

# UNIVERSITY TIME-TABLE SCHEDULING SYSTEM: DATA-BASES DESIGN

Dr. Samson Oluwaseun Fadiya, Management Information System (PhD)  
Girne American University, Mersin 10 via Turkey, Email: [samsonfadiya.gau.edu.tr](mailto:samsonfadiya.gau.edu.tr)

Nwabuzor E. Iruoma, Management Information System (BSc)  
Girne American University, Mersin 10 via Turkey, Email: [iruoma@gmail.com](mailto:iruoma@gmail.com)

Received: April, 25.2015.

Accepted: June, 29.2015.

Articles

**Abstract.** Automated System Preparing Time-tables (ASPT) system was designed to assist the organizations in organising and maximizing their time. Interactive system ASPT forms a part of a bigger project that concerns time-table scheduling system in higher learning.

The aim of this project is to prepare a computerized and automated system of preparing timetables for the business faculty in order to save time, prevent clashes, ensure maximum use of the facilities and also create a better learning environment for the student and a more suitable teaching environment for the lecturers.

**Keyword:** Automation, database, SQL, schedule, system, management.

## 1. INTRODUCTION

A multi-continental university in different countries is very robust in nature and in handling files. The most popular of its faculties and also the biggest of them is the business faculty which has a huge number of students and even more coming each year. With the large number of students and also the business and engineering faculty sharing some of the same classes and even some of other faculties taking some of its courses in the same facility, the preparation of the business faculty timetable could be very hectic and tiresome because of the manual way it is being done.

Some of this Universities uses a manual system for the selection of courses and preparation of timetables, thereby resulting in clashes when done by the lecturers. But with the increasing number of students every semester, this doesn't just cause clashes but also wastes time.

Corresponding Author

Dr. Samson Oluwaseun Fadiya, Management Information System, Girne American University, Mersin 10 via Turkey, Email: [samsonfadiya.gau.edu.tr](mailto:samsonfadiya.gau.edu.tr)

## 2. DESIGN

A database package would be needed in order to store all the details about the lecturer, students, classes, etc. For the back end there are many possibilities for the database such as Microsoft Access, MySQL, SQL etc.

MySQL is popular for handling very powerful and large databases but it also has a couple of setbacks some of which are it does not support transactions or concurrent update of complex table which if used would have become a huge hindrance to the development of the system in the making. Oracle is very well known for the use of SQL statements. It is one of the most expensive databases on the market. There is no graphical interface for creation or modification of tables. This implies that the use of this database is very time consuming

But for the database of this project we would be making use of Microsoft SQL. SQL is preferred out of the other databases due to some reasons. Being one of the database pioneers, SQL has become more easy to use as well as learning because of this it make developing a new database system much quicker compared to others such as Oracle, My sql etc. It is a cheaper software and also more affordable. Also it is easier to learn.

For the front end of this project, we would be making use of Visual C#, this is because visual C# is a widely available, commercial interpreter that are available for most personal computers. Also the interface includes very familiar GUI controls such as list boxes, buttons, combo boxes, etc. and with just a few mouse clicks, one could add a combination of these GUI controls to the program

making it easier and faster to design the user interface of the program.

## 2.1. Database Design

The database would consist of eight (8) entities;

- FACULTY,
- DEPARTMENT,
- LECTURER,
- CLASSROOM,
- CLASS,
- COURSES,
- LOCATION,
- USERS.

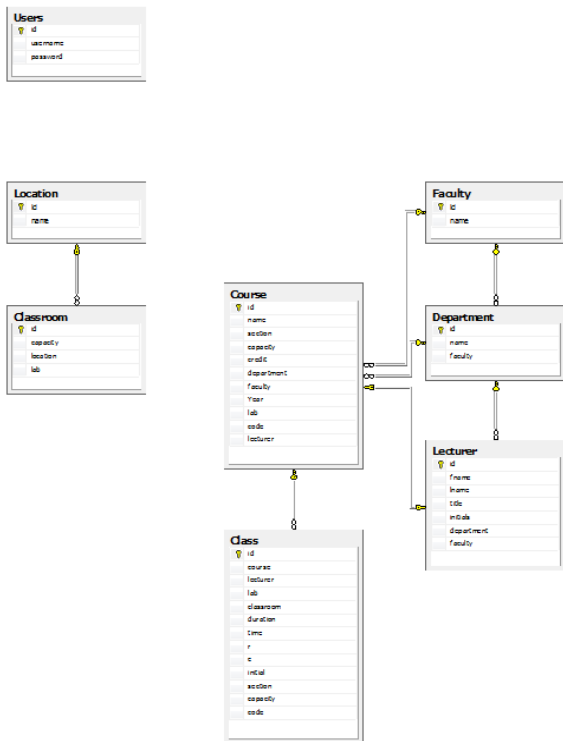


Figure 1. Database Diagram

## 2.2. Definition of Data Requirements

Table 1. Faculty Table

Attribute Name	Comments	Data Type and Length	Validation
ID	Just one faculty Automatically incremented	Nvarchar(50)	Unique primary key
Name		Varchar(100)	

Table 2. Department Table

Attribute Name	Comments	Data Type and Length	Validation
ID	Automatically incremented	Integer	Unique primary key
Name		Varchar(100)	
Faculty		Nvarchar(50)	Must exist on FACULTY table

Table 3. Lecturer Table

Attribute Name	Comments	Data Type and Length	Validation
ID	Automatically Incremented	Integer	Unique primary key
Initials		Varchar(5)	Unique
Title		Nvarchar(30)	
Lname		Varchar(50)	
Fname		Varchar(50)	
Faculty		Nvarchar (50)	Must exist on FACULTY table
Department		Integer	Must exist on DEPARTMENT table

Table 4. Course Table

Attribute Name	Comments	Data Type and Length	Validation
ID		Varchar(10)	Unique primary key
Name		Varchar(100)	
Section		Integer	
Capacity		Integer	
Credit		Integer	
Department		Integer	Must exist on DEPARTMENT table
Faculty		Nvarchar(50)	Must exist on FACULTY table
Year		Integer	
Lab	Check if the class is a lab or not	Bit	
Code		Varchar(10)	
Lecturer		Varchar(5)	Must exist on LECTURER table

Table 5. Location Table

Attribute Name	Comments	Data Type and Length	Validation
ID	Automatically incremented	Integer	Unique primary key
Name		Varchar(20)	

Table 6. Classroom Table

Attribute Name	Comments	Data Type and Length	Validation
ID		Varchar(10)	Unique primary key
Lab	Check if the class is a lab or not	Bit	
Capacity		Integer	
Location		Integer	Must exist on LOCATION table

Table 7. Class Table

Attribute Name	Comments	Data Type and Length	Validation
ID	Automatically incremented	Integer	Unique primary key
Course		Varchar(10)	Must exist on COURSE table
Initials		Varchar(5)	Must exist on LECTURER table
Lab		Bit	
Classroom		Varchar(10)	Must exist on CLASSROOM table
Duration	Duration of the class	Integer	
Time	Represents the ID of a cell on the table	Nchar(20)	
R	Represents the ID of a row on the table	Integer	
C	Represents the ID of a column on the table	Integer	
Code		Varchar(10)	

Attribute Name	Comments	Data Type and Length	Validation
ID	Automatically incremented	Integer	Unique primary key
Username		Varchar(200)	
Password		Varchar(20)	

## 2.3. Design of Pages

### 2.3.1. Login Page

This page is used for the login of the Admin (who is in charge of the program). The admin is strictly in charge of the preparation of the timetable and the editing of the data that would be used in the preparation of the timetable. He is the only one for now who makes use of the software.

### 2.3.2. Lecturers Icon

In this icon we have three controls under it, one of it is to add lecturers, the other is to view lecturers and the last one is to search for a lecturer and either edit or delete the lecturer's details from the database.

Adding lecturers: This form would be used for several purposes:

- To input a lecturer's ID: this is validated to only input a maximum amount of numbers.
- To input a lecturers initials:
- To input a lecturer's title: that could either be PHD, Asst, Dr. and the likes. There would be a drop down menu where the admin can choose from.
- To input a lecturer's name
- To input a lecturer's surname
- To input a lecturer's department: here there is also a drop down menu where the lecturer can choose his or her department.
- To input a lecturer's preferable days: also here the lecturer can only choose one preferable day he wishes to teach.
- To input a lecturer's preferable time: also here the lecturer can only choose one preferable time he wishes to teach either; Morning, After-

noon or Evening.

- Then there is an add button that is being clicked on only when all the details are being inputted unless it brings an error message.
- 
- Search, edit, and delete a lecturer: This page is to search, edit or delete a lecturer
- Here one can search for a lecturer either with the lecturer's ID or name.
- Then the list of the lecturers are displayed on a table under
- To edit a lecturer's details, you have to click on it and change the details and then click on the edit button.
- To delete a lecturer's details, one has to click on the lecturer's details and click on the delete button.

### 2.3.3. Classroom Icon

In this icon we have two controls under it also either to add a classroom or to search/edit or delete a classroom.

Adding a classroom: This form would be used for several purposes

- To input a classroom name: here the admin enters a unique classroom name for a class
- To input a classroom location: this is to input where the classroom is located in the school either techno park or spectrum etc.
- To select if it is a lab or not.
- To input the capacity of each classroom.

Search, edit, and delete a lecturer: This page is to search, edit or delete a lecturer

- Here one can search for a classroom with the classroom's name.
- Then the lists of the classroom are displayed on a table below.
- To edit a classroom's details, you have to click on it and change the details and then click on the edit button.

- To delete a classroom's details; click on the details and click on the delete button.

### 2.3.4. Courses Icon

In this icon we have two controls under it either to add courses or to search/edit or delete a course.

Adding courses: This form would be used for several purposes

- To input a course code: this is validated to only input a unique course code.
- To input course name: here, the name of the course is being inputted.
- To input the number of sections: this is to select how many sections this course would have
- To input the maximum number of students: Here the maximum number of students for the course is inputted
- To input the total teaching hours: The total teaching hours for the course is selected, usually it's a maximum of three hours and a minimum of one hour.
- To input a course department: This is used to select which department the course falls under.

Search, edit, and delete a course: This page is to search, edit or delete a course

Here one can search for a course either with the course ID or name.

Then the course/courses are displayed on a table below

To edit a course details, you have to click on it and change the details and then click on the edit button.

To delete a course details; click on the course details and click on the delete button.

### 2.3.5. Timetable Icon

Here the time slots are shown, and when a slot is selected, if it for the time one can choose the start or end time, if it is for the courses, a popup menu comes up and then the

course code, classroom and lecturer is being chosen for that slot. Whenever a constraint is being breached, an error message is displayed

## 2.4. Diagrams of Pages

### 2.4.1. Login Page

Figure 2. Login Page

### 2.4.2. Add Courses Page

Figure 3. Add Courses Page

### 2.4.3. Search Courses Page

Course Code	Course Name	No of Sections	Department	Max no of students	Total class hours

Figure 4. Search courses Page

### 2.4.4. Add Lecturers' Page

Figure 5. Add Lecturers' Page

2.4.5. Search Lecturers' Page

Figure 6. Search Lecturers' Page

2.4.6. Add classroom page

Figure 7. Add Classroom Page

2.4.7. Search Classroom Page

Figure 7. Add Classroom Page

Figure 8. Search Classroom Page

2.4.8. Timetable page

Figure 9. Timetable

2.4.9. Popup menus for add class

Figure 10. Popup menu for editing a class information

2.4.9.1. Use Case Diagram

USE CASE DIAGRAM: UNIVERSITY TIME-TABLE SCHEDULING

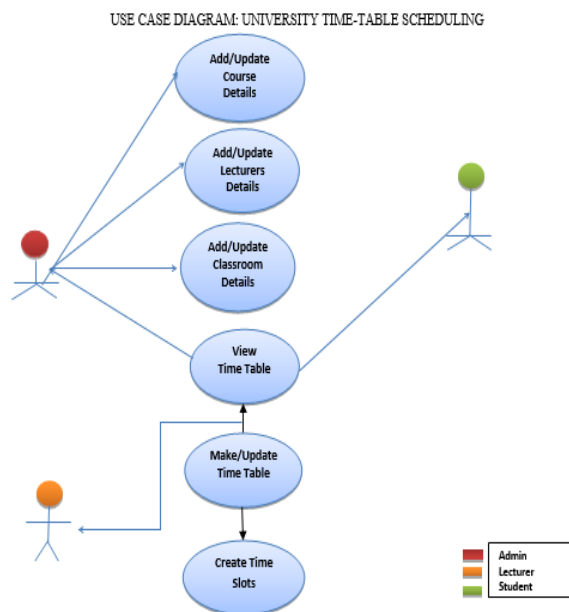


Figure 11. Use Case Diagram

### 2.4.9.2. Menu Design

In the design of the menu, all items are linked to each other; one can connect to any other page from any page. But on the Timetable page you can only log out. You can't go back to adding courses again.

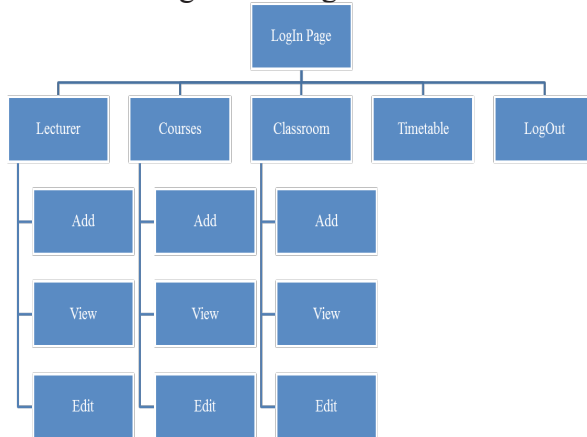


Figure 12. Menu Design

### 2.4.9.3. Test Strategy

I will use user feedback, valid data, invalid data, special cases/unusual values and boundary value testing to reveal the existence of errors.

I plan to test the user interface first.

- I will elicit feedback on the appropriateness of the user interface from potential users:
- Is colour scheme appropriate?
- Is structure of layout appropriate?
- Can the elements that make up the user interface be read easily?
- Does the user interface offer sufficient guidance on its use for the given task?

I will test the effectiveness of the security on the user interface:

- Valid password gains access to the system
- Invalid password rejected and system access denied

### Functional Testing

- I will write test cases for the menu and toolbar user interface to test that:
- The menu and toolbar options are displayed correctly
- All options on the menu and toolbar can be selected
- Invalid selections are dimmed or rejected with an error message

### Logical Testing

I will write test cases that will:

- Check that the correct input screen is produced for each option
- Check that the data can be entered as expected
- Check that invalid data is rejected/ check that valid data is accepted
- Check that when output from the system is expected that output is produced
- Check that an appropriate error message is displayed when the user attempts an invalid operation or tries to enter invalid data

I will also write test cases to test that:

- Details for each lecturer, course and classroom can be entered and then stored
- It is possible to select for viewing/editing/updating the details of a course, lecturer and classroom
- It is possible to delete a lecturer, course or a classroom from the database
- Department, faculties and lecturers are automatically generated in the drop down box after they are being added.
- Timetables can be printed and viewed
- Constraints for a lecturer is being checked
- Constraints for classrooms are observed
- Constraints for courses and their sections are being observed.

## **System Testing**

When the testing is complete, a whole range of tests is conducted to ensure that the whole system is working properly.

## **Recovery Testing**

Here we would reboot the computer while the database is open; this is to check that in case of a power failure the data would not be lost.

## **Acceptance Testing**

In this case, the user is brought in and asked to check the entire program to make sure all the functionalities is present and working properly as planned. This kind of testing might result to further changes depending on the user.

### **2.4.9.4. Security**

The whole program is properly secured and the Database could be secured by adding a password in the SQL program.

## **3. CONCLUSION**

The teaching staffs usually spend a lot of time in timetable generation and timetable management. The aim of this project is to enable the process of timetable generation to be done faster and more efficient. There are so many ways the program could be developed to become a better application to be used in the university.

### **Conflict of interests**

Author declare no conflict of interest.

## **REFERENCE**

- [1] Bouvier. Versions and standards of HTML *ACM SIGAPP Applied Computing Review*, 3:9-15, 1995.
- [2] F. Sueychyun, University Course Scheduling System (UCSS): A UML Application with Database and Visual Programming, *Journal of Computing Sciences in Colleges*, 20:160-169, 2005.
- [3] G. M. White. Constrained satisfaction not so constrained satisfaction and the timetabling problem. In PATAT'00, volume 1, Konztanz, Germany, August 2000.

