

Chapter Four:

Implementations and Testing

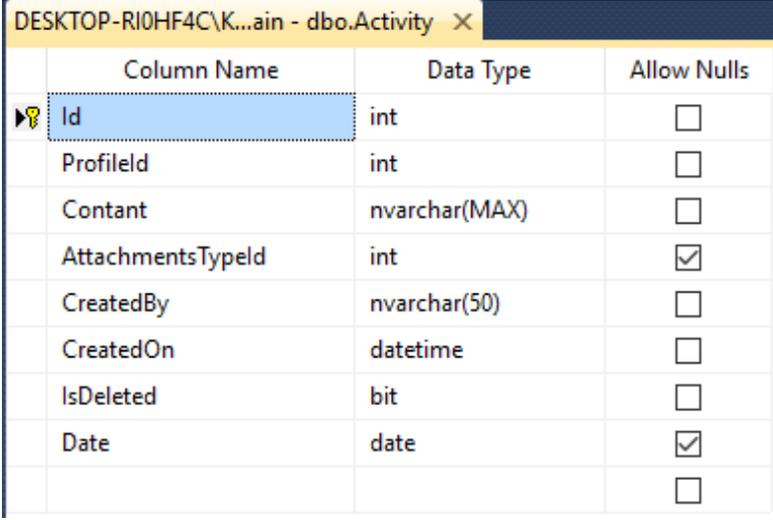
4.1 Database Mapping (Schema Diagram):



Figure 1 - Schema Diagram (Database)

The above figure shows the schema diagram of the system database.

4.2 Tables Descriptions:



The screenshot shows a table named 'Activity' in a database. The table has the following columns and properties:

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
ProfileId	int	<input type="checkbox"/>
Content	nvarchar(MAX)	<input type="checkbox"/>
AttachmentTypeId	int	<input checked="" type="checkbox"/>
CreatedBy	nvarchar(50)	<input type="checkbox"/>
CreatedOn	datetime	<input type="checkbox"/>
IsDeleted	bit	<input type="checkbox"/>
Date	date	<input checked="" type="checkbox"/>

Figure 2- Activity table

Activity table which identifies the information of activity: : Id of activity , Profile Id(the person who created the activity) , Content (The description of activity) , AttachmentTypeId(foreign key that refers to the type of attachment image , video or text) , CreatedBy(indicates who created the activity) , CreatedOn(indicates who created the activity), Is Deleted(Identifies if the activity is deleted where the deletion is logically) , Date(refers to the date that the activity ad happened on

4.3 Architecture/technology Description:

In the beginning our architecture of the system depends on different tiers.

At first, we have the website project where it consists of three main layers as the following:

- Controllers.
- Manger.
- Repository.

In addition to that the website is connected to the SQL database where only the repository can communicate with the DB.

A Blob storage where the controller can access it using a helper.

Moreover, we have a web API that provide services for mobile application through providing a communication to the database of the system.

Our mobile applications are connected to the web API through http protocol where there are no sessions on the other hand it's just a onetime request.

The figure below provides more explanation about our system architecture:

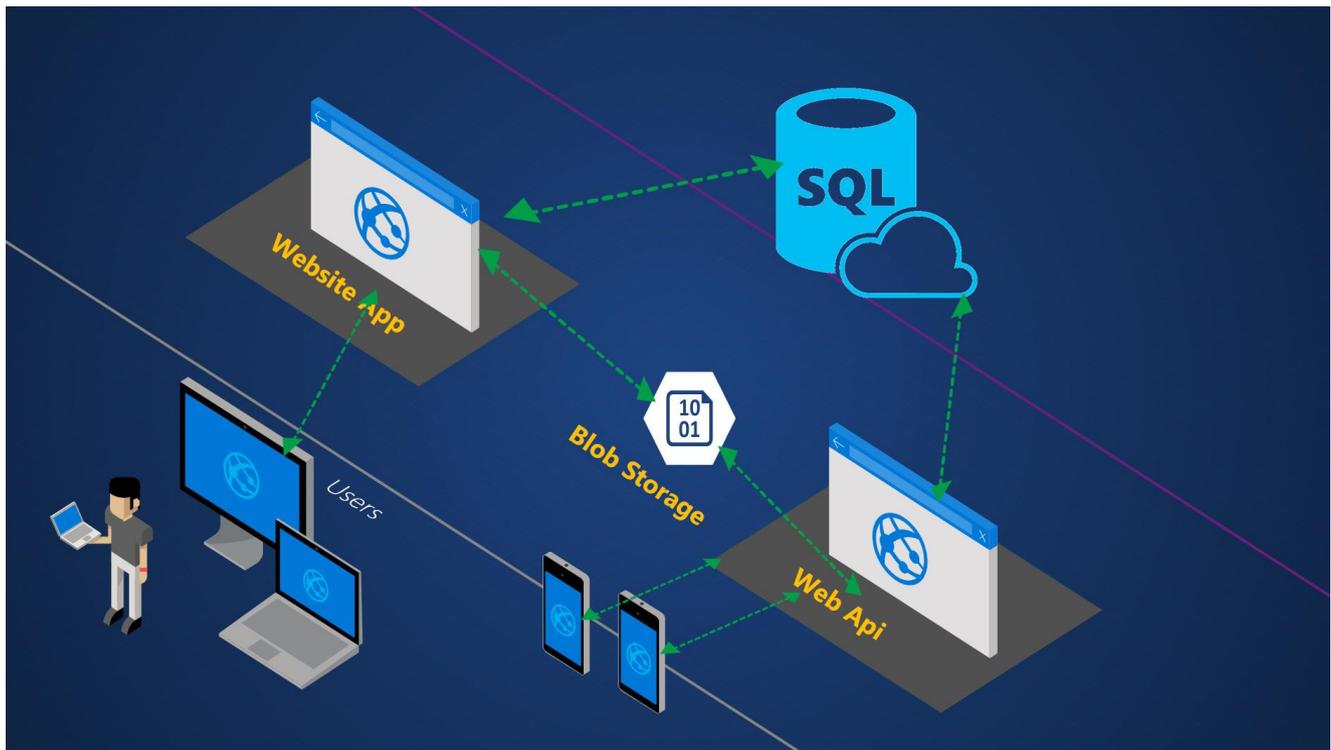


Figure 3 - System Architecture

As for the technologies that are used in in our project are described as the following:

At first we have MVC framework which stands for Model View Controller, where the view contains all the tags that are related to interface such as HTML , Bootstrap . In addition to all the logic needed to arrange the data displayed to the user.

The model consists of all the data that we want to display to the user where we don't have to return all the elements to the user, therefore it gives the ability to control all what we want to display.

Finally, we have the controller which it connects the view with the model and all the algorithms are used in it.

In Conclusion, we have used the MVC framework because it gives us the ability to work concurrently without decencies.

Secondly, we used Ajax requests that give the ability to post data silently through using JSON objects or calling a controller from the view.

We have used Ajax requests in order to increase user experience.

Xamrine android is one of the best technologies, where using C# language we were able to build an android application in addition to connecting it to web API service.

WUP which stands for Universal windows platform, where we have the ability to implement our application once and it works on different platforms such as windows 10, windows phone & windows tablet.

OAuth is one of the technologies that provide us the ability to manage the authorization and authentication in our project, where we used it in order to authorize all the users that sign in. In addition, to authenticate them through providing different roles depending on the user role.

Blob storage technology provided us with usability to manage all the media that is related to our project, where it take over all reading and writing files on the database. Only the URL of the media file is stored in the database. Moreover, it helps us increasing the efficiency of our system through retrieving all the needed media in a fast manner.

LINQ is one of the most valuable technologies that is used in our project. Where it stands for language Integrated Query. It provides usability in connecting with the SQL database.

4.4 Algorithm for major functions:

4.4.1 Add User Algorithm

```
01 CLASS AddUser_form
02 Form_Entry()
03  Get the user's id
04  Get the user's password
05  Get the user's type
06  Get the user's name
07    IF submitted
08    IF email-id already exists in database THEN
09  Prompt admin for another id
10  Get another id from the admin
11    ELSE
12    IF fields validated THEN
13  Add new user
14  Save the record to database
15  Call Form_Show() method
16    ELSE
17  Call Form_Entry() method
18  END IF
19  END IF
20  END IF
21      Form_Show()
22  Display the user's id
23  Display the user's password
```

- 24 Display the user's type
- 25 Display the user's name

Figure 4 - Add Activity Algorithm

4.5 Sample code for main functions:

4.5.1 Administration Index Controller

```
// GET: Administration
/// <summary>
/// Indexes this instance.
/// </summary>
/// <returns>System.Threading.Tasks.Task<ActionResult>.</returns>
0 references | Rami Qasem, 2 days ago | 3 authors, 6 changes
public async System.Threading.Tasks.Task<ActionResult> Index()
{
    try
    {
        AdminDashboredViewModel model = new AdminDashboredViewModel();
        var temp = ProfileManager.GetAllProfiles();
        //Gets Newest Users registered and saves them in the model
        model.NewestUsers = (from p in temp
                            orderby p.CreatedOn descending
                            select p).Take(10).ToList();
        //Counts the number of students & alumnis registered in the system and saves it in the model
        model.StudentsAndAlumniCount = temp.
            Where(p => p.ProfileTypeID == (int)ProfileTypes.Student).Count() +
            temp.Where(p => p.ProfileTypeID == (int)ProfileTypes.Alumni).Count();
        //Counts the number of companies registered in the system and saves it in the model
        model.CompaniesCount = temp.Where(p => p.ProfileTypeID == (int)ProfileTypes.Company).Count();
        //Counts the number of instructors registered in the system and saves it in the model
        model.instructorCount = temp.Where(p => p.ProfileTypeID == (int)ProfileTypes.Instructor).Count();
        //Counts the number of profiles that are pending for admin confirmation
        model.PendingCount = temp.Where(p => p.ProfileStatusID == (int)ProfileStatuses.Pending).Count();
        //Loop on all the pending profiles
        model.RegisterPendingUsers = new List<Profile>();
        foreach (var item in temp.Where(p => p.ProfileStatusID == (int)ProfileStatuses.Pending).Take(10).ToList())
        {
            //Checks if the profile is locked in order to exclude it from the list
            var locked = await UserManager.GetLockoutEndDateAsync(item.SecurityID);
            if (DateTime.Now >= locked)
            {
                model.RegisterPendingUsers.Add(item);
            }
        }
        return View(model);
    }
    catch (Exception ex)
    {
        //returs an empty view
        return View();
    }
}
}
```

Figure 5 - Administration Index Controller

4.6 Functional test Cases:

4.6.1 Login Function

Test Case ID	Inputs	Expected Output	Actual output	Fail/Pass
1	Email="" ; Password=""	Error Message	Error Message	Pass
2	Email="Rami";Password="123"	Error Message	Error Message	Pass
3	Email="Rami@Gmail.com";Password="123456"	Logged In	Logged In	Pass
4	Email="Rami.....(256Char)" ; Password="123456"	Message Error	Message Error	Pass
5	Email="Rami@Gmail.com";Password="4001Char"	Message Error	Message Error	Pass

Table 1 – Login Function

4.6.2 Register Function

Test Case ID	Inputs	Expected Output	Actual output	Fail/Pass
1	UniID="";AccountType="Student"; FullName="";Email="";Password=""; ConfirmPassword=""	Error Message	Error Message	Pass
2	UniID="201310626";AccountType="Student"; FullName="Rami";Email="Rami";Password="1234"; ConfirmPassword="1234"	Error Message	Error Message	Pass
3	UniID="";AccountType="Company"; FullName="Entermarkets"; Email="EnterMarket@Entermarkets.com"; Password="P@ssw0rd2016"; ConfirmPassword="P@ssw0rd"	Register Success	Register Success	Pass
4	UniID="";AccountType="Alumni"; FullName="Mahmoud...(101Char)"; Email="Mahmoud@hot...(257CHAR)"; Password="char(4001)"; ConfirmPassword="char(4001)"	Message Error	Message Error	Pass
5	UniID="";AccountType="Alumni"; FullName="Mustafa AlSnaid";Email="Mustatfa....Char(256)"; Password="Char()";ConfirmPassword="Char(4000)"	Register Success	Register Success	Pass

Table 2 – Register Function