03-135142-014   MUHAMMAD MUSAB SOHAIL

03-135142-007   HAFIZ AHMAD HASSAN BUTT

Autovisor

In partial fulfilment of the requirements for the degree of

Bachelor of Science in Information Technology

Supervisor: Mr. Taimoor Aamer

Department of Computer Sciences

Bahria University, Lahore Campus

June 2018

C e r t i f i c a t e

We accept the work contained in the report titled

"AUTOVISOR",

written by

MUHAMMAD MUSAB SOHAIL

HAFIZ AHMAD HASSAN BUTT

as a confirmation to the required standard for the partial fulfilment of the degree of Bachelor of Science in Information Technology.

Approved by:

Supervisor: Mr. Taimoor Aamer

____ (Signature)

June 4th, 2018

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-135142-014	MUHAMMAD MUSAB SOHAIL	
03-135142-007	HAFIZ AHMAD HASSAN BUTT	

Date : ____________________________

Specially dedicated to

my beloved grandmother, mother and father

(Muhammad Musab Sohail)

my beloved grandmother, mother and father

(Hafiz Ahmad Hassan Butt)

my beloved grandmother, mother and father

ACKNOWLEDGEMENTS

We would like to thank everyone who had contributed to the successful completion of this project. We would like to express my gratitude to my research supervisor, Mr. Taimoor Aamer for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, We would also like to express our gratitude to our loving parent and friends who had helped and given us encouragement.

Muhammad Musab Sohail
Hafiz Ahmad Hassan Butt

AUTOVISOR

ABSTRACT

Autovisor is providing a solution for student advisor related activities. The main objective of this project is to automate the activities of student advisor which are being done manually and to save the time and extra effort made by them. We have used different tools and platforms to develop this project which includes .Net framework and databases. Autovisor will make student advisors more productive as many of the time-consuming work will be done through this software. This project is being developed using Microsoft Visual Studio IDE.

This project has some of the processes which are almost handled manually at many places. But Autovisor aims to automate them and provides a solution which will change the way of doing the work. Those manual processes include, time tabling, managing makeup classes, sending notifications, keeping records and announcements. Now all these manual processes will be handled through a desktop application written in C#.Net language.

Autovisor can keep the records of students and teacher and can perform CRUD actions. Also it can keep the records of applications from students and on approval or disapproval it sends a notification to student. It can generate individual semester time tables by the help of algorithms. It can also search the individual room time table for every semester. It can also generate an individual time table for teachers. User can send emails and SMS by using this software. The audience of this system is specific as the system can only be used by student advisors. Recommendations for development in future and conclusions of the project are also included in the report.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

CRUD	Create, Remove, Update, Delete
AVS	AutoVisor
IDE	Integrated Development Environment
PDF	Portable Document Format
DB	DataBase
PC	Personal Computer
GUI	Graphical User Interface
C#	C Sharp language
RAM	Random Access Memory
API	Application Programming Interface
GB	GigaByte
AVS	AutoVisor
UML	Universal Model Language
ERD	Entity Relationship Diagram
UC	Usecase
ISO	International Standards Organization
ECMA	European Computer Manufacturers Association
SQL	Structured Query Language
ANSI	American National Standard Institute
IDE	Integrated Development Environment
DBMS	Database Management System
OS	Operating Systems

CHAPTER 1

INTRODUCTION

1.1 Background

Most of the processes in institutes are computerized but still some of them are still being done manually because of inherent difficulties. The manual work demands more effort and more time. So, in most institutes, different processes like time table scheduling and keeping the records are still mostly done manually. Student advisors does not find any particular place where they can integrate their tasks and do them on the go. They do have some very hectic routines because of the work load and the nature of their job as they work as a bridge between administration, teachers and students. Many solutions to this problem are already there but no solution was integrating two or more things

So, the main aim of this project is to first automate maximum processes and then integrate them at one place. Every institute has its own requirements and problems but we tried to make this software as general as possible so it can target maximum audience. Giving solution of the problem can save so much time and effort. Although many solutions are available to automate some of the processes but not a single one is available which could integrate all the processes. So, this project will be an initiative to integrate all those activities. [1]

1.2 Problem Statements

In a world, where from your phone to your watch, everything is getting smart but talking about your day to day activities which remain unjustly manual.

Here, the activities related to student advisors in most of the institutes are being executed manually, and are not integrated. Which costs a lot of time and effort. So keeping in view of these difficulties there should be a platform where all these situations could be cater i.e. automating the student advisor activities and integrating them in one place.

1.3 Aims and Objectives

The objectives of this project are shown as following:

i) To eliminate the previous manual system of student advisor.

ii) To get a platform to integrate all student advisor activities.

iii) To get a platform where all the processes are automated.

1.4 Scope of Project

Autovisor is a desktop application developed by using C#. Net language. Autovisor aims to provide solution for the manual activities related to student advisors. It has some of the features which have been automated and integrated at one place. First of all, Autovisor can register a student, can delete, search and update a student record in a database from the application. Also it can do same for teacher i.e. register, update, search and delete a teacher in database using the Autovisor application. And same goes for the courses, it can register, update and delete a course from DB and can view all offered courses. In addition to this, Autovisor can send emails and SMS to students, teachers and all concerned staff from within the application. It can also keep record of application received from students and on approval and disapproval it sends a message to the student. Also we can do announcement to specific audience through this application. Autovisor can also generate time table for a semester using algorithm which ensures to only have 2 classes for a subject in a week. After generating the time table, you can view it, export it in PDF, can print it and send it through email. A user can also view individual time tables for rooms and also for individual teachers. So, the basic aim or objective for this application is to introduce IT in manual processes and to integrate all those processes at one place to minimize the work load and save the time.

CHAPTER 2

SRS

2.1 Introduction

The purpose of this chapter is to elaborate an in depth description of automatic system “Autovisor”. it will justify the aim and features of the software system, the interfaces of the software system, what the software system can do and therefore the constraints beneath that it should operate. This chapter is meant for users of the software system and additionally potential developers.

2.2 Overall Description

2.2.1 User Classes and Characteristics

i) Typical User, like student advisors, who need to use Autovisor for allocating the class rooms, generating the time tables and keeping records.

ii) Indirect Users, such as student/ teachers and HOD, who want to use Autovisor for different purposes.

iii) Programmers, who are interested in working on the project by further developing it or fix existing bugs.

2.2.2 Operating Environment

Followings are the best operating environments for Autovisor.

i) Windows Vista.
ii) Windows 7.
iii) Windows 8.
iv) Windows 8.1.
v) Windows 10

2.2.3 Design and Implementation Constraints

Autovisor is developed in C#, it uses SQL for its database development and has been built using the Microsoft SQL Server. For C# development, it uses Microsoft Visual Studio as a platform. It uses a modular design where every feature is wrapped into a separate module and the modules depend on each other. It is a desktop application with a front-end application and a back-end, which is the database.

2.2.4 Assumptions and Dependencies

Autovisor is developed only for one platform which is windows. So that means it is platform dependent. The user who will be using windows on their PC, could be able to run the application. Also, the database will be placed on the same system. So, to see the records, database must be in the same PC.

2.3 External Interface Requirements

2.3.1 User Interfaces

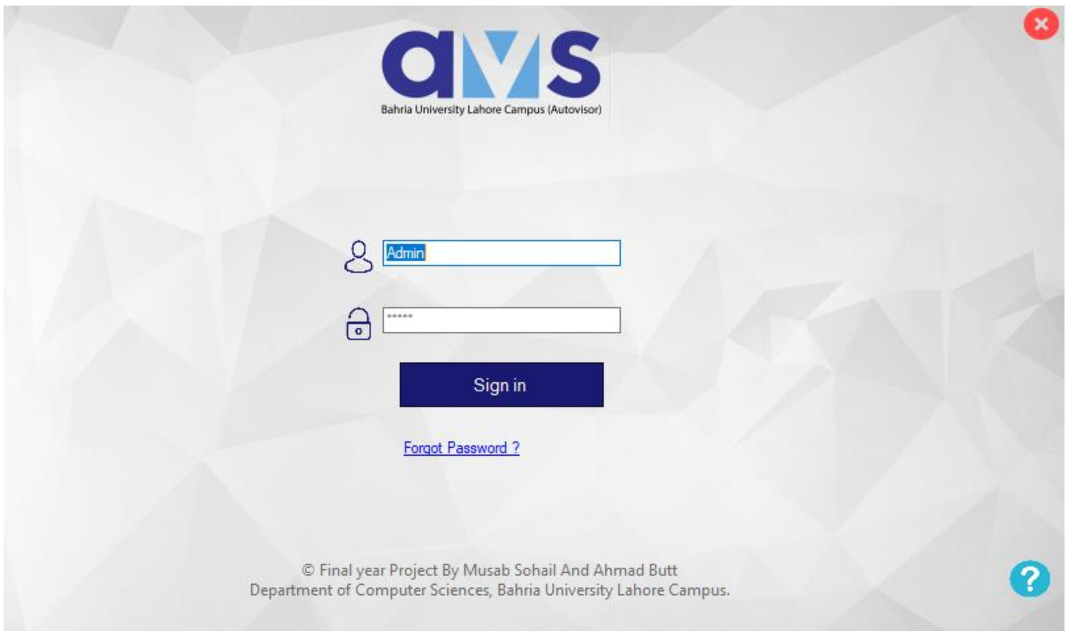

aMS
Bahria University Lahore Campus (Autovisor)
Admin
*****
Sign in
Forgot Password ?
© Final year Project By Musab Sohail And Ahmad Butt
Department of Computer Sciences, Bahria University Lahore Campus.

Figure 2.2.1: Login Screen

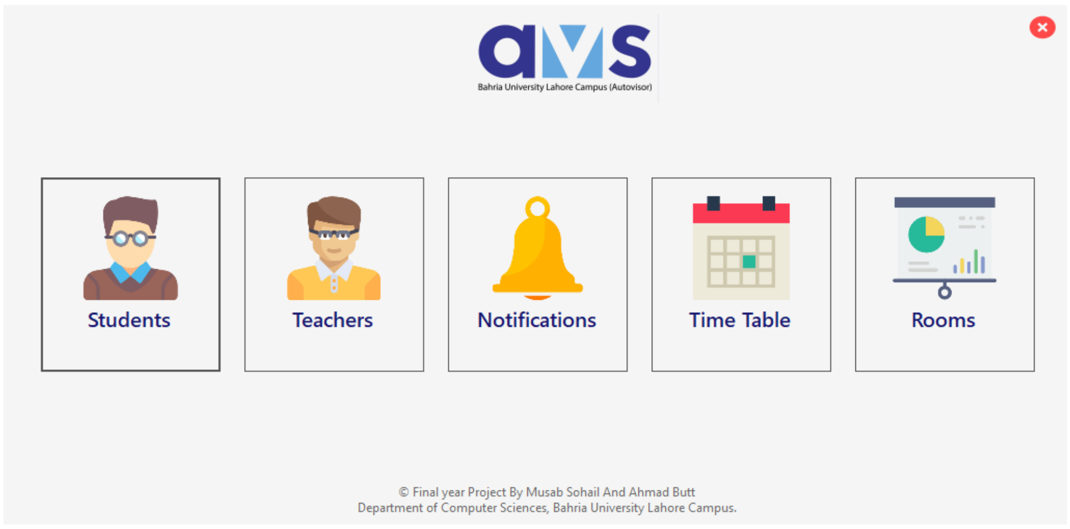
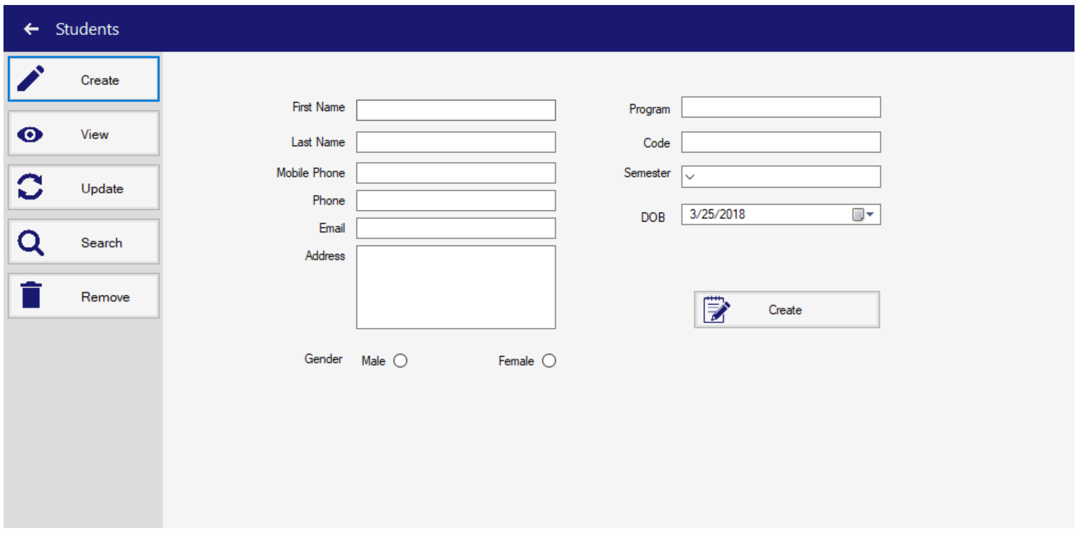

aMS
Bahria University Lahore Campus (Autovisor)
Students
Teachers
Notifications
Time Table
Rooms
© Final year Project By Musab Sohail And Ahmad Butt
Department of Computer Sciences, Bahria University Lahore Campus.

Figure 2.2: Main Menu



DOB 3/25/2018

Figure 2.3: Create Students

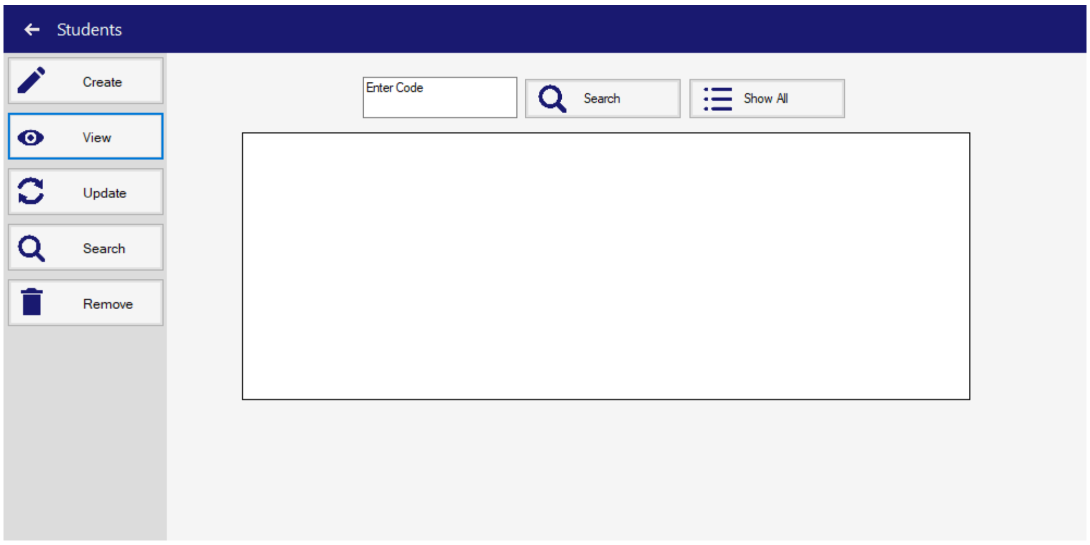


Figure 2.4: View Student

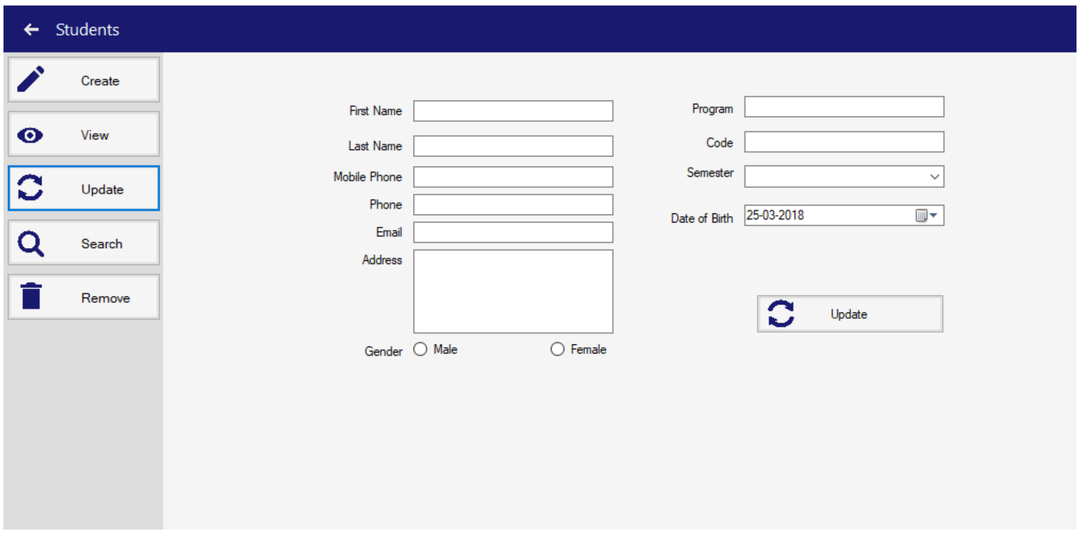


Figure 2.5: Update Student

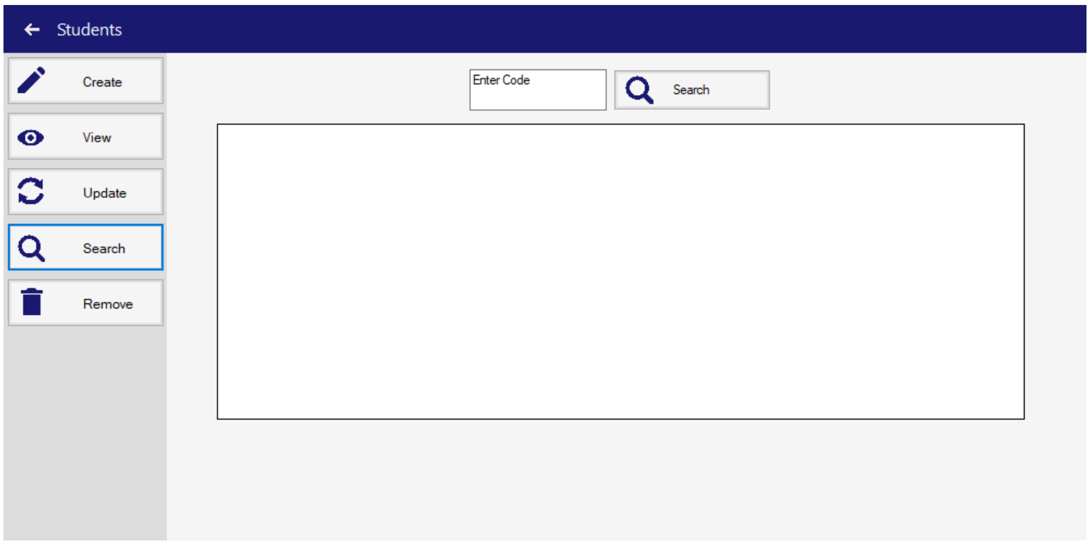


Figure 2.6: Search Student

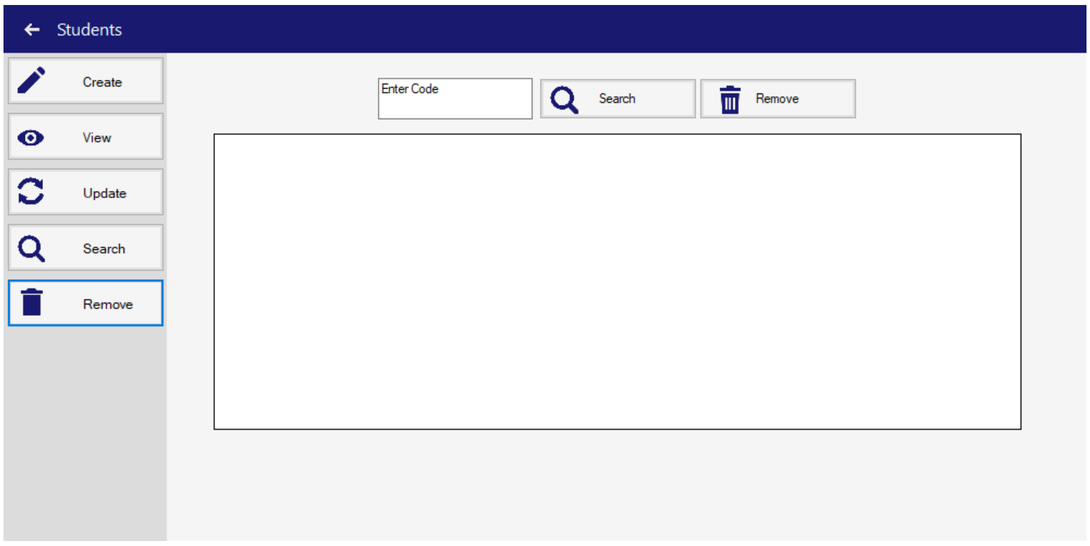


Figure 2.7: Remove Student

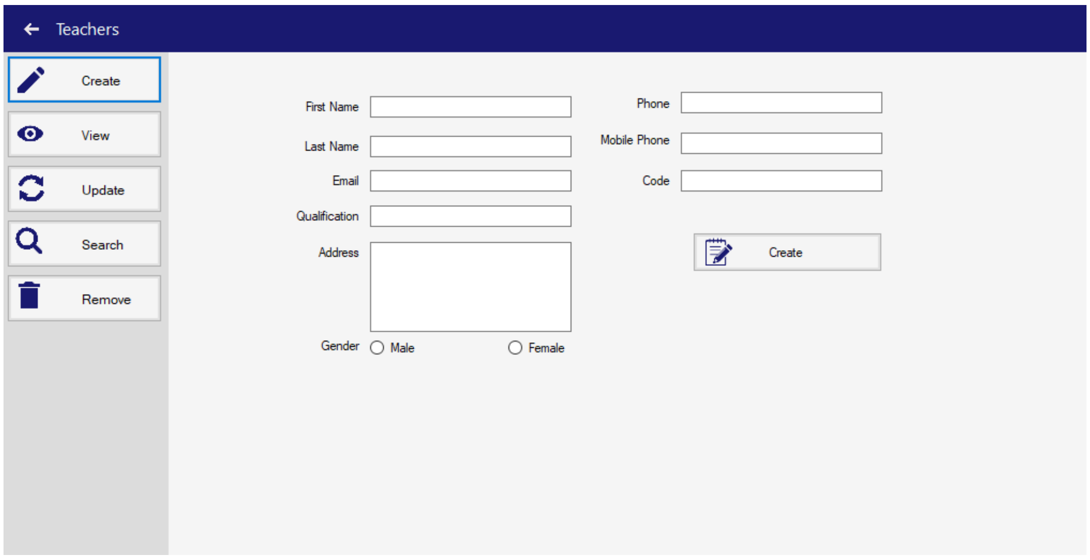
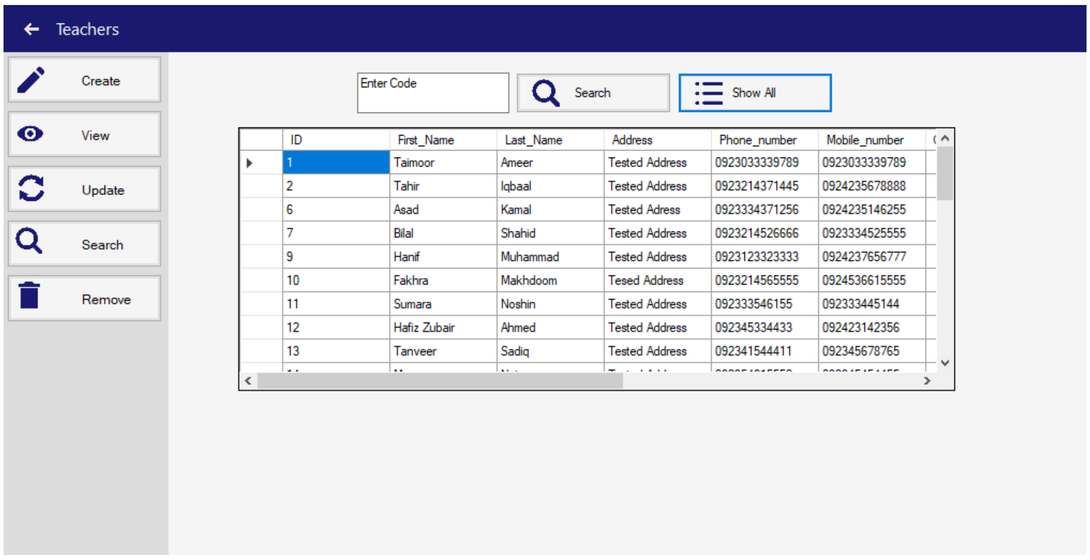


Figure 2.8: Create Teacher



ID	First_Name	Last_Name	Address	Phone_number	Mobile_number
1	Taimoor	Ameer	Tested Address	0923033339789	0923033339789
2	Tahir	Iqbaal	Tested Address	0923214371445	0924235678888
6	Asad	Kamal	Tested Adress	0923334371256	0924235146255
7	Bilal	Shahid	Tested Address	0923214526666	0923334525555
9	Hanif	Muhammad	Tested Address	0923123323333	0924237656777
10	Fakhra	Makhdoom	Tesed Address	0923214565555	0924536615555
11	Sumara	Noshin	Tested Address	092333546155	092333445144
12	Hafiz Zubair	Ahmed	Tested Address	092345334433	092423142356
13	Tanveer	Sadiq	Tested Address	092341544411	092345678765

Figure 2.9: View Teacher

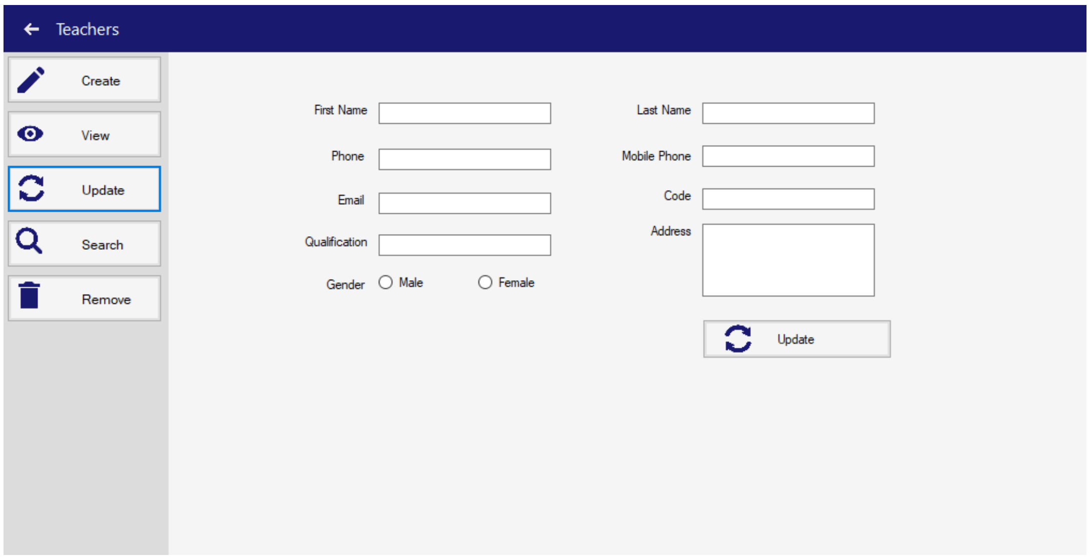


Figure 2.10: Update Teacher

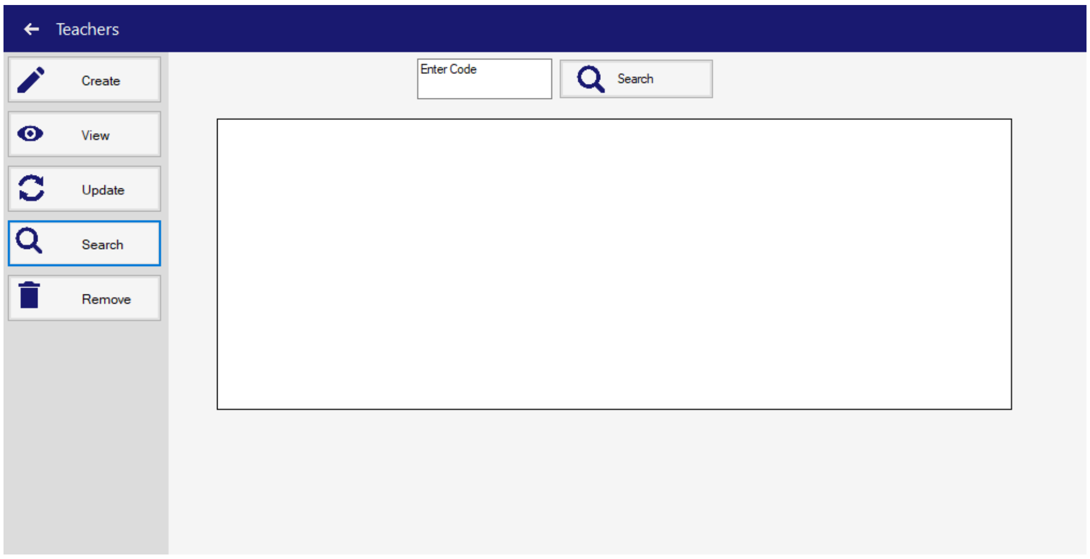


Figure 2.11: Search Teacher

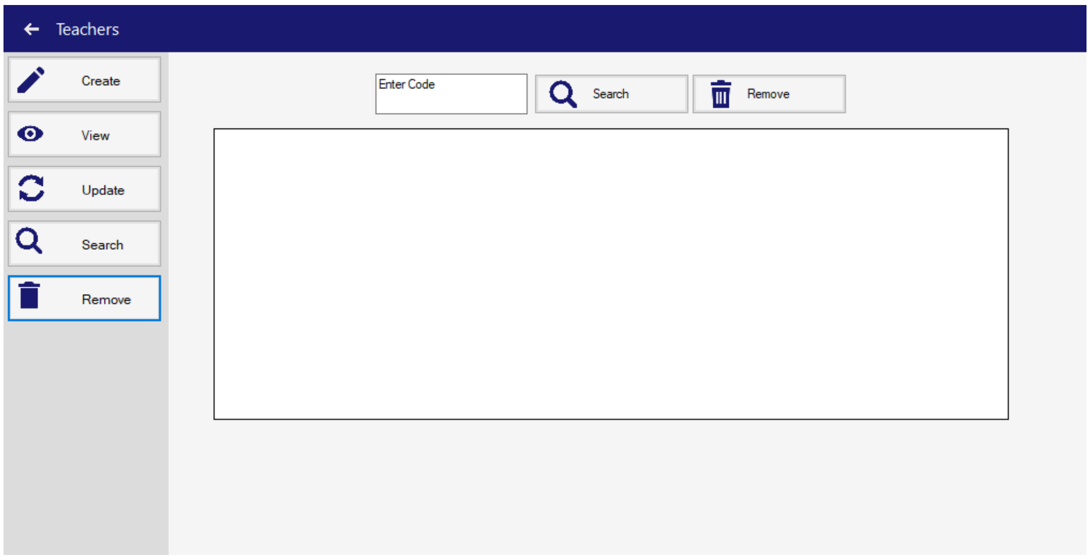
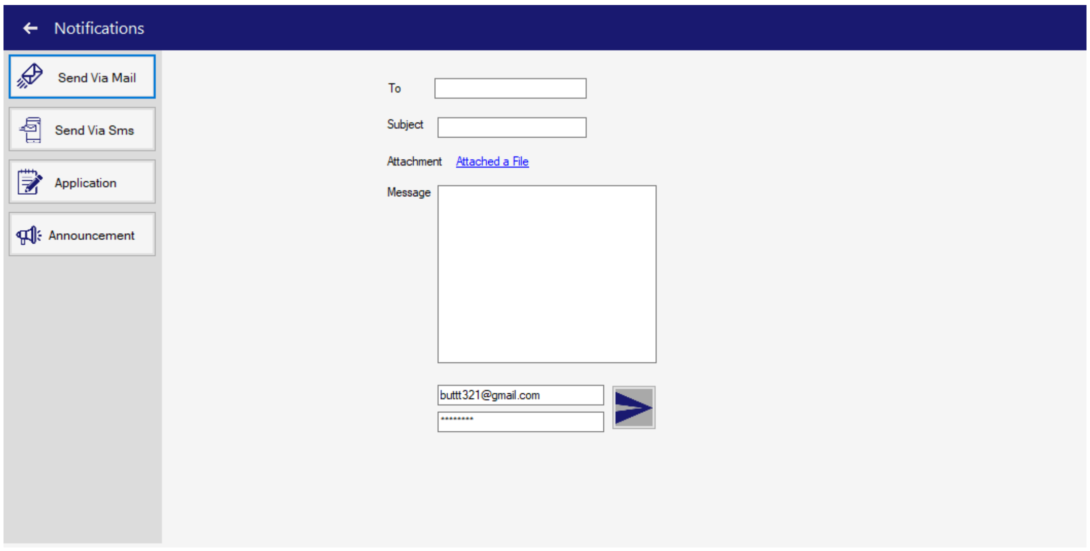


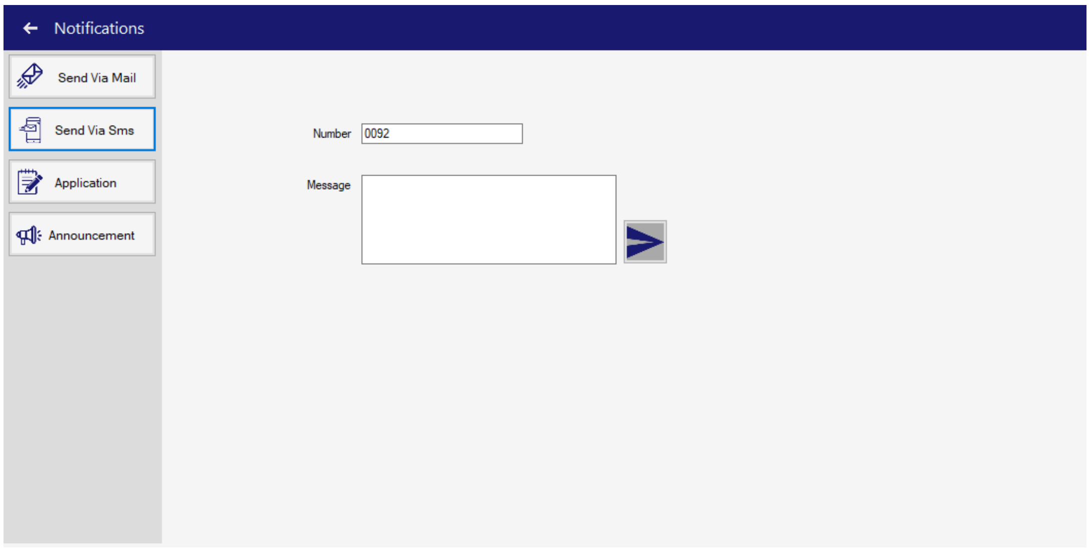
Figure 2.12: Remove Teacher



butt321@gmail.com

••••••••

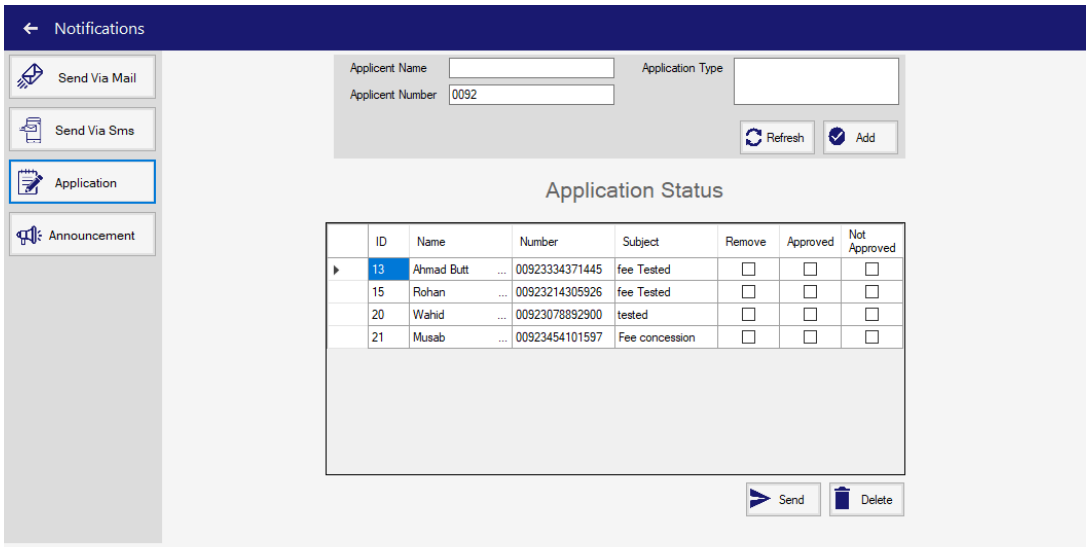
Figure 2.13: Send Email



Number 0092

Message

Figure 2.14: Send SMS



Applicent Name

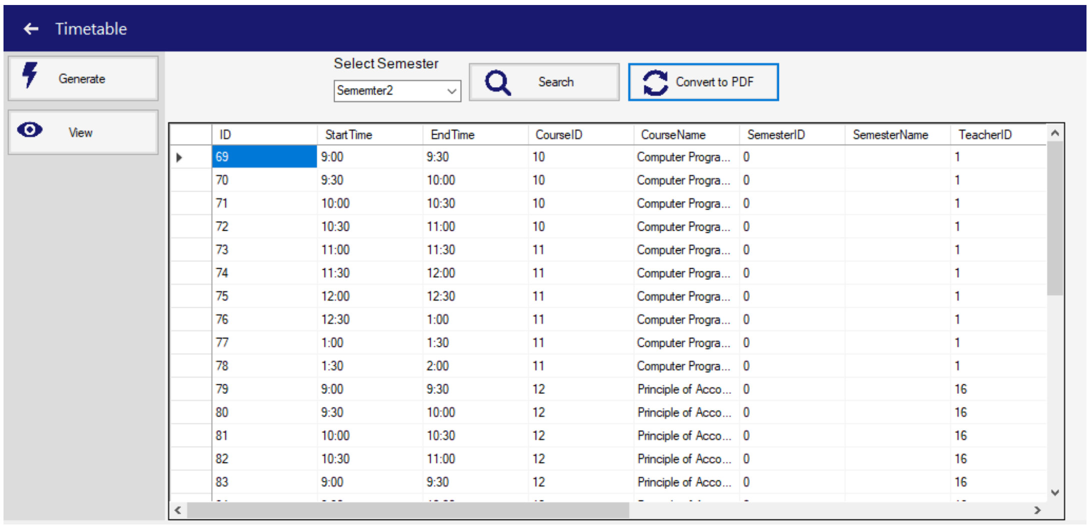
Applicent Number 0092

Application Type

Application Status

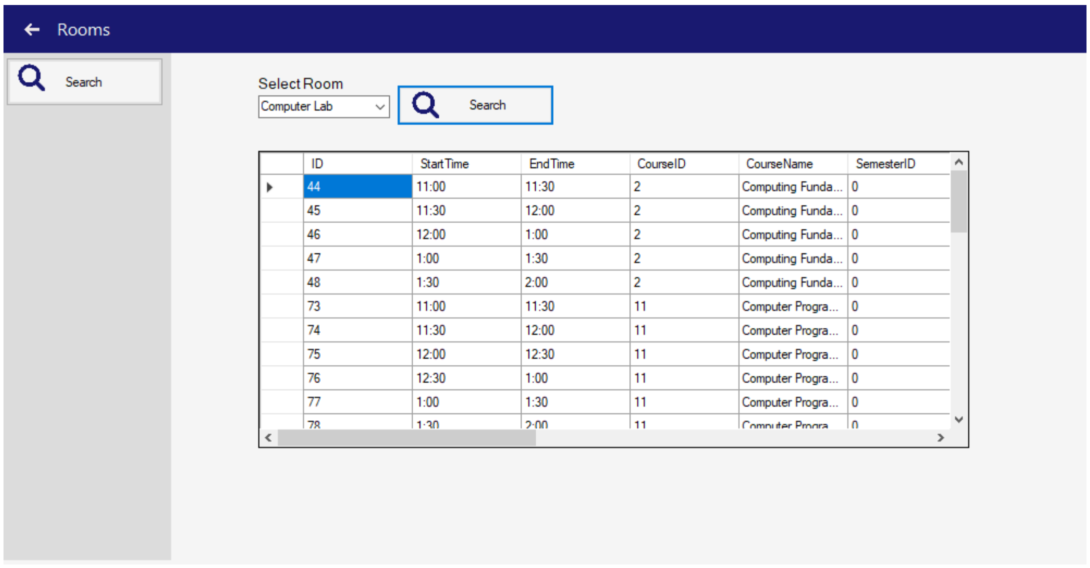
ID	Name	Number	Subject	Remove	Approved	Not Approved
13	Ahmad Butt	00923334371445	fee Tested	☐	☐	☐
15	Rohan	00923214305926	fee Tested	☐	☐	☐
20	Wahid	00923078892900	tested	☐	☐	☐
21	Musab	00923454101597	Fee concession	☐	☐	☐

Figure 2.15: Application Record



ID	StartTime	EndTime	CourseID	CourseName	SemesterID	SemesterName	TeacherID
69	9:00	9:30	10	Computer Progra...	0		1
70	9:30	10:00	10	Computer Progra...	0		1
71	10:00	10:30	10	Computer Progra...	0		1
72	10:30	11:00	10	Computer Progra...	0		1
73	11:00	11:30	11	Computer Progra...	0		1
74	11:30	12:00	11	Computer Progra...	0		1
75	12:00	12:30	11	Computer Progra...	0		1
76	12:30	1:00	11	Computer Progra...	0		1
77	1:00	1:30	11	Computer Progra...	0		1
78	1:30	2:00	11	Computer Progra...	0		1
79	9:00	9:30	12	Principle of Acco...	0		16
80	9:30	10:00	12	Principle of Acco...	0		16
81	10:00	10:30	12	Principle of Acco...	0		16
82	10:30	11:00	12	Principle of Acco...	0		16
83	9:00	9:30	12	Principle of Acco...	0		16

Figure 2.16: View Generated Time Table



ID	StartTime	EndTime	CourseID	CourseName	SemesterID
44	11:00	11:30	2	Computing Funda...	0
45	11:30	12:00	2	Computing Funda...	0
46	12:00	1:00	2	Computing Funda...	0
47	1:00	1:30	2	Computing Funda...	0
48	1:30	2:00	2	Computing Funda...	0
73	11:00	11:30	11	Computer Progra...	0
74	11:30	12:00	11	Computer Progra...	0
75	12:00	12:30	11	Computer Progra...	0
76	12:30	1:00	11	Computer Progra...	0
77	1:00	1:30	11	Computer Progra...	0
78	1:30	2:00	11	Computer Progra	0

Figure 2.17: View Room Time Table

2.3.2 Hardware Interface

Autovisor requires a system with windows vista or above. Having Dual core or above processors with above 2 GigaBytes of RAM.

2.3.3 Software Interface

Autovisor can be connected with a SQL database to import all the records of the students, teachers and other entities of system.

2.3.4 Network Interface

There are no such communication interfaces. As, Autovisor is a desktop application and only runs on the specific PC. Only requires internet when sending a mail or SMS.

2.4 Other Non-Functional Requirements

2.4.1 Performance Requirements

Autovisor requires a system with windows vista or above. Having dual core or above processors with above 2 GigaBytes of RAM

2.4.2 Safety Requirements

To ensure that no one of Autovisor users loses any data while using Autovisor (due to a crash or a bug of some kind) the developer team updates Autovisor regularly. So that in every new fix, the errors should be minimized.

2.4.3 Security Requirements

Autovisor is basically designed for student advisor, so that means the main user of the system is the student advisor and as far as security is concern. User must be register first to use the system and then logged in using username and password to allow system to give them rights and privilege

2.4.4 Software Quality Attributes

Autovisor provides the users with both simple and advanced features. Due to its well designed and easy to use interface and good usability design it can be used by both experts and typical users. However, users must already have a basic knowledge of desktop applications and computers before using it.

2.5 Other Requirements

2.5.1 Database Requirements

Autovisor is all based on getting the values and manipulating them. So, all the values must be store somewhere. For that, a well-designed database must be developed with new features and techniques and which is compatible on all the platforms on which Autovisor is compatible. Ensure the compatibility of database and the application.

2.5.2 API Requirements

Autovisor also uses some APIs to perform certain tasks. APIs are a very integral part of a program because it makes a program easy to handle and make it a well-designed written code.

2.6 System Requirement Chart

Following table shows the requirements for the SRS:

Table 2.1: System Requirements

ID	Type	Source	Description
AVS-01	Functional	Bahria University	User login
AVS-02	Non-Functional	Bahria University	A main menu with icons
AVS-03	Functional	Bahria University	View/Search items
AVS-04	Functional	Bahria University	Generate time table
AVS-05	Functional	Bahria University	Search the rooms
AVS-06	Functional	Bahria University	Allocate the rooms
AVS-07	Functional	Bahria University	Search the time slots
AVS-08	Functional	Bahria University	Student record
AVS-09	Functional	Bahria University	Teachers record
AVS-10	Functional	Bahria University	Offered courses
AVS-11	Functional	Bahria University	Allocate offered courses to teachers
AVS-12	Functional	Bahria University	Record keeping

AVS-13	Functional	Bahria University	Check status of application
AVS-14	Functional	Bahria University	Send SMS
AVS-15	Functional	Bahria University	Send Email
AVS-16	Functional	Bahria University	Generate Time Table PDF
AVS-17	Functional	Bahria University	Notify student about application status
AVS-18	Functional	Bahria University	Email including attachment
AVS-19	Functional	Bahria University	Make announcements

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Introduction

This chapter provides high level design which can provide aid in code development by providing the main points for the way the code is ought to be designed. This chapter provides narrative and graphical documentation of the code design for the project including use case models, domain model and different supporting requirement data. This chapter is all regarding the use case modeling and code design. In the previous chapter, analysis of the system is observed. Therefore, we tend to understand the present scenario of the problem domain. Following artifacts should be used in this deliverable.

i) System architecture
ii) Use case description
iii) Use case diagram
iv) Domain Model
v) Data Model

3.2 System Architecture

The system architecture of Autovisor is 2 tier application architecture. Where there is a back end which is database and a front end, which is application.

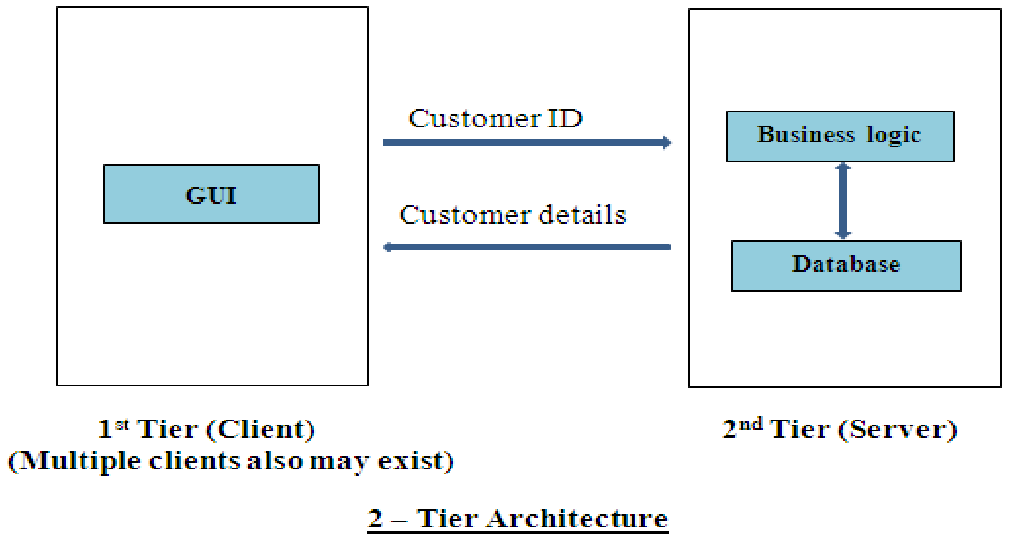


Figure 3.1: Autovisor 2-tier architecture Diagram

3.3 Use Case Description

Use case description usually contains the details description about all the use cases shown in the use case diagram. It usually describes the whole diagram narratively.

3.3.1 Generate Automatic Semester Time Table

Table 3.1: UC1

ID	UC1
Brief Description	User will generate the time table automatically and will get a proper designed time table
Preconditions	User is logged in
Basic Flow	i) User chooses the generate time table from menu. ii) User will choose the option by pressing generate time table button. iii) A generated time table will be displayed.
Alternate Flow	i) At step (ii) the database is not accessible.

	ii) An error message is displayed telling the user that some error occurred. iii) Return to step (ii)
Post Condition	Time table is generated

3.3.2 Individual Room Time Table

Table 3.2: UC2

ID	UC2
Brief Description	User can view the time table for an individual room, will get a proper designed time table
Preconditions	User is logged in & Time table is generated
Basic Flow	i) User chooses the time table from menu. ii) User will choose the option by pressing room time table button. iii) A generated time table will be displayed.
Alternate Flow	i) At step (ii) the database is not accessible. ii) An error message is displayed telling the user that some error occurred. iii) Return to step (ii)
Post Condition	Time table is shown

3.3.3 Individual Teacher Time Table

Table 3.3: UC3

ID	UC3
Brief Description	User can view the time table for an individual teacher, will get a proper designed time table
Preconditions	User is logged in & Time table is generated

Basic Flow	i) User chooses the time table from menu. ii) User will choose the option by pressing teachers time table button. iii) A generated time table will be displayed.
Alternate Flow	i) At step (ii) the database is not accessible. ii) An error message is displayed telling the user that some error occurred. iii) Return to step (ii)
Post Condition	Time table is shown

3.3.4 Offered Courses

Table 3.4: UC4

ID	UC4
Brief Description	User can register and view the registered courses
Preconditions	User is logged in
Basic Flow	i) User chooses offered option from menu. ii) User will choose the option of view all courses. iii) All the offered courses will be shown to the user.
Alternate Flow	i) At step (ii) there are no records in database or any error. ii) An error message is displayed telling the user that some error occurred. iii) Return to step (ii)
Post Condition	Offered courses will be shown

3.3.5 Teacher Registration

Table 3.5: UC5

ID	UC5
Brief Description	User can register a teacher in database
Preconditions	User is logged in & courses are registered
Basic Flow	i) User chooses the teacher option from menu. ii) User will choose the option of register teacher. iii) User will fill the particulars. iv) User will click the save button. v) Teacher will be registered.
Alternate Flow	iv) At step (iii) the user make mistake in filling. v) An error message is displayed telling the user that some error occurred. vi) Return to step (iii)
Post Condition	Teacher should be registered

3.3.6 Student Registration

Table 3.6: UC6

ID	UC6
Brief Description	User can register a student in database
Preconditions	User is logged in
Basic Flow	i) User chooses the student option from menu. ii) User will choose the option of register students. iii) User will fill the particulars. iv) User will click the save button. v) Student will be registered.
Alternate Flow	i) At step (iii) the user make mistake in filling. ii) An error message is displayed telling the user that some error occurred. iii) Return to step (iii)

Post Condition	Student should be registered

3.3.7 Record Keeping

Table 3.7: UC7

ID	UC7
Brief Description	User can keep the records of applications
Preconditions	User is logged in
Basic Flow	i) User chooses the notification option from menu. ii) User will choose the option of application. iii) User will fill the particulars. iv) User will click the save button. v) Record will be saved.
Alternate Flow	i) At step (iii) the user make mistake in filling. ii) An error message is displayed telling the user that some error occurred. iii) Return to step (iii)
Post Condition	Application record is saved

3.3.8 Application Status

Table 3.8: UC8

ID	UC8
Brief Description	User can view the status of application and can change it
Preconditions	User is logged in & application record is saved
Basic Flow	i) User chooses the notification option from menu. ii) User will choose the option of application. iii) User can see the status in the grid view check boxes.

Alternate Flow	i) At step (ii) the database is not accessible. ii) An error message is displayed telling the user that some error occurred. iii) Return to step (ii)
Post Condition	User have seen the status or changes it

3.3.9 Application Status Notification

Table 3.9: UC9

ID	UC9
Brief Description	User can send a notification of approval and disapproval to the applicant
Preconditions	User is logged in & application is saved
Basic Flow	i) User chooses the notification option from menu. ii) User will choose the option of application. iii) User can see the status in the grid view check boxes. iv) User will press send button against an application. v) Notification is sent.
Alternate Flow	i) At step (iv) the gateway is not accessible. ii) An error message is displayed telling the user that some error occurred. iii) Return to step (iv)
Post Condition	A notification is sent

3.3.10 Application Records

Table 3.10: UC10

ID	UC10
Brief Description	User can keep the records of applications
Preconditions	User is logged in
Basic Flow	i) User chooses the notification option from menu. ii) User will choose the option of application. iii) User will fill the particulars. iv) User will click the save button. v) Record will be saved.
Alternate Flow	i) At step (iii) the user make mistake in filling. ii) An error message is displayed telling the user that some error occurred. iii) Return to step (iii)
Post Condition	Application record is saved

3.3.11 Notifications

Table 3.11: UC11

ID	UC11
Brief Description	User can send notifications to people
Preconditions	User is logged in & internet connectivity
Basic Flow	i) User choses the notification option from menu ii) User will then choose the mode of notification iii) Send the notification
Alternate Flow	i) At step (iii) the internet is not accessible. ii) An error message is displayed telling the user that some error occurred. iii) Return to step (iii)
Post Condition	Notification is sent

3.3.12 Send SMS

Table 3.12: UC12

ID	UC12
Brief Description	User can send SMS to people
Preconditions	User is logged in & internet connectivity
Basic Flow	i) User choses the notification option from menu ii) User will then choose the SMS option iii) User write the number and SMS iv) User press Send button v) SMS is sent
Alternate Flow	i) At step (iv) the gateway is not accessible. ii) An error message is displayed telling the user that some error occurred. iii) Return to step (iv)
Post Condition	SMS is sent

3.3.13 Send Email

Table 3.13: UC13

ID	UC13
Brief Description	User can send emails to people
Preconditions	User is logged in & internet connectivity
Basic Flow	i) User choses the notification option from menu ii) User will then choose the email option iii) User write the email and message iv) User press Send button v) Email is sent
Alternate Flow	i) At step (iv) the internet is not accessible.

	ii) An error message is displayed telling the user that some error occurred. iii) Return to step (iv)
Post Condition	Email is sent

3.4 Use case Diagram

The use case diagram for Autovisor is as follows:

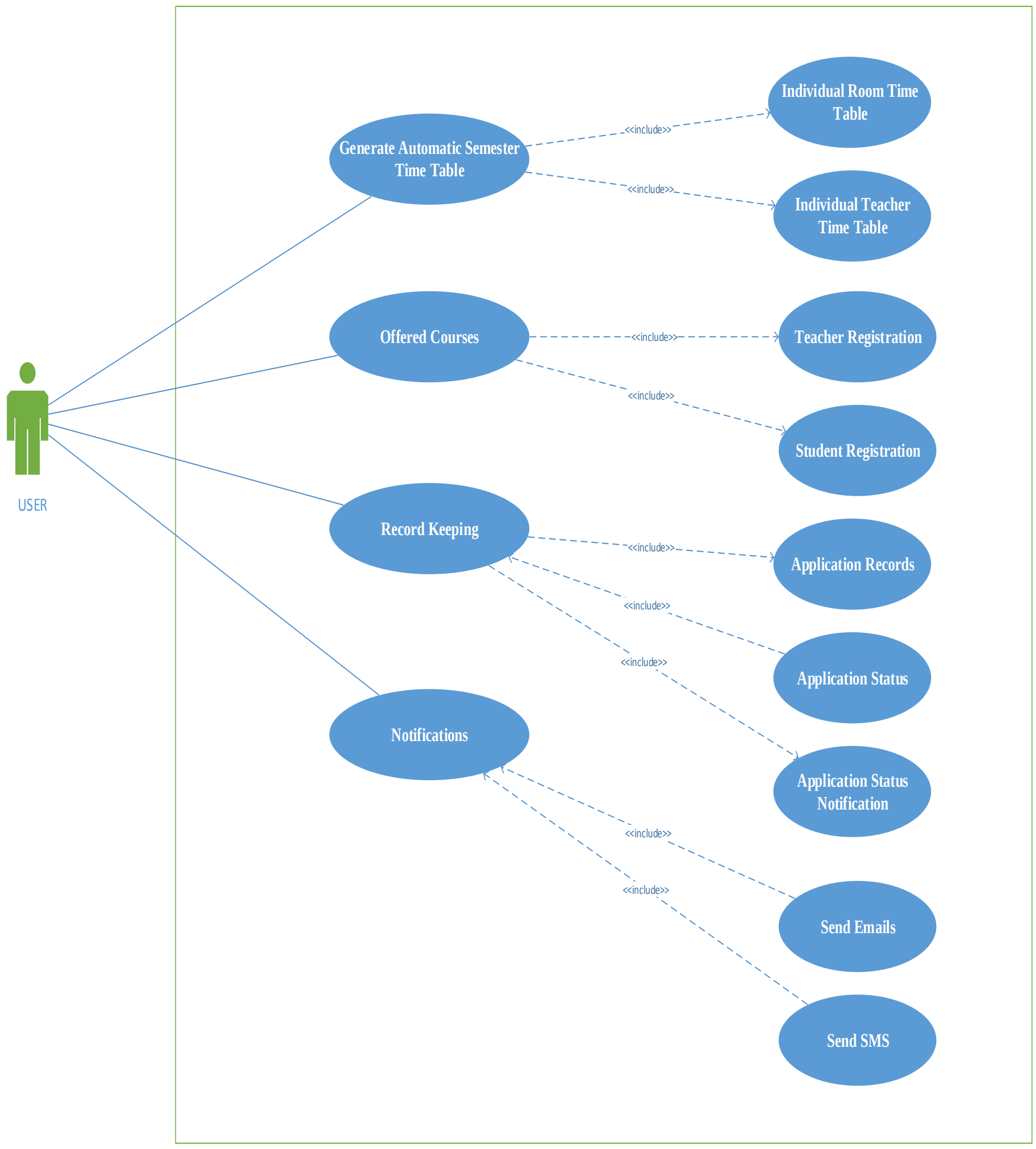


Figure 3.2: Autovisor Usecase Diagram

3.5 Domain Model

In the UML, a class diagram is used to represent domain model. The domain model of AVS is shown in the following figure:

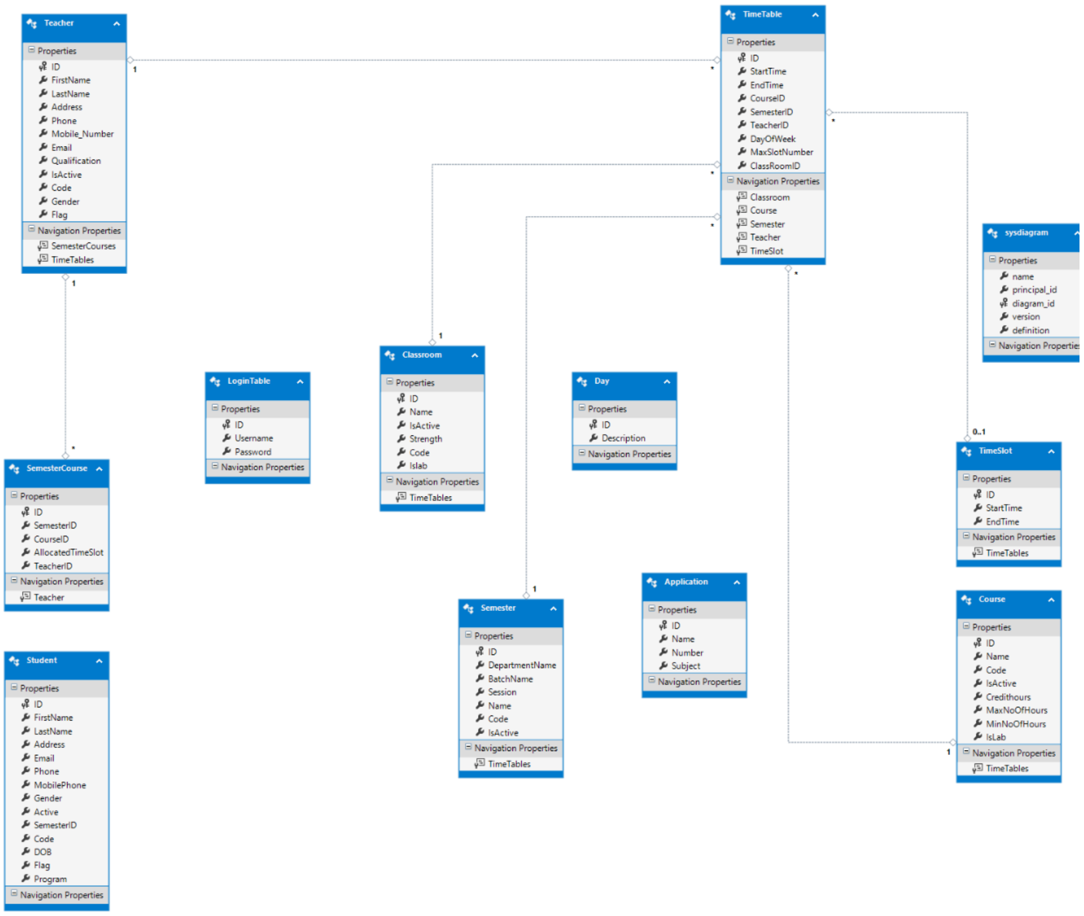


Figure 3.3: Autovisor Domain Model

3.6 Data Model

The data model in a UML can be shown by using an ERD. So, the ERD for Autovisor is as follows:

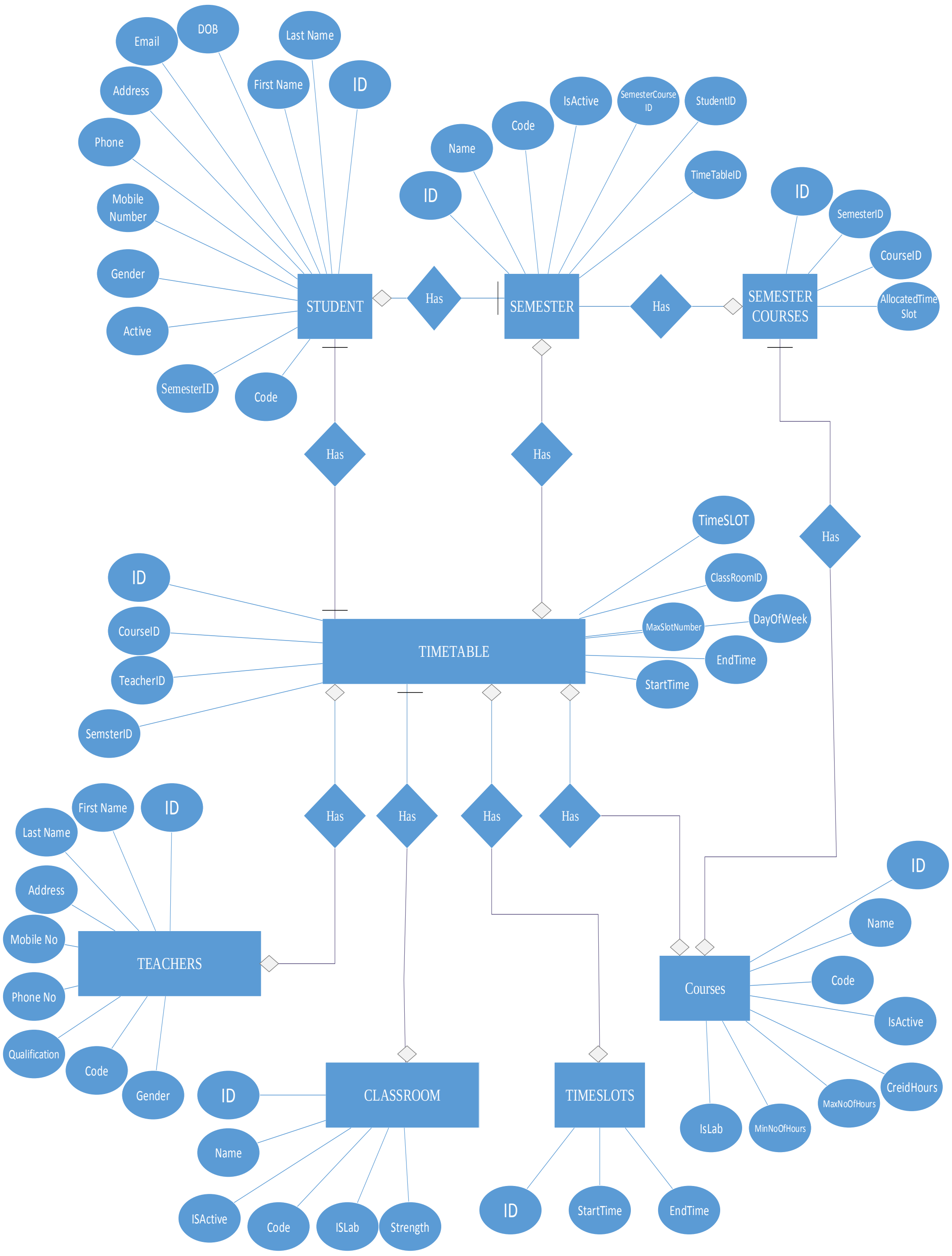


Figure 3.4: Autovisor Data Model

CHAPTER 4

IMPLMENTATION

4.1 Technologies used

4.1.1 C#.Net

C# is a programming language very much based on C and C++. It is and object-oriented language and also has the support for visual and event driven programming. This is a language created by Microsoft under their .Net framework. This language is approved by ISO and ECMA. It has some very strong programming features, making it one of the top language.

The reason of using this language is that it is very good for the desktop application and the nature of our project states that it is a desktop application. So, C# tends to be the best possible choice for the development of this project. [2]

4.1.2 SQL

SQL stands for Structures Query Language. This language allows you to manipulate and access your databases. It is approved by ANSI. By using this language, you can create, remove tables in database and also you can add, delete and update records in that tables. This language is most popular to handle the databases. That is why it is being used in this project. [3]

4.1.3 Entity Framework

This framework allows .Net developers to manipulate and handle many datasets or databases through a very minimal code. And is very easy to use. The version we have used is 6.1.3.

4.2 Tools used

4.2.1 Visual Studio

Visual Studio is an IDE, developed by Microsoft. It is used to develop applications for different platforms like mobile app, desktop app, web app etc. It has a very large support for different languages and has a compiler and debugger built into it. [4]

We have used this IDE because we are developing Autovisor by using C# language. So, that means we have to use a compiler which is best supports this language. So, using the product of Microsoft was best possible idea because C# is also developed by Microsoft.

4.2.2 Microsoft SQL Server

SQL server is developed by Microsoft and is a very popular platform to handle databases. It is a relational database management system. With a full support to be connected with Visual Studio. This is the reason why this DBMS was preferred over any other.

4.3 Database Design

4.3.1 Table Schema

There are 8 tables which have been designed for the database of Autovisor. The name if the tables are as follows:

i) SemesterCourse
ii) Course
iii) Student
iv) Semester
v) Teacher
vi) TimeSlot
vii) TimeTable
viii) Classroom

4.4 API Used

4.4.1 TextLocal

Text local is an API used to get a gateway and send text messages through it. This API is very useful and easy to use. All the SMS sending through Autovisor are using this TextLocal API.

4.4.2 DGVprinterHelper

DGWprint is an API used to print or convert a data grid view to PDF. Autovisor used this API to print or convert the time tables in PDF format.

CHAPTER 5

USER MANUAL

5.1 System Requirement

The most feasible system requirement is given in the below table:

Table 5.1: Feasible System Requirements

Device	Operating System (OS)	OS Version	RAM
PC,Tablet	**Windows**	**Vista,7,8,8.1,10**	**1GB or above**

5.2 Pre-requisite

i) User must be registered with a valid username to use Autovisor.
ii) User must have a valid password to use Autovisor.

5.3 Installation

To install the application in any PC or windows tablet device. User needs to run the setup package named as “avs.exe”. This application can be stored anywhere in the primary storage. It needs a very minimal amount of storage space. Following steps should be followed:

i) Tap the folder where the "avs.exe" is saved.
ii) Tap the application package "avs.exe".
iii) After selecting the package, application lets participant to install.
iv) Tap install.
v) Once the installation is done the application is ready to use.

5.4 Getting Started

Tap the icon to open the application.

avs

Bahria University Lahore Campus (Autovisor)

Figure 5.1: Autovisor icon

5.5 Opening the application/ Login

Once you open the application, you will see this login screen. Here you must enter a valid username and password. Then press sign in button to open the main menu

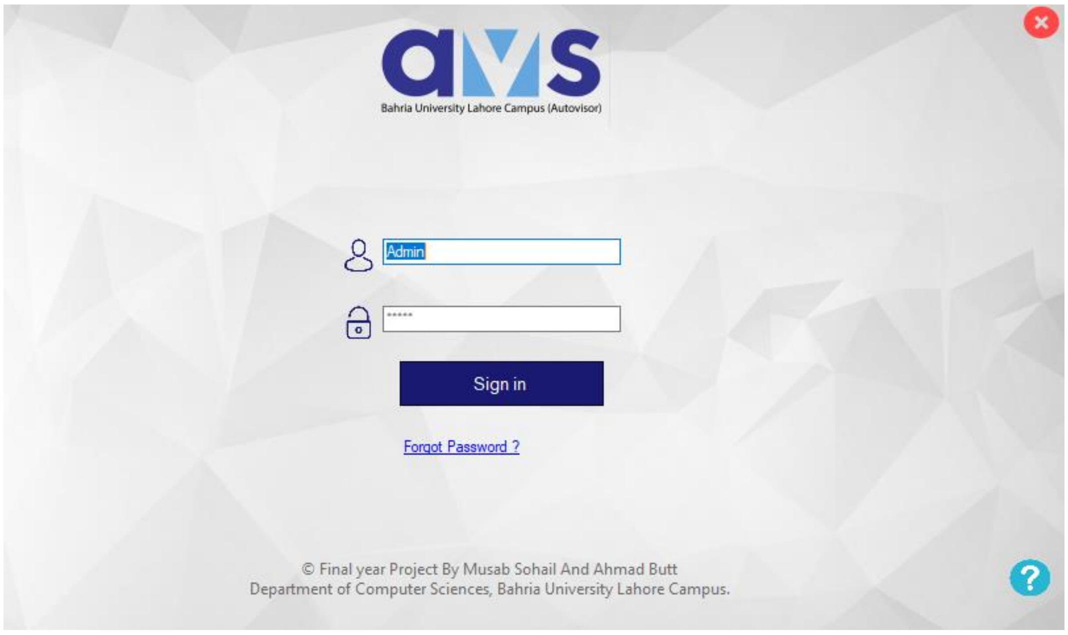

Bahria University Lahore Campus (Autovisor)
Admin
Sign in
Forgot Password ?
© Final year Project By Musab Sohail And Ahmad Butt
Department of Computer Sciences, Bahria University Lahore Campus.

Figure 5.2: Login

5.6 Home/ Main Menu

This is the main screen of the Autovisor. Here you can find the different features or menu items. Tap your desired menu item to enter that feature.

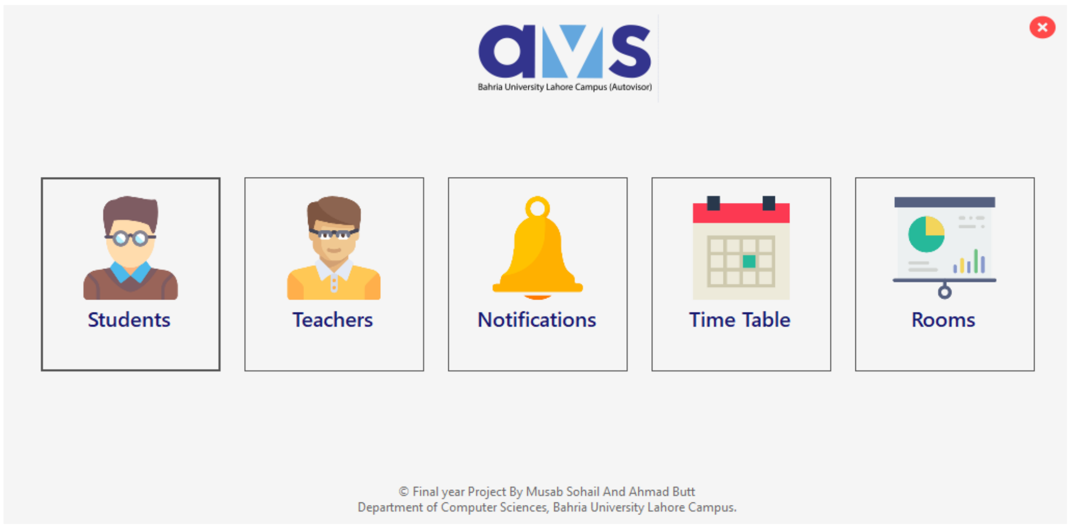

Bahria University Lahore Campus (Autovisor)
Students
Teachers
Notifications
Time Table
Rooms
© Final year Project By Musab Sohail And Ahmad Butt
Department of Computer Sciences, Bahria University Lahore Campus.

Figure 5.3: Home

5.7 Students

This section helps you performing activities related to students, like registering a student, accessing the record and manipulating those records etc.

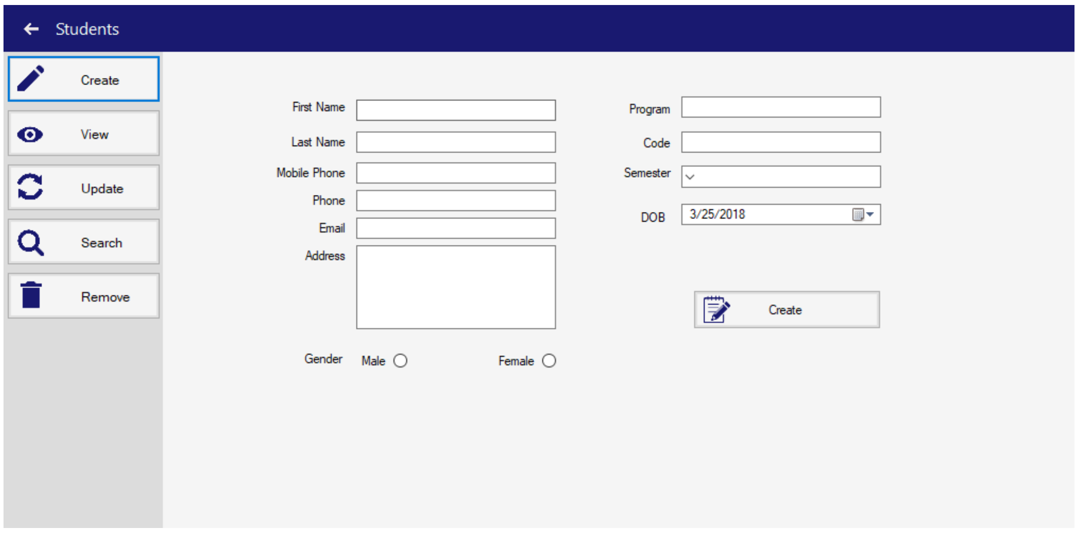


Figure 5.4: Student Section

5.8 Teachers

This section helps you performing activities related to teachers, like registering a teacher, accessing the record and manipulating those records etc.

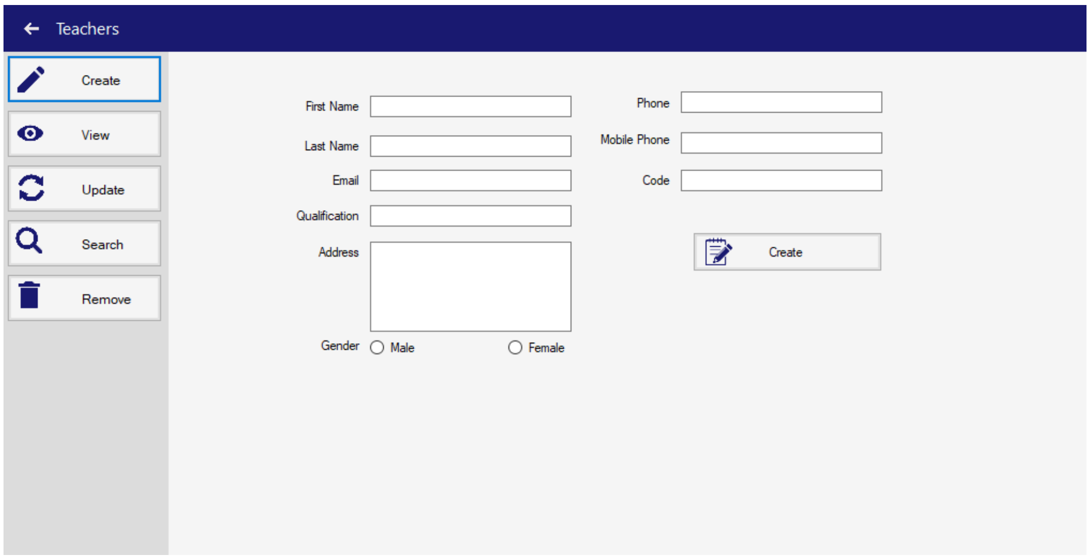
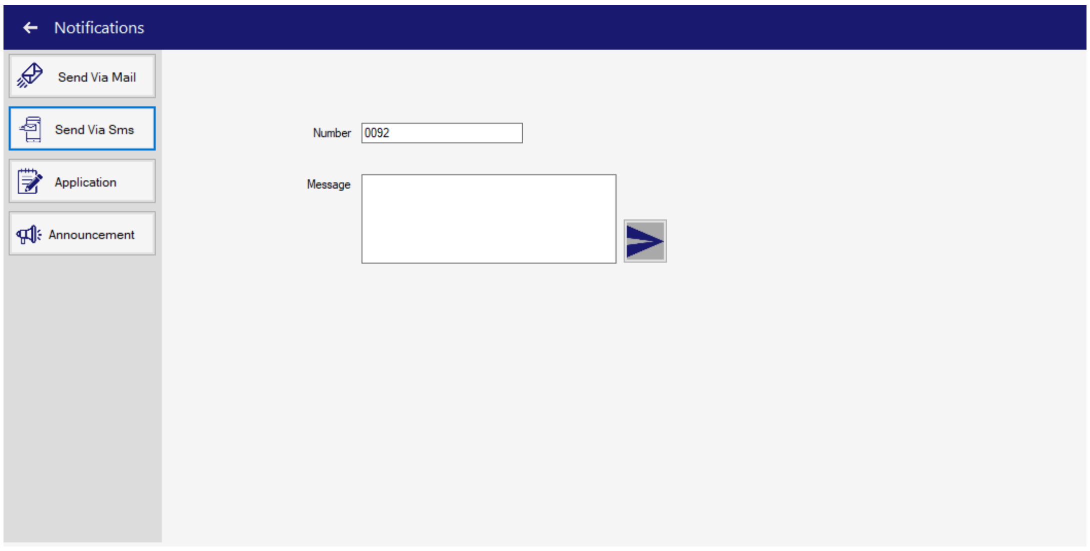


Figure 5.5: Teachers Section

5.9 Notifications

This section helps you performing activities related to notifications, like sending a message, sending an email, making announcements and keeping records of applications and sending a notification of their approval.

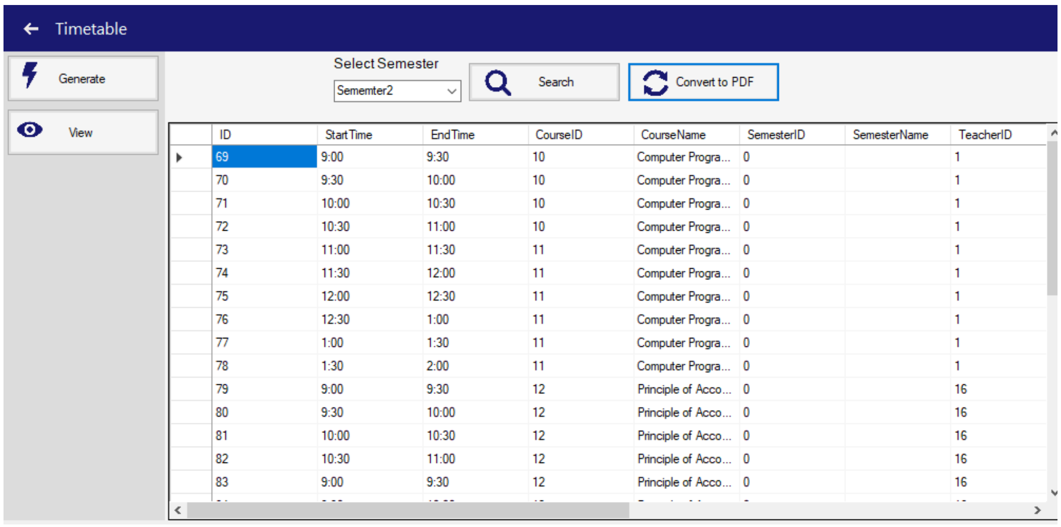


Number: 0092

Figure 5.6: Notifications Section

5.10 Time Table

This section helps you performing activities related to time table, like generating a time table and viewing time tables for different semesters.



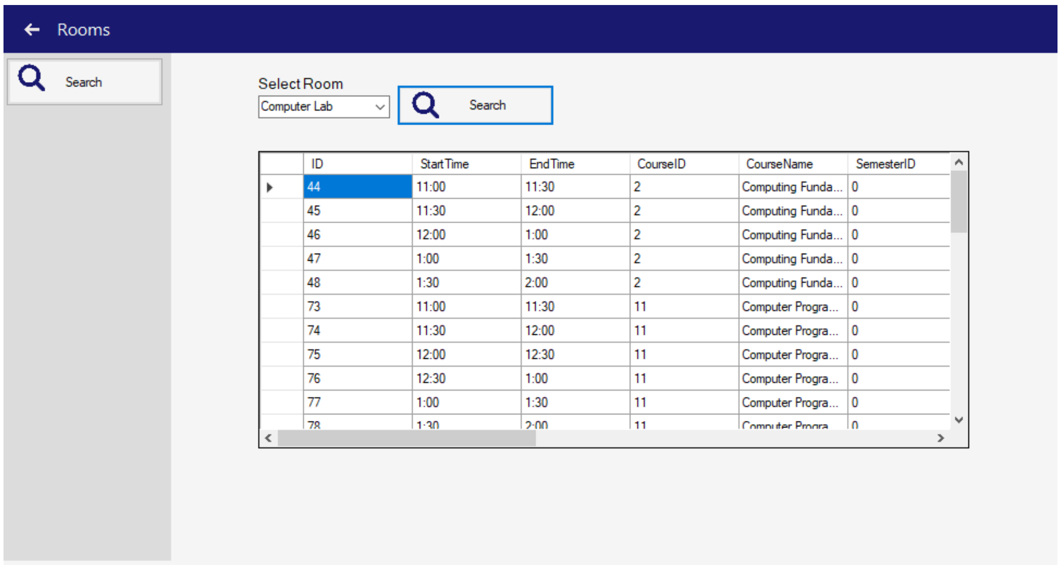
Select Semester: Sememter2

ID	StartTime	EndTime	CourseID	CourseName	SemesterID	SemesterName	TeacherID
69	9:00	9:30	10	Computer Progra...	0		1
70	9:30	10:00	10	Computer Progra...	0		1
71	10:00	10:30	10	Computer Progra...	0		1
72	10:30	11:00	10	Computer Progra...	0		1
73	11:00	11:30	11	Computer Progra...	0		1
74	11:30	12:00	11	Computer Progra...	0		1
75	12:00	12:30	11	Computer Progra...	0		1
76	12:30	1:00	11	Computer Progra...	0		1
77	1:00	1:30	11	Computer Progra...	0		1
78	1:30	2:00	11	Computer Progra...	0		1
79	9:00	9:30	12	Principle of Acco...	0		16
80	9:30	10:00	12	Principle of Acco...	0		16
81	10:00	10:30	12	Principle of Acco...	0		16
82	10:30	11:00	12	Principle of Acco...	0		16
83	9:00	9:30	12	Principle of Acco...	0		16

Figure 5.7: Time Table Section

5.11 Rooms

This section helps you performing activities related to time table, like viewing time tables for different rooms.



ID	StartTime	EndTime	CourseID	CourseName	SemesterID
44	11:00	11:30	2	Computing Funda...	0
45	11:30	12:00	2	Computing Funda...	0
46	12:00	1:00	2	Computing Funda...	0
47	1:00	1:30	2	Computing Funda...	0
48	1:30	2:00	2	Computing Funda...	0
73	11:00	11:30	11	Computer Progra...	0
74	11:30	12:00	11	Computer Progra...	0
75	12:00	12:30	11	Computer Progra...	0
76	12:30	1:00	11	Computer Progra...	0
77	1:00	1:30	11	Computer Progra...	0
78	1:30	2:00	11	Computer Progra	0

Figure 5.8: Rooms Section

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

In the end, what I have got is the solution to a problem, which was causing a lot of effort and time of some student advisors by not having a place where there day to day processes are not integrated. So according to my research, many people tried and have been successful making the student advisor activities automatic. But none of them have made such platform where all those activities or processes are integrated. So, Autovisor is very small effort to make that thing happen. So that we can have all the processes together in a platform.

The things which did not have been addressed in this paper or our project is that we have made it a general application with some general features. Because the time of this project and research did not allowed us to go beyond that thing. Every institute have their own processes and it is difficult to cater all those processes. But we are hoping to see that in future.

6.2 Recommendations

As Autovisor is a general-purpose software, so in future it can be expanded for different institutes and made it specific for every other institute. Also, in this project only a desktop app has been made. So, in future it can be expanded to multi-platform support like mobile app and web app. Also, many processes still can be added into this app. It can also be made as a portal where teacher, student and student advisors can meet.

REFERENCES

Journal Papers:

[1]. Khaled Mahar, ‘’AUTOMATIC GENERATION OF UNIVERSITY TIMETABLES: AN EVOLUTIONARY APPROACH’’, *College of Computing and Information Technology, Arab Academy for Science and Technology*May 2006.

Electronic Sources from Internet:

[2] TutorialsPoint, “C# - Overview”, 2018, Available at: https://www.tutorialspoint.com/csharp/csharp_overview.htm

[3] W3Schools, “Introduction to SQL”, 2018, Available at: https://www.w3schools.com/sql/sql_intro.asp

[4] Wikipedia, “Microsoft Visual Studio”, 2018, Available at: https://en.wikipedia.org/wiki/Microsoft_Visual_Stud