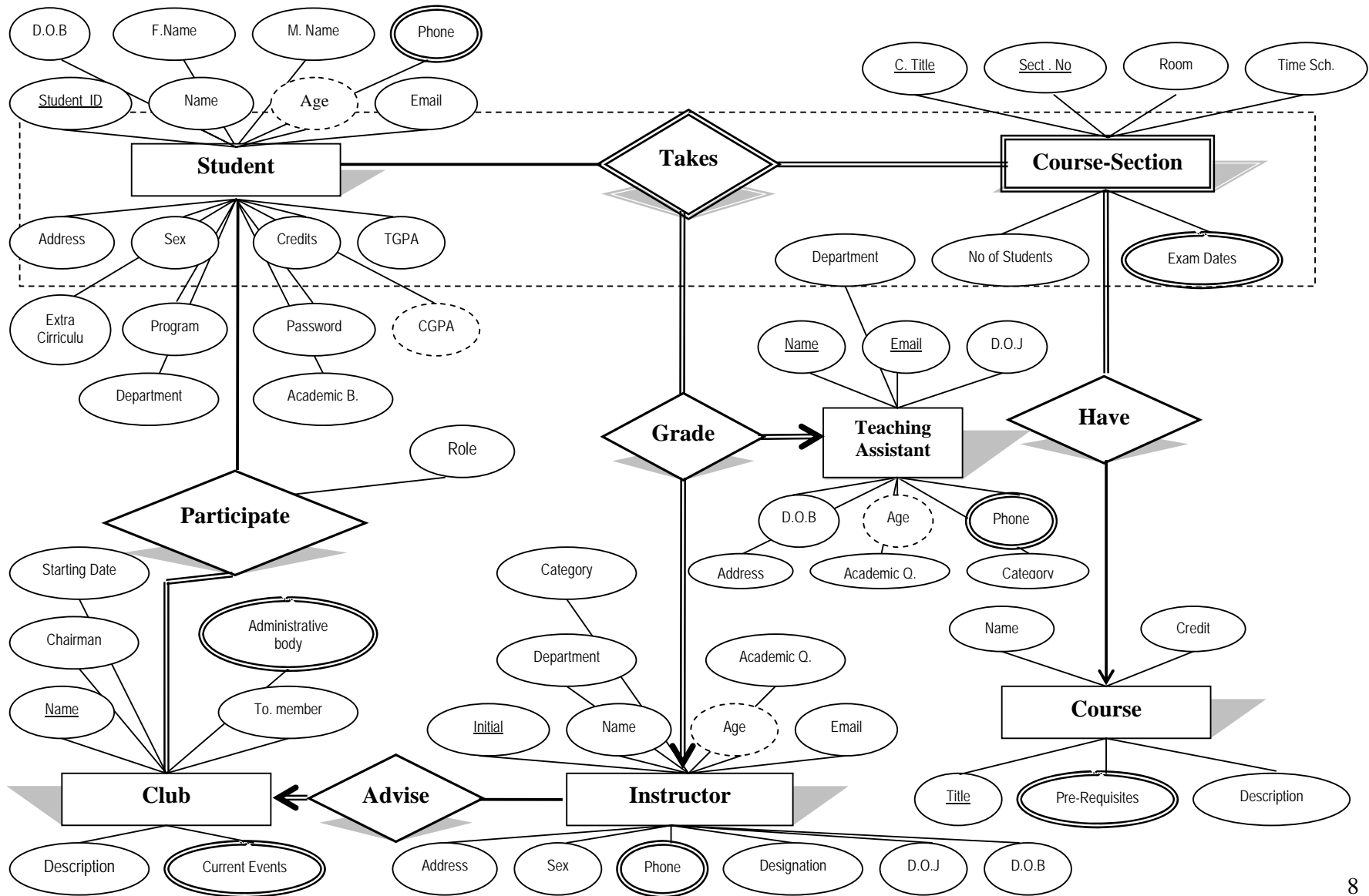
Section is a weak entity set with the following attributes:

- (i) Section number.
- (ii) Course Title.
- (iii) Room number.
- (iv) Time schedule.
- (v) No of students.
- (vi) Exam dates.

6.2 Relationship set designation

No	Name of Relation	Related Entities	Cardinality	Participation
01	Takes	section & students	Many to many.	Student side partial, section side total.
02	Participate	Club & students	Many to many.	Student side partial, club side total.
03	Advise	Club & instructors	Many to one.	Faculty side partial, club side total.
04	Grades	Student, section, TA & Instructor.	Many, one, one, one.	All side total

4.3 Entity Relationship Diagram (ERD)



6.4 Reduce Table from ERD

Student <Table>

Name	<u>Student ID</u>	Password	Date of Birth	Age	Sex	Father's Name	Mother's Name
------	-------------------	----------	---------------	-----	-----	---------------	---------------

Address	Email address	Credits Completed	TGPA	CGPA	Extra curriculum Activities	Program
---------	---------------	-------------------	------	------	-----------------------------	---------

Department	Academic Background
------------	---------------------

Student_Phone <Table>

<u>Student ID</u>	<u>Phone</u>
-------------------	--------------

Instructor <Table>

Name	<u>Initial</u>	Department	Designation	Date of Birth	Age
------	----------------	------------	-------------	---------------	-----

Address	Email	Date of join	Academic qualification	Category	Sex
---------	-------	--------------	------------------------	----------	-----

Instructor_Phone <Table>

<u>Initial</u>	<u>Phone</u>
----------------	--------------

Teaching Assistant <Table>

<u>Name</u>	Department	Date of Birth	Age	<u>Email</u>
-------------	------------	---------------	-----	--------------

Address	Date of join	Academic Qualification	Category
---------	--------------	------------------------	----------

Teaching Assistant_Phone <Table>

<u>Name</u>	<u>Email</u>	<u>Phone</u>
-------------	--------------	--------------

Course <Table>

Course name	<u>Title</u>	Credit	Description
-------------	--------------	--------	-------------

Course prerequisite <Table>

<u>Title</u>	<u>Prerequisite</u>
--------------	---------------------

Club <Table>

<u>Name</u>	Chairman	Starting Date	Description	No of Members
-------------	----------	---------------	-------------	---------------

Club administrative body <Table>

<u>Name</u>	<u>Administrative Body</u>
-------------	----------------------------

Club current events <Table>

<u>Name</u>	<u>Current event</u>
-------------	----------------------

Course_Section <Table>

<u>Section_No</u>	<u>Title</u>	<u>Room_No</u>	<u>Time Schedule</u>	<u>Number of Students</u>
-------------------	--------------	----------------	----------------------	---------------------------

Course Section Exam dates <Table>

<u>Section_No</u>	<u>Title</u>	<u>Exam date</u>
-------------------	--------------	------------------

Grade <Table>

<u>Student_ID</u>	<u>Title</u>	<u>Section_No</u>	<u>Initial.</u>	<u>Name.(TA)</u>
-------------------	--------------	-------------------	-----------------	------------------

<u>Email.(TA)</u>	<u>Semester</u>	<u>Grade</u>
-------------------	-----------------	--------------

Grade quiz marks<Table>

<u>Student_ID</u>	<u>Title</u>	<u>Section_No</u>	<u>Quiz marks</u>
-------------------	--------------	-------------------	-------------------

Grade exam marks<Table>

<u>Student_ID</u>	<u>Title</u>	<u>Section_No</u>	<u>Exam marks</u>
-------------------	--------------	-------------------	-------------------

Grade assignment marks<Table>

<u>Student_ID</u>	<u>Title</u>	<u>Section_No</u>	<u>Assignment marks</u>
-------------------	--------------	-------------------	-------------------------

Participate <Table>

<u>Student_ID</u>	<u>Club_Name</u>	<u>Role</u>
-------------------	------------------	-------------

6.5 Relational Schema

Student

(Name, Student ID, Password, Date of Birth, Age, Sex, Father's Name, Mother's Name, Address, Email address, Credits Completed, TGPA, CGPA, Extra curriculum Activities, Program Department, Academic Background)

Student_Phone

(Student ID, Phone)

Instructor

(Name, Initial, Department, Designation, Date of Birth, Age, Address, Email, Date of join, Academic qualification, Category)

Instructor_Phone

(Initial, Phone)

Teaching Assistant

(Name, Department, Date of Birth, Age, Email, Address, Date of join, Academic Qualification, Category)

Teaching Assistant _Phone

(Name, Email, Phone)

Course

(Course name, Title, Credit, Description)

Course prerequisite

(Title, prerequisite)

Club

(Name, Chairman, Starting Date, Description, No of Members)

Club administrative body

(Name, Administrative Body)

Club current events

(Name, Current events)

Course_Section

(Section_No, Title, Room_No, Time Schedule, Number of Students)

Course Exam dates

(Section_No, Title ,Exam date)

Grade

(Student ID, Title, Section No, Initial, Name.(TA), Email.(TA), Semester, Grade)

Grade quiz marks

(Student ID, Title, Section No, Quiz marks)

Grade exam marks

(Student ID, Title, Section No, Exam marks)

Grade assignment marks

(Student ID, Title, Section No, Assignment marks)

Participate

(Student ID, Name, Role)

4.6 Schema Diagram

Student

PK	<u>Student_ID</u>
	Name Password Date of Birth Age Sex Father's Name Mother's Name Address Email Credits Completed Request Extra Curriculum- Activities Program Department Academic Background

Student_Phone

PK1	<u>Student_ID</u>
PK2	<u>Phone</u>

Course Prerequisite

PK1	<u>Title</u>
PK2	<u>Prerequisite</u>

Course

PK	<u>Title</u>
	Name Credit Description

Course_Section

PK1	<u>SectionNo</u>
PK2	<u>Title</u>
	Room_Number Time Shcedule Number of Students

Grade

PK1	<u>Student_ID</u>
PK2	<u>Title</u>
PK3	<u>Section_No</u>
Pk4	<u>Initial</u>
PK5	<u>Email TA</u>
	Semester Grade

CourseSectionExam Dates

PK1	<u>Section_No</u>
PK2	<u>Title</u>
PK3	<u>Exam Date</u>

Instructor

PK	<u>Initial</u>
	Name Password Department Designation Date of Birth Age Sex Address Email Date of join Academic Qualification Category

Grade assignment marks

PK1	<u>Student_ID</u>
PK2	<u>Title</u>
PK3	<u>Section_No</u> <u>AssignmentNo.</u>
	AssignmentMark Weight

Club Current Events

PK1	<u>Name</u>
PK2	<u>Currents Events</u>

Club Administrative body

PK1	<u>Name</u>
PK2	<u>Adminstrative body</u>

Instructor_Phone

PK1	<u>Initial</u>
PK2	<u>Phone</u>

GradeQuizMarks

PK1	<u>StudentID</u>
PK2	<u>Title</u>
PK3	<u>SectionNo</u>
PK4	<u>QuizNo</u>
	QuizMark Weight

Club

PK	<u>Name</u>
	Chairman Starting Date Description Advisor Message

Teaching Assistant(TA)

PK	<u>Email</u>
	Name Password Department Date of Birth Date of join Age Sex Phone Address Academic Qualification Category

Participate

PK1	<u>Student_ID</u>
PK2	<u>Name</u>
	Role

Teaching_Assistant_Phone

PK1	<u>Email</u>
PK2	<u>Phone</u>

Course_Section_Message

PK1	<u>Section_No</u>
PK2	<u>Title</u>
PK3	<u>MessageNo</u>
	PostedBy Message

GradeExamMarks

PK1	<u>StudentID</u>
PK2	<u>Title</u>
PK3	<u>SectionNo</u>
PK4	<u>ExamNo</u>
	ExamMark Weight

APPENDIX A: DATABASE SCHEMA

#Table Structure Of Student

```
create table Student
(
    Name varchar2(30) not null,
    StudentId char(9) not null,
    Password varchar2(20) not null,
    DateOfBirth date not null,
    Age number(2),
    sex varchar2(6),
    FatherName varchar2(30) not null,
    MotherName varchar2(30) not null,
    Address varchar2(50),
    Email varchar2(30),
    CreditsCompleted number(3),
    ExtraCurriculumActivities varchar2(50),
    Program varchar2(13) not null,
    Department char(3)not null,
    AcademicBackground varchar2(50),
    Request varchar2(255),
    primary key(StudentId),
    constraint studentCreditTest check(CreditsCompleted >= 0 ),
    constraint studentAgeTest check(age >= 16),
    constraint studentProgramTest check(Program in
('UnderGraduate','Graduate') ),
    constraint studentSexTest check(sex in ('Male','Female') ),
    constraint studentEmailTest check((email like '%@%') and (email like '%.%'))
),
    constraint studentDepartmentTest check(Department in
('CSC','CEG','ENV','ENG','ETE','BBA') ) );
```

#Table Structure Of StudentPhone

```
create table StudentPhone
(
    StudentId char(6) not null,
    Phone varchar2(10) not null,
    primary key (StudentId,Phone),
    foreign key (studentId) references Student(studentId) on delete cascade
);
```

#Table Structure Of Instructor

```
create table Instructor
(
    Name varchar2(30) not null,
    Ini_tial char(3) not null,
    Password varchar2(20) not null,
    Department char(3) not null,
    Designation varchar2(10) not null,
    DateOfBirth date not null,
    DateOfJoin date not null,
    Age number(2),
    sex varchar2(6),
    Address varchar2(50),
```

```

    Email varchar2(30),
    AcademicQualification varchar2(50),
    catagory char(8),
    primary key(Ini_tial),
    constraint instructorAgeTest check(age >= 30),
    constraint instructorSexTest check(sex in ('Male','Female') ),
    constraint instructorCatagoryTest check(catagory in ('HalfTime','FullTime') ),
    constraint instructorDepartmentTest check(Department in
('CSC','CEG','ENV','ENG','ETE','BBA') ) ) ;

```

#Table Structure Of InstructorPhone

```

create table InstructorPhone
(
    Ini_tial char(3) not null,
    Phone varchar2(10) not null,
    primary key (Ini_tial,Phone),
    foreign key (Ini_tial) references Instructor(Ini_tial) on delete cascade
);

```

#Table Structure Of TeachingAssistant

```

create table TeachingAssistant
(
    Name varchar2(30) not null,
    Password varchar2(20) not null,
    Department char(3) not null,
    DateOfBirth date not null,
    Age number(2),
    Email varchar2(30),
    Address varchar2(50),
    DateOfJoin date not null,
    sex varchar2(6),
    AcademicQualification varchar2(50),
    catagory char(8),
    primary key(Email),
    constraint taAgeTest check(age >= 30),
    constraint taSexTest check(sex in ('Male','Female') ),
    constraint taCatagoryTest check(catagory in ('HalfTime','FullTime') ),
    constraint taDepartmentTest check(Department in
('CSC','CEG','ENV','ENG','ETE','BBA') ) ) ;

```

#Table Structure Of TeachingAssistantPhone

```

create table TeachingAssistantPhone
(
    Email varchar2(30),
    Phone varchar2(10) not null,
    primary key (Email,Phone),
    foreign key (Email) references TeachingAssistant(Email) on delete cascade
);

```

#Table Structure Of Course

```
create table Course
(
    CourseName varchar2(30) not null,
    Title char(6) not null,
    Credit number(1) not null,
    Description varchar2(50),
    primary key (Title),
    constraint creditTest check(Credit >= 0) );
```

#Table Structure Of CoursePrerequisite

```
create table CoursePrerequisite
(
    Title char(6) not null,
    Prerequisite char(6) not null,
    primary key (Title, Prerequisite),
    foreign key (Title) references Course(Title) on delete cascade,
    foreign key (Prerequisite) references Course(Title) on delete cascade
);
```

#Table Structure Of Course_Section

```
create table Course_Section
(
    SectionNo number(1) not null,
    Title char(6) not null,
    Room_No varchar2(7) not null,
    TimeSchedule varchar2(6) not null,
    No_ofStudents number(3),
    primary key (Title, SectionNo),
    foreign key (Title) references Course(Title) on delete cascade,
    constraint studentTest check(No_ofStudents > 0) );
```

#Table Structure Of Course_Section_message

```
create table Course_Section_message
(
    SectionNo number(1) not null,
    Title char(6) not null,
    message_No number(1) not null,
    PostedBy varchar2(30) not null,
    primary key (Title, SectionNo, message_No),
    foreign key (Title, SectionNo) references Course_Section(Title, SectionNo) on
delete cascade
);
```

#Table Structure Of Course_SectionExamDates

```
create table Course_SectionExamDates
(
    SectionNo number(1) not null,
    Title char(6) not null,
    ExamDate date not null,
    primary key (Title, SectionNo, ExamDate),
    foreign key (Title, SectionNo) references Course_Section(Title, SectionNo) on
delete cascade
);
```

#Table Structure Of Grade

```
create table Grade
(
    StudentId char(9) not null,
    Title char(6) not null,
    SectionNo number(1) not null,
    Email varchar2(30),
    Ini_tial char(3) not null,
    Semester varchar2(6) not null,
    Grade char(2),
    Gpa number(2,1),
    primary key (StudentId,Title,SectionNo),
    foreign key (StudentId) references Student(StudentId) on delete cascade,
    foreign key (Title,SectionNo) references Course_Section(Title,SectionNo) on
delete cascade,
    foreign key (Email) references TeachingAssistant(Email) on delete cascade,
    foreign key (Ini_tial) references Instructor(Ini_tial) on delete cascade
);
```

#Table Structure Of GradeQuizMark

```
create table GradeQuizMark
(
    StudentId char(9) not null,
    Title char(6) not null,
    SectionNo number(1) not null,
    QuizNo number(2) not null,
    QuizMark number(2) not null,
    weight number(2) not null,
    primary key (StudentId,Title,SectionNo,QuizNo),
    foreign key (StudentId) references Student(StudentId) on delete cascade,
    foreign key (Title,SectionNo) references Course_Section(Title,SectionNo)on
delete cascade,
    constraint QuizTest check(QuizMark >=0) );
```

#Table Structure Of GradeExamMark

```
create table GradeExamMark
(
    StudentId char(9) not null,
    Title char(6) not null,
    SectionNo number(1) not null,
    ExamNo number(1) not null,
    ExamMark number(2) not null,
    weight number(3) not null,
    primary key (StudentId,Title,SectionNo,ExamNo),
    foreign key (StudentId) references Student(StudentId) on delete cascade,
    foreign key (Title,SectionNo) references Course_Section(Title,SectionNo) on
delete cascade,
    constraint ExamTest check(ExamMark >=0) );
```

#Table Structure Of AssignmentMark

```
create table AssignmentMark
(
    StudentId char(9) not null,
    Title char(6) not null,
    SectionNo number(1) not null,
    AssignmentNo number(3) not null,
    AssignmentMark number(2) not null,
    weight number(3) not null,
    primary key (StudentId,Title,SectionNo,AssignmentNo),
    foreign key (StudentId) references Student(StudentId) on delete cascade,
    foreign key (Title,SectionNo) references Course_Section(Title,SectionNo) on
delete cascade,
    constraint AssignmentTest check(AssignmentMark >=0) );
```

#Table Structure Of Club

```
create table club
(
    Chairman char(9) not null,
    Name varchar2(30) not null,
    Advisor char(3) not null,
    StartingDate date not null,
    Description varchar2(50),
    message varchar2(255),
    primary key (Name),
    foreign key (Chairman) references Student(StudentId) on delete cascade,
    foreign key (Advisor) references Instructor(Ini_tial) on delete cascade
);
```

#Table Structure Of clubAdministrativeBody

```
create table clubAdministrativeBody
(
    Name varchar2(30) not null,
    AdministrativeBody char(9),
    primary key (Name,AdministrativeBody),
    foreign key (AdministrativeBody) references Student(StudentId)on delete
cascade,
    foreign key (Name) references Club(Name)on delete cascade
);
```

#Table Structure Of clubCurrentEvents

```
create table clubCurrentEvents
(
    Name varchar2(30) not null,
    CurrentEvent varchar2(50),
    primary key (Name,CurrentEvent),
    foreign key (Name) references Club(Name) on delete cascade
);
```

#Table Structure Of participate

```
create table Participate
(
    StudentId char(9) not null,
    Name varchar2(30) not null,
    Role varchar2(15),
    primary key (Name,StudentId),
    foreign key (StudentId) references Student(StudentId) on delete cascade,
    foreign key (Name) references Club(Name) on delete cascade
);
```

APPENDIX B: LIMITATIONS

The developed system lacks few aspects which should be taken care of. Few of these things are described here.

Multi login

We could not able to prevent multi login for either of the students, Teaching Assistants and Instructors. By using normal and web application multiple login with same user name can be possible.

Derived Attributes

The system is supposed to develop the derived attributes using embedded sql. We tried to overcome this problem using front end calculation which is not suitable for all situation .

Synchronization

The system is not synchronized properly. So data write in same field of a table of the database is possible at a time which may occur inconsistent situation.

APPENDIX C: NEXT VERSION FEATURE

Next version will include few features which will prove handy. Also, existing features could be modified time to time to meet specific requirement. Here is a list of features that could be implemented in the next release.

Mail Facility

The system doesn't include any mail facility between the students. Emailing facility would make the interaction with the users more active.

Advising Facility

By giving the advisors/Instructors more flexible writing facility this system can be easily converted to an application with advising facility.

More Login type

Now the system just have the student login for the web application. In next version we will try to include instructor and teaching assistant login for the web application.

APPENDIX D: PROGRAMS LIST

The programs List for front end are given bellow. More about it is included in Programmer's manual and user's manual.

Application programs

LoginScreen.java
StudentScreen.java
MainScreen.java
StudentDataHandler.java
AssistantScreen.java
AssistantDataHandler.java
FacultyScreen.java
FacultyDataHandler.java
Constants.java
DBaseConnector.java
Button2D.java
TextField2D.java
PassField2D.java
TextArea2D.java
ChartPanel.java
MessageBox.java
PassordFrame.java
ResultSetTable.java
Tools.java
UserButton.java