**Criterion C-Development**

The deadline manager is divided into 3 Roles (Admin, Teacher and students). Client adds the users to application and keeps deadline for submission, the client can add or delete project guidelines, delete the deadline, and generate the reports. Teachers send supporting documents for assessments.

To develop the application based on client’s specifications the following techniques were used:

-Database tables

-User Login

-User interface

-Database connectivity with C#[[1]](#footnote-1)

-Encapsulation[[2]](#footnote-2)

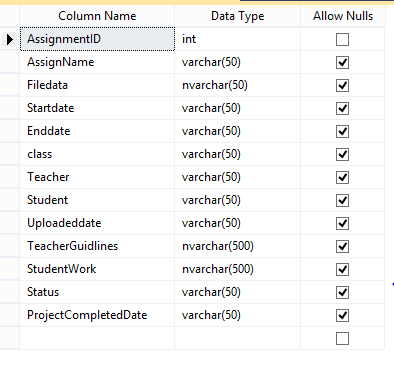
-Polymorphism[[3]](#footnote-3)

-Conditional Statements

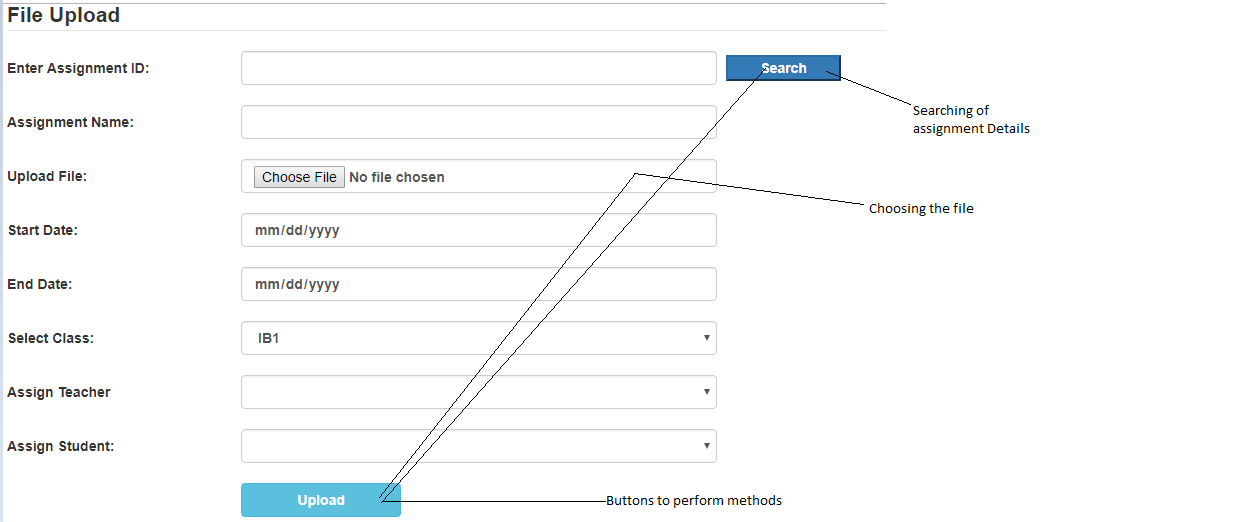
-GUI tabs and Popup menus.

-Saving to a file.

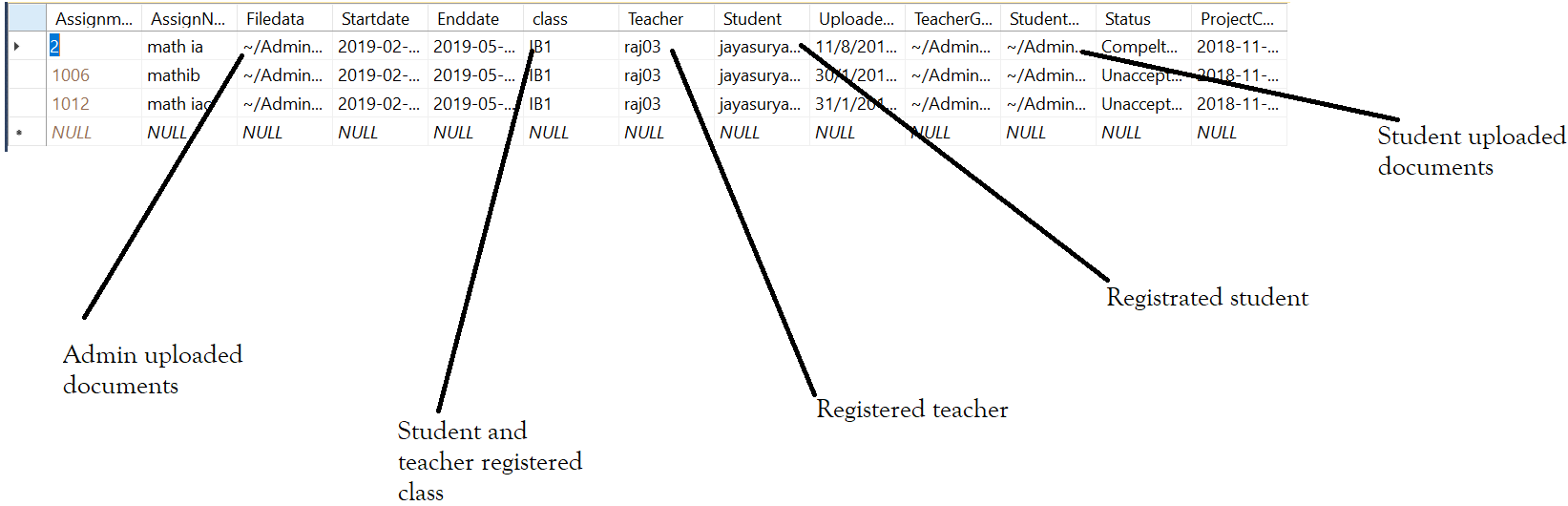
**Database tables**



This facilitates the storage of assigning specific teacher, student, class for creating and assignment submission and dates. Client uploads the data in front end and data will be stored in the database.



**Use of database tables**



Admin uploaded documents and assigns to teacher and students. Students uploaded completed documents and teachers upload their guidelines to the students. [[4]](#footnote-4)

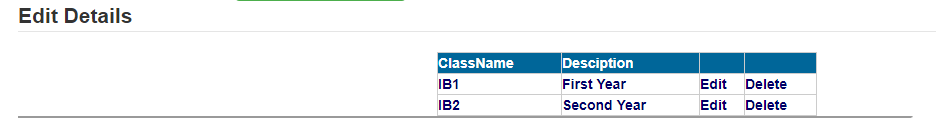


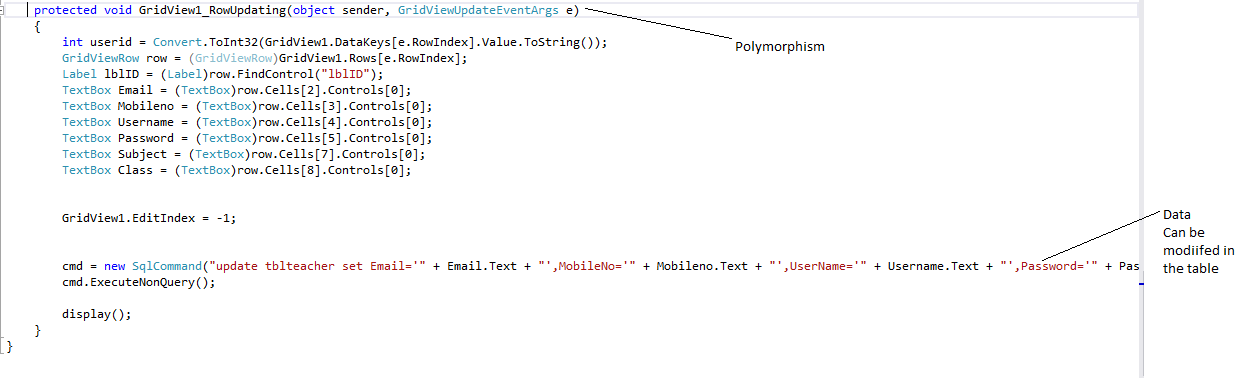
**Encapsulation**

Encapsulation utilized for data hiding. This is also supporting abstraction as it removes complexity, which provides a client an easier approach to the problem. I had used encapsulation to prevent unauthorized access and to protect the data from accidental alterations.



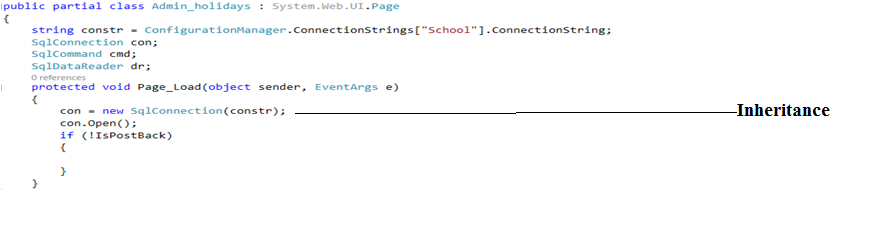
**Polymorphism**





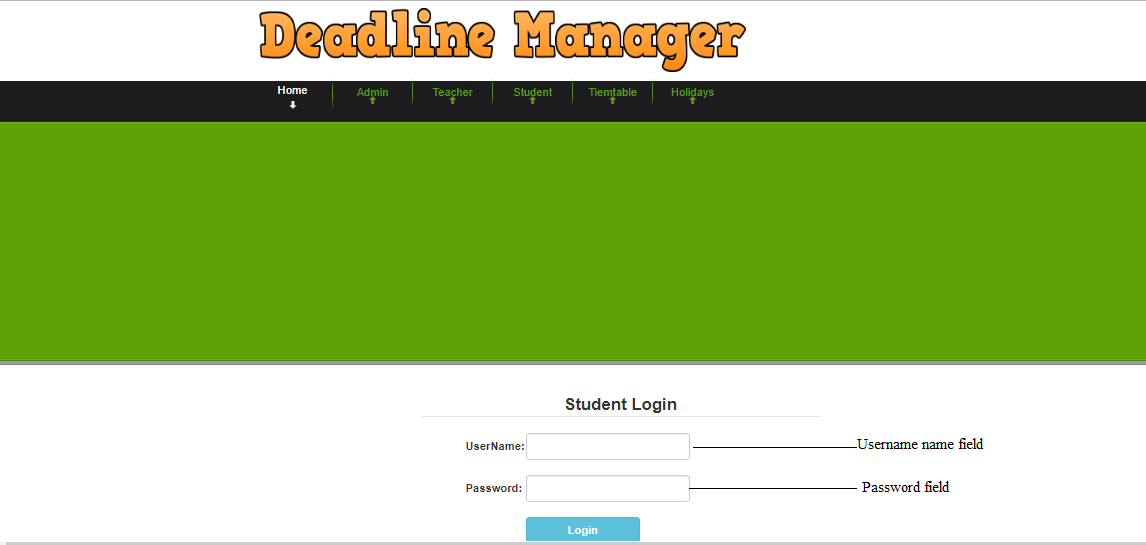
Polymorphism helps the client to modify or edit the details in the application for edit the details and update the details of specific record. Polymorphism improves performance and work efficiency of the program in comparison to using various methods different respective names.

**Inheritance**

****

I have thought abstractly when planning how partial class would be represented, I have decided to use Ms-Sqlserver the data can be connected from front end to backend.

**User Login**



In the application this login page for student, in this page the students enter their user id and password if username and password was correct the view their deadlines and submit their project. Teacher and admin have same procedure for login

The following code below shows how the function will be called, and condition will be checked.

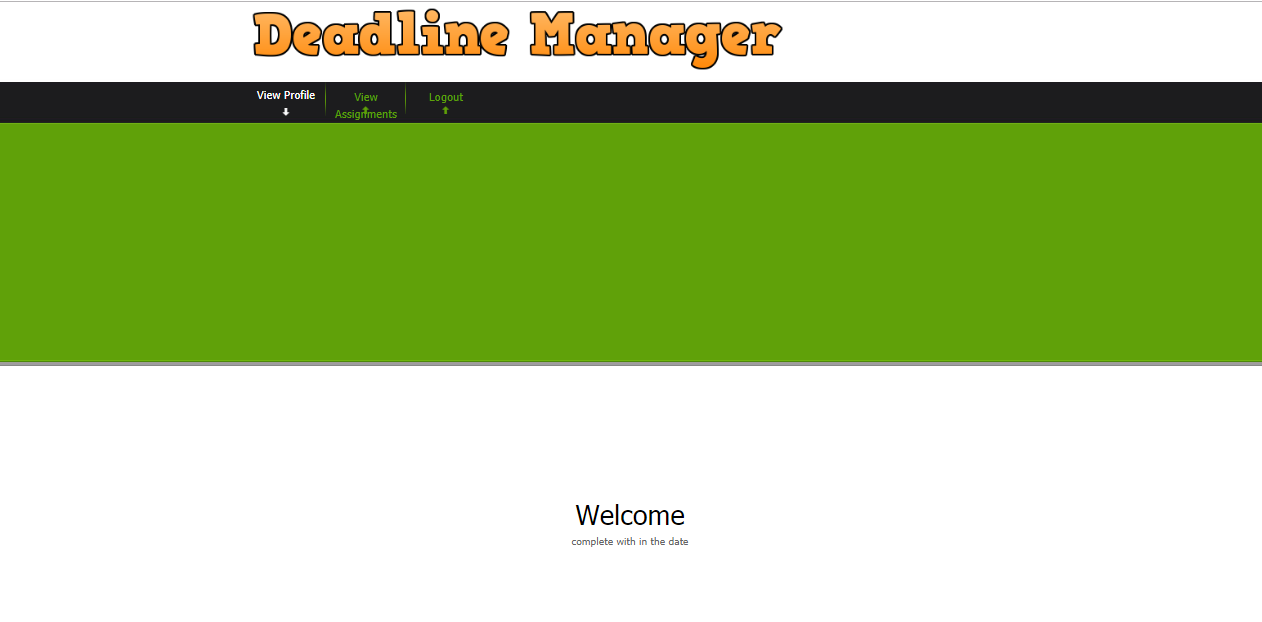


Once function executes the following sqlquery will trigger and return the result.



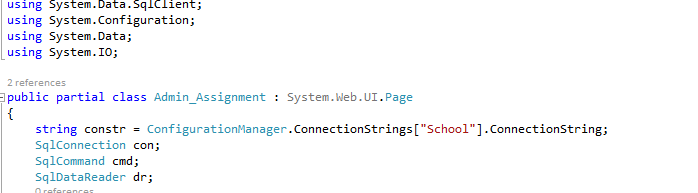


If wrong credentials were entered, they can’t access the application, this ensures proper validations for all forms.

**

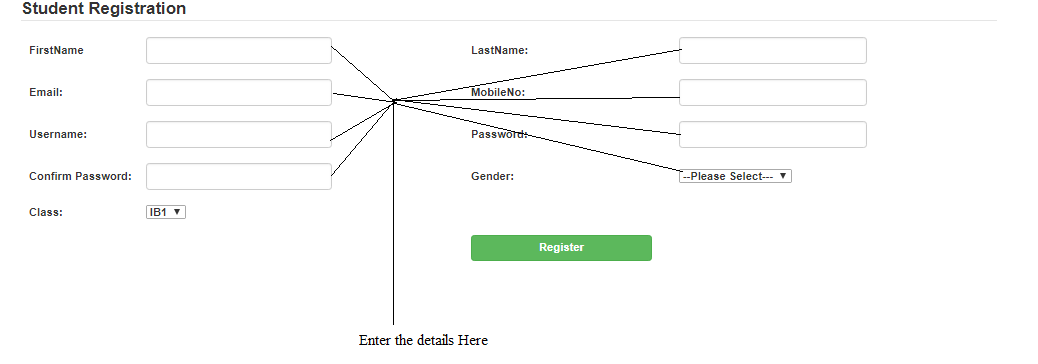
If exact username and password found within database, the message will be displayed and redirect to the main menu.

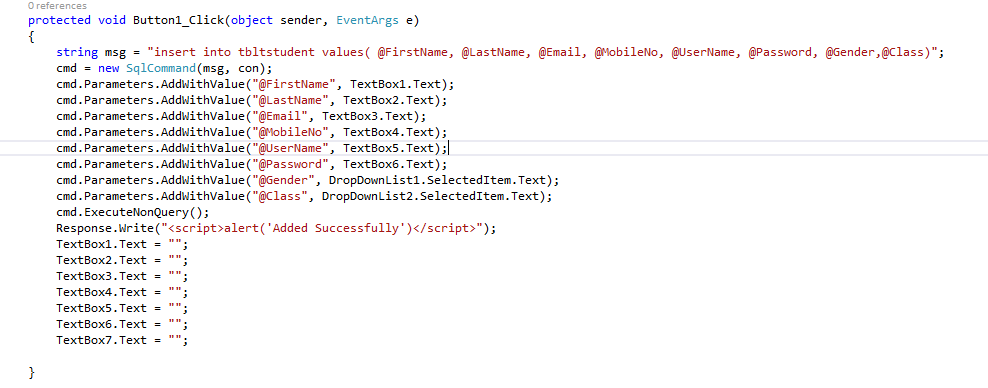
**Database connectivity with C#**

****

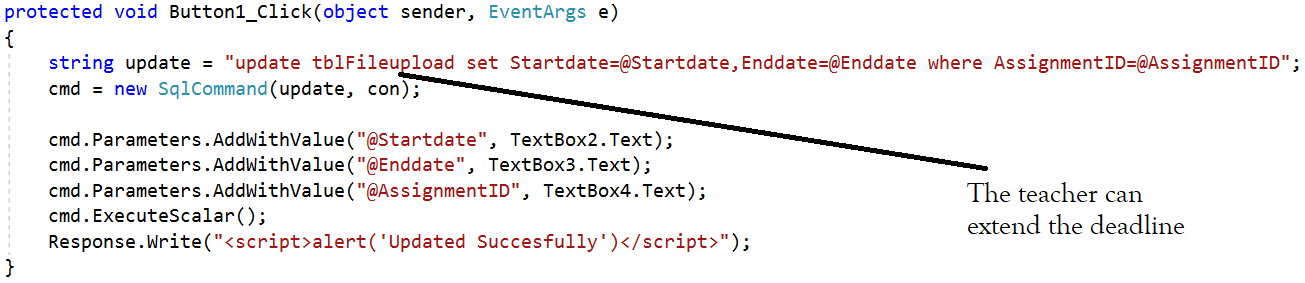
This is used for connection; in this we have to include a header file (Using System. Data, Using System.Data.SqlClient, Using System. Configuration), this makes the connection if user has enter the details and the values what the client has enter the data will be saved in to database.

Once the details had been entered in the textboxes and Admin click on Register Button the values has been inserted in to database.

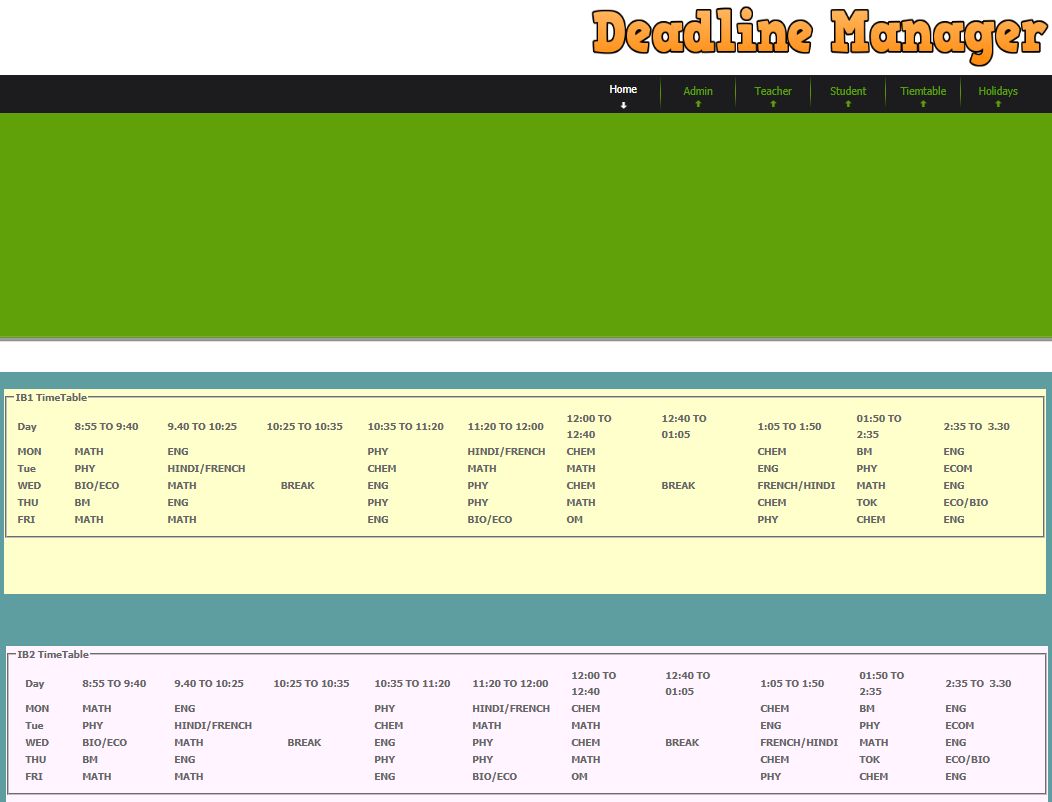
[[5]](#footnote-5)



The teacher can extend the deadlines



All the users can view the timetable in the home page



**Conditional Statements**

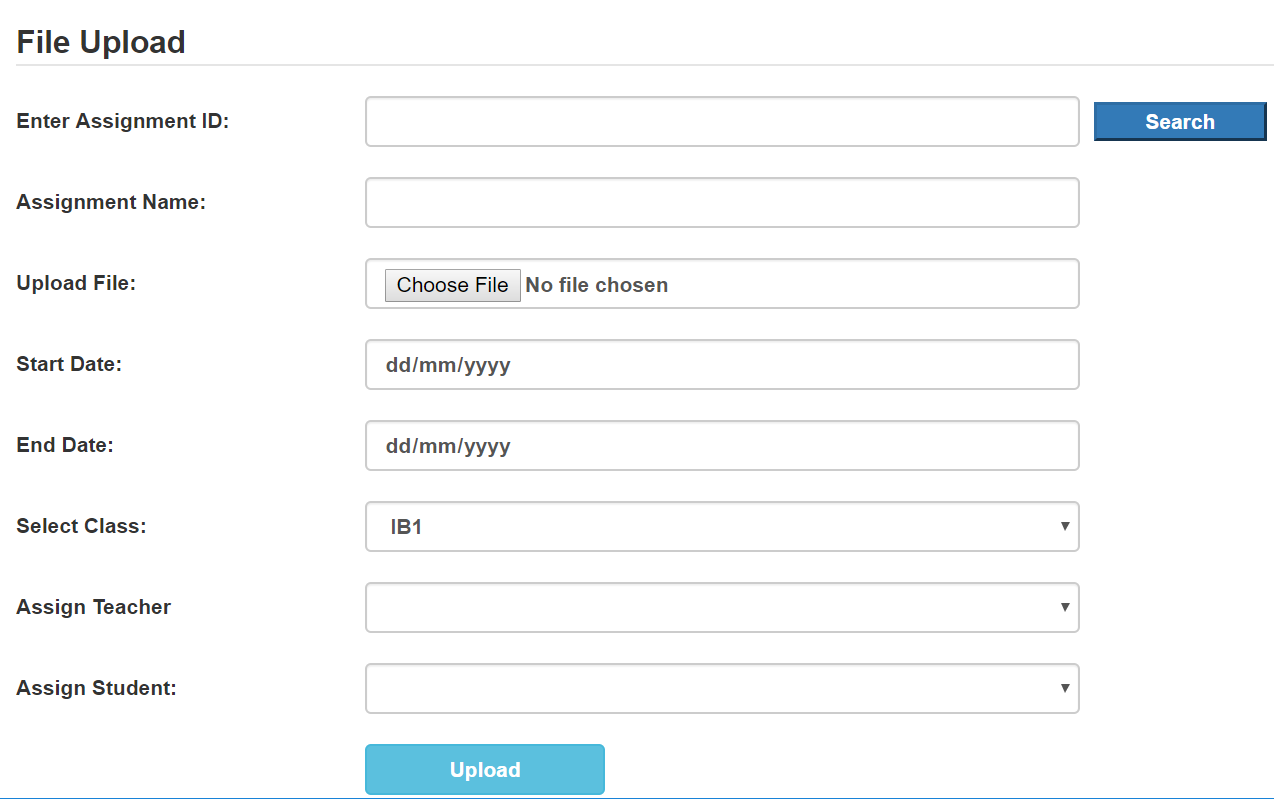
In this page” if” condition is implemented to check the user has registered in the application or not

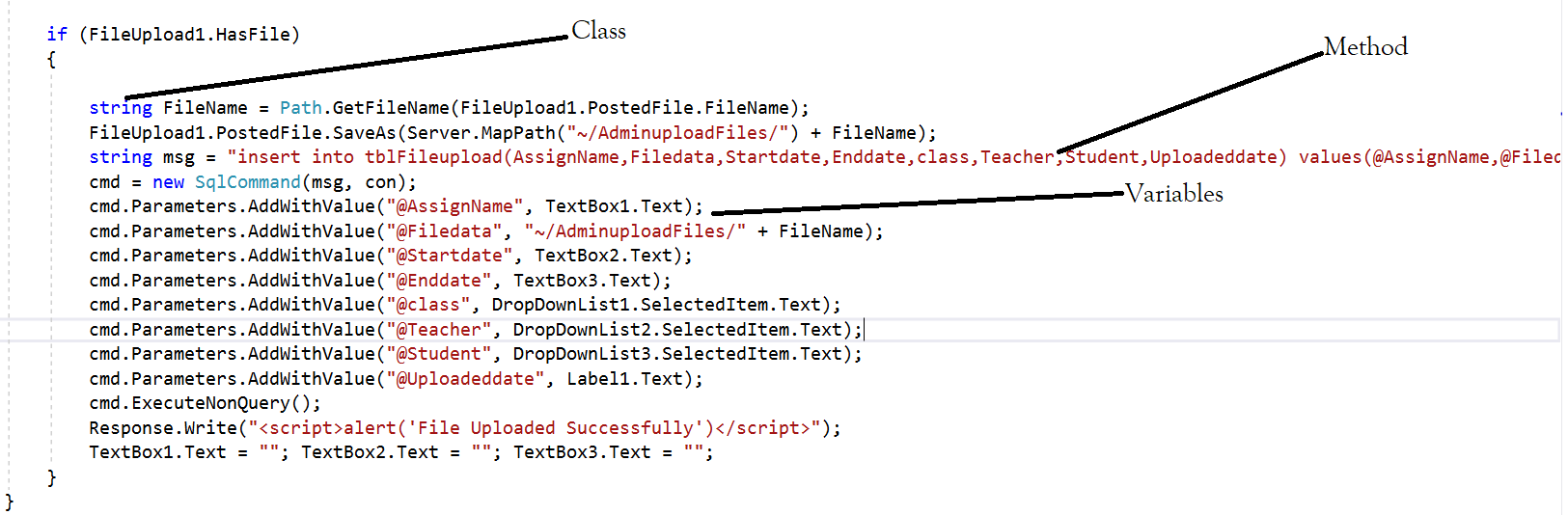
The condition states that if the rows in the table has the values then it checks the user entered details if row has found and the user credentials are correct, then redirects their functional page otherwise the else statement will be returned.



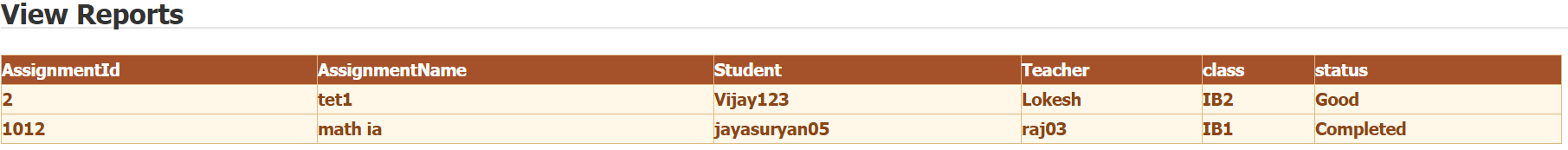
**Deadline**

The client can create an assignment and the admin, or the teacher can the upload the supporting documents. It allows the allocation of students and teachers for the client to assign the work. Using strings and classes the client would be able to give the file name insert the data (from the string) into the database. By using the method, the data can be stored in the backend. The variables store the data entered in the textbox.





**Report generation**





The data is retrieved from the backend technology.

**User interface**

While designing the forms of this project, different graphical entities/tools have been used. For instance,

-Labels are used for recognisation of the field like (First name, Last name), etc.

-Text Fields to input data from keyboard.

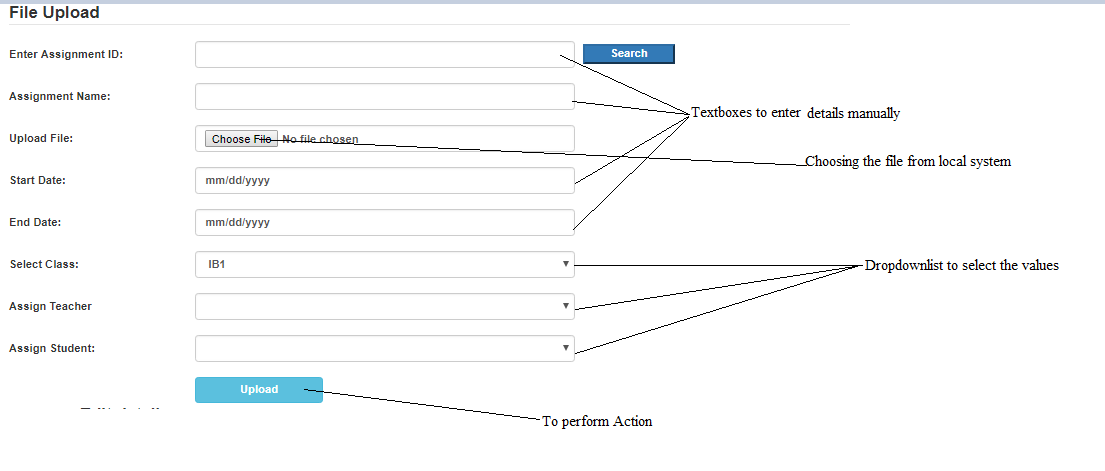
-Dropdown list is used for selecting the value (these are used in Student registration for -Select the class and Gender and same used for Teacher registration, and used in adding --Assignment Page to select the Class, Teacher, Student)

-Buttons are used for performing action.

-File Upload are used to upload the files

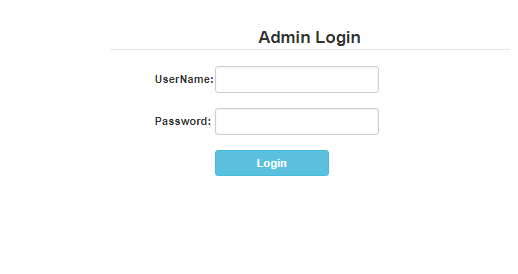
In this page the admin enters the project details like project name, Upload file, start date and end date

And select the class and assign the teacher and student for the assignment and click on Upload button the fields what’s admin has enter the date will be stored in the database.



**GUI tabs and Popup menus.**

In this page the admin enters the Username and password if username and password had wrong the popup window be raised.





**Saving to a file.**

In this application admin upload the files will be stored in the local folder and path will be stored and this path inserted into the database. When the Teacher, Student click for downloading the file will be downloaded from the folder.

****

**Program Developing and Prototyping**

By using OOP while developing the application. It improved the overall usability and extensibility of the application. Method e.g., to include information which could be called during multiple occasions and restrictions were removed.

I realized its effectiveness by approaching via OOP using multiple classes that made my program modular which made it easier for me and for the client to maintain the application which increases the extensibility. Additionally, due to the size and number of the OOP classes had made the programming to be considerably simpler and enabled many complex methods to be utilized in different scenarios.

Since I had organized most aspects of my solution in order to address success criteria. This made my development to be smooth and enabled my approach towards prototyping my solution. Since I made use of modular programming the solution was broken down the solution into 3 parts, I developed the prototype in 3 main stages. I utilized high fidelity and evolutionary prototyping in which the client could understand the features of the application and any feedback given could be addressed with improvising the prototype that integrated into the entire program. I used spiral development to prevent the waterfall to assure that the client works with me during the development of the project and that program could developed with successive iterations.

**Bibliography**

1. Freeman, Adam, et al. Pro ASP.NET 4.5 in C#. 7th ed., Apress, 2013.
2. Gosselin, Don. ASP .NET Programming with C# & SQL Server. Cengage Learning, 2009.
3. Khan, Mudassar Ahmed. "Upload and Download Files from SQL Server Database in ASP.Net." ASPSnippets, 19 June 2013, www.aspsnippets.com/Articles/Upload-and-Download-files-from-SQL-Server-Database-in-ASPNet.aspx. Accessed 22 Sept. 2018.

**Word Count**: 1000 words

1. "16.1 JDBC | Java Database Connectivity Theory Tutorial." YouTube, Telusko, 29 Mar. 2016, www.youtube.com/watch?v=y\_YxwyYRJek. Accessed 14 Sept. 2018. [↑](#footnote-ref-1)
2. "Encapsulation In C#." C# Corner - Community of Software and Data Developers, www.c-sharpcorner.com/article/encapsulation-in-C-Sharp/. Accessed 14 Sept. 2018. [↑](#footnote-ref-2)
3. Wagner, Bill. "Object-Oriented Programming (C#)." Technical Documentation, API, and Code Examples | Microsoft Docs, Microsoft, 20 July 2015, docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/object-oriented-programming. Accessed 12 Sept. 2018. [↑](#footnote-ref-3)
4. Please refer to appendix C for the code [↑](#footnote-ref-4)
5. Please refer to appendix C for the code [↑](#footnote-ref-5)