Student Database Management System

Ajay Shankar Bidyarthy*, Abhishek Kumar**
b.ajay@iitg.ernet.in* kumar.ak@iitg.ernet.in**

November 13, 2012

Abstract: An organized and systematic office solution is essential for all universities and organizations. There are many departments of administration for the maintenance of college information and student databases in any institution. All these departments provide various records regarding students. Most of these track records need to maintain information about the students. This information could be the general details like student name, address, performance, attendance etc or specific information related to departments like collection of data. All the modules in college administration are interdependent. They are maintained manually. So they need to be automated and centralized as, Information from one module will be needed by other modules. For example when a student needs his course completion certificate it needs to check many details about the student like his name, reg number, year of study, exams he attended and many other details. So it needs to contact all the modules that are office, department and examination and result of students.

With that in mind, we overhauled the existing Student Database Management System and made necessary improvement to streamline the processes. Administrators using the system will find that the process of recording and retrieving students information and managing their classes, including marking of attendance, is now a breeze. In general, this project aims to enhance efficiency and at the same time maintain information accurateness. Later in this report, features and improvement that allow achievement to this goal will be demonstrated and highlighted.

Our work is useful for easy user interface. We are planning to uttilize the powerful database management, data retrieval and data manipulation. We will provides more ease for managing the data than manually maintaining in the documents. Our work is useful for saving valuable time and reduces the huge paper work.

Keywords: Education, Database managements, students database, students academic performance, students health record.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Student Management System deals with all kind of student details, academic related reports, college details, course details, curriculum, batch details and other resource related details too. It tracks all the details of a student from the day one to the end of his course which can be used for all reporting purpose, tracking of attendance, progress in the course, completed semesters years, coming semester year curriculum details, exam details, project or any other assignment details, final exam result etc.

Our design can facilitate us to explore all the activities happening in the college, even we can get to know which faculty is assigned to which course, the current status of a student, attendance percentage of a student and upcoming requirements of a student. The student management system is an automated version of manual Student Management System. It can handle all details about a student. The details include college details, subject details, student personnel details, academic details, exam details etc.

In case of manual system they need a lot of time, manpower etc. Here almost all work is computerized. So the accuracy is maintained. Maintaining backup is very easy. It can do with in a few minutes. Our system has two type of accessing modes, administrator and user. Student management system is managed by an administrator. It is the job of the administrator to insert update and monitor the whole process. When a user log in to the system. He/she would only view details of the student. He/she can't perform any changes. Our system has seven modules, they are administrator, student, course, department, exam, attendance, and section. These modules and its attributes with entity relationship module presented in the ER diagram secion.

Chapter 2

Project Review

2.1 Project management

Project management skills are put to good use for this project. Having gone through project management modules in Time Series Analysis, Optimization and with two interns Project Management for Business and IT respectively, they enhanced my knowledge on managing a project. Project management focuses on achieving the objectives by applying five processes presented in Figure below.

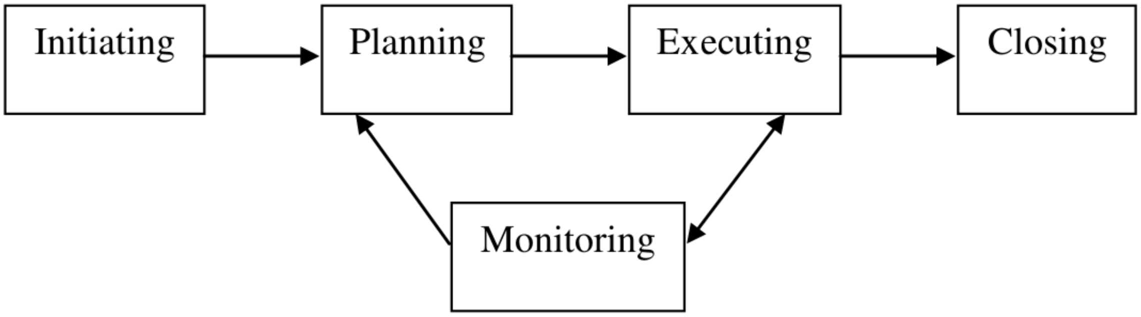


Figure 2.1: Project Development Phases

2.2 Student management system

There are many software development companies that offer student management system for schools in the market. There are records on the past years projects on student management system is done by students. Through the researches, it is observed that there are features where this project can adopt and implement. One of it will be with addition of new course or class, or even upgrading of students to the next level, the school administrator can easily register all of them within a particular class into the new one using just a page and not having to register one at a time. With this feature, it helps administrator to save time as well as increase their efficiency.

Chapter 3

Resources

3.1 System Development lyfe cycle

Systems Development Life Cycle (SDLC) is the most common process adopted to develop a project and not surprisingly, this project is following this model too. To be precise, waterfall model is being applied. Waterfall model is a sequential model process where the input of a phase actually results from the previous phase.

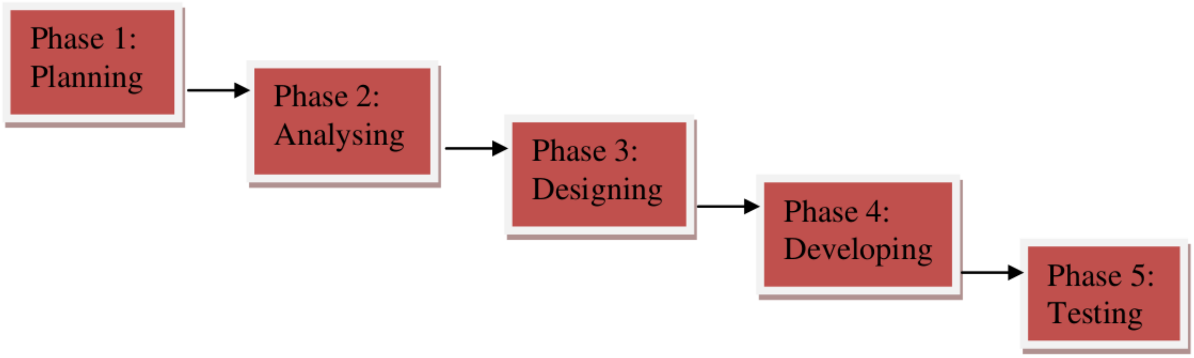


Figure 3.1: SDLC Phases

There are five phases in this model and the first phase is the planning stage. The planning stage determines the objectives of the project and whether the project should be given the green light to proceed. This is where the proposal submission comes into picture. After obtaining the approval, the next phase is analysis. Gathering and analysing the system and user requirements is essential for entry to the design step.

With the user requirements gathering completed, there is a need to prepare the resources for the project. Be it software or hardware components, careful consideration and selection is to be taken care at this stage. The decision on the appropriate resources to be used is further elaborated under the subsections below. The next step is to design the system and database structure.

Results from the analysis and preparation that were concluded from the previous stage are put into action. With the user requirements in mind, the flow of the system is planned and the user interface is designed to suit their easy navigation needs. In addition, the number of tables, attributes, primary and unique keys of the database is listed.

After completing the design, actual coding begins. Database is created and codes are written. Some of the codes required amendments and improvement to it so these are being developed at this fourth stage of the waterfall model. With the development completed, testing will begin. The codes and database are tested to ensure the results obtained are as intended. More time is spent on both development and testing stages because it is inevitable to have errors and issues and buffer time is allocated for troubleshooting.

3.2 Scripting language selection

There are many scripting languages available in the market. VBScript, Perl, JSP (Java Server Pages), ASP (Active Server Pages) and PHP (Hypertext Pre-processor) are some of those commonly used. Yet for this project, PHP is the language that is utilised for the coding piece because it is a server-side, embeddable HTML language. Being a widely-used open source scripting language, it is free for everyone to use and is especially suited for web development. On top of that, the existing system is already using PHP. There are many advantages for using PHP thus no need for the switch to another scripting language.

Other than being a freeware, there are many free upgrade packages easily available. The other benefit of choosing PHP is the ease in installation. It can run as a plug in on quite a number of web servers such as the Apache. On the other hand, JSP requires J2EE server to run and because it is a Java coded language, it is therefore more complex to understand and to do coding

Further exploring on the processing speed against ASP, PHP is interpreted at runtime and not compiled into memory whereas ASP is more memory intensive with each ASP language compiler running in its own processes. This results in slower processing speed for ASP. In addition, ASP runs more reliably only on Microsoft Windows-based web servers than other web servers

In conclusion, PHP is the preferred selection due to the ease of usage and it can be uploaded and run on another platform with minimal change required to be done to the script. Beyond and above, the compiling time and speed for PHP is faster and more efficient.

3.3 Database selection

There are a variety of databases that we can select from the market. The widely used databases are Microsoft Access, Microsoft SQL, Oracle and MySQL. Looking at Microsoft

Access, it does not encourage concurrent usage and it may be inefficient, as the database needs to be saved into one file. It is also unable to process high speed and large size database as compared to MySQL.

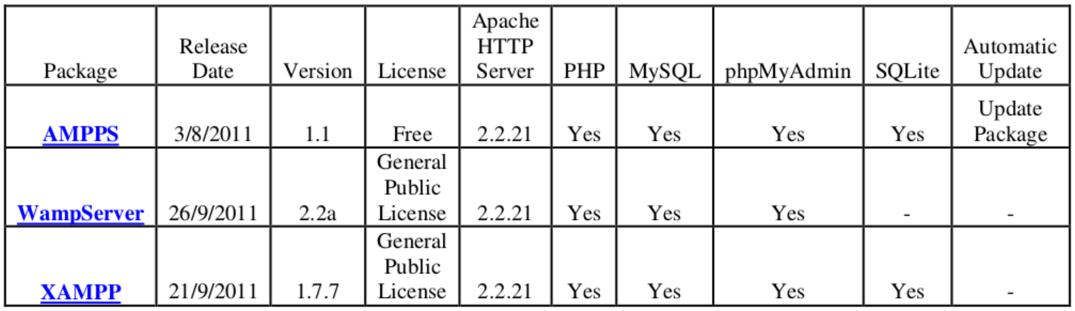
In terms of costs, Oracle database requires a licensing fee but MySQL database is a freeware. In addition, MySQL database is easy to install, user friendly, reliable and is able to run on different platforms. Moreover PHP can access MySQL database directly without the need to go through ODBC (Open Database Connectivity).

To conclude, PHP script is able to run faster with MySQL database and the processing time will definitely be shorter. The pre-school does not require complex and costly software for its database management system hence MySQL is the ideal database for this project.

3.4 Web server selection

After deciding on the scripting language and database, next is to select the web server that can support them. Web server is necessary for the delivery of web content to the web browser. As such, Apache HTTP server which has performance similar with other 'high-performance' server is considered

Thereafter, research and actual testing have been performed to see the outcome of the various servers listed in the Figure below. These servers include PHP and MySQL in their installation packages thus allowing smoother and simpler download process. However, based on the performance and interface, Wamp or LAMP server is the preferred choice.



Package	Release Date	Version	License	Apache HTTP Server	PHP	MySQL	phpMyAdmin	SQLite	Automatic Update
AMPPS	3/8/2011	1.1	Free	2.2.21	Yes	Yes	Yes	Yes	Update Package
WampServer	26/9/2011	2.2a	General Public License	2.2.21	Yes	Yes	Yes	-	-
XAMPP	21/9/2011	1.7.7	General Public License	2.2.21	Yes	Yes	Yes	Yes	-

Figure 3.2: Comparison of web servers

Chapter 4

System Analysis and Design

4.1 Use case analysis

In order to provide a clearer picture of the functionality provided by the student management module, we have done a use case analysis. Figure below is a use case diagram to present the functionality in the student management module.

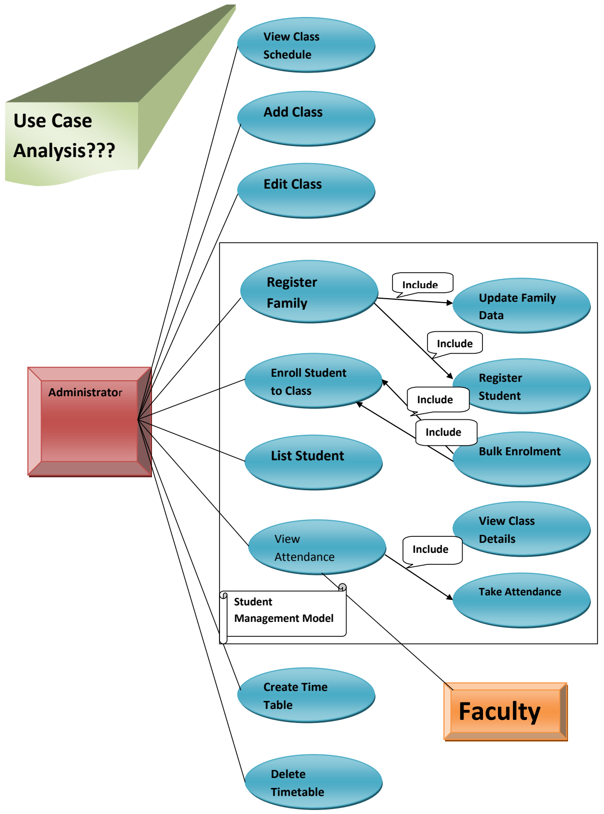
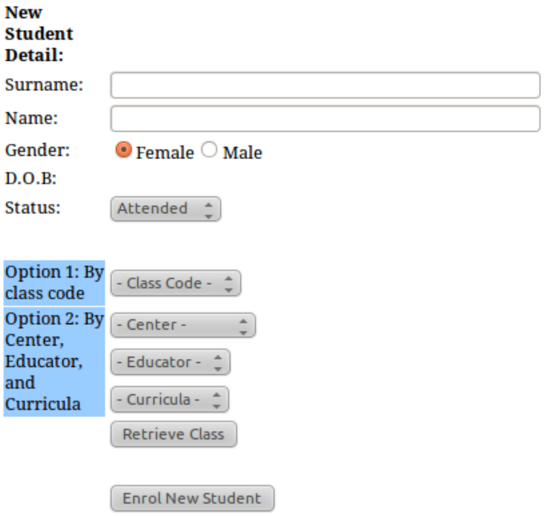


Figure 4.1: use case analysis

4.2 System Design

Under the existing student management module, the administrator has to first register the family. This includes the parent's particulars and home details. After submitting these data, the administrator will be directed to the student page whereby the student information is to be submitted.

Figure below provides the interface of the existing student registration site. For easy navigation, the registration of the student and enrolment to the class has been combined into one process.



New Student Detail:

Surname:

Name:

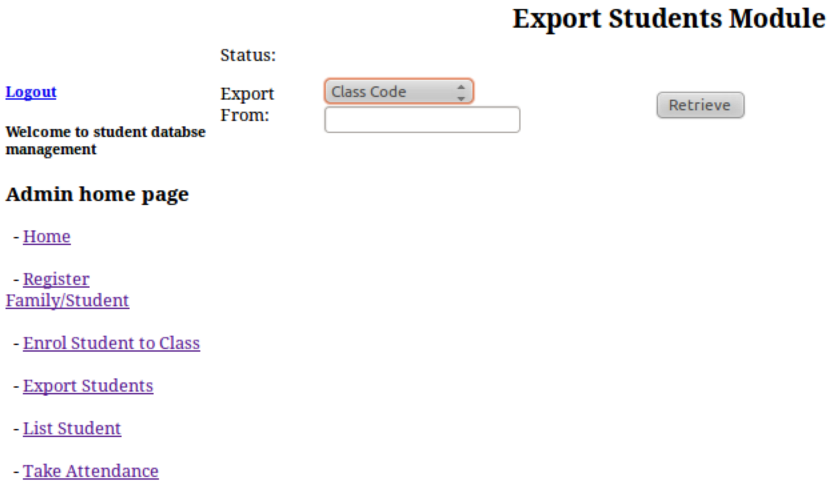
Gender: ● Female ○ Male

D.O.B:

Status: Attended

Figure 4.2: Student registration process

Besides the registration for a single student, there is a new feature added to the system. That is to 'export' bulk students from a class to another. This new feature is created to minimize the time required to register students to a new class. The school has many programs ranging from beginner to intermediate levels thus with this new feature, the administrator can easily enrol those students to their next level. The interface of this function is shown in Figure below.

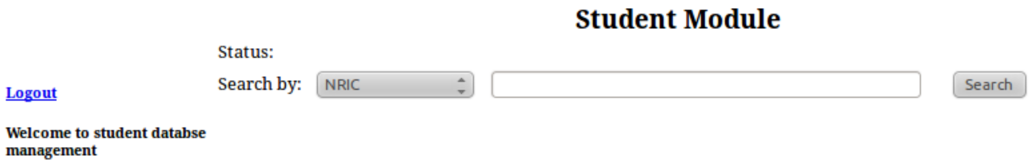


Export From: Class Code

Figure 4.3: Export students module

Other than the 'export' feature, the system has been modified to allow a couple of search attributes at the family registration part. The name attribute includes the first and last name whereas contact number means mobile, office and home number.

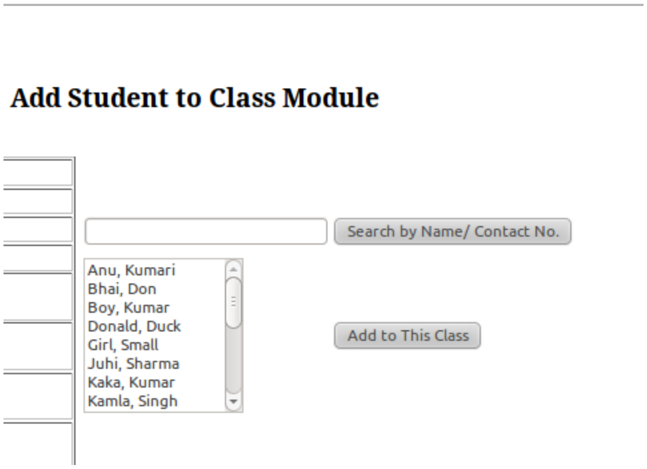
Figure below present functionality at registration page.



Search by: NRIC

Figure 4.4: Search attributes for registration process

Figure below present the functionality to add student to class module from the existing one.

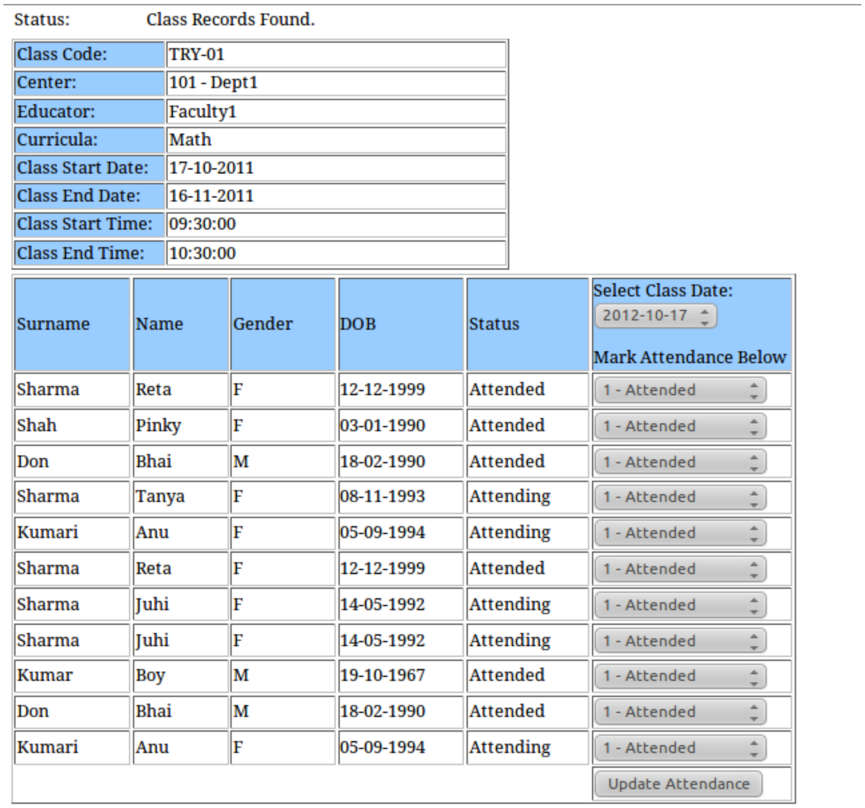


Add Student to Class Module

Anu, Kumari
Bhai, Don
Boy, Kumar
Donald, Duck
Girl, Small
Juhi, Sharma
Kaka, Kumar
Kamla, Singh

Figure 4.5: Enrol existing student to class

For the attendance marking system the following functionality has been added



Status: Class Records Found.

Class Code:	TRY-01
Center:	101 - Dept1
Educator:	Faculty1
Curricula:	Math
Class Start Date:	17-10-2011
Class End Date:	16-11-2011
Class Start Time:	09:30:00
Class End Time:	10:30:00

Surname	Name	Gender	DOB	Status	Select Class Date: 2012-10-17 Mark Attendance Below
Sharma	Reta	F	12-12-1999	Attended	1 - Attended
Shah	Pinky	F	03-01-1990	Attended	1 - Attended
Don	Bhai	M	18-02-1990	Attended	1 - Attended
Sharma	Tanya	F	08-11-1993	Attending	1 - Attended
Kumari	Anu	F	05-09-1994	Attending	1 - Attended
Sharma	Reta	F	12-12-1999	Attended	1 - Attended
Sharma	Juhi	F	14-05-1992	Attending	1 - Attended
Sharma	Juhi	F	14-05-1992	Attending	1 - Attended
Kumar	Boy	M	19-10-1967	Attended	1 - Attended
Don	Bhai	M	18-02-1990	Attended	1 - Attended
Kumari	Anu	F	05-09-1994	Attending	1 - Attended
					Update Attendance

Figure 4.6: Legend and attendance system

In order to fulfil the second objective on minimizing human error, there are a few error checking functions being implemented for the student management module. Firstly, at the family registration page, all the fields except the entry date, referral and remarks fields have been coded to do validation check. This is to ensure that all the relevant information are collected and inserted into the database in an orderly manner.

The contact fields are designed to prompt an error message if none of the three fields has been input. The school requires at least one valid number. In addition, contact number should not contain alphabets so this has been taken into consideration for the error checking. As for surname, name and nationality fields, checks are implemented to make sure the fields are not empty. On the other hand the postal code which should not contain alphabet; there will be message to alert any error as well.

Other than error checks on family registration page, the student registration page also checks for the surname and name fields. After retrieving the class details at the same page, users will be alerted if no class is selected for the enrolment. This is also implemented at the bulk student enrolment process. If the targeted class code is not provided, error message will appear.

The codes for the system design are provided under Appendix A.

Chapter 5

Database Design and Development

5.1 Database design

Database is critical for all businesses. A good database does not allow any form of anomalies and stores only relevant information in an ordered manner. If a database has anomalies, it is affecting the efficiency and data integrity. For example, delete anomaly arise upon the deletion of a row which also forces other useful data to be lost. As such, the tables need to be normalised. This fulfils the last objective of ensuring data are accurate and retrieved correctly. For the database of this project, the tables are normalised to BCNF as shown below.

admin(id, name, password)
faculty(id, name, password)
ATTENDANCE (student-class-id, class-date, status)
Where ATTENDANCE.student-class-id must exist in STUDENT-CLASS.id
ATTENDANCE-CODE (attendance-code, attendance-status)

CLASS-SCHEDULE (id, code, clientcode, clientname, educatorname, educatornric, curriculatheme, startdate, enddate, starttime, endtime)
Where CLASS-SCHEDULE.code must exist in CLIENT.code,
CLASS-SCHEDULE.clientname must exist in CLIENT.name,
CLASS-SCHEDULE.educatornric must exist in EDUCATOR.nric,
CLASS-SCHEDULE.educatorname must exist in EDUCATOR.name
And CLASS-SCHEDULE.curriculatheme must exist in CURRICULA.theme

CLASS-TIMETABLE (class-schedule-id, class-date)
Where CLASS-TIMETABLE.class-schedule-id must exist in CLASS-SCHEDULE.id
CLIENT (code, name)
CURRICULA (grp, theme)
EDUCATOR (nric, name)
(nric: National Registration Identity Card)

FAMILY (id, entry-date, reg-fee, status, name, surname, nric, address1, address2, postal-code, contact-home, contact-office, contact-hp, email, nationality, referred-by, remarks)

STUDENT (id, family-id, name, surname, sex, dob, status)
Where STUDENT.family-id must exist in FAMILY.id
STUDENT-CLASS (id, student-id, class-schedule-id, remark)
Where STUDENT-CLASS.student-id must exist in STUDENT.id
And STUDENT-CLASS.class-schedule-id must exist in CLASS-SCHEDULE.id

5.2 Database development

fter identifying the tables and columns of the database, the next step is to create them. Basically there are two ways to do so. The first option is to create them using commands. Below are some of the ’create’ and ’insert’ statements. The complete database code is provided under Appendix B.

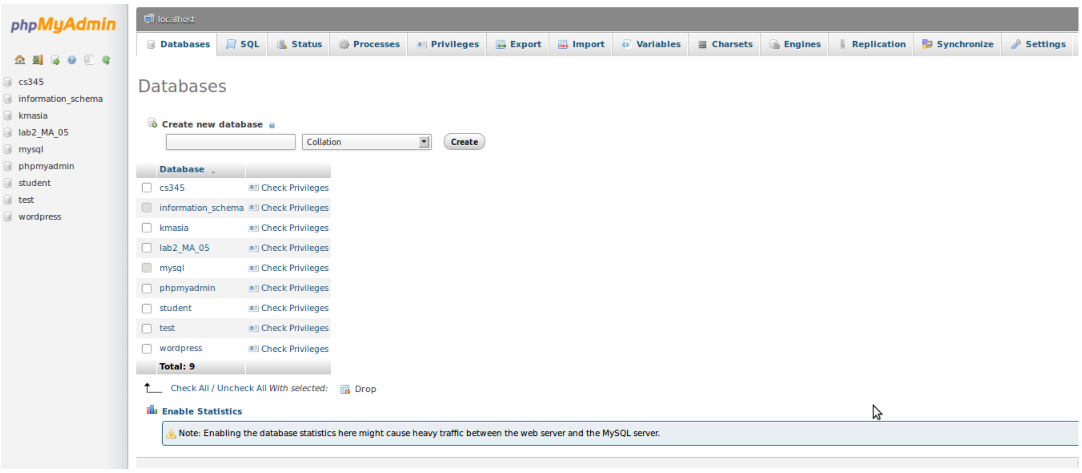
```
create table if not exists admin(aid char(50) not null primary key, aname char(50), apwd char(50));
insert into admin values(’admin’, ’admin’, password(’admin’));
create table if not exists faculty(fid char(50) not null primary key, fname char(50), fpwd char(50));
insert into faculty values(’faculty’, ’faculty’, password(’faculty’));
```

```
CREATE TABLE IF NOT EXISTS curricula (grp varchar(50) NOT NULL, theme varchar(50) NOT NULL, PRIMARY KEY (grp) ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
INSERT INTO curricula (grp, theme) VALUES (’1’, ’Math’), (’2’, ’Phy’), (’3’, ’Che’), (’4’, ’CSE’);
```

The second option is to use phpMyAdmin. This is a Graphical User Interface (GUI) interface for building and maintaining the database which is included in the web server, XAMPP package. This is a simpler way to create the tables if one does not know the usage of commands.

The first step is to create the database as shown in Figure below. The default storage engine in this MySQL server is InnoDB which has commit, rollback, and crash-recovery capabilities to protect user data. Following that is to create the tables in the database.



Databases

Create new database

Collation — Create

Database	
cs345	Check Privileges
information_schema	Check Privileges
kmasia	Check Privileges
lab2_MA_05	Check Privileges
mysql	Check Privileges
phpmyadmin	Check Privileges
student	Check Privileges
test	Check Privileges
wordpress	Check Privileges
Total: 9	

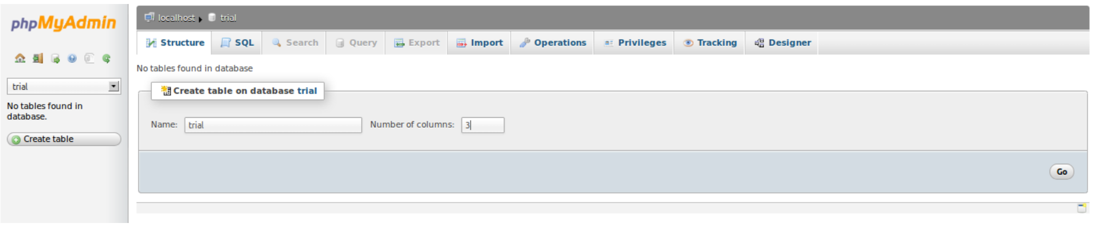
Check All / Uncheck All *With selected:* Drop

Enable Statistics

Note: Enabling the database statistics here might cause heavy traffic between the web server and the MySQL server.

Figure 5.1: Database creation in phpMyAdmin

Figure below gives a view of the table creation.



No tables found in database.

No tables found in database

Create table on database trial

Name: trial Number of columns: 3

Figure 5.2: Table creation in phpMyAdmin

Figure below shows the interface for creating the attributes in the table.

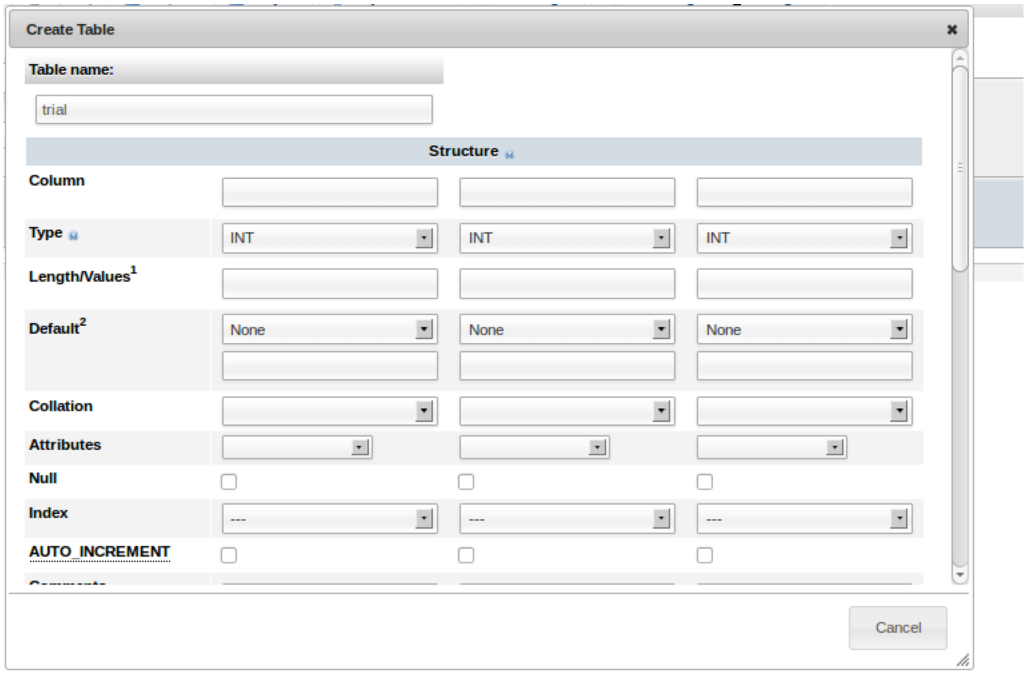


Table name: trial

Structure			
Column			
Type	INT	INT	INT
Length/Values[1]			
Default[2]	None	None	None
Collation			
Attributes			
Null	□	□	□
Index	---	---	---
AUTO_INCREMENT	□	□	□

Figure 5.3: Attribute creation in phpMyAdmin

After completing the creation, Figure below depicts the contents of the database, providing information like the number of tables in the database and the number of records in each of them. The MySQL server, phpMyAdmin, has many other features such as the deletion of tables, attributes and database. It also provides a mean for exporting or importing data into another database easily just by selecting the desired option seen at the top of the table in Figure below.

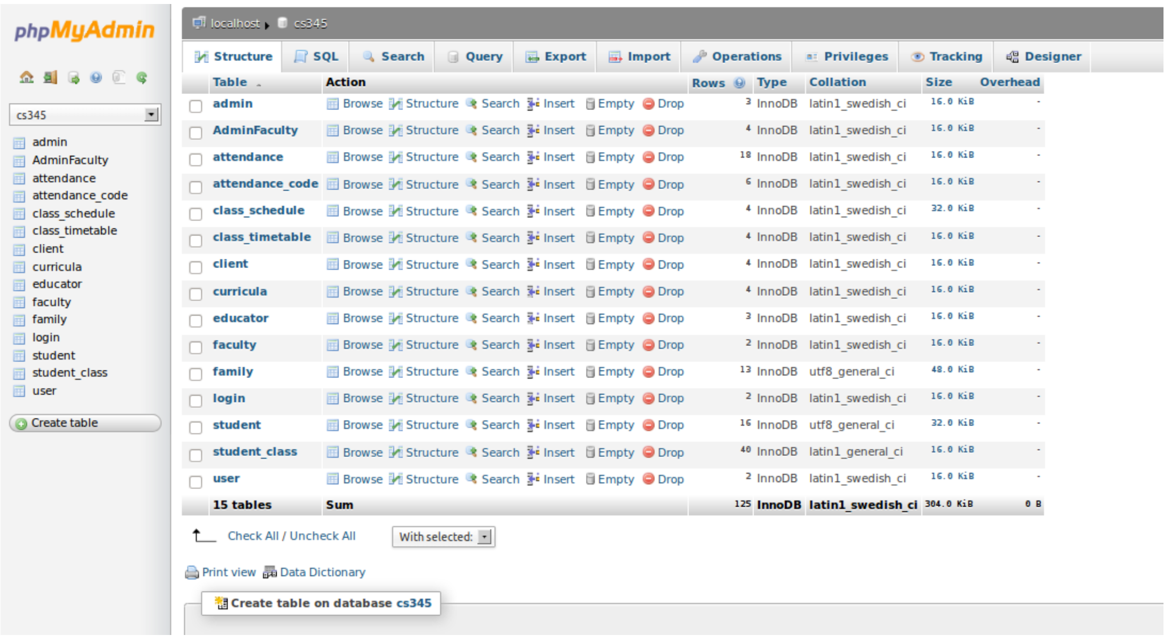
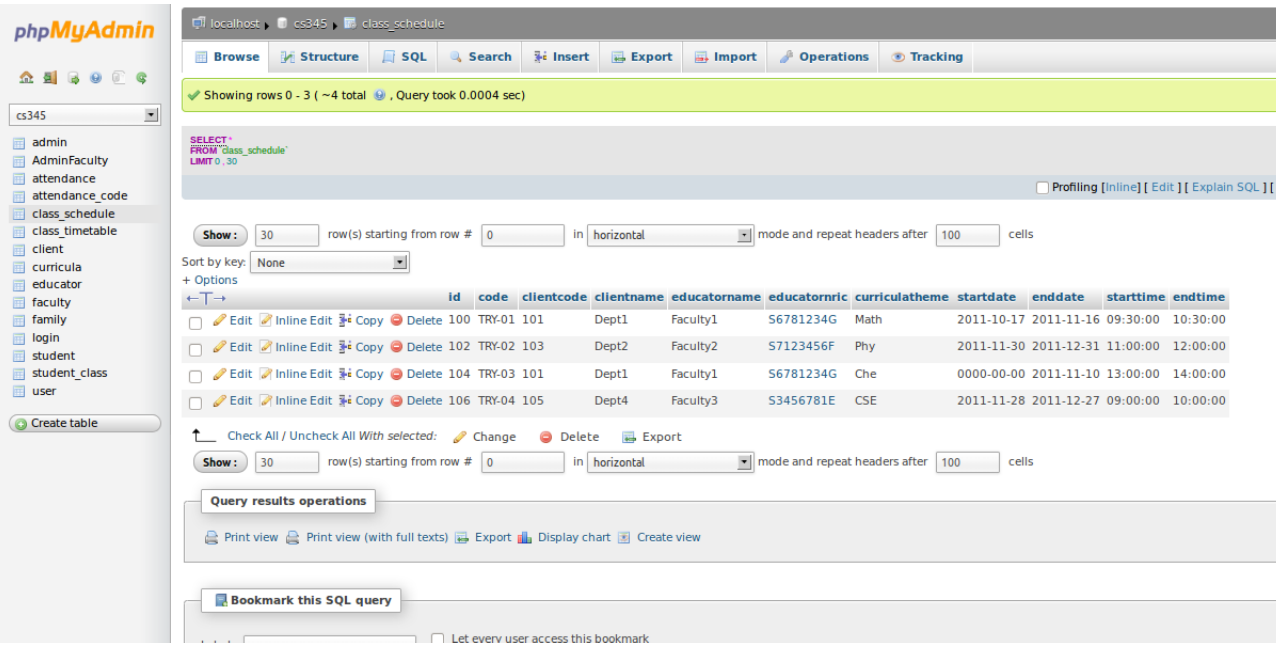


	Table	Action	Rows	Type	Collation	Size	Overhead
□	admin	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	AdminFaculty	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	attendance	Browse Structure Search Insert Empty Drop	18	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	attendance_code	Browse Structure Search Insert Empty Drop	6	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	class_schedule	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	32.0 KiB	-
□	class_timetable	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	client	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	curricula	Browse Structure Search Insert Empty Drop	4	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	educator	Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	faculty	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	family	Browse Structure Search Insert Empty Drop	13	InnoDB	utf8_general_ci	48.0 KiB	-
□	login	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB	-
□	student	Browse Structure Search Insert Empty Drop	16	InnoDB	utf8_general_ci	32.0 KiB	-
□	student_class	Browse Structure Search Insert Empty Drop	40	InnoDB	latin1_general_ci	16.0 KiB	-
□	user	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16.0 KiB	-
	15 tables	**Sum**	**125**	**InnoDB**	**latin1_swedish_ci**	**304.0 KiB**	**0 B**

Figure 5.4: Total number of tables in a database

To view the data in one of the table, this can be done by clicking on the table name listed on the left menu as per Figure below. The figure illustrates the educator table and there are three records in it.



✓ Showing rows 0 - 3 (~4 total , Query took 0.0004 sec)

```
SELECT *
FROM `class_schedule`
LIMIT 0 , 30
```

□ Profiling [Inline] [Edit] [Explain SQL] [

Show : 30 row(s) starting from row # 0 in horizontal mode and repeat headers after 100 cells

Sort by key: None

+ Options

		id	code	clientcode	clientname	educatorname	educatornric	curriculatheme	startdate	enddate	starttime	endtime
□	Edit Inline Edit Copy Delete	100	TRY-01	101	Dept1	Faculty1	S6781234G	Math	2011-10-17	2011-11-16	09:30:00	10:30:00
□	Edit Inline Edit Copy Delete	102	TRY-02	103	Dept2	Faculty2	S7123456F	Phy	2011-11-30	2011-12-31	11:00:00	12:00:00
□	Edit Inline Edit Copy Delete	104	TRY-03	101	Dept1	Faculty1	S6781234G	Che	0000-00-00	2011-11-10	13:00:00	14:00:00
□	Edit Inline Edit Copy Delete	106	TRY-04	105	Dept4	Faculty3	S3456781E	CSE	2011-11-28	2011-12-27	09:00:00	10:00:00

Check All / Uncheck All *With selected:* Change Delete Export

Show : 30 row(s) starting from row # 0 in horizontal mode and repeat headers after 100 cells

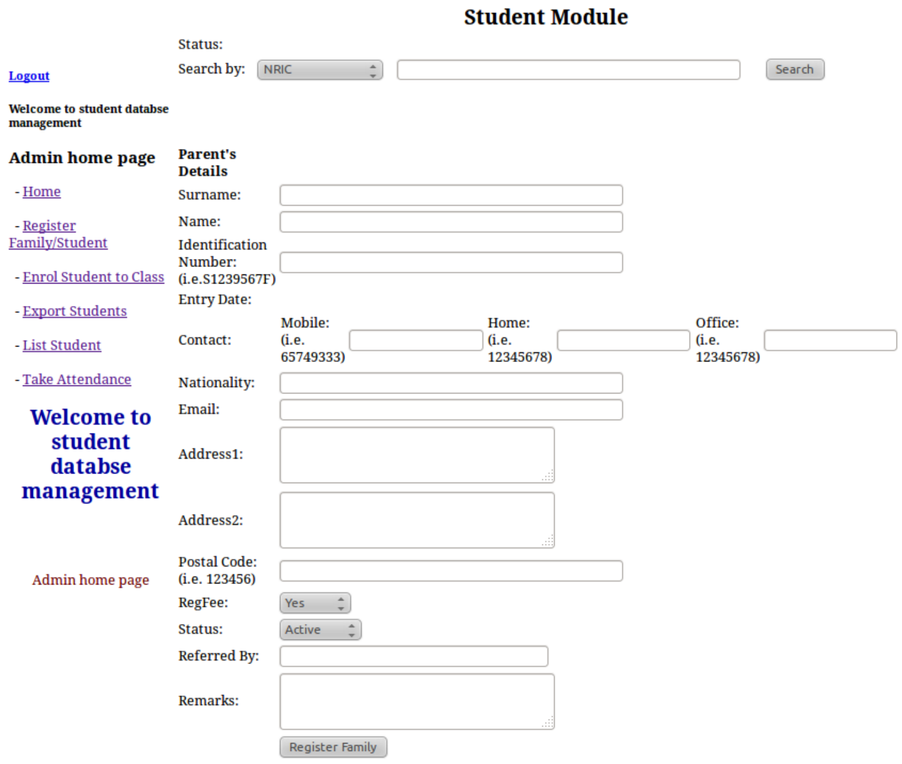
Figure 5.5: Data of one table

Chapter 6

System and Database Testing

6.1 System and database testing

Testing is the last phase of this project. This is the stage where it shows whether the system and database are working as desired. Otherwise, further rectification is required. Tests have been conducted and recorded below to show the results.



Student Module

Status:

Search by: NRIC

Parent's Details

Surname:

Name:

Identification Number: (i.e.S1239567F)

Entry Date:

Contact: Mobile: (i.e. 65749333) Home: (i.e. 12345678) Office: (i.e. 12345678)

Nationality:

Email:

Address1:

Address2:

Postal Code: (i.e. 123456)

RegFee: Yes

Status: Active

Referred By:

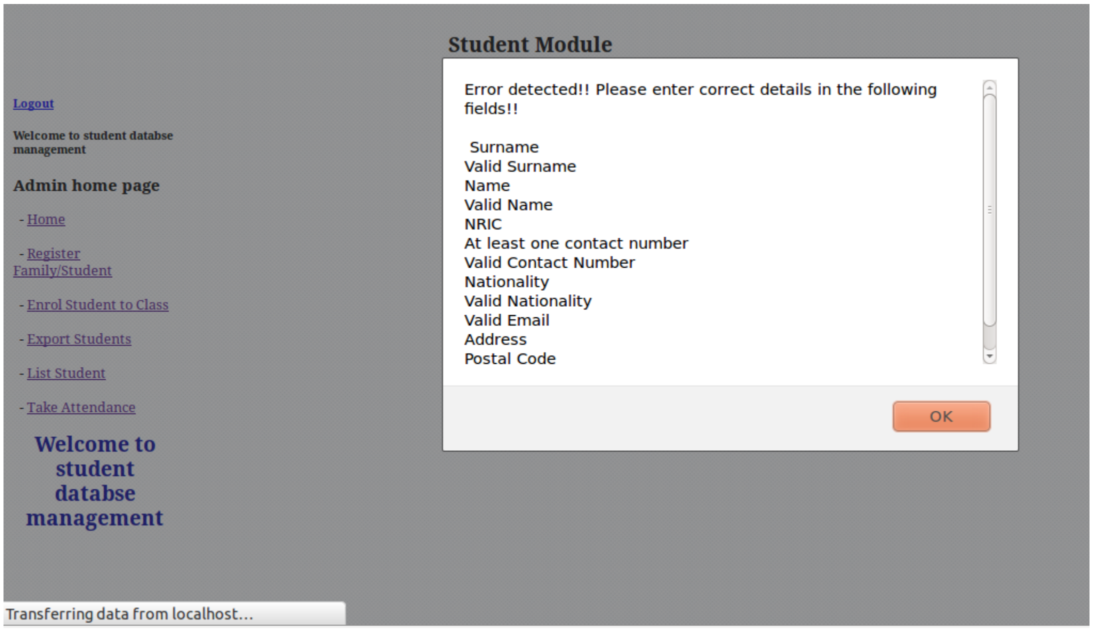
Remarks:

Figure 6.1: Test

Test 1

Purpose: Test the error checking functions
Location: Family Registration
Condition: All mandatory fields need to be filled up

Result



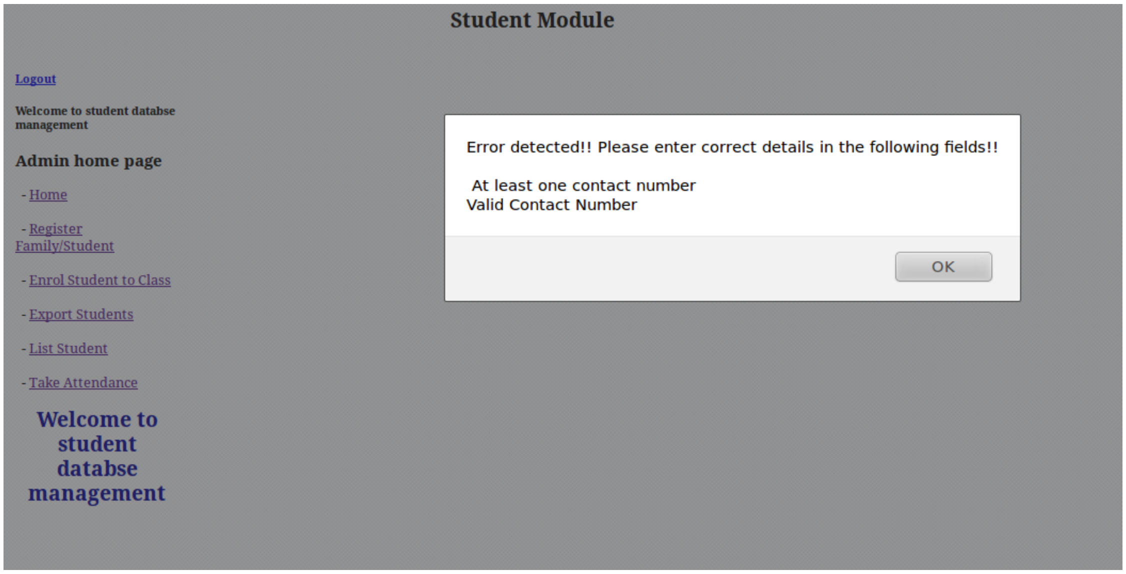
Student Module
Logout
Welcome to student databse management
Admin home page
- Home
- Register Family/Student
- Enrol Student to Class
- Export Students
- List Student
- Take Attendance
Welcome to student databse management
OK
Transferring data from localhost...

Error detected!! Please enter correct details in the following fields!!

Surname
Valid Surname
Name
Valid Name
NRIC
At least one contact number
Valid Contact Number
Nationality
Valid Nationality
Valid Email
Address
Postal Code

Figure 6.2: Test1

Test 2
Purpose: Test the error checking functions
Location: Family Registration
Condition: At least one contact number is required
Result:



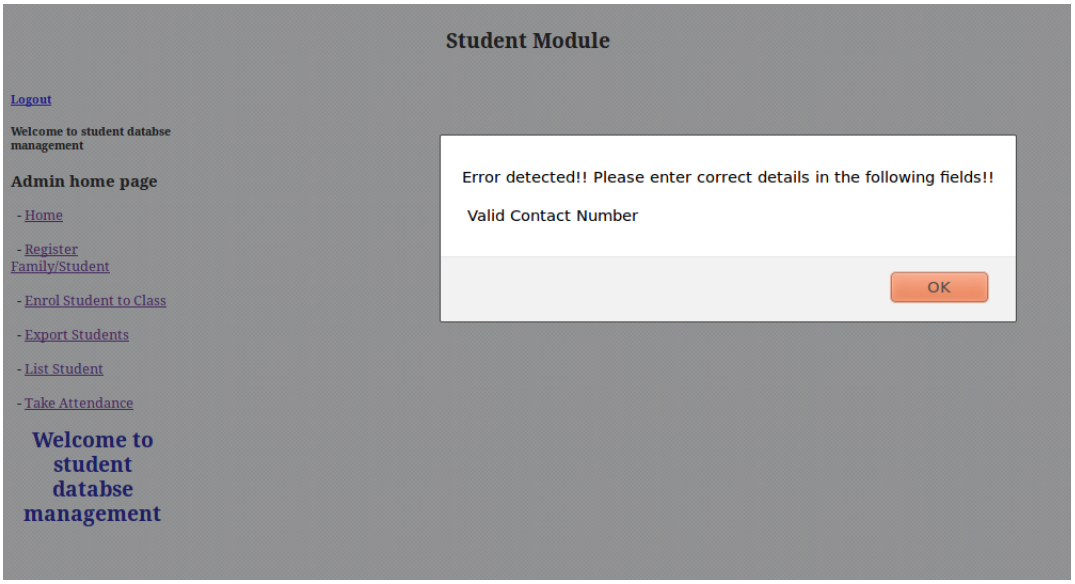
Error detected!! Please enter correct details in the following fields!!

At least one contact number
Valid Contact Number

Figure 6.3: Test2

Test 3

Purpose: Test the error checking functions
Location: Family Registration
Condition: Contact number should contain numbers only
Result:



Error detected!! Please enter correct details in the following fields!!

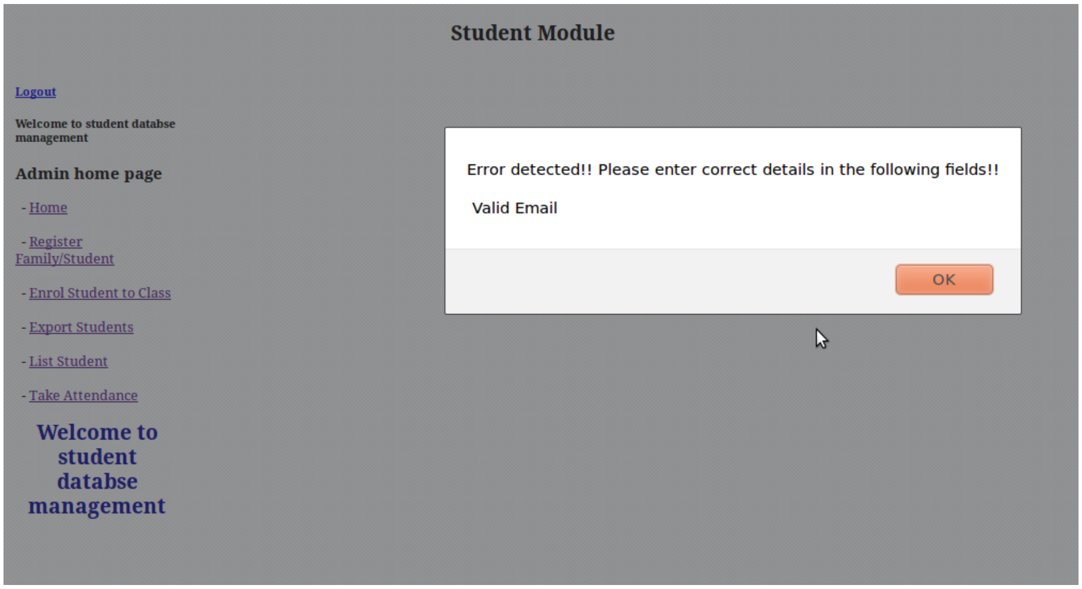
Valid Contact Number

Figure 6.4: Test3

Test 4

Purpose: Test the error checking functions
Location: Family Registration

Condition: Email address must be valid
Result:

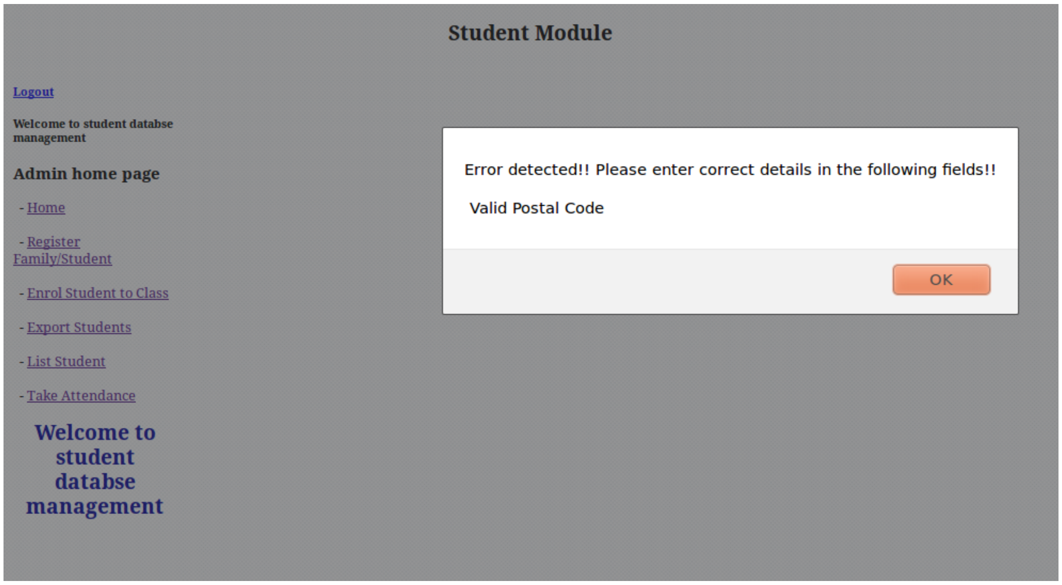


Error detected!! Please enter correct details in the following fields!!

Valid Email

Figure 6.5: Test4

Test 5
Purpose: Test the error checking functions
Location: Family Registration
Condition: Postal Code should contain numbers only
Result:



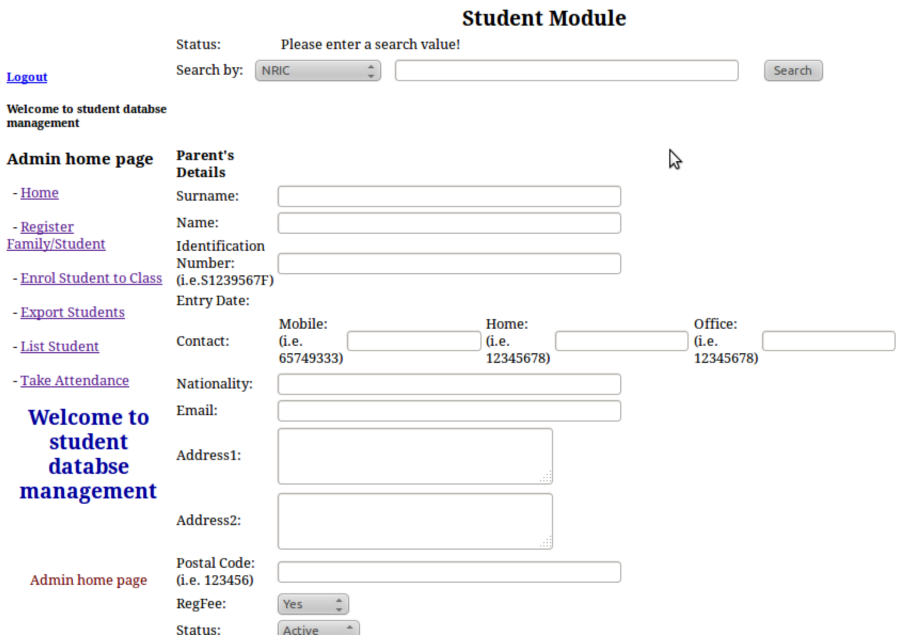
Error detected!! Please enter correct details in the following fields!!

Valid Postal Code

Figure 6.6: Test5

Test 6

Purpose: Test the Search function
Location: Family Registration
Condition: No search value is input
Result:



Student Module

Status: Please enter a search value!

Search by: NRIC

Logout

Welcome to student databse management

Admin home page

- Home
- Register Family/Student
- Enrol Student to Class
- Export Students
- List Student
- Take Attendance

Welcome to student databse management

Admin home page

Parent's Details

Surname:

Name:

Identification Number: (i.e.S1239567F)

Entry Date:

Contact: Mobile: (i.e. 65749333) Home: (i.e. 12345678) Office: (i.e. 12345678)

Nationality:

Email:

Address1:

Address2:

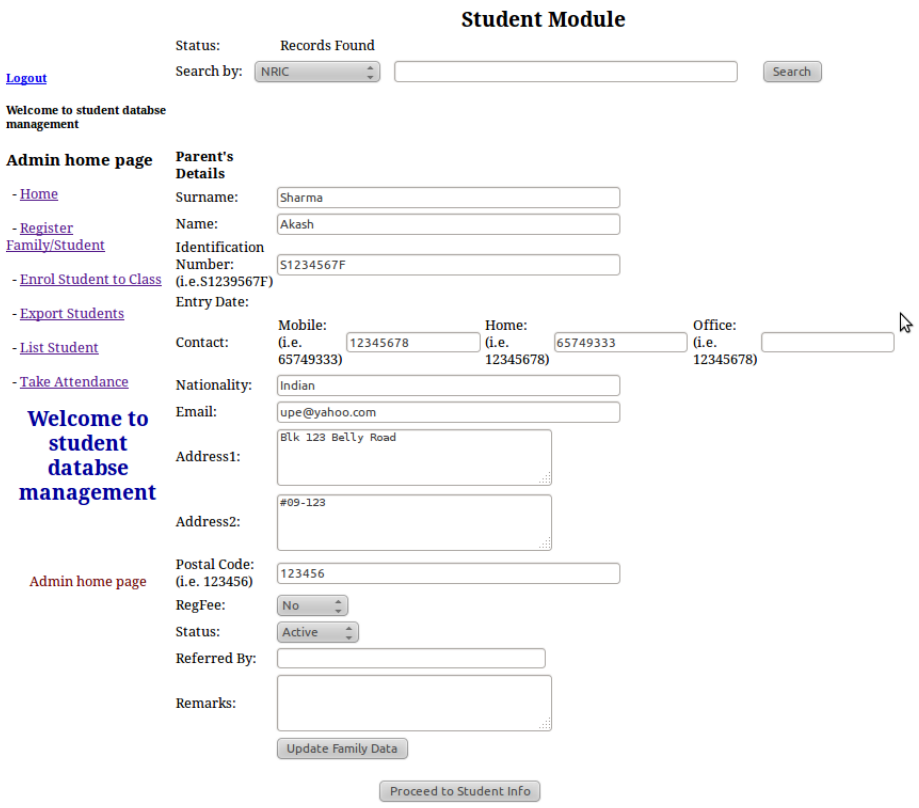
Postal Code: (i.e. 123456)

RegFee: Yes

Status: Active

Figure 6.7: Test6

Test 7
Purpose: Test the Search function
Location: Family Registration
Condition: Search using NRIC
Result:



Student Module

Status: Records Found

Search by: NRIC

Logout

Welcome to student databse management

Admin home page

- Home
- Register Family/Student
- Enrol Student to Class
- Export Students
- List Student
- Take Attendance

Welcome to student databse management

Admin home page

Parent's Details

Surname: Sharma

Name: Akash

Identification Number: (i.e.S1239567F) S1234567F

Entry Date:

Contact: Mobile: (i.e. 65749333) 12345678 Home: (i.e. 12345678) 65749333 Office: (i.e. 12345678)

Nationality: Indian

Email: upe@yahoo.com

Address1: Blk 123 Belly Road

Address2: #09-123

Postal Code: (i.e. 123456) 123456

RegFee: No

Status: Active

Referred By:

Remarks:

Figure 6.8: Test7

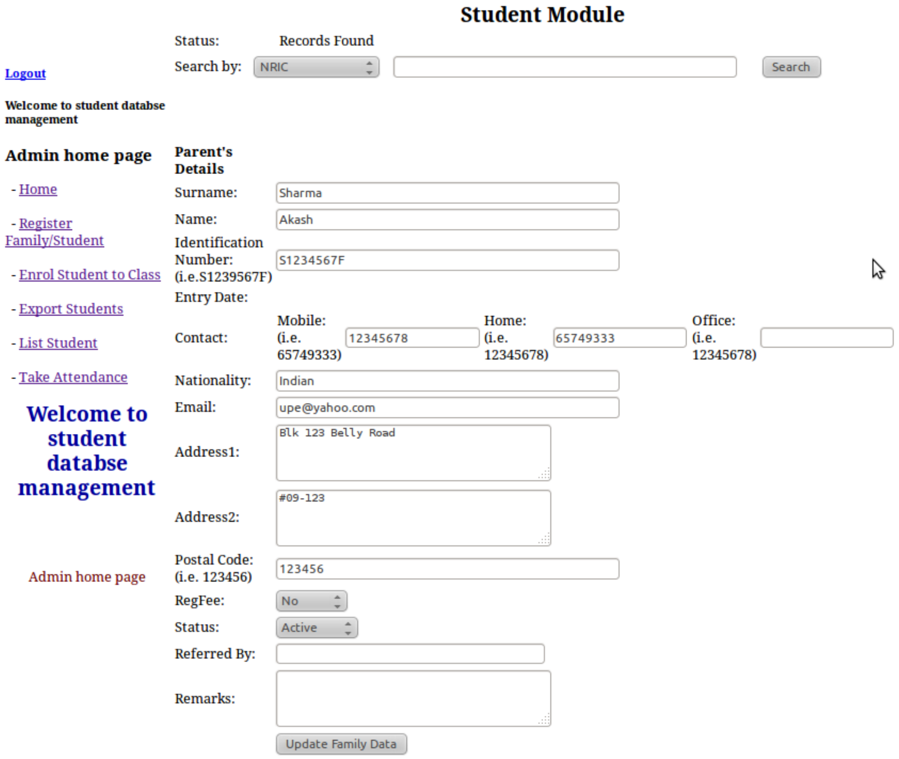
Test 8

Purpose: Test the Search function

Location: Family Registration

Condition: Search using any contact number

Result:



Student Module

Status: Records Found

Search by: NRIC

Parent's Details

Surname: Sharma

Name: Akash

Identification Number: (i.e.S1239567F) S1234567F

Entry Date:

Contact: Mobile: (i.e. 65749333) 12345678 Home: (i.e. 12345678) 65749333 Office: (i.e. 12345678)

Nationality: Indian

Email: upe@yahoo.com

Address1: Blk 123 Belly Road

Address2: #09-123

Postal Code: (i.e. 123456) 123456

RegFee: No

Status: Active

Referred By:

Remarks:

Figure 6.9: Test8

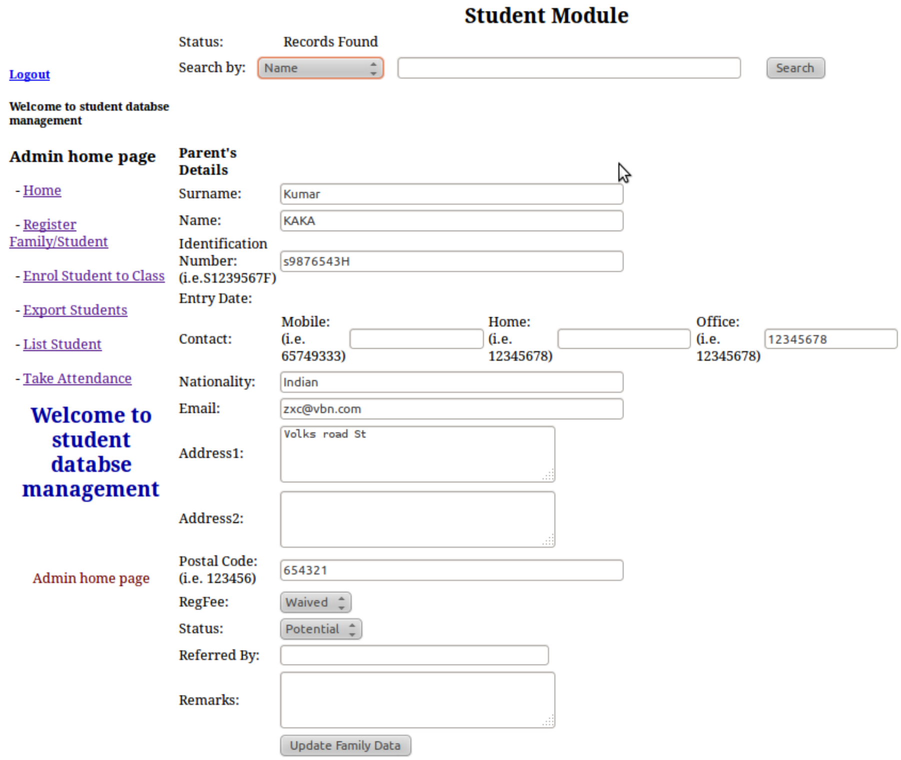
Test 9

Purpose: Test the Search function

Location: Family Registration

Condition: Search using name

Result:



Student Module

Status: Records Found

Search by: Name

Parent's Details

Surname: Kumar

Name: KAKA

Identification Number: (i.e.S1239567F) s9876543H

Entry Date:

Contact: Mobile: (i.e. 65749333) Home: (i.e. 12345678) Office: (i.e. 12345678) 12345678

Nationality: Indian

Email: zxc@vbn.com

Address1: Volks road St

Address2:

Postal Code: (i.e. 123456) 654321

RegFee: Waived

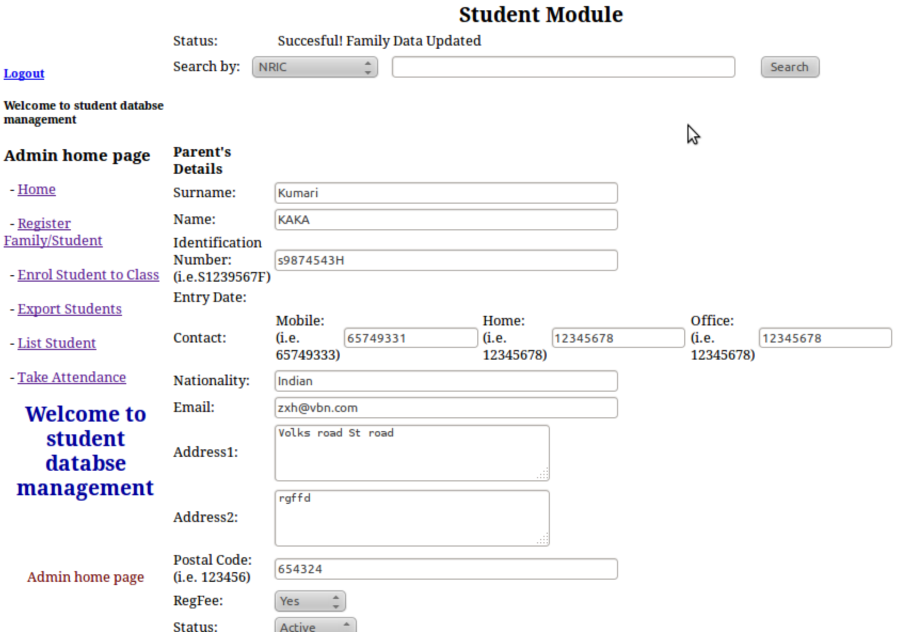
Status: Potential

Referred By:

Remarks:

Figure 6.10: Test9

Test 10
Purpose: Update of family details
Location: Family Registration
Database: Family table
Condition: All data are valid
Result: In the system:



Status: Succesful! Family Data Updated

Search by: NRIC

Parent's Details

Surname: Kumari

Name: KAKA

Identification Number: (i.e.S1239567F) s9874543H

Entry Date:

Contact: Mobile: (i.e. 65749333) 65749331 Home: (i.e. 12345678) 12345678 Office: (i.e. 12345678) 12345678

Nationality: Indian

Email: zxh@vbn.com

Address1: Volks road St road

Address2: rgffd

Postal Code: (i.e. 123456) 654324

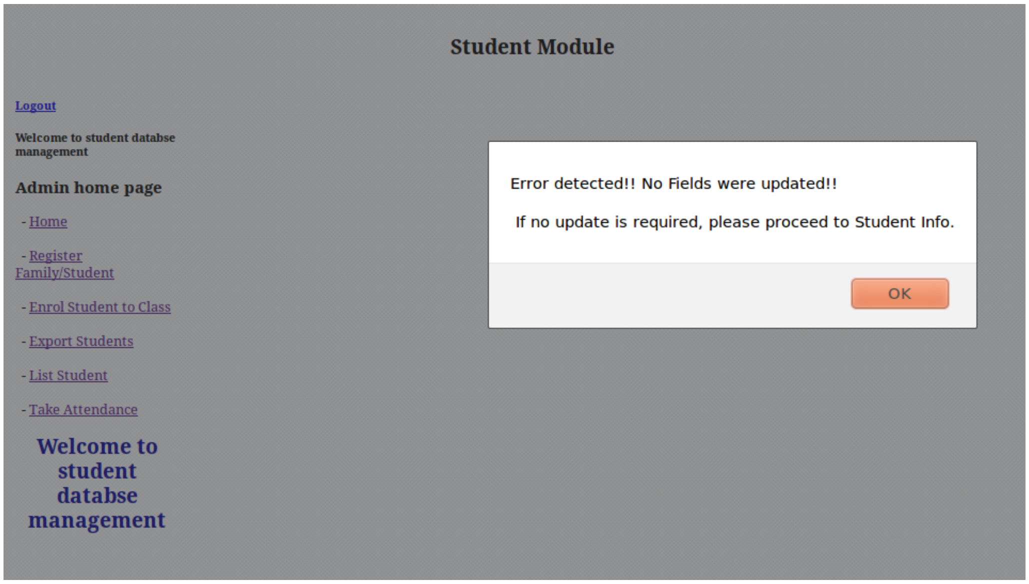
RegFee: Yes

Status: Active

Figure 6.11: Test10

Test 11

Purpose: Update of family details
Location: Family Registration
Database: Family table
Condition: No update is done (status quo)
Result: In the system:



Error detected!! No Fields were updated!!

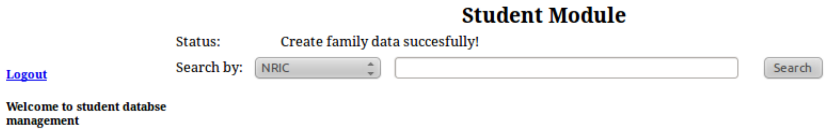
If no update is required, please proceed to Student Info.

Figure 6.12: Test11

Test 12

Purpose: Create family
Location: Family Registration
Database: Family table
Condition: All data are valid

Result:In the system:



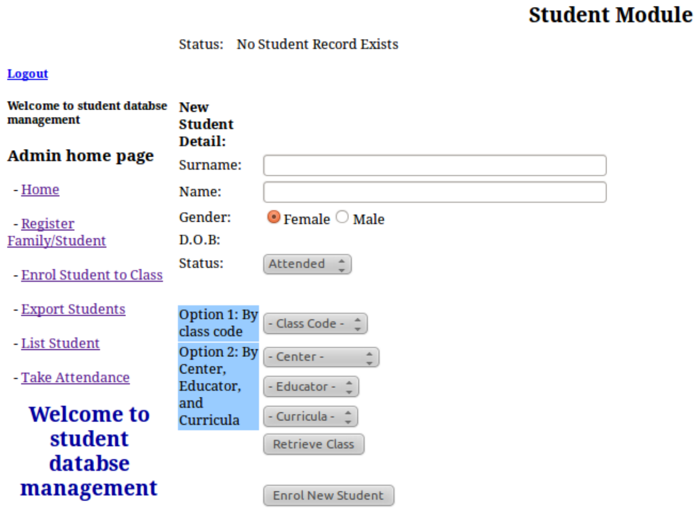
Student Module

Status: Create family data succesfully!

Figure 6.13: Test12

Test 13

Purpose: Create student (After creating the family)
Location: Student Registration
Condition: All mandatory fields are required



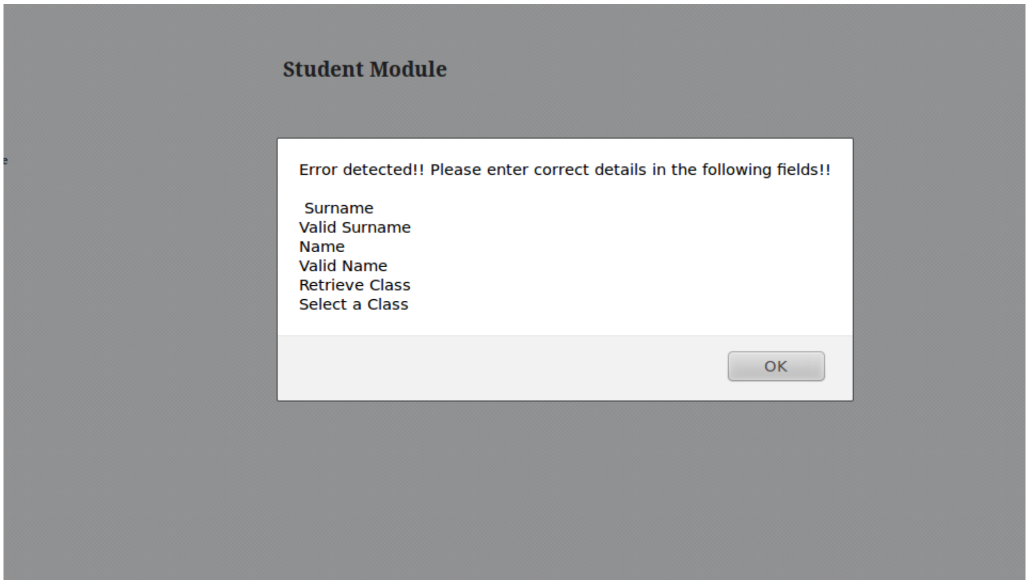
Student Module

Status: No Student Record Exists

Figure 6.14: Test13

Test 14

Purpose: Create student (After creating the family)
Location: Student Registration
Condition: All mandatory fields are required
Result:

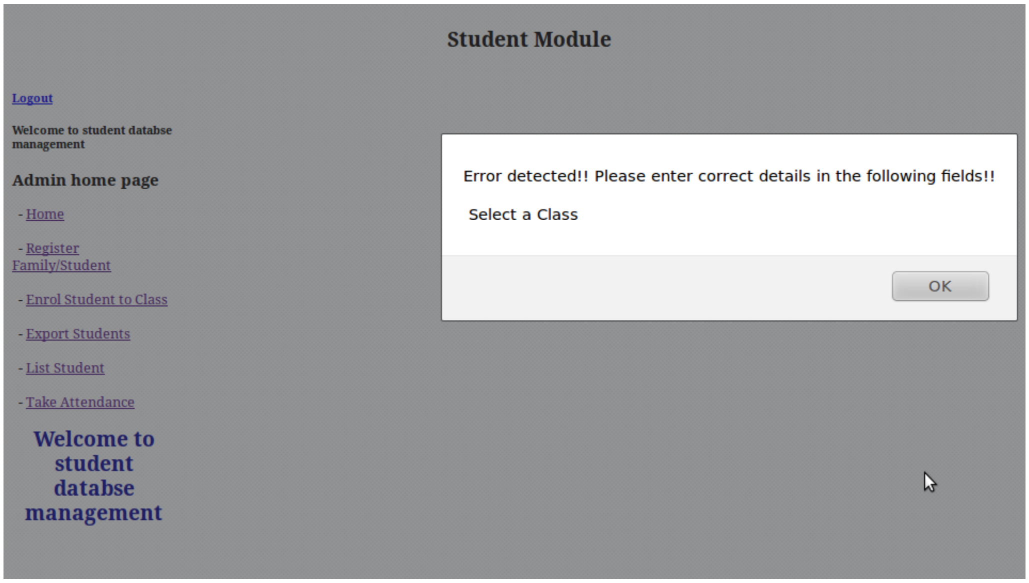


Error detected!! Please enter correct details in the following fields!!

Surname
Valid Surname
Name
Valid Name
Retrieve Class
Select a Class

Figure 6.15: Test14

Test 15
Purpose: Create student (After creating the family)
Location: Student Registration
Condition: Need to select the class after retrieve class
Result:



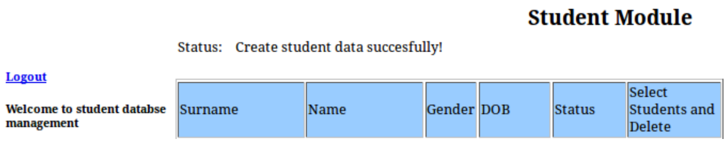
Error detected!! Please enter correct details in the following fields!!

Select a Class

Figure 6.16: Test15

Test 16
Purpose: Create student (After creating the family)
Location: Student Registration

Database: Student and Student-class tables
Condition: All data are valid
Result: In the system:



Student Module

Status: Create student data succesfully!

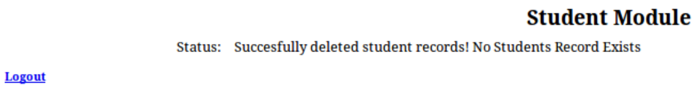
Logout

Welcome to student databse management

Surname	Name	Gender	DOB	Status	Select Students and Delete

Figure 6.17: Test16

Test 17
Purpose: Delete student
Location: Student Registration
Database: Student and Student-class tables
Result: In the system:



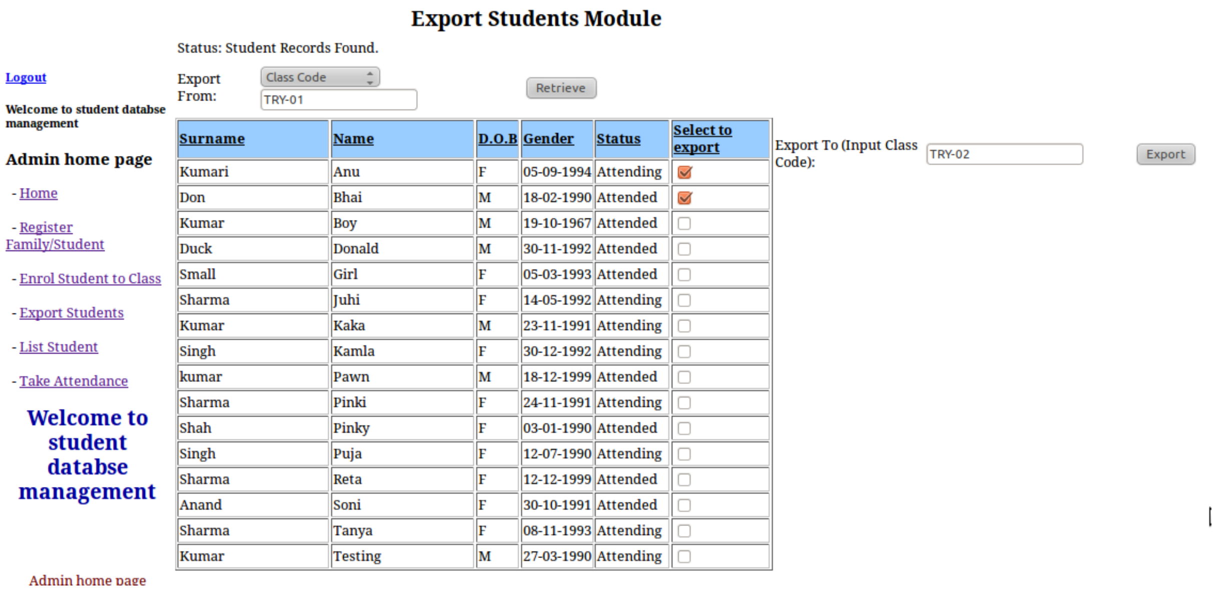
Student Module

Status: Succesfully deleted student records! No Students Record Exists

Logout

Figure 6.18: Test17

Test 18
Purpose: To test the export function
Location: Export students
Condition: A valid class code is required.
Result:



Export Students Module

Status: Student Records Found.

Logout

Welcome to student databse management

Admin home page

- Home
- Register Family/Student
- Enrol Student to Class
- Export Students
- List Student
- Take Attendance

Welcome to student databse management

Admin home page

Export From: Class Code TRY-01 Retrieve

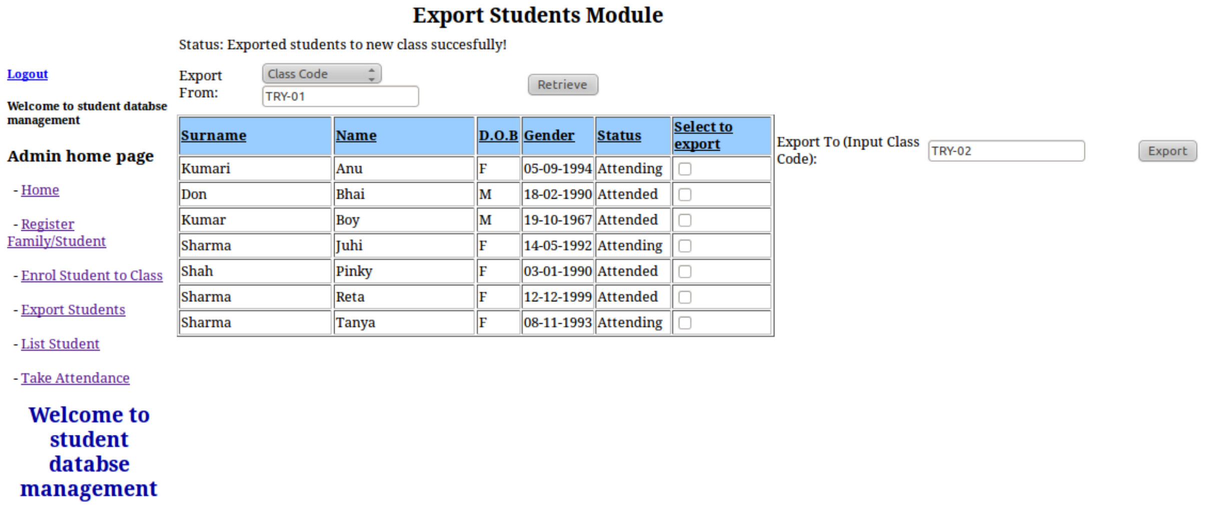
Surname	Name	D.O.B	Gender	Status	Select to export
Kumari	Anu	F	05-09-1994	Attending	☑
Don	Bhai	M	18-02-1990	Attended	☑
Kumar	Boy	M	19-10-1967	Attended	□
Duck	Donald	M	30-11-1992	Attended	□
Small	Girl	F	05-03-1993	Attended	□
Sharma	Juhi	F	14-05-1992	Attending	□
Kumar	Kaka	M	23-11-1991	Attending	□
Singh	Kamla	F	30-12-1992	Attending	□
kumar	Pawn	M	18-12-1999	Attended	□
Sharma	Pinki	F	24-11-1991	Attending	□
Shah	Pinky	F	03-01-1990	Attended	□
Singh	Puja	F	12-07-1990	Attending	□
Sharma	Reta	F	12-12-1999	Attended	□
Anand	Soni	F	30-10-1991	Attended	□
Sharma	Tanya	F	08-11-1993	Attending	□
Kumar	Testing	M	27-03-1990	Attending	□

Export To (Input Class Code): TRY-02 Export

Figure 6.19: Test18

Test 19

Purpose: To test the export function
Location: Export students
Database: Student-class table
Condition: Selected student to be exported from TRY-01 to class TRY-02
Result: In the system



Export Students Module

Status: Exported students to new class succesfully!

Logout

Welcome to student databse management

Admin home page

- Home
- Register Family/Student
- Enrol Student to Class
- Export Students
- List Student
- Take Attendance

Welcome to student databse management

Export From: Class Code TRY-01 Retrieve

Surname	Name	D.O.B	Gender	Status	Select to export
Kumari	Anu	F	05-09-1994	Attending	□
Don	Bhai	M	18-02-1990	Attended	□
Kumar	Boy	M	19-10-1967	Attended	□
Sharma	Juhi	F	14-05-1992	Attending	□
Shah	Pinky	F	03-01-1990	Attended	□
Sharma	Reta	F	12-12-1999	Attended	□
Sharma	Tanya	F	08-11-1993	Attending	□

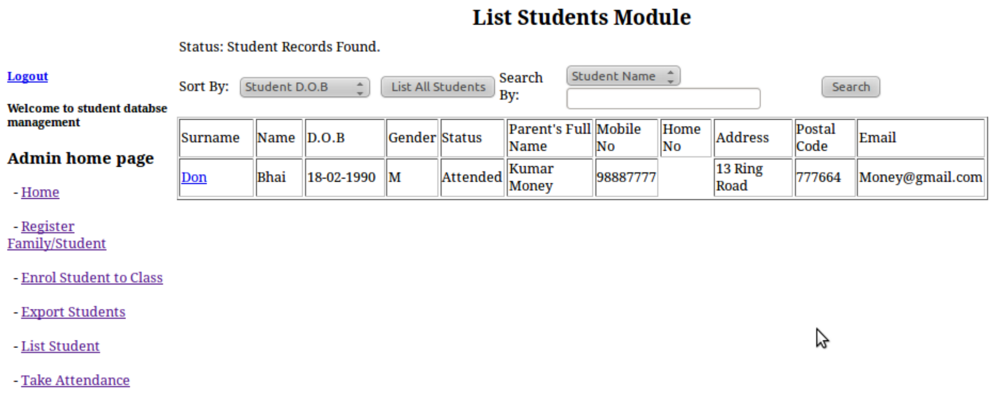
Export To (Input Class Code): TRY-02 Export

Figure 6.20: Test19

Test 20

Purpose: To test the list students function
Location: List Student

Database: Student and Family tables
Condition: Search for students with name 'Bhai' and list out the family information
Result:



List Students Module

Status: Student Records Found.

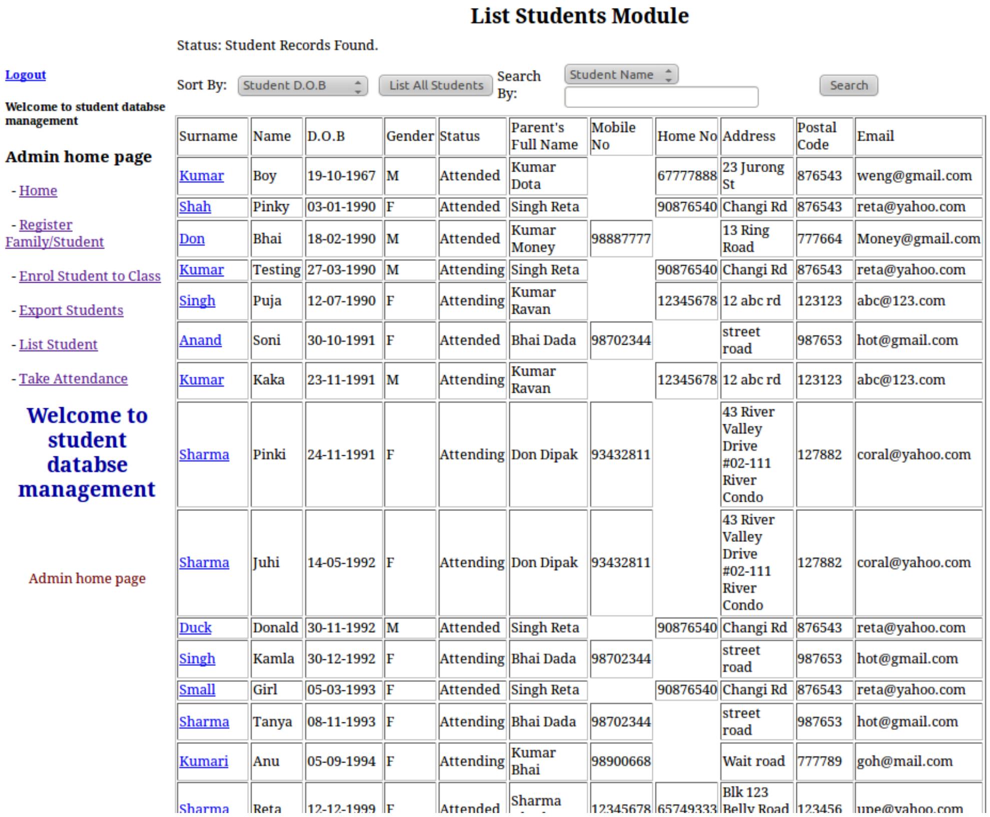
Sort By: Student D.O.B

Search By: Student Name

Surname	Name	D.O.B	Gender	Status	Parent's Full Name	Mobile No	Home No	Address	Postal Code	Email
Don	Bhai	18-02-1990	M	Attended	Kumar Money	98887777		13 Ring Road	777664	Money@gmail.com

Figure 6.21: Test20

Test 21
Purpose: To test the list students function
Location: List Student
Database: Student and Family tables
Condition: List all students in the school by the order of Date of Birth
Result:



List Students Module

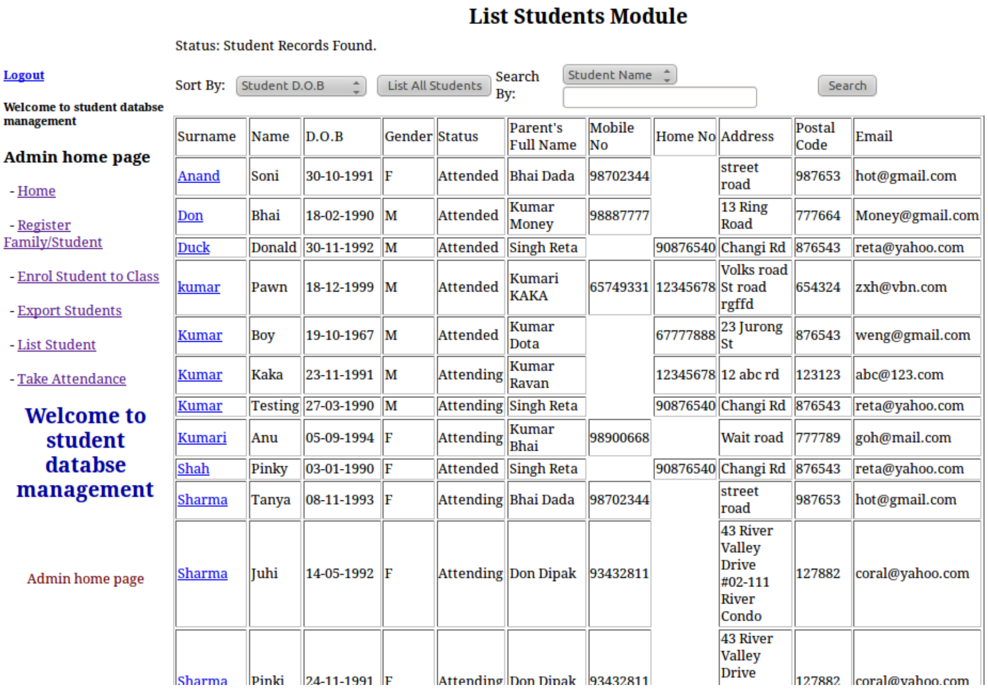
Status: Student Records Found.

Surname	Name	D.O.B	Gender	Status	Parent's Full Name	Mobile No	Home No	Address	Postal Code	Email
Kumar	Boy	19-10-1967	M	Attended	Kumar Dota		67777888	23 Jurong St	876543	weng@gmail.com
Shah	Pinky	03-01-1990	F	Attended	Singh Reta		90876540	Changi Rd	876543	reta@yahoo.com
Don	Bhai	18-02-1990	M	Attended	Kumar Money	98887777		13 Ring Road	777664	Money@gmail.com
Kumar	Testing	27-03-1990	M	Attending	Singh Reta		90876540	Changi Rd	876543	reta@yahoo.com
Singh	Puja	12-07-1990	F	Attending	Kumar Ravan		12345678	12 abc rd	123123	abc@123.com
Anand	Soni	30-10-1991	F	Attended	Bhai Dada	98702344		street road	987653	hot@gmail.com
Kumar	Kaka	23-11-1991	M	Attending	Kumar Ravan		12345678	12 abc rd	123123	abc@123.com
Sharma	Pinki	24-11-1991	F	Attending	Don Dipak	93432811		43 River Valley Drive #02-111 River Condo	127882	coral@yahoo.com
Sharma	Juhi	14-05-1992	F	Attending	Don Dipak	93432811		43 River Valley Drive #02-111 River Condo	127882	coral@yahoo.com
Duck	Donald	30-11-1992	M	Attended	Singh Reta		90876540	Changi Rd	876543	reta@yahoo.com
Singh	Kamla	30-12-1992	F	Attending	Bhai Dada	98702344		street road	987653	hot@gmail.com
Small	Girl	05-03-1993	F	Attended	Singh Reta		90876540	Changi Rd	876543	reta@yahoo.com
Sharma	Tanya	08-11-1993	F	Attending	Bhai Dada	98702344		street road	987653	hot@gmail.com
Kumari	Anu	05-09-1994	F	Attending	Kumar Bhai	98900668		Wait road	777789	goh@mail.com
Sharma	Reta	12-12-1999	F	Attended	Sharma	12345678	65749333	Blk 123 Belly Road	123456	upe@yahoo.com

Figure 6.22: Test21

Test 22

Purpose: To test the list students function
Location: List Student
Database: Student and Family tables
Condition: List all students in the school by the order of Surname
Result:



List Students Module

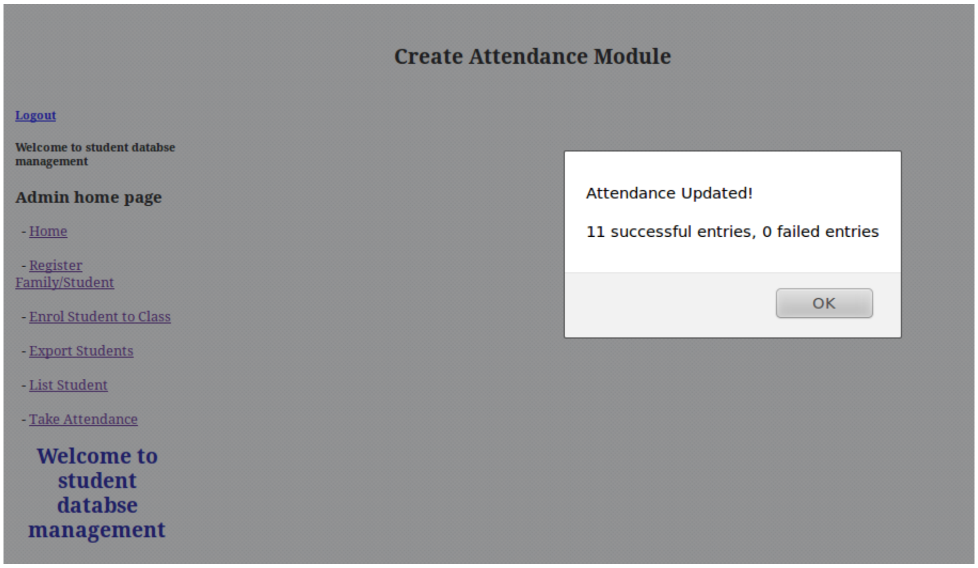
Status: Student Records Found.

Surname	Name	D.O.B	Gender	Status	Parent's Full Name	Mobile No	Home No	Address	Postal Code	Email
Anand	Soni	30-10-1991	F	Attended	Bhai Dada	98702344		street road	987653	hot@gmail.com
Don	Bhai	18-02-1990	M	Attended	Kumar Money	98887777		13 Ring Road	777664	Money@gmail.com
Duck	Donald	30-11-1992	M	Attended	Singh Reta		90876540	Changi Rd	876543	reta@yahoo.com
kumar	Pawn	18-12-1999	M	Attended	Kumari KAKA	65749331	12345678	Volks road St road rgffd	654324	zxh@vbn.com
Kumar	Boy	19-10-1967	M	Attended	Kumar Dota		67777888	23 Jurong St	876543	weng@gmail.com
Kumar	Kaka	23-11-1991	M	Attending	Kumar Ravan		12345678	12 abc rd	123123	abc@123.com
Kumar	Testing	27-03-1990	M	Attending	Singh Reta		90876540	Changi Rd	876543	reta@yahoo.com
Kumari	Anu	05-09-1994	F	Attending	Kumar Bhai	98900668		Wait road	777789	goh@mail.com
Shah	Pinky	03-01-1990	F	Attended	Singh Reta		90876540	Changi Rd	876543	reta@yahoo.com
Sharma	Tanya	08-11-1993	F	Attending	Bhai Dada	98702344		street road	987653	hot@gmail.com
Sharma	Juhi	14-05-1992	F	Attending	Don Dipak	93432811		43 River Valley Drive #02-111 River Condo	127882	coral@yahoo.com
Sharma	Pinki	24-11-1991	F	Attending	Don Dipak	93432811		43 River Valley Drive	127882	coral@yahoo.com

Figure 6.23: Test22

Test 23

Purpose: To test the attendance taking function
Location: Take Attendance
Database: Attendance table
Result: In the system:



Create Attendance Module

Attendance Updated!

11 successful entries, 0 failed entries

Figure 6.24: Test23

Chapter 7

Project Management

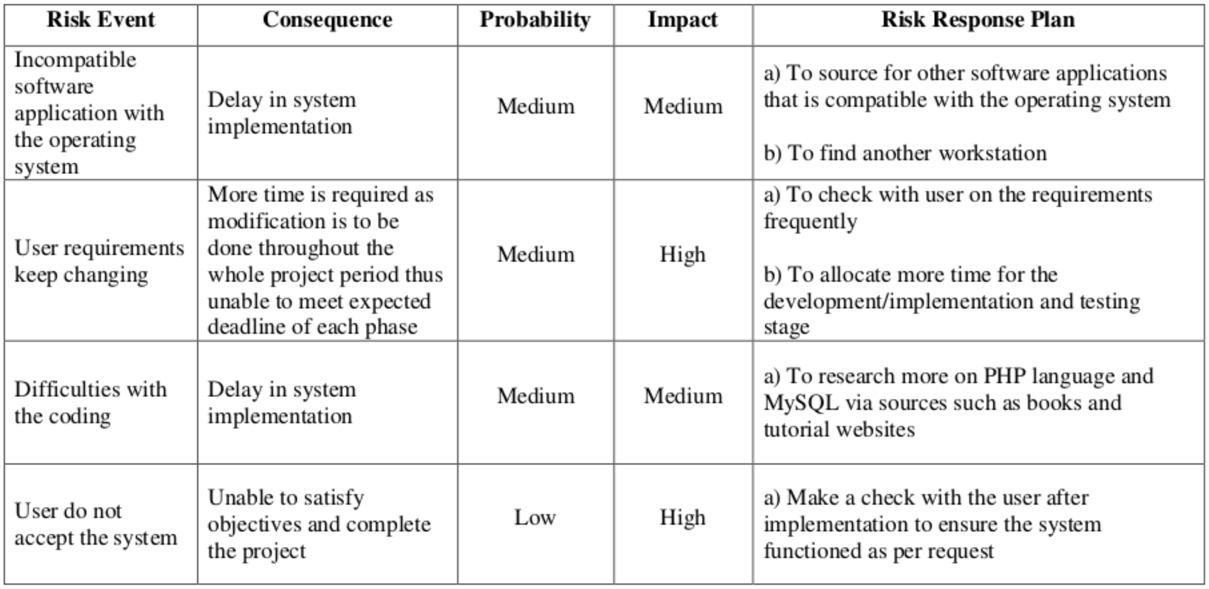
7.1 Work breakdown structure

The Work Breakdown Structure provides a clear picture of the items that are required to be completed for the project work scope. It is arranged in a tree structure manner to show all sub-sections required to be accomplished in each phase throughout the project.

7.2 Risk management

Every project has uncertainties and they may affect the project success. These uncertainties are known as risk. There is a need to determine the risks involved in the project to reduce the chance of its occurrence as well as develop a plan to either mitigate or avoid the risks that have been identified.

After identifying and determining the risks, they are recorded in a table called the risk assessment matrix. This matrix lists the severity which is also known as the impact level in terms of High, Medium and Low. In addition to those details, the action plan that can be taken for each of the risks is clearly stated too.



Risk Event	Consequence	Probability	Impact	Risk Response Plan
Incompatible software application with the operating system	Delay in system implementation	Medium	Medium	a) To source for other software applications that is compatible with the operating system b) To find another workstation
User requirements keep changing	More time is required as modification is to be done throughout the whole project period thus unable to meet expected deadline of each phase	Medium	High	a) To check with user on the requirements frequently b) To allocate more time for the development/implementation and testing stage
Difficulties with the coding	Delay in system implementation	Medium	Medium	a) To research more on PHP language and MySQL via sources such as books and tutorial websites
User do not accept the system	Unable to satisfy objectives and complete the project	Low	High	a) Make a check with the user after implementation to ensure the system functioned as per request

Figure 7.1: Risk Assessment Matrix

Chapter 8

Conclusions

8.1 Conclusions

Simplicity is never simple. As we have seen in this project, the process of creating a user-friendly and straightforward platform that facilitates the administrator's job is one filled with complexity. From understanding user requirements to system design and finally system prototype and finalization, every step requires in-depth understanding and commitment towards achieving the objectives of the project.

Although the student database management module is not fully integrated to the system and used on real time, the system prototype demonstrates easy navigation and data are stored in a systematic way. Overall, efficiency has improved and work processes simplified. Although all the objectives have been met, the system still has room for improvement. The system is robust and flexible enough for future upgrade using advanced technology and devices.

Chapter 9

Critical Reviews

9.1 Critical reviews

In the process of doing this project, obstacles have been encountered. I have learnt to be more flexible in resolving issues. One of the situations that I have faced was when I had installed MySQL server, but it did not work. Then I tried to troubleshoot, research online and did a couple of re-installation but to no avail. Then I installed XAMPP, it did not work out too. Some how I did get rid of this problem but I did not like XAMPP server. Therefore I decided to install LAMP server. Yes it worked fine. At the end I found that I had installed multiple server and hence system was confused which server is to be used i.e. conflicts with MySQL. Then I stopped all server running on my laptop and decicded to start only LAMP server and it worked fine.

Being a novice in PHP language, I had difficulty in understanding the basic to write php files. Therefore I took longer time to understand how and why PHP works. I researched too much online. Went through several examples, tutorials etc.

Still I am not good enough in PHP language. In this entire project I have used only PHP and HTML languages. but later I realized that there are multiple laguages which can support in designing a good database project. Those are CSS, JavaScript. There may be more but trully speeking I really don't know any more.

Eventually with more hands-on exposure throughout this project, I managed to gain some understanding of PHP language.

Discipline played an important role for the completion of this project. Using the project management skills taught in class, I was able to apply them to this project.

Besides project management skills, I had also applied the database knowledge that I had learnt in the class to this project. From analysing of database structure and their relations to creating the database and tables, all these have strengthened my understanding. However when doing part of the codes, I would still research online and find solutions, for example, the usage of 'Ignore' command to avoid duplication of data from inserting into the database. This command did not work on MyISAM storage engine thus I had to change my database storage to InnoDB in order for that to work.

Overall though the project was filled with challenges, it had given me opportunity to apply and strengthen my knowledge as well as gain new knowledge and skill through researches and experienced professionals whom I am fortunate enough to encounter.