University Scheduling

TEAM: -

Supervisor: Dr. Mohammed Hashim

Participants:

1-Assim Essam Farahat Shaaban

2-Mostafa Hassan Mohamed Hassan Helal

3-Muhannad Ibrahim Ali Alsayed

4-Omar Khaled Seif ElNasr Mahmoud

Acknowledgements

We would like to express our special thanks of gratitude to our supervisor Dr. Mohammed Hashim as well as our Vice Dean of our college Dr. Mohammed Khaled and Dr. Ashraf Baghdadi who gave us the opportunity to do this important project on the topic university scheduling, which also helped us in doing a lot of Research and we came to know about so many new things we are really thankful to them.

We would also like to thank Mrs. Marwa the secretary of vice dean of the College who helped us a lot in providing us with the information needed to finalize this project within the limited time frame.

CONTENTS

Project Specification

1- Description

The Project “University Scheduling” job is to schedule university courses into classrooms under some constraints: -

1-Classroom time, capacity and resources

2-Programs schedule

2- Objectives

The project objective is to successfully place all given courses into lectures/sections/labs without any conflicts in the constraints in a minimal time.

3- Scope

The app takes (courses, programs, classrooms and resources) as input to schedule courses depends on how the app distributes the courses in suitable classroom.

4- Outcomes

The courses schedule for each classroom and program.

Existing systems

1-Manually: -

The existing system in collage right now is to schedule the courses manually using human help.

2-ROZ: -

A Paid system that doesn't fit our collage custom needs like high program counts and skipping doctors schedule.

Used by Arab Academy For Science, Technology And Maritime Transport (AASTMT).

Requirements Gathering and Analysis

1- The Project is a collage requirement to help schedule the courses faster.
2- Searching for an existing algorithm to see if the app can be implemented.
3- Searching for other existing systems to get more ideas for the implementation.
4- Failing to find an existing system that fits collage requirements and starting to find ways to implement our own methods.
5- Meet up with Dr. Mohammed Khaled and then Mrs. Marwa to see how her mind works while scheduling the tables and how she priorities courses to schedule first over other courses.
6- Brain storming with Dr. Mohammed Hashim that provides us with technical papers about genetic algorithm and helps us to go to AASTMT because of their system
7- We decided to partition ourselves 2 by 2 to work with the genetic algorithm and build our algorithm which is greedy algorithm-based on priority.
8- We decided to complete our algorithm because the genetic algorithm has a high complexity and we couldn’t customize it to fit our needs.

Our algorithm

The base idea of the Algorithm is a greedy priority based algorithm.

The algorithm takes the courses then gives them weights according to the hardness of scheduling each course and its importance.

The weighting factors are: -

Actual hours need to be scheduled weight factor = 10

Program count weight factor = 5

Credit hours weight factor = 2

Student Count = Student Count/20

then start scheduling from the Highest to Lowest Weight conflicts: -

1- If the app failed to schedule a course because the student counts is too large the course is split into two parts each part(slot) has its own weight.
2- If it just failed because any other reason like time or resources it just gives higher weight to the course and start rescheduling from the given weight to the smaller weights.

Input:

- Courses => list of courses that needs to be scheduled and programs of each course.
- Classrooms => available classrooms that can be used.
- max_hours => max number of hours that could be used for scheduling e.g. 10 hours.
- max_days => max number of days that could be used for scheduling e.g. 6 days.

Output:

- List of Reservations.

Algorithm:

1. Make simple data validations

    1.1. Make sure that for each course there exist at least one classroom that satisfies its required resources.

    1.2. Make sure that available lecture hours, lab hours in classroom and lab is enough for the existing courses.

2. Initialization

    2.1. Making "SLOTS": -

    A Slot is a part of the course as lecture, Practice, Lab or any of the previous split into many parts, that needs to be scheduled

        2.1.1. Giving Weight to each slot

        Weight += slot.programs.Count * programWF;

        Weight += slot.creditHours * hourWF;

        Weight += slot.hours * actualHourWF;

        Weight += slot.studentCount / 20;

3. Reserving and conflicts resolving

    3.1. Reserving slots from highest weight to Lowest weight (a weight maps to a list of slots).

    For each slot S in slots

    Loop through available days in the form of

(FirstDay, LastDay, FirstDay+1, LastDay-1, FirstDay+2,...) helps spread the weight across available days

For each classroom C in [select classrooms where (if slot is lab get lab rooms, else get non lab rooms) and select from them classrooms that satisfies the slot resources required]

If slot.studentCount > C.capacity find next classroom

For time from 0 to max_hours

If classroom at time is empty

If time + slot.hours > max_hours

break and find another classroom

For k from time+1 to time+slot.hours

If classroom is not free at k

break and find another hour starting from time = k+1

For each program P in slot.programs

For k from time+1 to time+slot.hours

If P is not free at k

break and find another hour starting from time = k+1

//All Above Guards are passed we can now make a Reservation for each slot program and classroom select.

3.2. In case of conflict the slots of the selected weight won't be reserved and the reserve function will return conflict type and slot that caused the conflict then there is two ways to handle the conflict

3.2.1:- capacity conflict(no classroom big enough for slot program)

If slot had many program the programs will be split into two slots with new weights

If slot had one program the slot student count is just divided by two

3.2.2:- other conflicts (Resources, programs time, class time, day limit) the weight of the slot increases to join slots of higher weight and then all slots of higher weight are rescheduled

Technology on which SW project carried out

1. C#

   Which is the basic language used to build our application using the environment visual studio (at least version 2015).

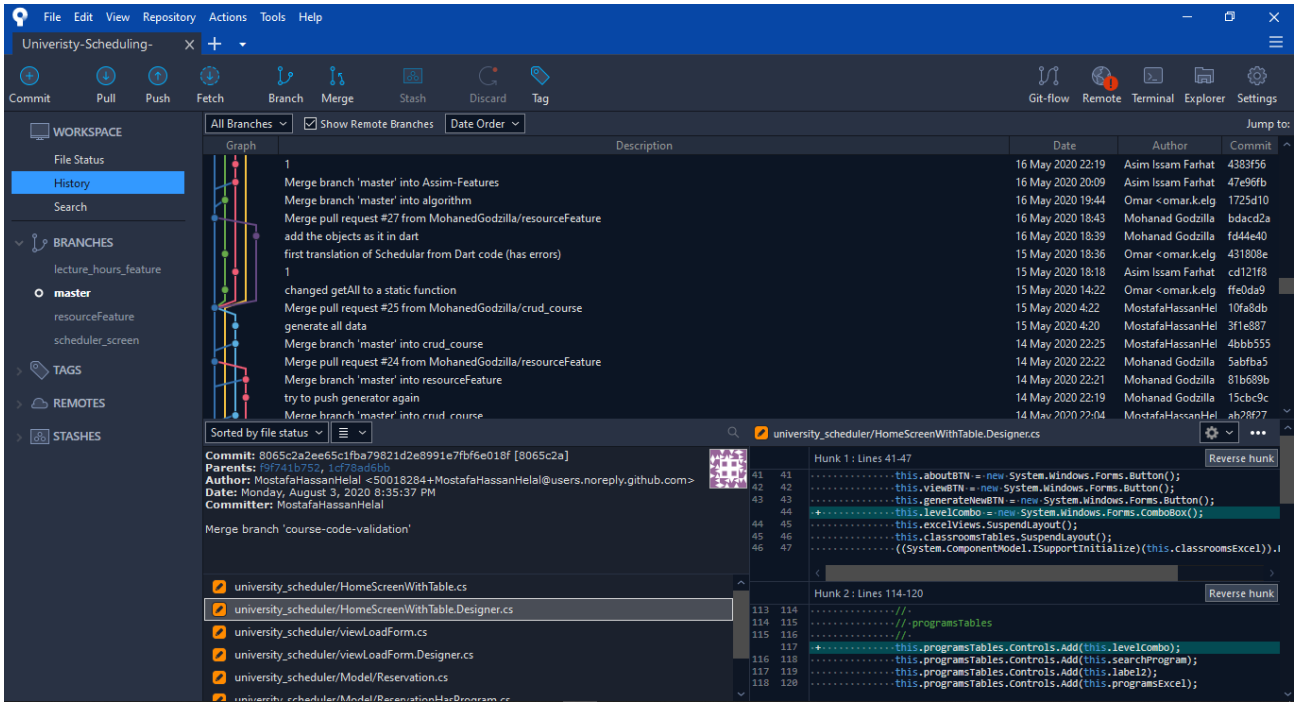
2. Sql server

   Recommend to use Microsoft SQL Server Express Edition (64-bit) version 15.0.2000 RTM

3. Dart

   It is a client-optimized programming language for apps on multiple platforms. It is developed by Google and is used to build mobile, desktop, server, and web applications. And it is an object-oriented, class-based, garbage-collected language with C-style syntax. It's used to design our algorithm and test it with random data.

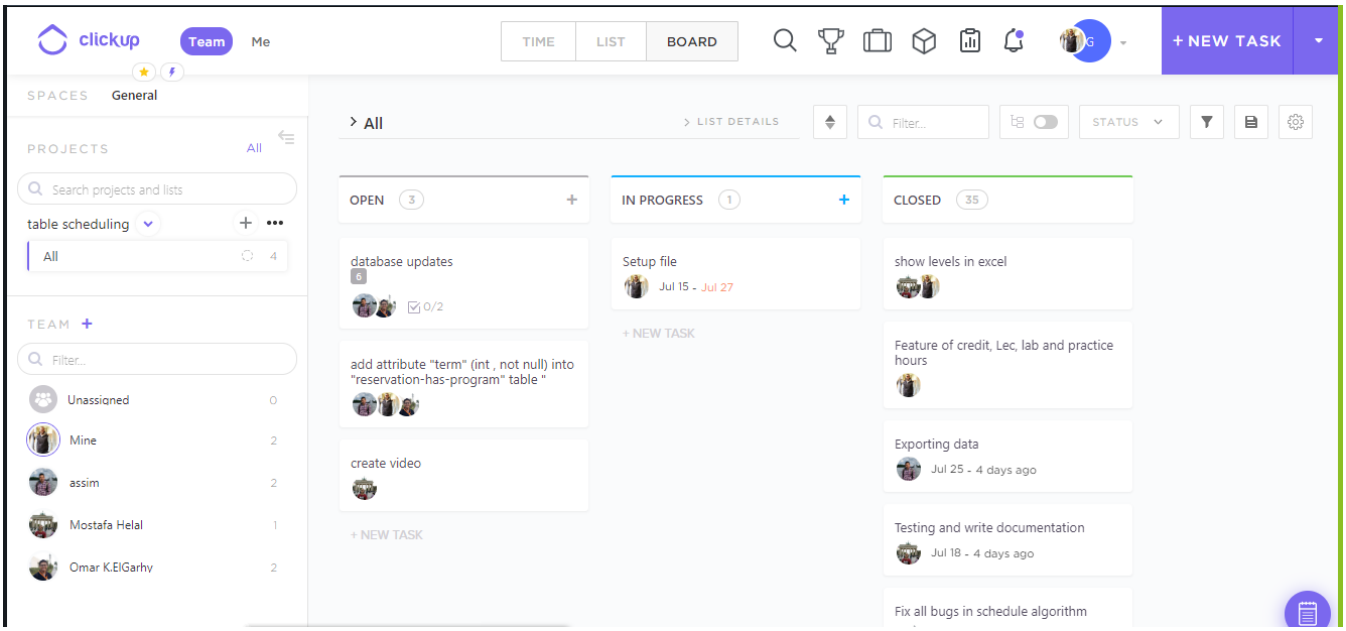
4. Git source control
   a. Host: GitHub.
   b. Desktop client source tree.



Process model adapted for SW project development

We worked through the application using the AGILE method as the following:

- First, we decided to use the click up application for planning, assigning tasks and know what features need to be done.



- We designed the database for the application.

- We designed a pseudo code for our algorithm.

- We implemented the pseudo code as a program using dart programming language to test it.

- The App features were split into 3 sprints each sprint process was (Analysis=>UI/UX Design Script Prototype on Paint=>Implement the features=>Test & Fix=>Close Tasks).

- Sprint period varied depending on features

    First Sprint (Entering the data): -

        The sprint aimed to finish features related to entering data like courses, programs, classrooms, resources, start time & day and end time & day.

    Second sprint (Generating timetable)

        Trying to perform the algorithm on the input data to test its efficiency when working on random data which is close as much as the real data.

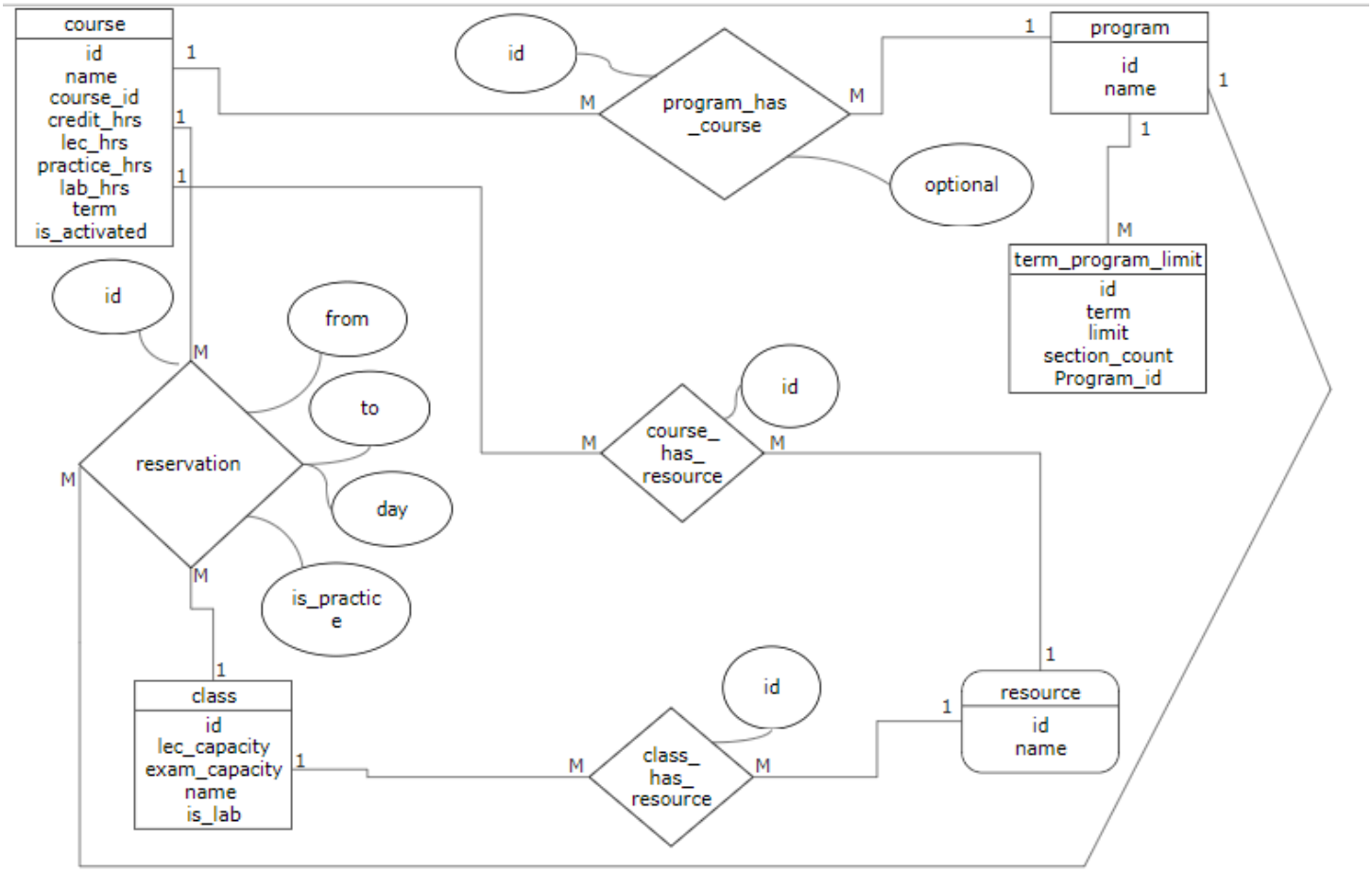
    Third sprint (representing the data)

        After algorithm has been created the timetable. We represent the timetable as classrooms and programs which user can click on it to view the timetable within the Excel program for specific program or classroom and he can switch between the Excel sheets to see the other programs or classrooms. And he has the ability to generate a new timetable within the UI.

Data dictionary

The database contains 5 essential tables which is course, program, class, resource and reservation specific for program or classroom.

And, you can view the relationship **model** (ER **model**) from the following:



System architecture

The system uses MVC design pattern as Windows Form Application Projects done by C# has a great and ready to go base code to follow the MVC Pattern as it generates the View and Starter Controller code automatically.

Models(M): -

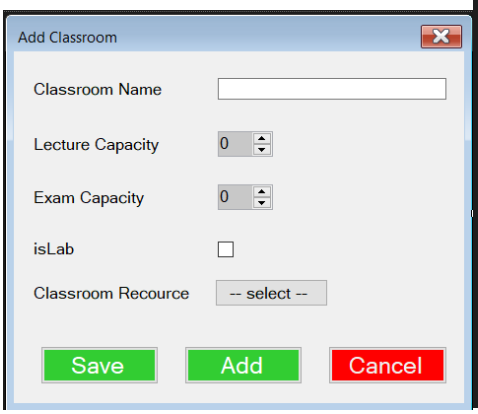
There is almost a model for each table that contains its own data members and functions that directly communicate with the database to fetch the required data.

```
28 references
class Course {

    members

    0 references
    public static int insertCourse(string dummyName, string codeNI, int crH, dou
    1 reference
    public static List<Course> getAll()...
    0 references
    public static List<Course> getAll(string dummyName)...
    1 reference
    public static List<Course> getCoursesByTerm(int term) ...
    1 reference
    public static Course getCourseById(int courseId)...
    4 references
    static Course setCourse(SqlDataReader reader) ...
    2 references
    private static List<Program> getCoursePrograms(int courseID) ...
    1 reference
    public List<Program> getCoursePrograms() ...
    1 reference
    public List<Resource> getCourseResources() ...
    1 reference
    public static DataTable search()...
}
```

Model (A) can communicate with Model (B) in the background if Model (A) has a relationship with Model (B)

View (V): -

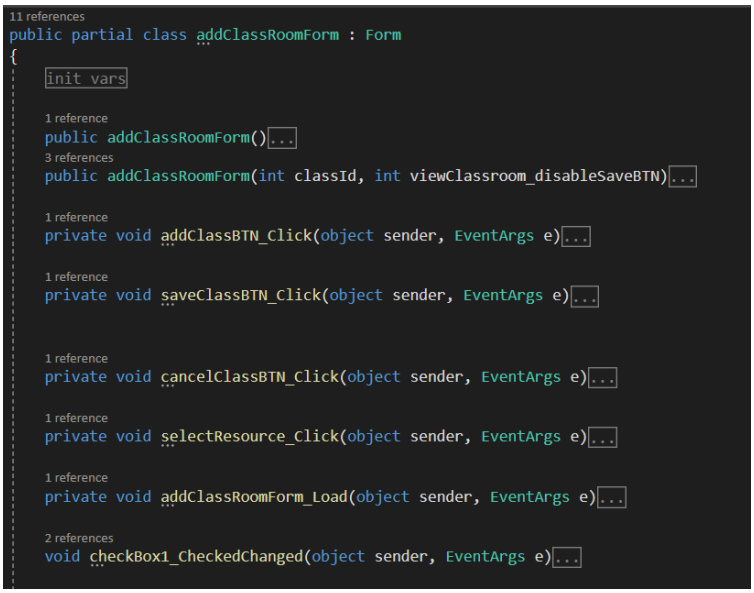
The App is developed using visual studio which offers a drag and drop Windows Form Designer tool which generates its view code




```
partial class addClassRoomForm
{
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should
    14 references
    protected override void Dispose(bool disposing)...

    Windows Form Designer generated code

    private System.Windows.Forms.Button saveClassBTN;
    private System.Windows.Forms.Button addClassBTN;
    private System.Windows.Forms.Button cancelClassBTN;
    private System.Windows.Forms.NumericUpDown examCounter;
    private System.Windows.Forms.Label label3;
    private System.Windows.Forms.Label label2;
    private System.Windows.Forms.Label label1;
    private System.Windows.Forms.NumericUpDown lecCounter;
    private System.Windows.Forms.TextBox className;
    private System.Windows.Forms.Button selectResource;
    private System.Windows.Forms.Label label4;
    private System.Windows.Forms.Label label5;
    private System.Windows.Forms.CheckBox isLab;
}
```

Controller (C): -

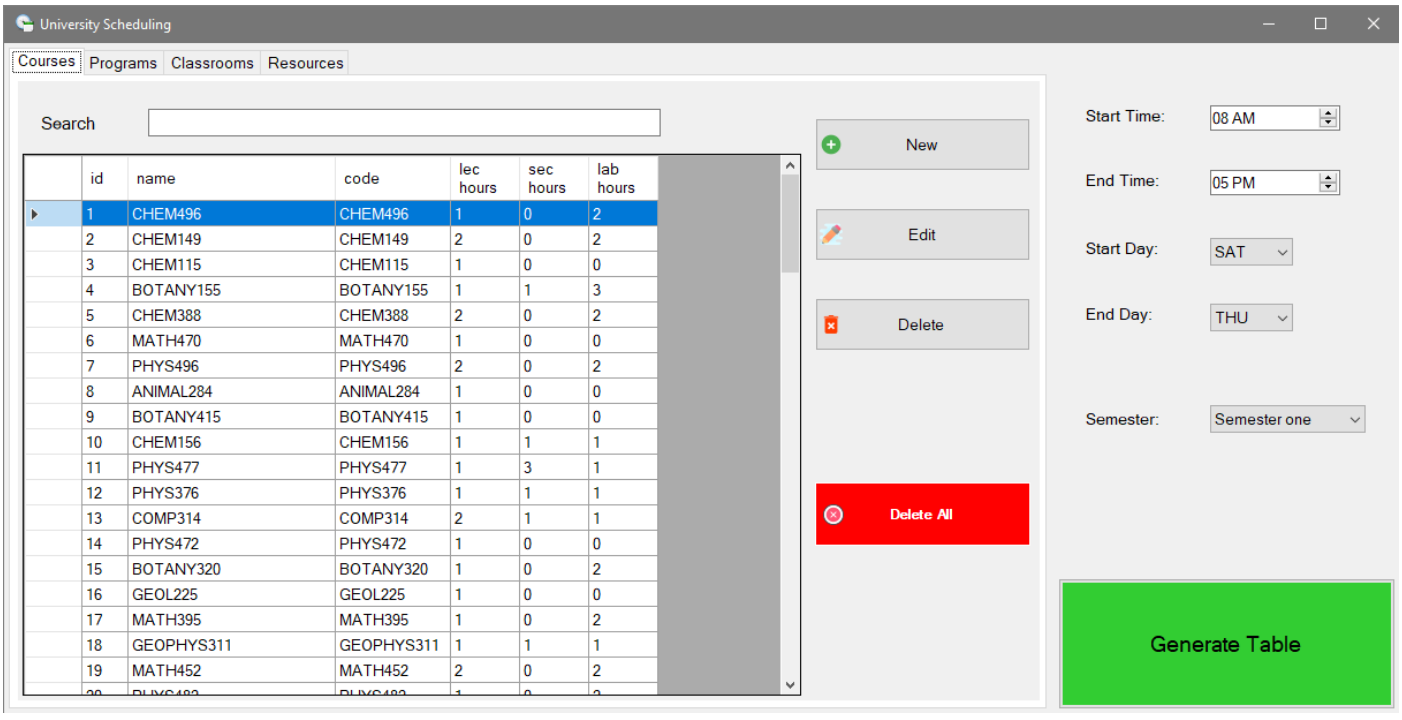
A starter code is also generated with each form then all event handlers like (OnMouseClick, and OnValueChanged) are added to the controller



User interface design and screen layouts

Our application is divided into 5 essential windows (or Forms) which are one for each view (course- program- class- resource) and the last essential window is the form that appears when the table is reserved and it's used to view the scheduled table for (program- classroom) and user can view the timetable within the excel program after double clicked on the program or classroom that he wants.

First we will represent the main window for the app



- As you see, we have 4 tab controls (courses, programs, classrooms, resources) at each one of them we have the same buttons (new, edit, delete, delete all).

- **Recommended**, to fill (Resources => classrooms => programs => courses).
- we have same other important information about the start time that the program should start scheduling from at every day and the end time that should the program end scheduling at every day.

- The screen includes also the start day and the end day for specifying the first day in the week and the last day in the week, respectively.

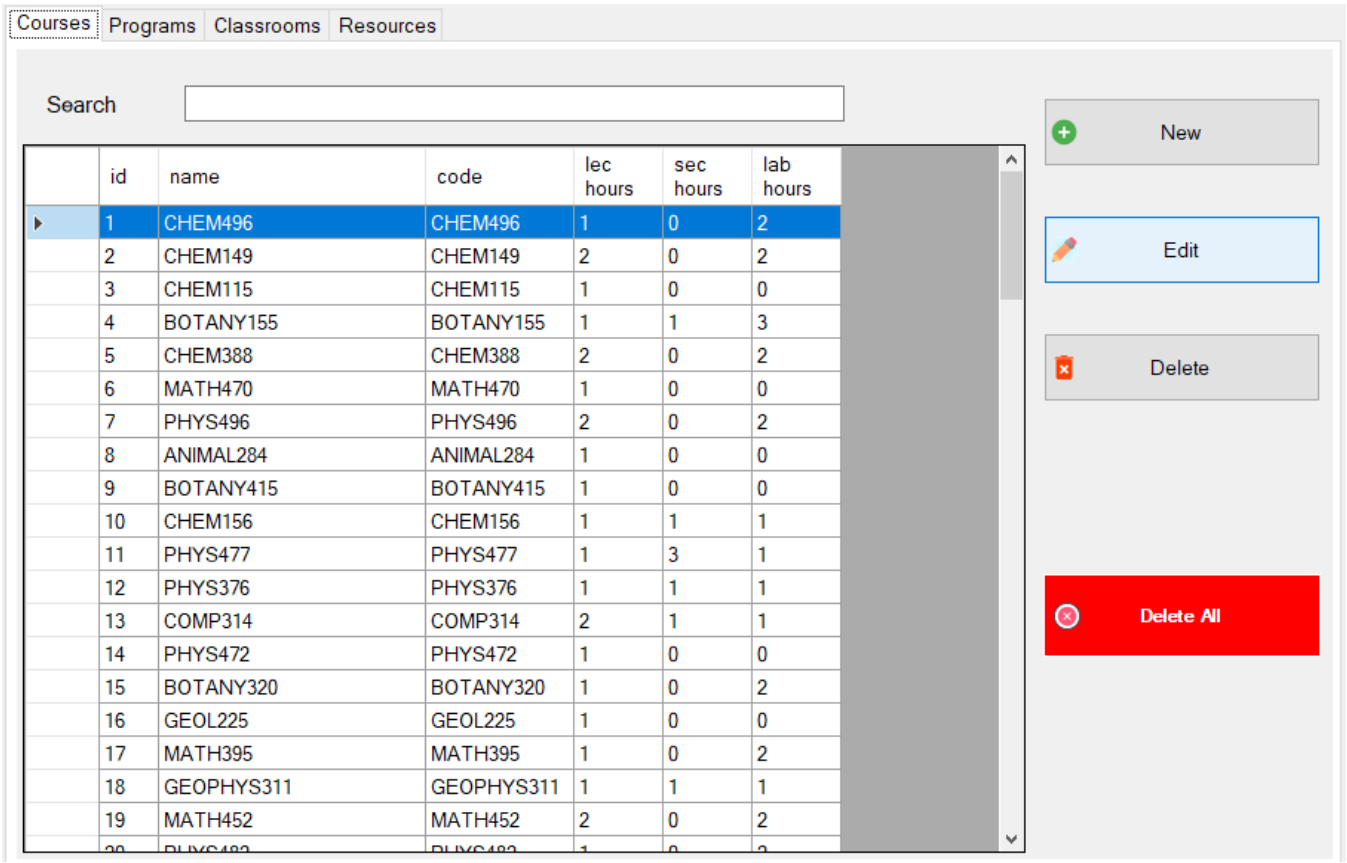
- The semester combo box is to select what is the semester that the program will generate table for.

- And the generate table button for generating the table.

Second, we will represent each of the essential windows and the dependent sub-windows: -

1- Course view window

Here, there are 4 buttons new, edit (we will explain it with more details), delete (for deleting a specific course) and delete all for erase all courses.

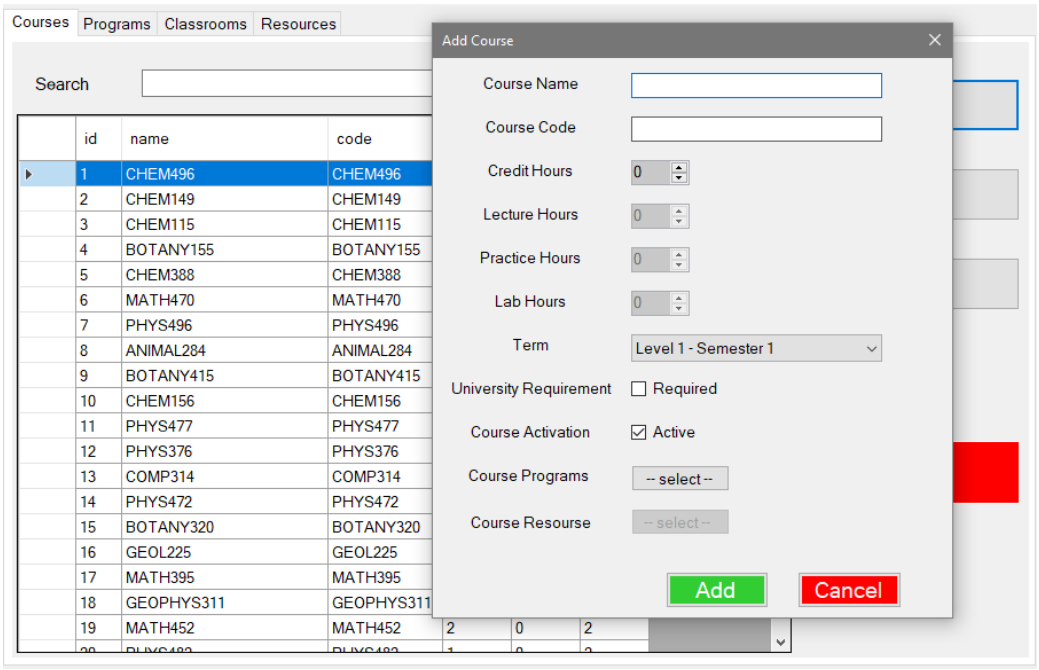


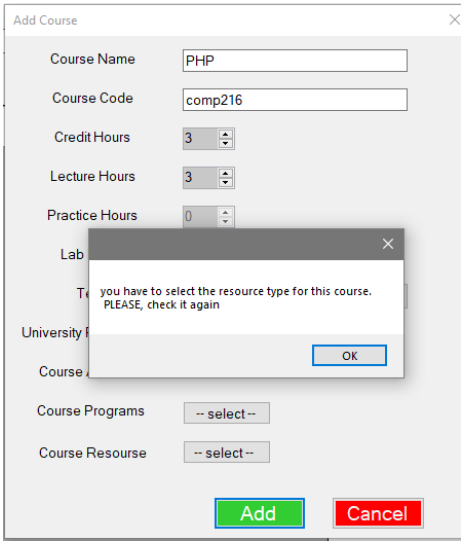
	id	name	code	lec hours	sec hours	lab hours
▶	1	CHEM496	CHEM496	1	0	2
	2	CHEM149	CHEM149	2	0	2
	3	CHEM115	CHEM115	1	0	0
	4	BOTANY155	BOTANY155	1	1	3
	5	CHEM388	CHEM388	2	0	2
	6	MATH470	MATH470	1	0	0
	7	PHYS496	PHYS496	2	0	2
	8	ANIMAL284	ANIMAL284	1	0	0
	9	BOTANY415	BOTANY415	1	0	0
	10	CHEM156	CHEM156	1	1	1
	11	PHYS477	PHYS477	1	3	1
	12	PHYS376	PHYS376	1	1	1
	13	COMP314	COMP314	2	1	1
	14	PHYS472	PHYS472	1	0	0
	15	BOTANY320	BOTANY320	1	0	2
	16	GEOL225	GEOL225	1	0	0
	17	MATH395	MATH395	1	0	2
	18	GEOPHYS311	GEOPHYS311	1	1	1
	19	MATH452	MATH452	2	0	2

1-1- Adding new course window

After clicking on the new button the following window is appearing and it allows user to insert a new course with the required fields (name of the course (can be Arabic), code of course (can be Arabic), credit, lecture, practice, lab hours, term that the course is taken in, university requirement to see if the course is required course form the university or not, Activation for selecting is the course active or not, course programs for selecting the programs that take this course

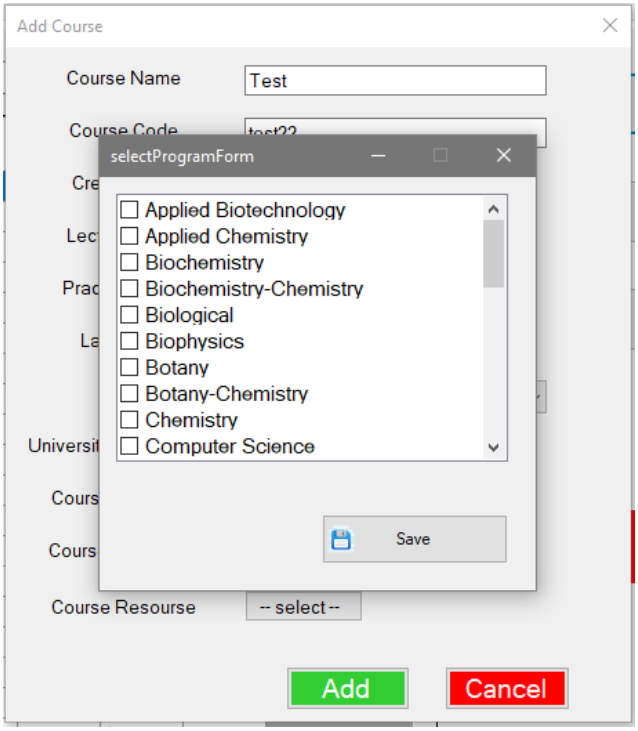
and course resources is activated when there's lab hours for this course and it's allow user to select type of lab needed for these lab hours) and if user add some lab hours without adding a resource type for it, he will receive an alert to add some resources then click on the add button.





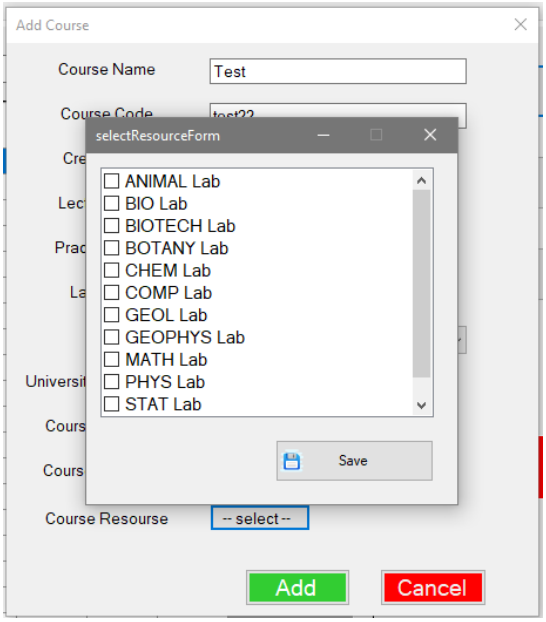
1-1-1- Select programs for this course.

Here, you have to select the programs that takes this course (it's arranged alphabetical).



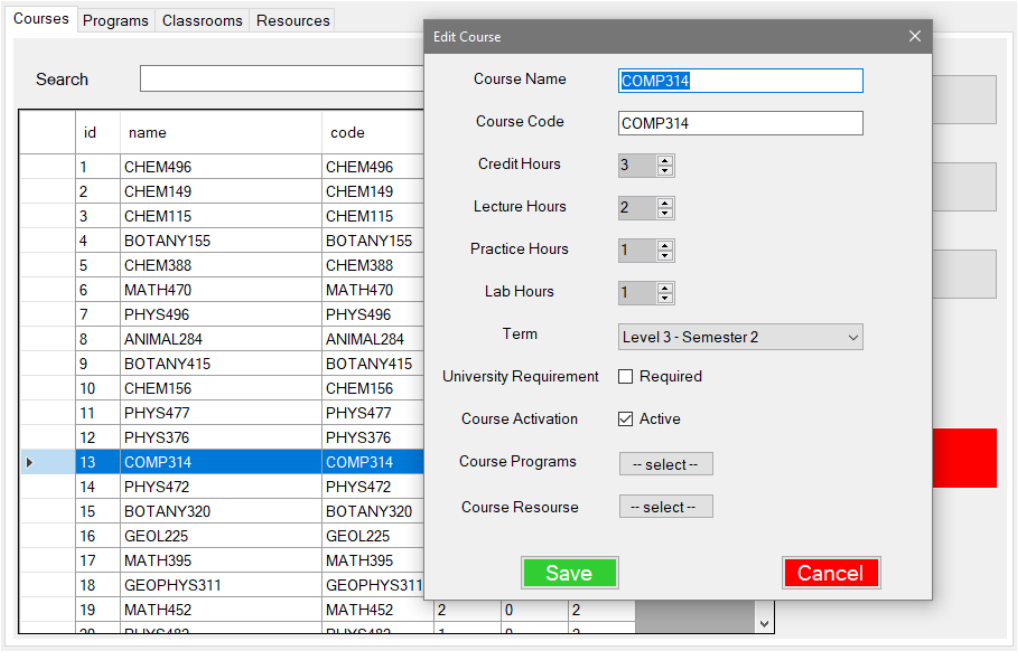
1-1-2- Select resources for this course.

Here, the select resource form will be active when the course has lab hours and you have to select the type of resources required for this course (it's arranged alphabetical) and then click on the save button.



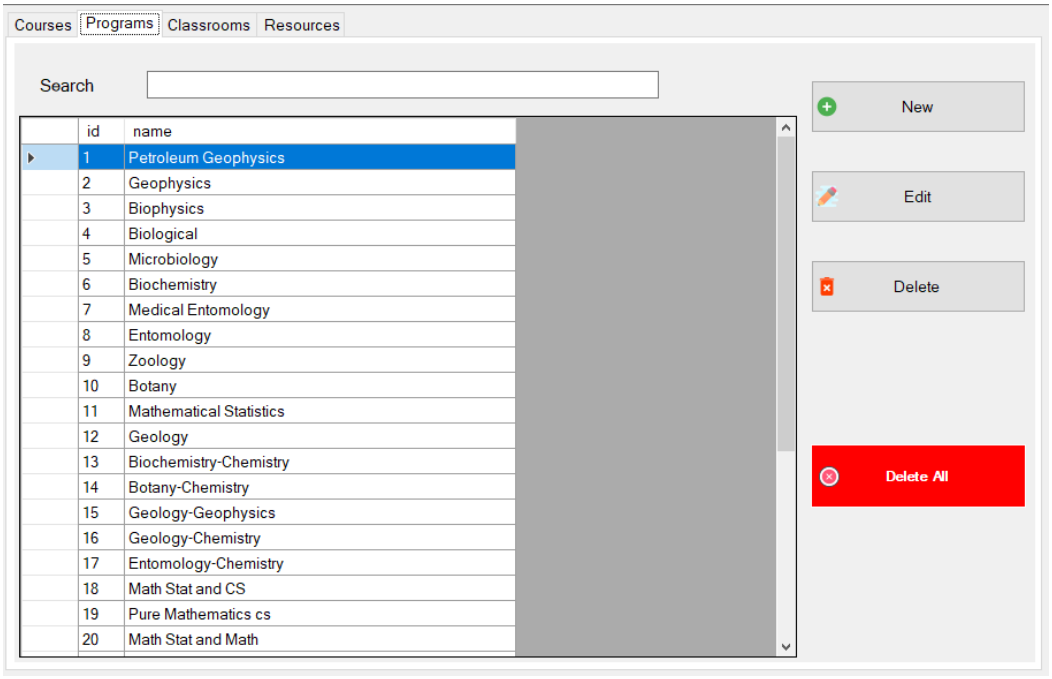
1-2- Edit specific course window

This form appears when user select a specific course and then click on the Edit button or double click on it. After clicking the form appears with complete information for this course and you can edit what you want and then click on the save button.



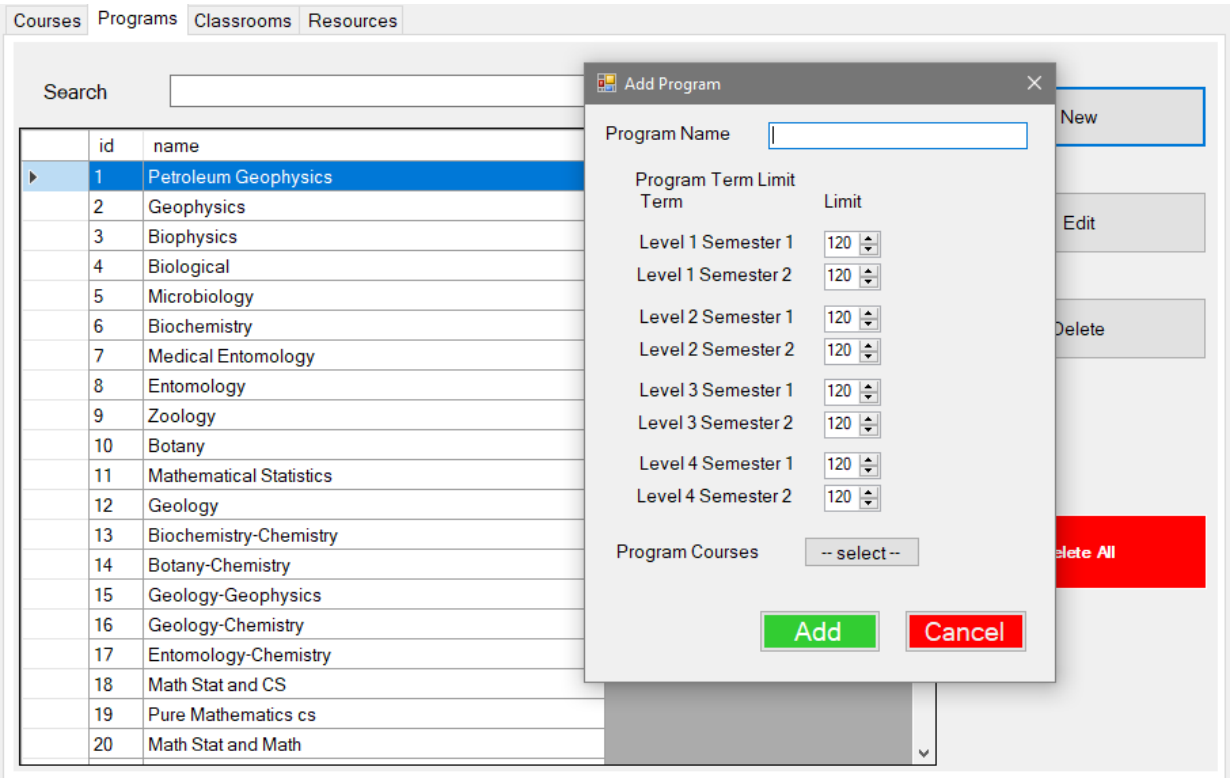
2- Program view window

It's the same view as course window but the information is related to programs.



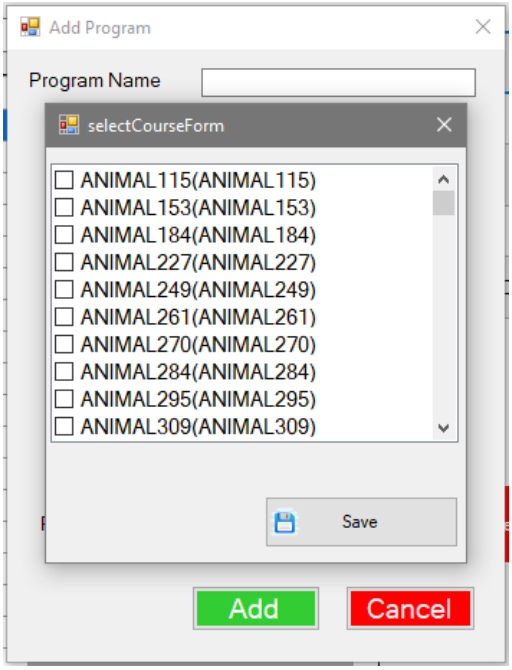
2-1- Adding new program window

After clicking on the new button the following window is appearing and it allows user to insert a new program with the required fields (program name (can be Arabic), program term limit is to set limit to every program at every term, program courses is an optional selection for selecting specific courses to this program because it will help user after fill all data as the recommend way to assign new courses to specific program) and then click on the add button.



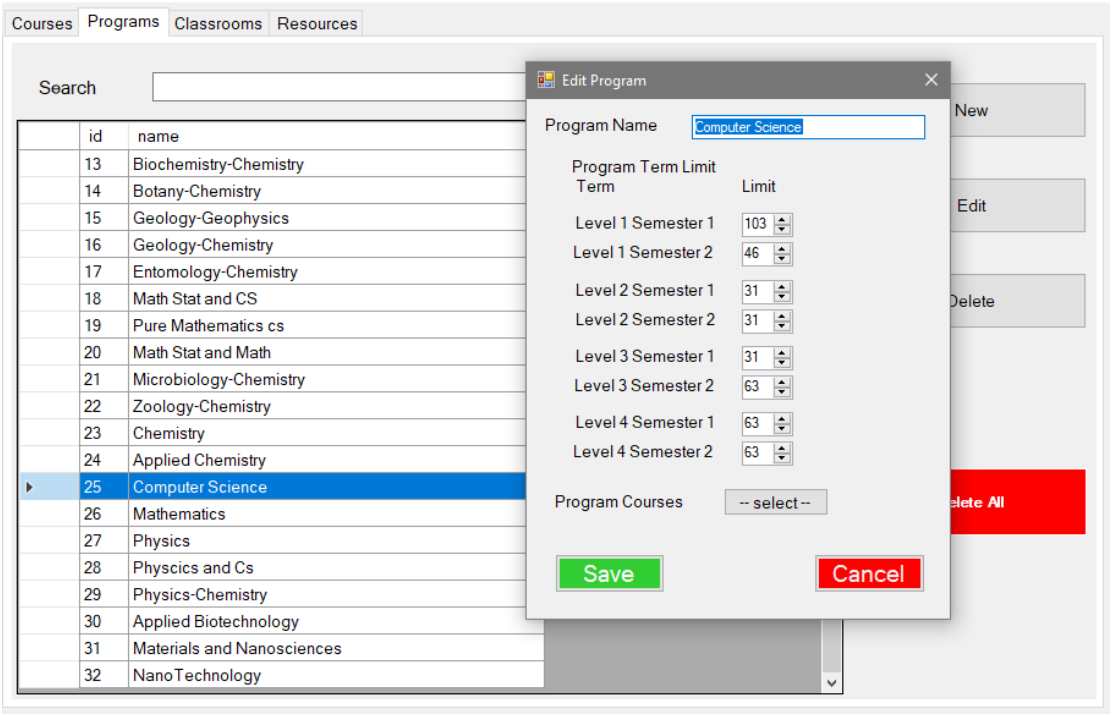
2-1-1- Select courses for this program

In this form it's optional to assign courses to this program.



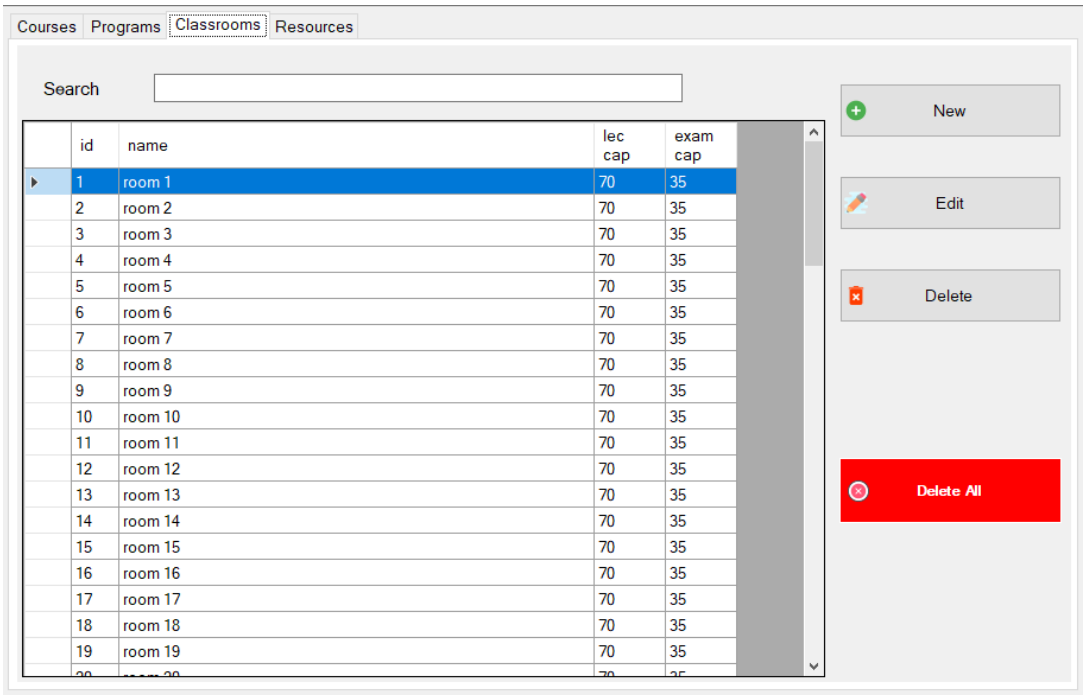
2-2- Edit specific program window

It’s the same as Edit course window.



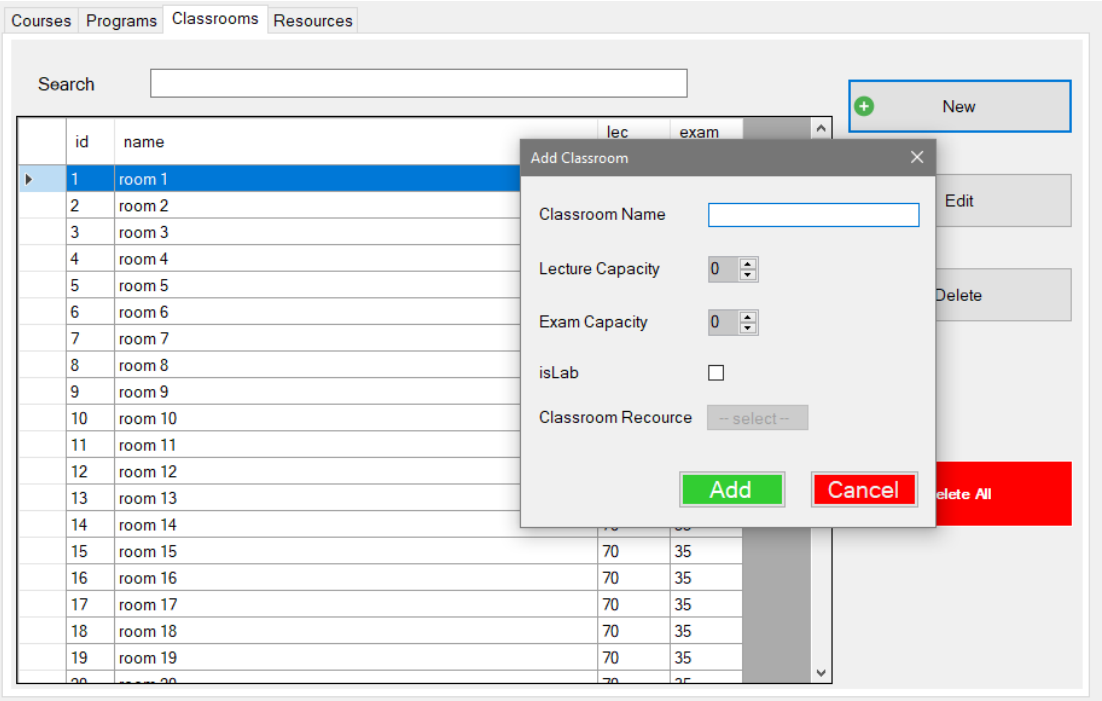
3-Classroom view window

It’s the same view as course window but the information is related to classrooms.



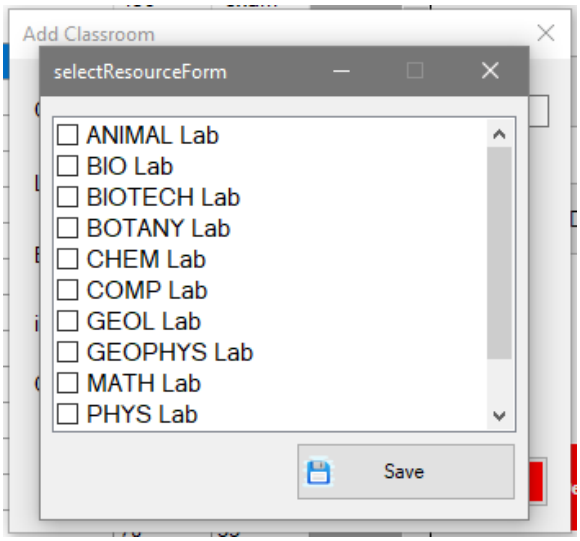
3-1- Adding new classroom window

After clicking on the new button the following window is appearing and it allows user to insert a new classroom with the required fields (classroom name (can be Arabic), lecture capacity for this classroom, exam capacity, isLab is a check box to differ between labs and halls, classroom resource is activated when the isLab is checked to select the type of lab) and then click on the add button.



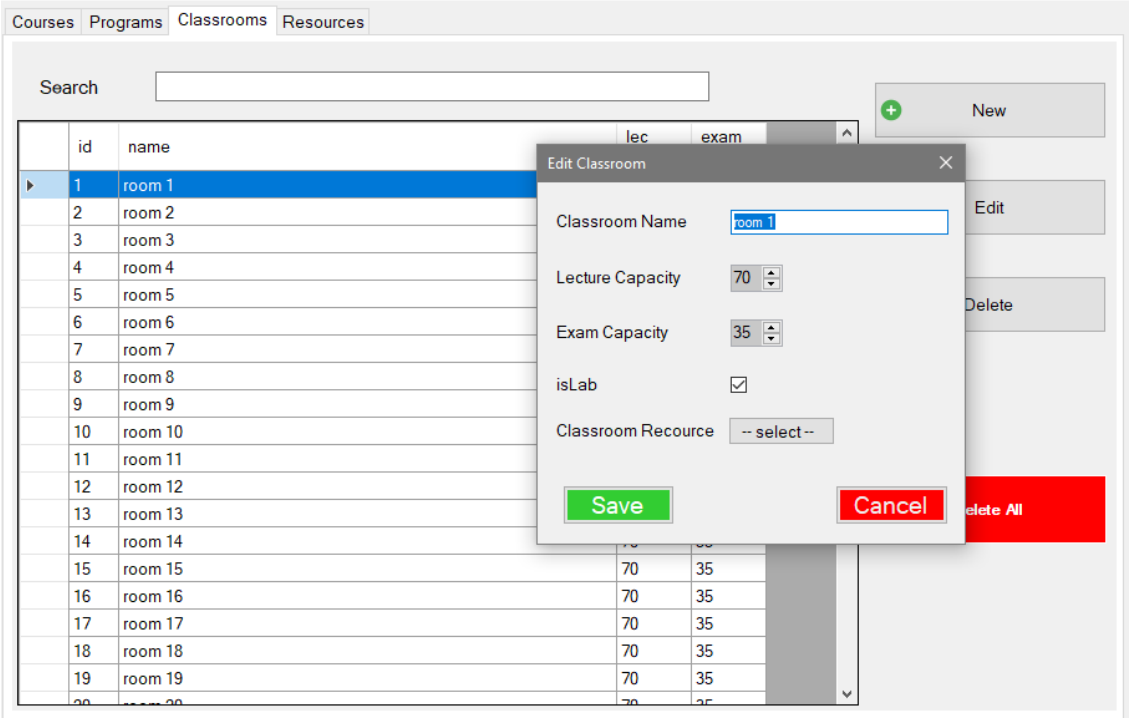
3-1-1- Select resources for this classroom

Here, the select resource form will be active when the classroom is lab and you have to select the type of resources required for this classroom (it's arranged alphabetical) and then click on the save button.



3-2- Edit specific classroom window

It’s the same as Edit course window.

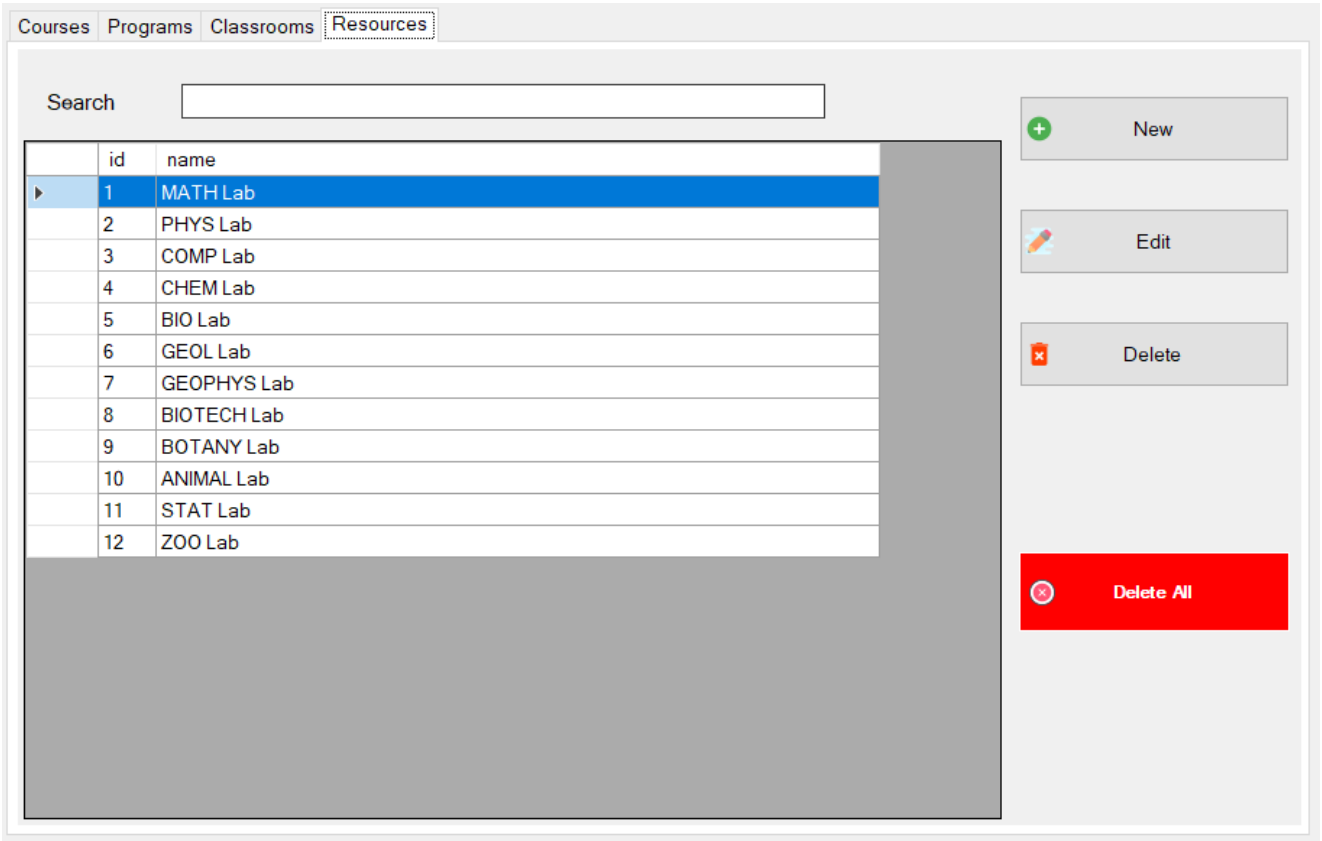


4- Resource view window

It’s the same view as course window but the information is related to resource. And we mean by resources the type of labs (E.g. the computer science program can’t take their section in bio lab).

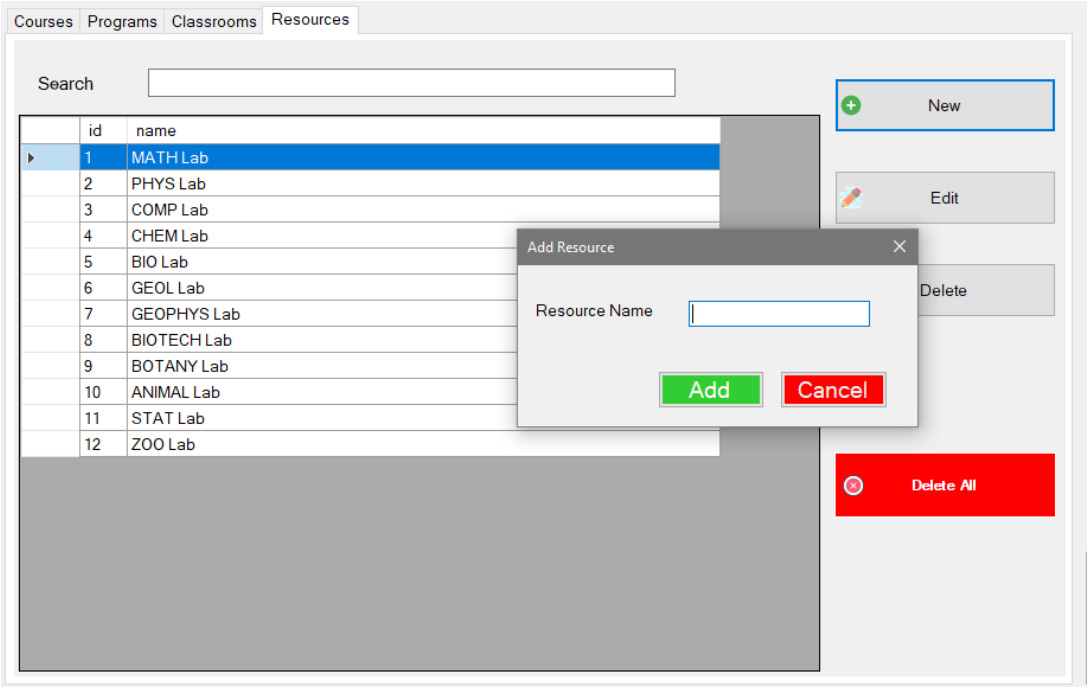
So recommended that user insert the resources as in the screen. And if he wants to insert some other things (ex: lab contains a data show or white board) this makes it harder for the scheduler to find a suitable room.

ALERT: IF A COURSE NEEDS A RESOURCE AND THE RESOURCE DOESN’T EXIST IN ANY ROOM.
THE SCHEDULER WILL FAIL TO PLACE THIS COURSE!!!



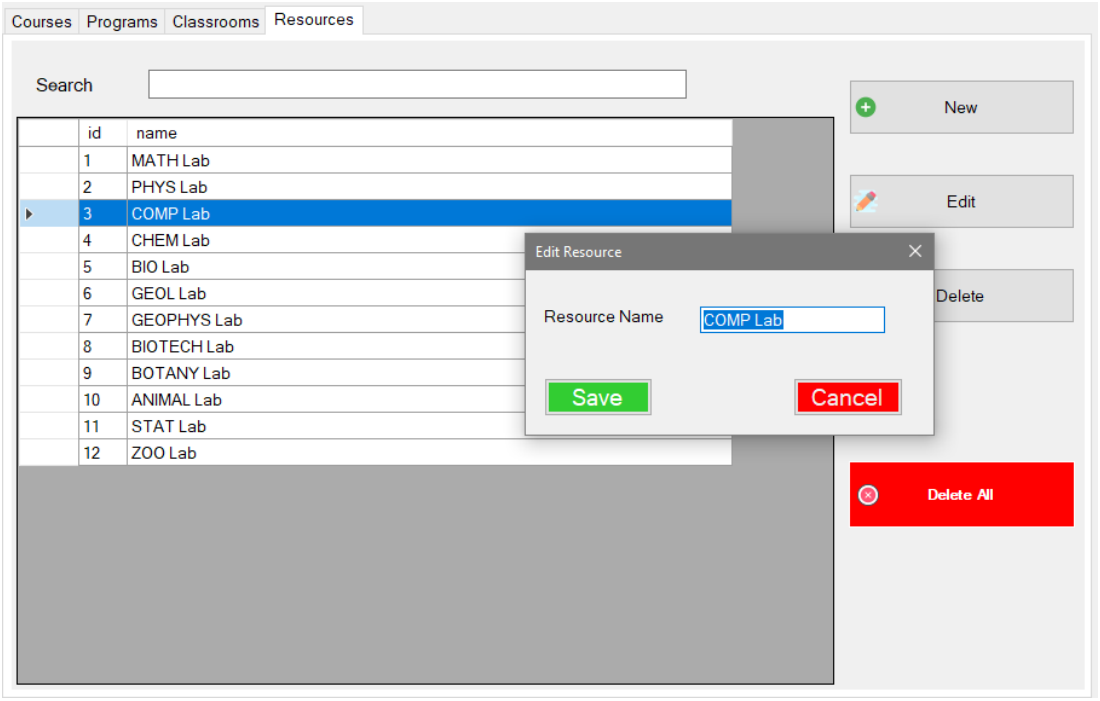
4-1- Adding new resource window

After clicking on the new button the following window is appearing and it allows user to insert a new resource with the required fields (resource name (can be Arabic))and then click on the add button.



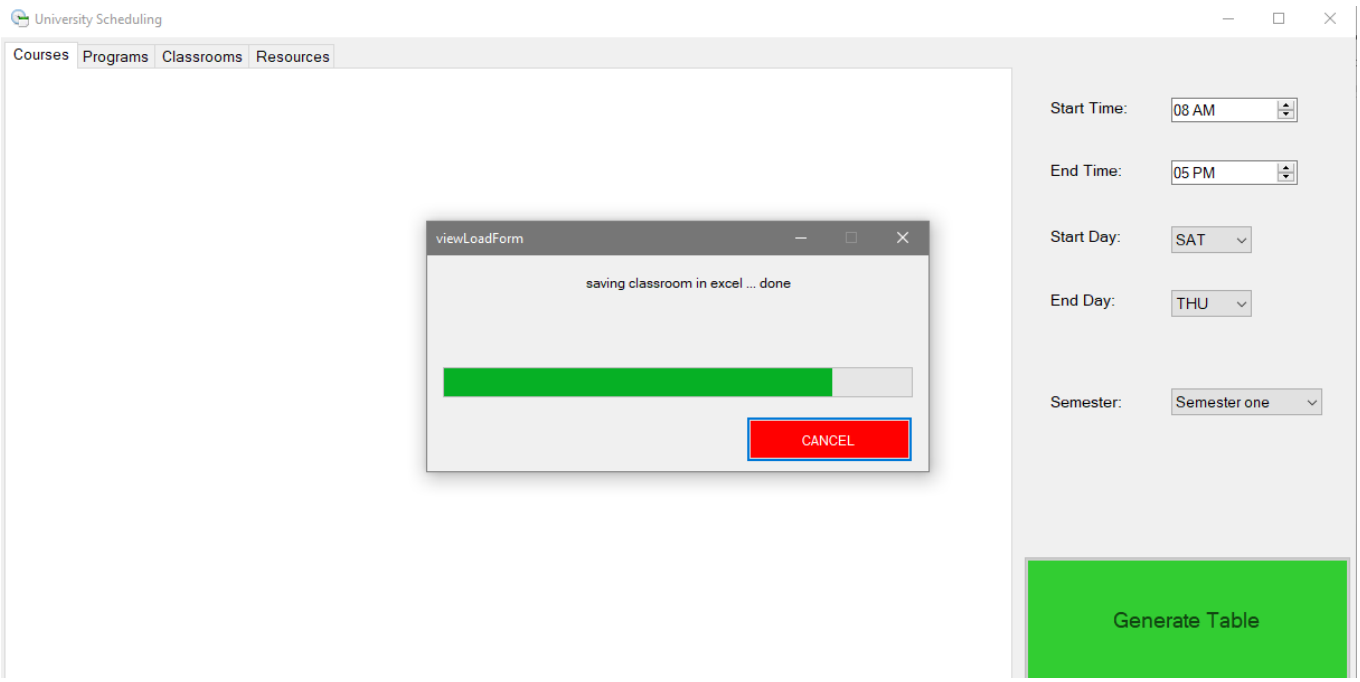
4-2- Edit specific resource window

It’s the same as Edit course window.



<u>Third, we will represent the loading dialog that shows the progress bar for work that have been done!.</u>

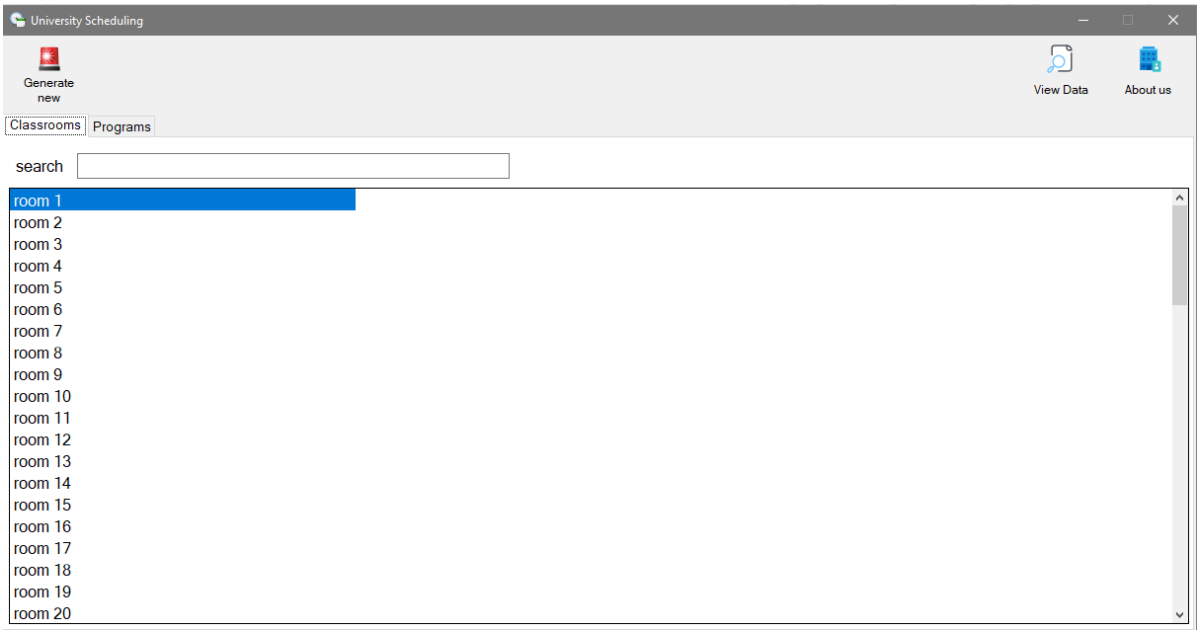
The progress bar shows the number of courses that is reserved and some other information related to the excel sheet that have been created.



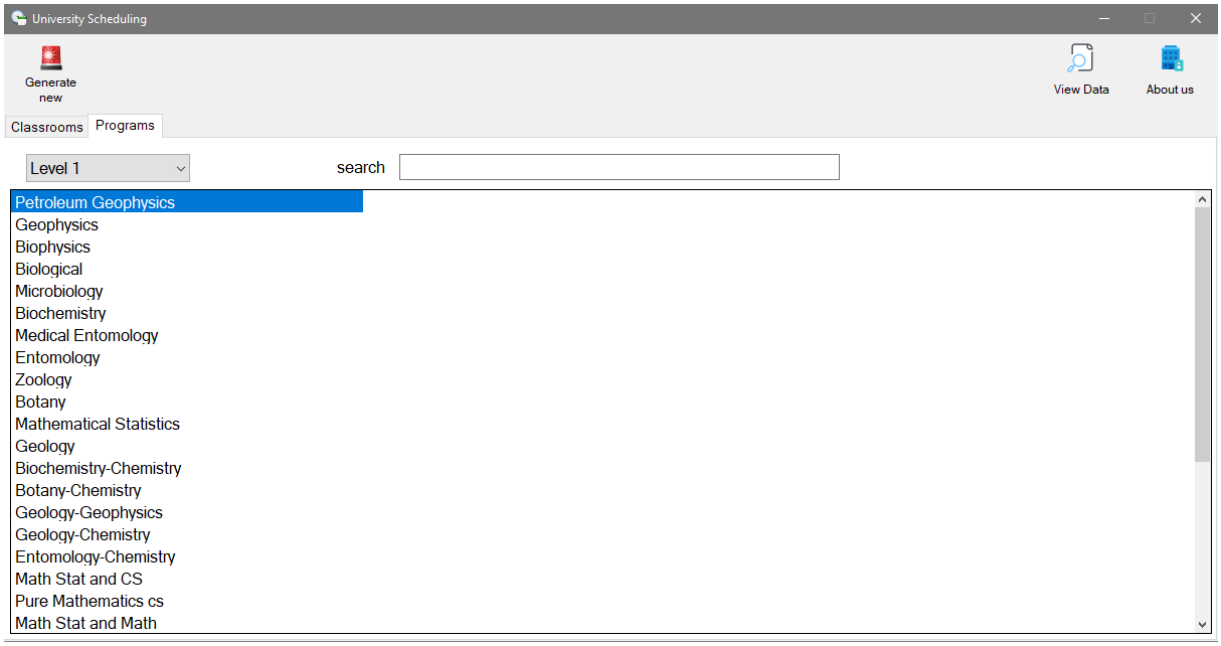
Fourth, we will represent the window after the table is reserved: -

Here, the timetable is created and user can view the table for classrooms or programs within search for the classroom or the program he wants and then double click on it to open the excel book where he can print the timetable or view the other classrooms or programs within the excel sheets.

* Classrooms is just represented as its name as the following:

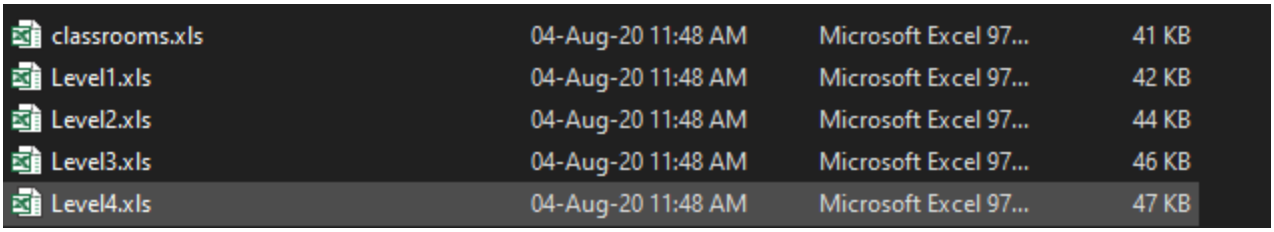


* Programs is shown as:

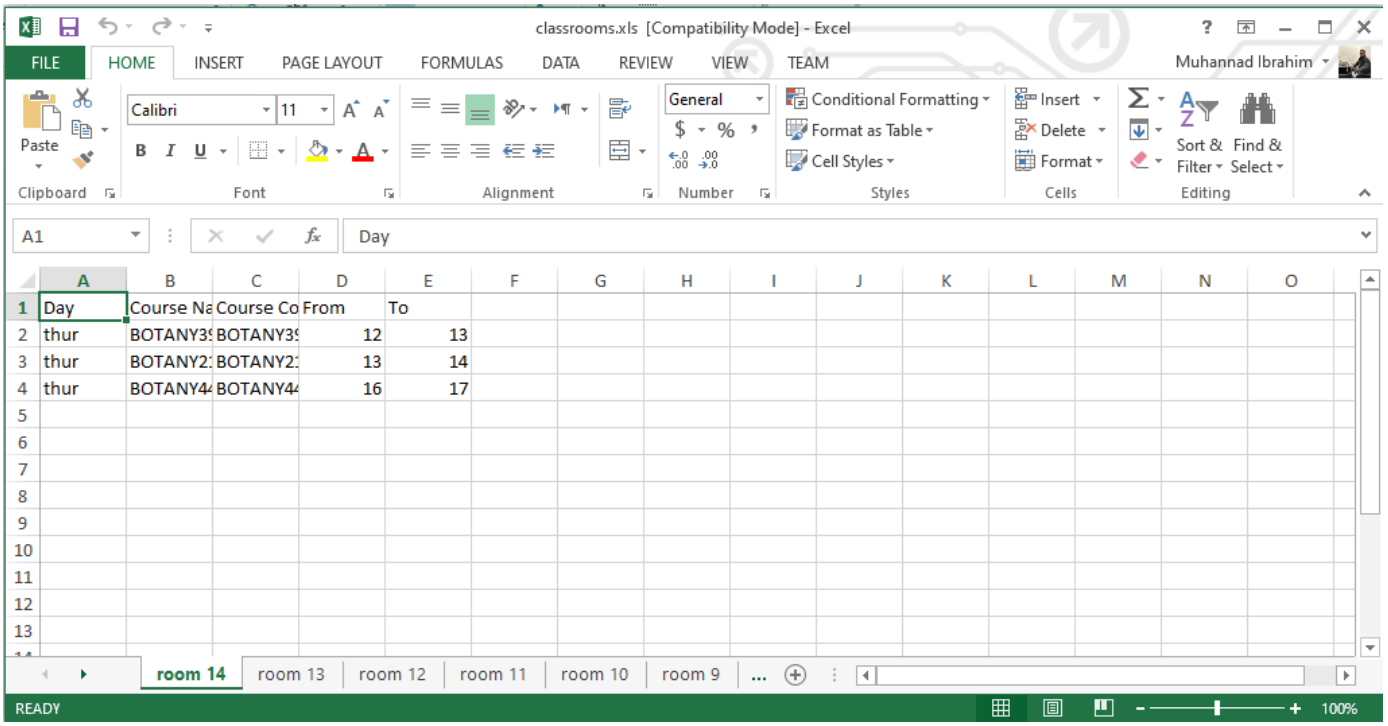


This interface allows user to view the time table for specific program in specific level by selecting the level and then double click on the program.

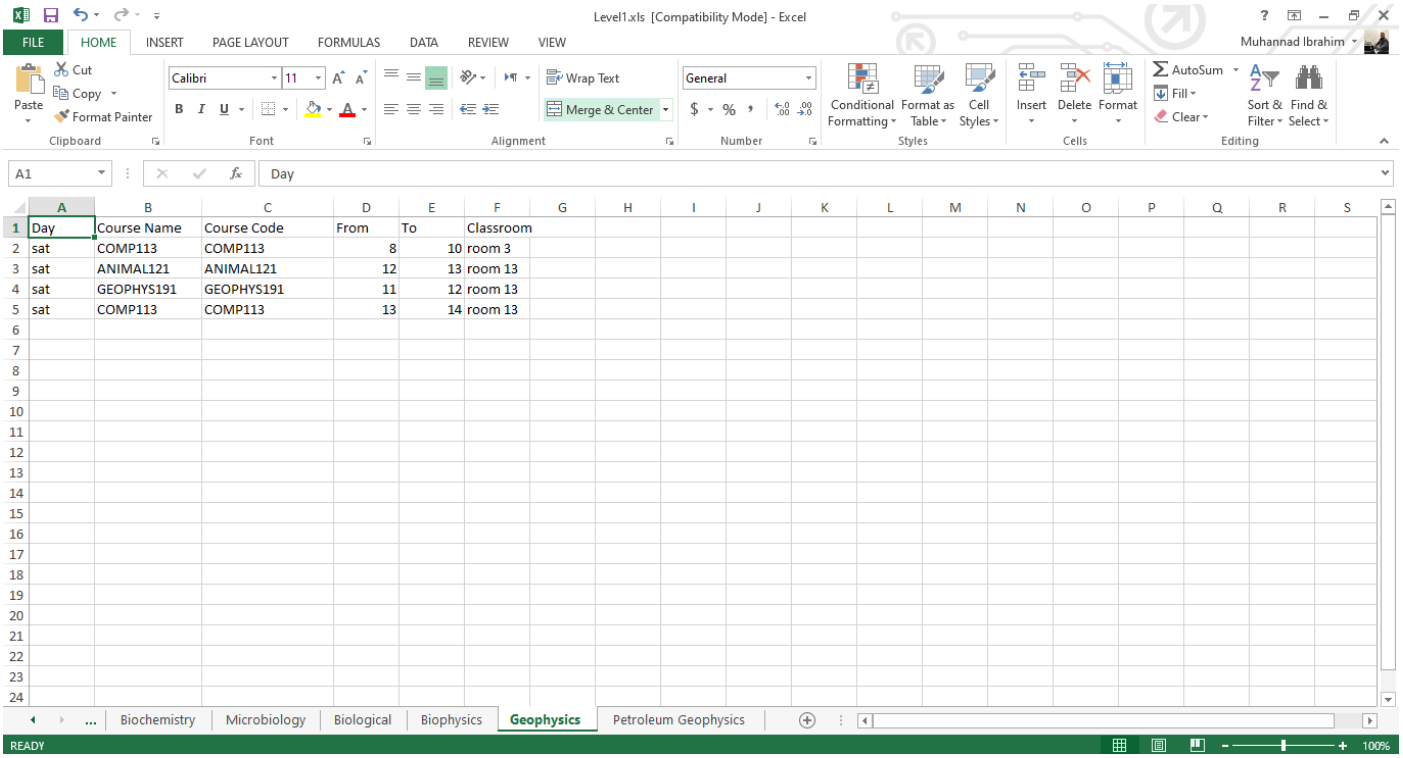
All timetables is saved in the folder of the project as 5 excels (one for classrooms and four for Levels for each program).



* Excel of the classrooms:

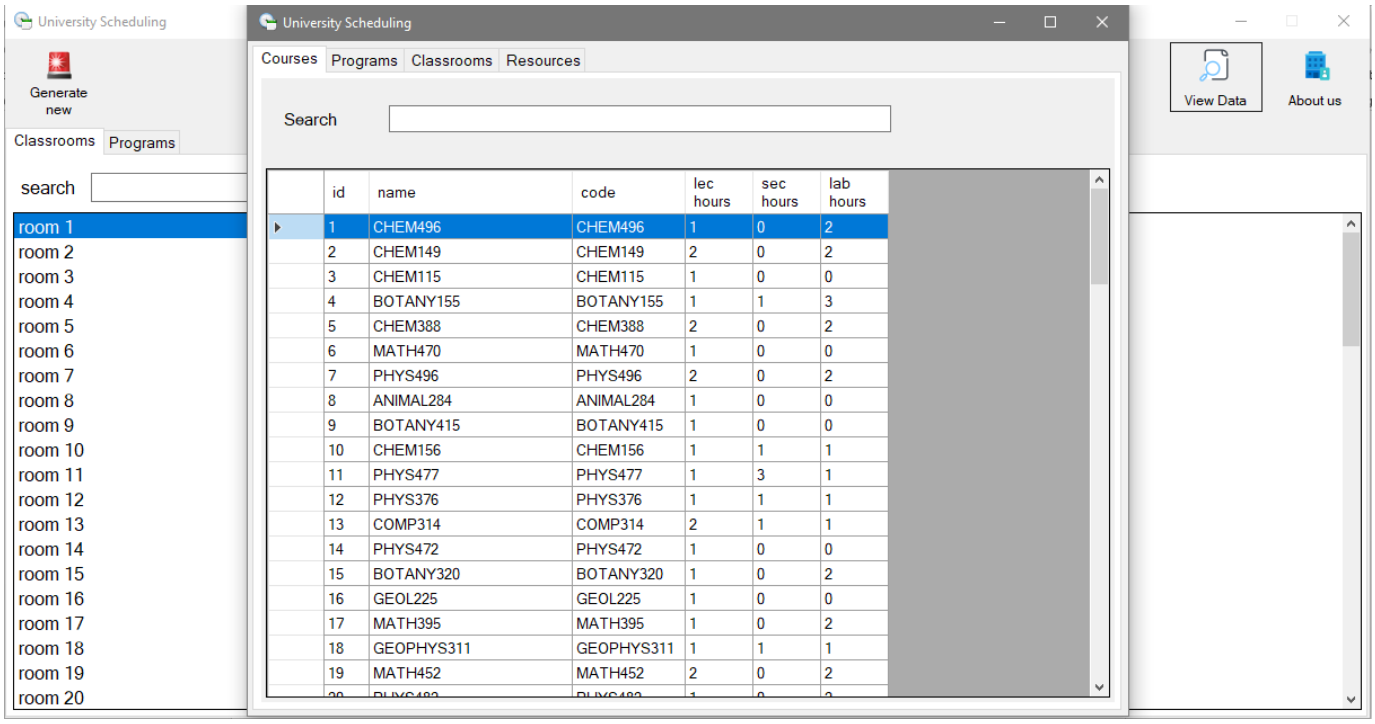


* Excel for Levels of the programs is shown as:



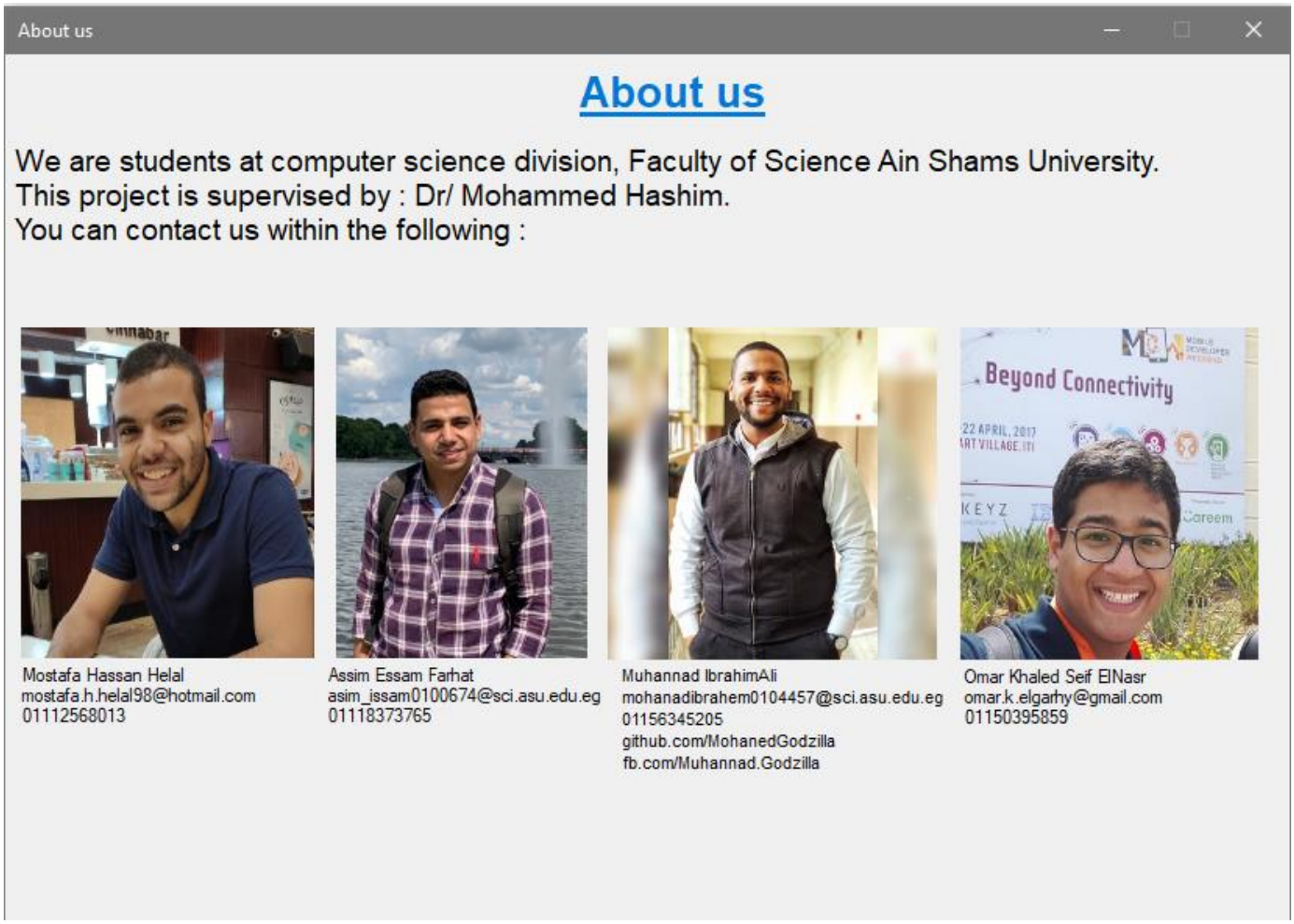
	A	B	C	D	E	F
1	Day	Course Name	Course Code	From	To	Classroom
2	sat	COMP113	COMP113	8	10	room 3
3	sat	ANIMAL121	ANIMAL121	12	13	room 13
4	sat	GEOPHYS191	GEOPHYS191	11	12	room 13
5	sat	COMP113	COMP113	13	14	room 13

1. The view Data button is used to review the entered data

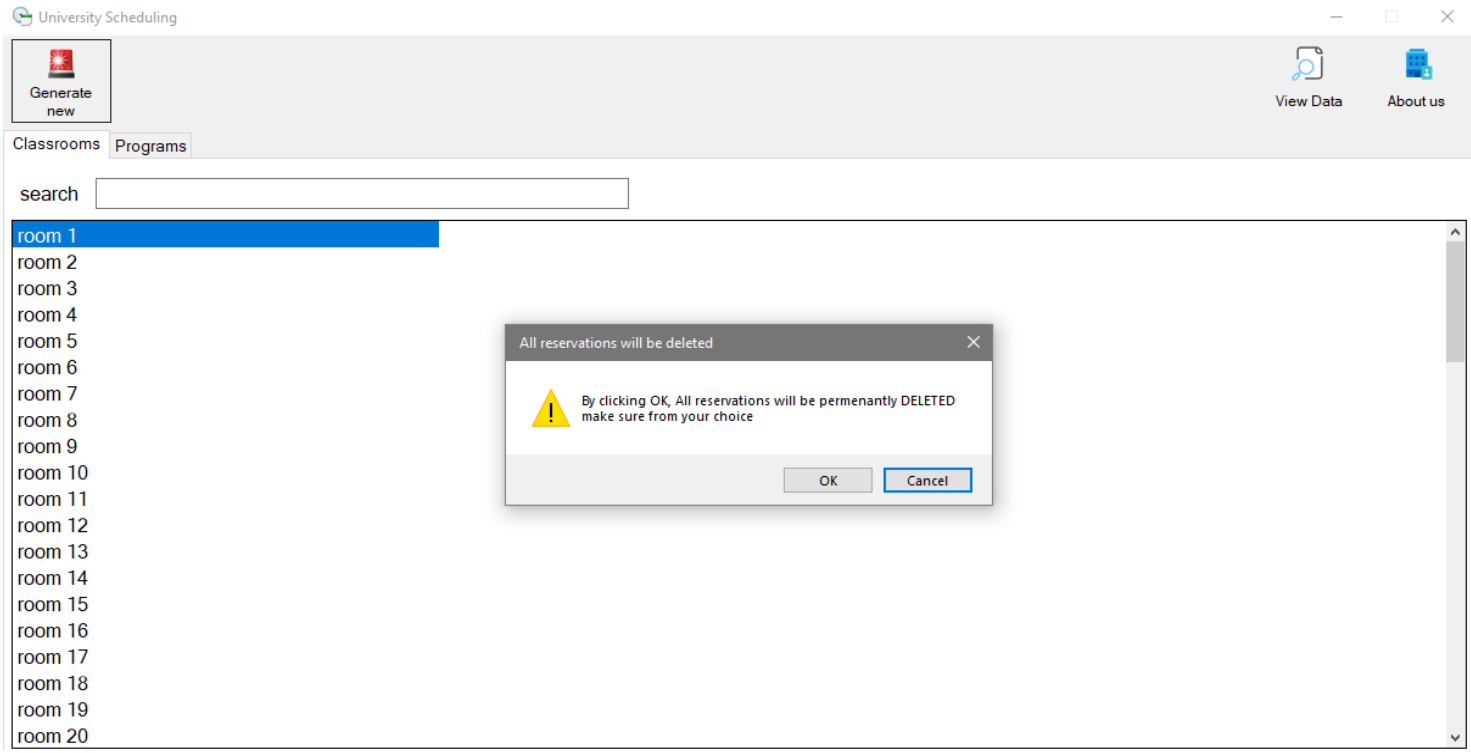


	id	name	code	lec hours	sec hours	lab hours
▸	1	CHEM496	CHEM496	1	0	2
	2	CHEM149	CHEM149	2	0	2
	3	CHEM115	CHEM115	1	0	0
	4	BOTANY155	BOTANY155	1	1	3
	5	CHEM388	CHEM388	2	0	2
	6	MATH470	MATH470	1	0	0
	7	PHYS496	PHYS496	2	0	2
	8	ANIMAL284	ANIMAL284	1	0	0
	9	BOTANY415	BOTANY415	1	0	0
	10	CHEM156	CHEM156	1	1	1
	11	PHYS477	PHYS477	1	3	1
	12	PHYS376	PHYS376	1	1	1
	13	COMP314	COMP314	2	1	1
	14	PHYS472	PHYS472	1	0	0
	15	BOTANY320	BOTANY320	1	0	2
	16	GEOL225	GEOL225	1	0	0
	17	MATH395	MATH395	1	0	2
	18	GEOPHYS311	GEOPHYS311	1	1	1
	19	MATH452	MATH452	2	0	2

2. The about us button is to show some contacts about us



3. The generate new button is for generating new table
   If user click on the button, the program will give him alert to be carefully that all reserved data will be deleted.



Testing techniques and test data

We tested the algorithm and data generator first without any UI using Dart Programming language for its simplicity to see if the concept of the algorithm works then we translated it to C#.

The algorithm had to be challenged with a large amount of data that had to simulate a real-life data.

We used the collage data as a reference to make the "data generator" more real.

The Data Generator generates Programs with their Courses(with its resources), Classrooms for lectures and labs with variant sizes and resources.

Future work

UI/UX Enhancements

- Show reservations in app instead of exporting it in excel files for better searching and analysis
- Allow adding class blocked hours from UI

Futures that could be added using the same algorithm concepts

- Generate a timetable for exams period: -

  Redesign the Algorithm to take the student as a main input with the required information needed to generate the timetable for the exams.

References

Main Idea of the algorithm is based on **Mrs.Marwa** Prioritizing method.

Algorithms: -

- Papers from Dr. Muhammed Hashim: -

  The papers contains different algorithms based on genetic algorithm, AI, Micro Services and others that are used to schedule timetable.

  Article: A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems

UI/C#/Database/Excel: -

YouTube links: -

- How to Create Simple C# Desktop Application?
- C# Tutorial - Insert Update Delete Search data from local database
- C# Tutorial - How to pass data from one Form to another Form
- C# Tutorial - Load Form in Panel C# Application
- How to connect SQL server database with visual Studio C#
- How to Export Data to Excel
- Search data in database and Filter in datagridview or table in C#
- How to Create Setup.exe file in Visual Studio 2017 with SQL database