***DATABASE PROJECT REPORT***

***University Management System***

***Members of Group “T.Y.A”:***

***Yousuf Moiz Ali ya03482***

***Muhammad Talha Shahab ms03495***

***Arsalan Ahmed aa03980***

**INTRODUCTION**

We have chosen to implement a university management system for our database project. Database is a storehouse of information and treated as a unit for information retrieval purpose. Database software is a set of one or more programs that enables users to create, maintain, organize, and retrieve data from database. It is widely used today in organization to maintain employee data, customer data, accounting data etc.

Universities are very large and complex organizational structures with many employees and many different sorts of operations to be executed. Our university management system will be used to create, manage and perform all the activities related to the database of a given university. The database consists of information about the university and all of its associated people, including the students, staff and faculty. Starting from registration of a new student in the college, it maintains all the details regarding the attendance and marks of the students. For faculty, we also maintain important details like the courses being taught, the status of the faculty and the salary being offered. Relevant details for all the people associated with the university can also be update according to our needs. The aim of this project is to simplify the admission process as well as be enough for changes required in the database system.

**BUSINESS SCENARIO**

As Habib University is growing in size, we need a faster and more reliable solution for handling its needs. Each year a larger batch is coming into Habib so the administrative load will increase each year. The purpose of such a system is that all the facilities for students, faculty and staff have a common system. All the facilities will be online and there will be no need for any sort of written documents in the administrative department. It also provides some additional features, which will help in better management and also make processes easier, such as the option to add and delete events. Less human effort is required and workload is also reduced sufficiently. There are no special requirements for this system as well so it does not require any drastic changes to the system in place.

**MODULES IMPLEMENTED**

We have implemented many different functionalities in our project which were briefly discussed above. For students, we have implemented the options of showing all the university related activities of a student including his basic information, courses he is taking, his course history and their associated grades, and his financial record. On the metacurricular side, his participation in events is also shown and the option for him to register in upcoming events is also provided.

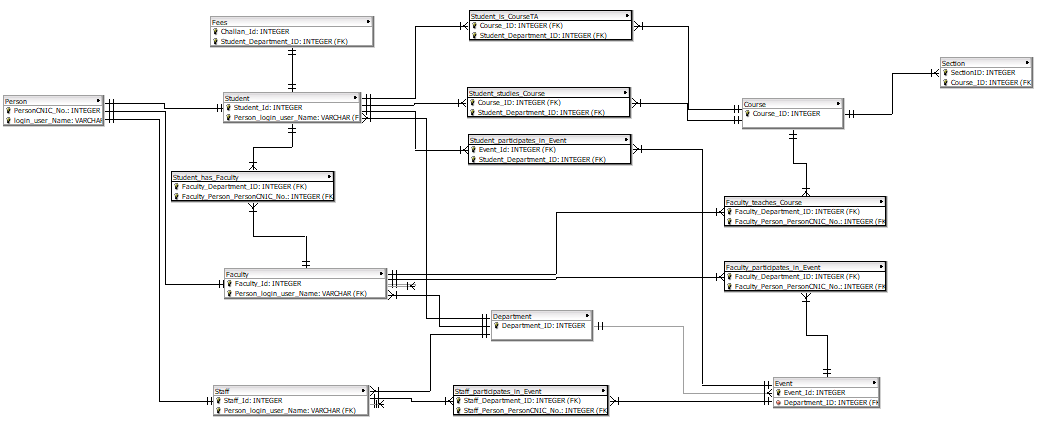
For faculty, we also have some basic information like their department name, the courses they are teaching and have taught, and their salary. Moreover, for each course they have taught, we have computed some logistics for those courses like the number of students in a particular offering and the average grade.

For staff, we have also implemented a similar scheme. Furthermore, the most important feature of our project is the option to add and remove different object with ease. For example, we can add courses, events, faculty members or students after filling out the required information in the Windows forms. We also have the facility of deleting students, faculty and events.

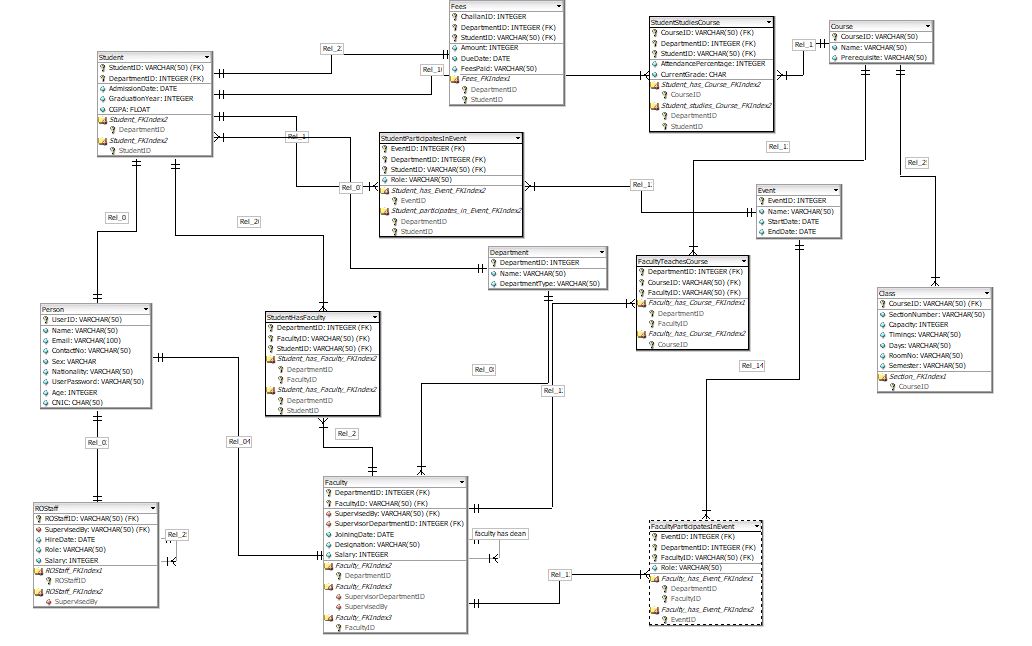
We also implemented a search so that any member in the database can have access to basic information of any other member in case he may need to schedule a meeting for example or send an urgent email. The search can be either by ID or by name and we have used DataGrid to display the data retrieved from the query in the search form.

**ERD:**

**Initial ERD:**



**Final ERD:**

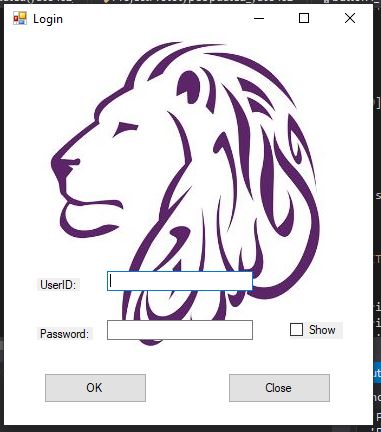


Doing this project over time, we realized that some of the changes were required in the initial ERD diagram. Many of the changes made were more technical rather than structural. For example, we changed the attributes of some types of variables so that they would take up less space and others were changed so that the types were such that they would be more understandable to the user and would parallel the existing university management system at Habib better.

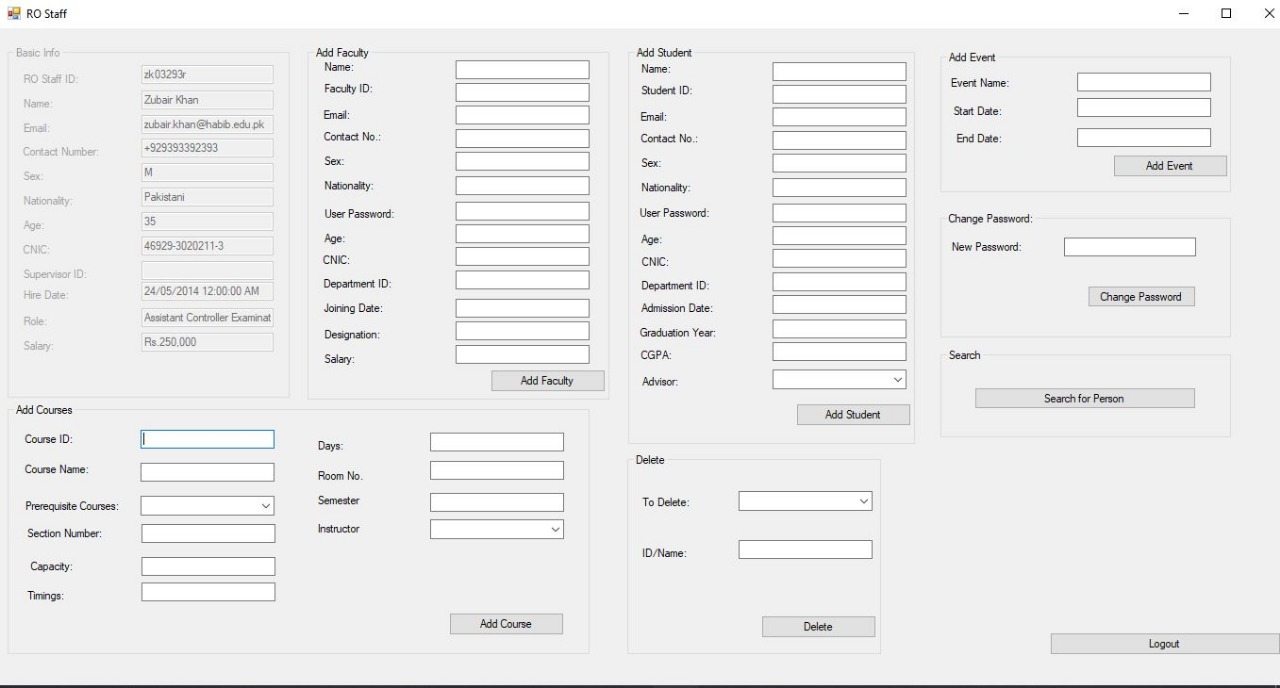
In some tables, we adjusted the auto increment property of the primary keys so for example StudentID is supposed to have the format of the initials of his first and last name plus a 5-digit number so it is not auto incremented. Initially, we had planned to incorporate all staff but due to some ambiguities in their actual roles, we reverted to just including the RO staff. Another feature to be discussed here is that the days of the Class entity will be entered so that they contain the initials of the days on which the pertinent class will be held. For example, MWR means classes will be held on Monday, Wednesday and Thursday.

**GUI**

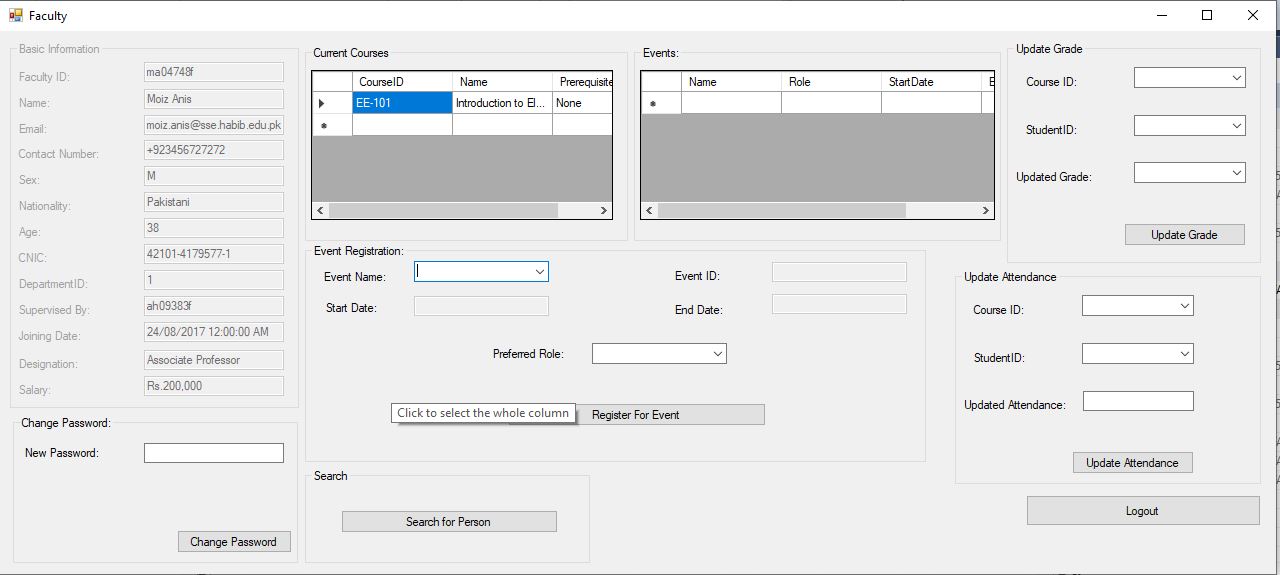
**Login**

****

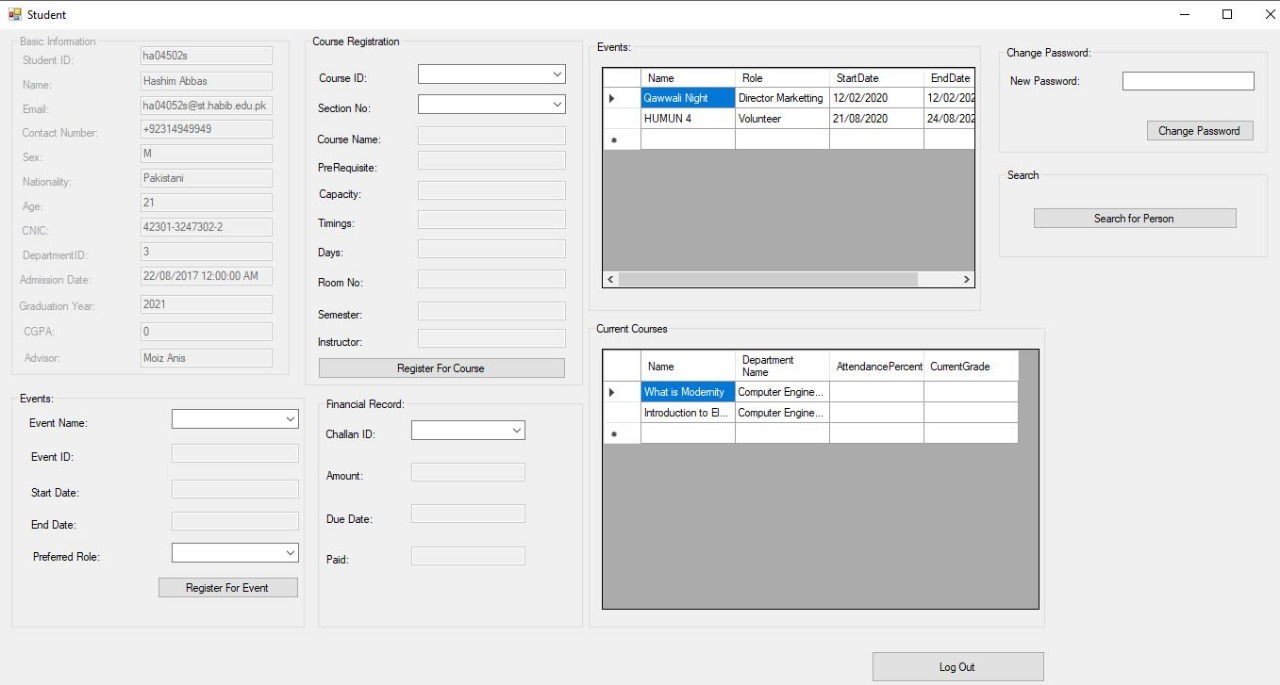
**Staff**

****

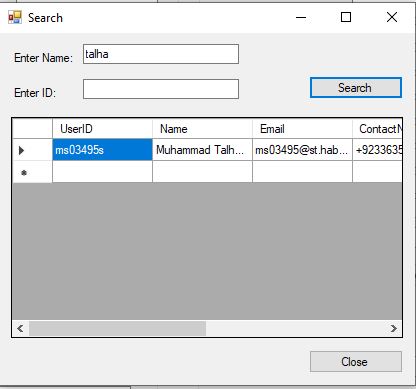
**Faculty**

****

**Student**

****

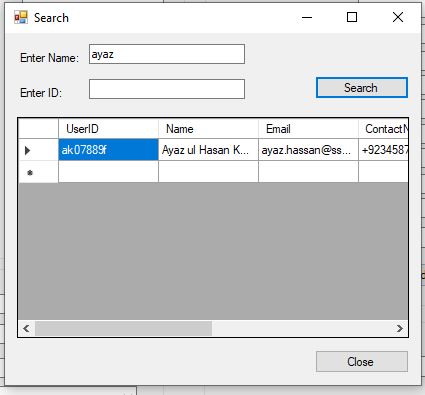
**Search**

****

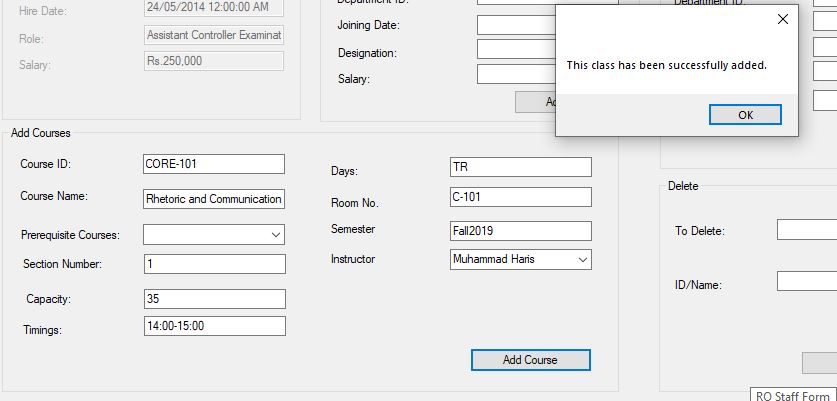
The user interface is implemented using simple Windows forms. Initially, we had created a different form for each of students, faculty and staff but after the instructor’s suggestion that the users should have to make minimal clicks and the interface should be as user-friendly as possible, we implemented a dashboard approach so that all the functionalities pertaining to a certain group are available in a single form. Thus, all the forms have its related queries linked to it on the click of a button. For example, the “Add Course” button has an insert query on the back end linked to the relevant table(s).

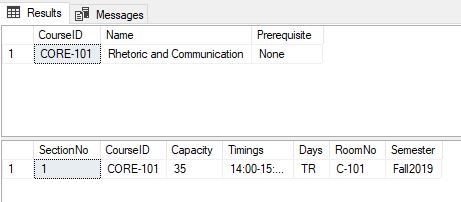
**TEST CASES CHECKED**

**Test Case 1: Searching**

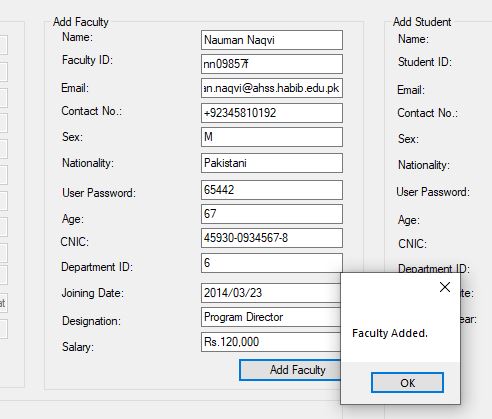
****

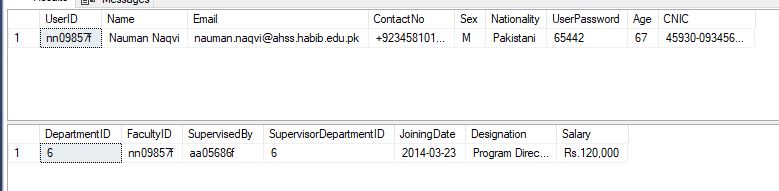
**Test Case 2: Adding Course**

****

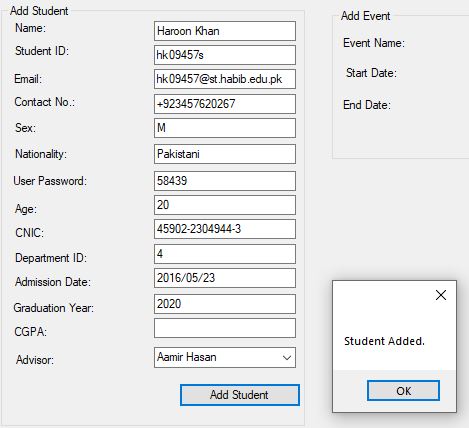
****

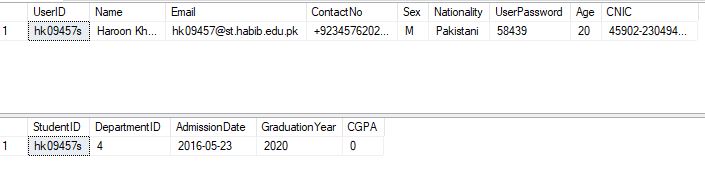
**Test Case 3: Adding Faculty**

****

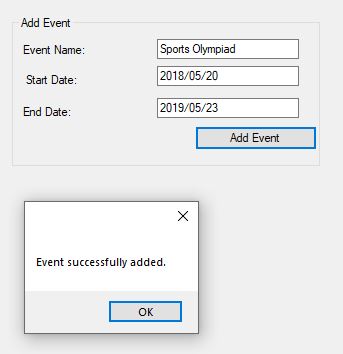
****

**Test Case 4: Adding Student**

****

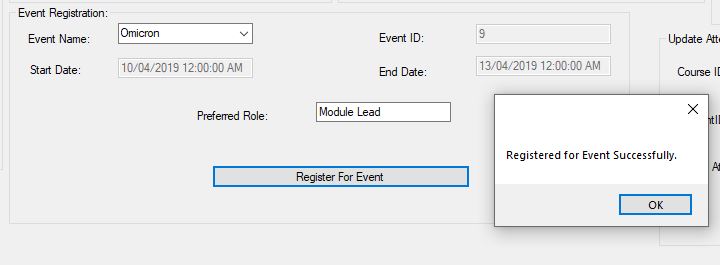
****

**Test Case 5: Add Event**

****

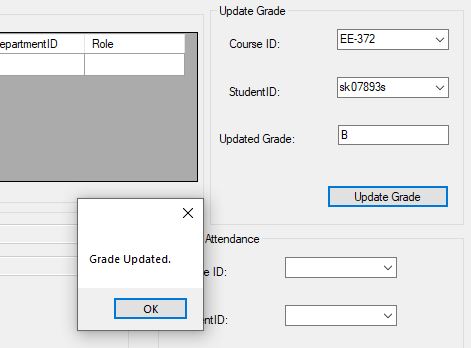
****

**Test Case 6: Event Registration**

****

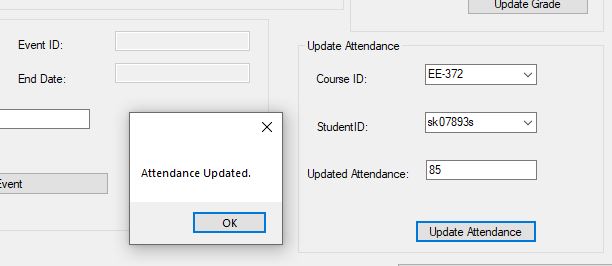
****

**Test Case 7: Updating Grades**

****

****

**Test Case 8: Updating Attendance**

****

****