# **Design**

Since the project is following waterfall methodology design is the 2nd phase of the project where the key features, structure and major deliverables are all panned out. A project design will generate many outputs which will contain different diagram such as flowcharts, class diagram, ER diagram, activity diagram that will help to proceed further to the next level

Different types of design are explained below:

## **Structural modelling**

Structural model will represent the framework for the project where all the associated components where all its components will exist. This model does not describe the dynamic behavior of the system but captures the static features. The class diagram and data flow diagram are shown below to describe this method.

### **Data flow diagram**

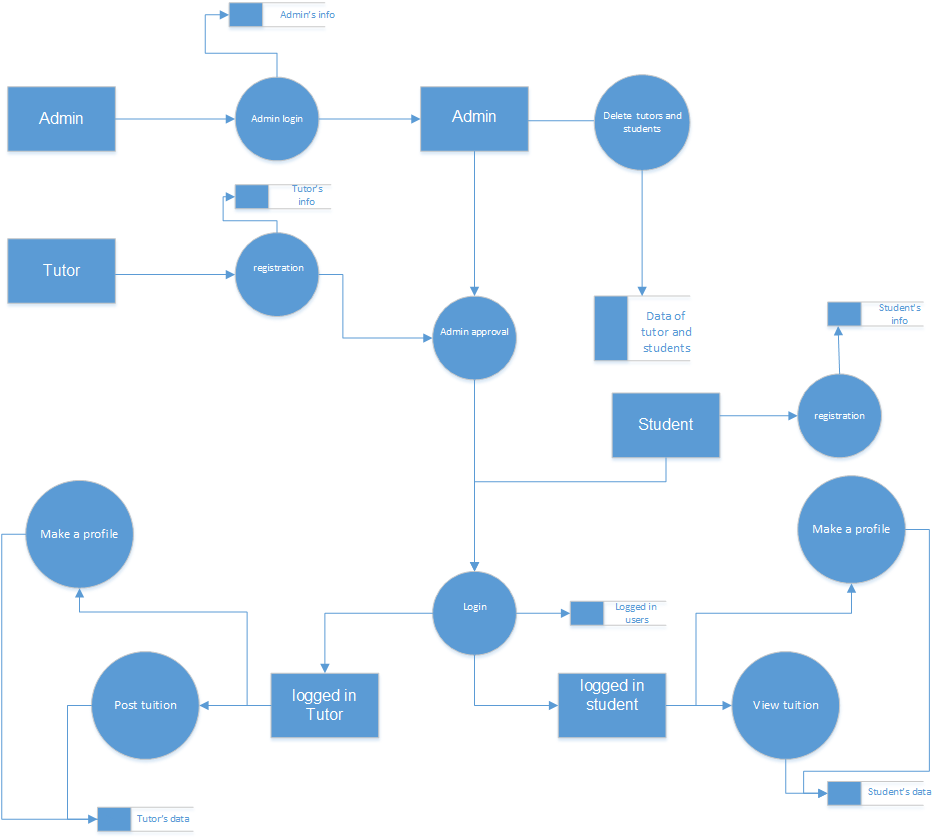


Illustration 8: DFD of tutor student portal

The above data flow diagram shows the activity and overall process of the tutor student portal. At the very beginning the admin has logged in the system and the teachers and students will register by filling up a registration form and await the admin’s approval. Once the admin has approved the registered users, they can login to the system where they will create a profile about themselves. In the profile the teachers will post their resume, qualification and what subject they are able to teach with their contact information similarly the student will also make a profile where they will post about in which subject they are in need of tuition. The profile data of both teacher and student will be stored in different database. The admin also has the ability to list, update and delete both teacher and student respectively.

## **Behavioral design**

This is the model where the communication of the system will be shown. Here the interaction among the structural diagrams will be shown. This is the model that will represent the dynamic nature of the project with the help of an activity diagram and sequence diagram respectively.

### **Class diagram**

A class diagram is a type of diagram that is created to offer a theoretical model and architecture of the system that is being built. A class diagram defines and suggests the summary and structure of the system via classes, attributes and methods with the relationship between different classes

Since MVC pattern was not used therefore the final class diagram is made on the basis of the pages of the website and the library classes that was used.

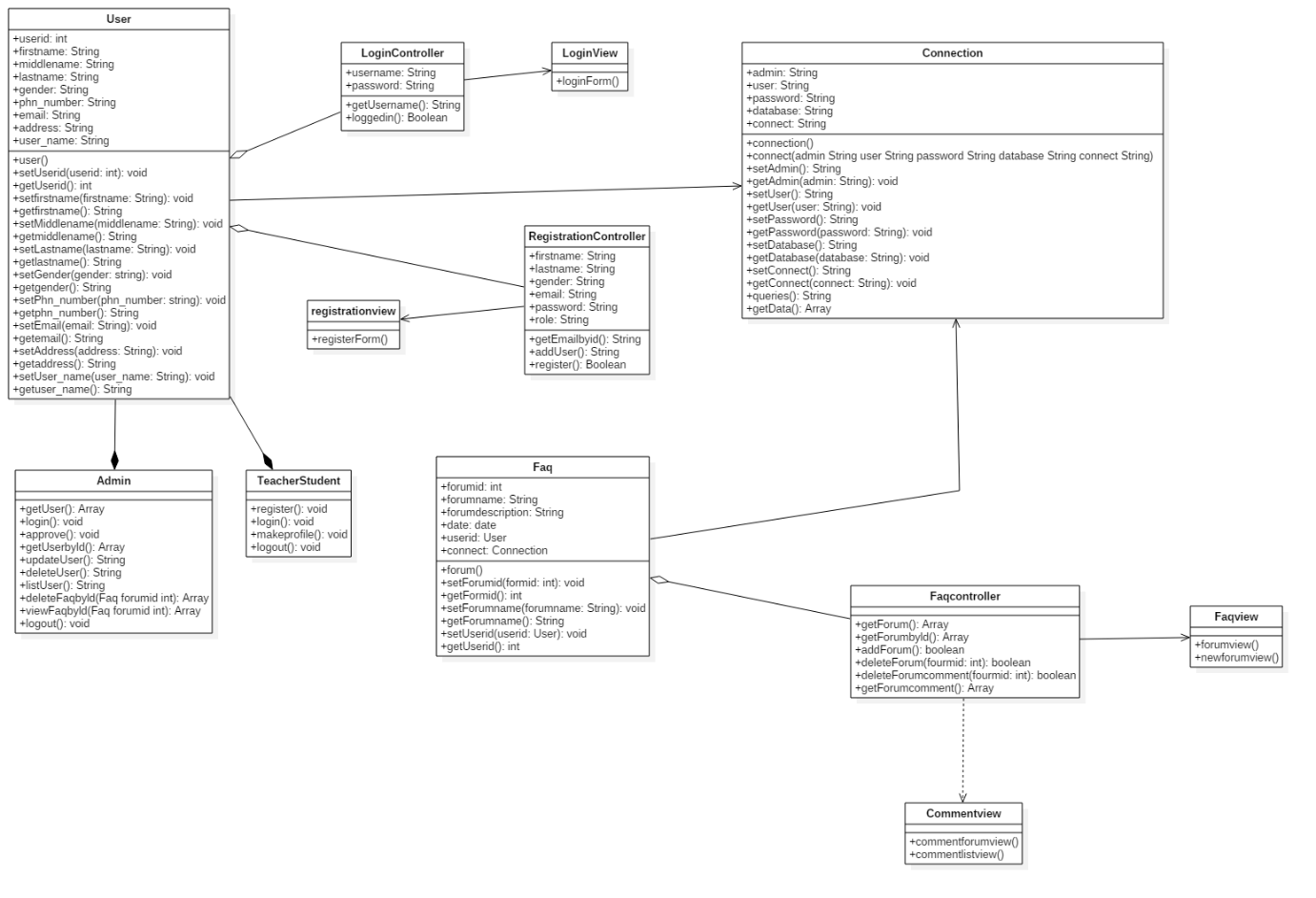


Illustration 9: class diagram of tutor student portal

### **Activity diagram**

Activity diagram is a graphical representation of executed set of procedural system activities that is used for modelling any activity's sequential work flow by concentrating on the action sequences and its initiating conditions.

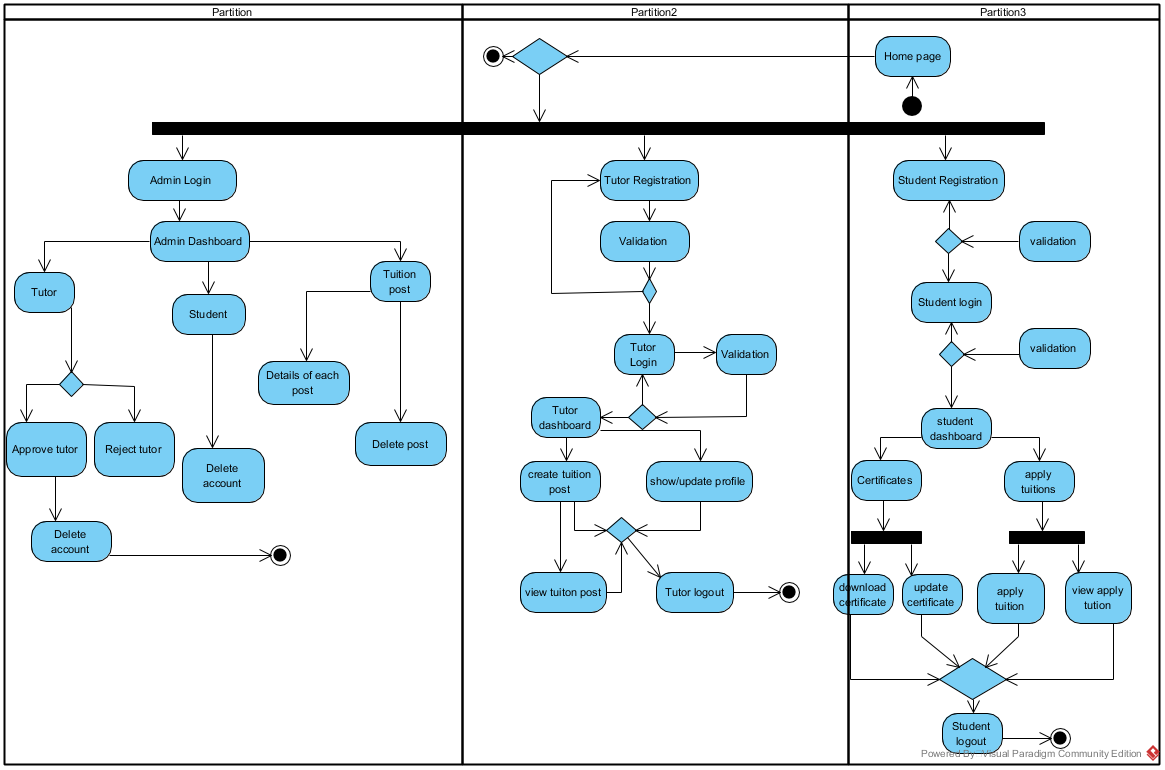
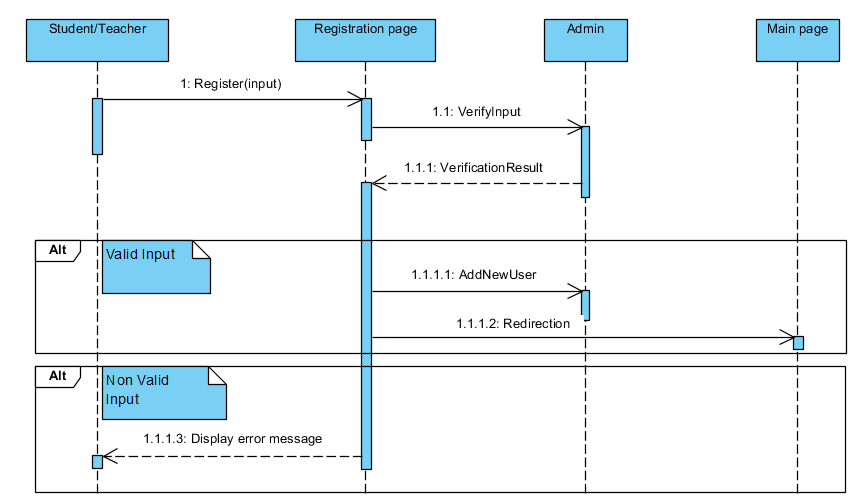


Illustration 10: activity diagram for tutor student portal

The above diagrams show the activities that takes place on the tutor student portal website. This diagram shows the flow of activity of three different entities that are involved in the website. Firstly, admin logs in to the website and then he has the authority to approve or reject tutors who have registered to the site. The admin can also delete the tutors, student and the tuition post. Admin can also view the number of students applied on each course. The tutor and student will both go through the same registration process but in order to login the tutor will have to wait for admin’s approval. After they have been logged into the website the tutor will be able to create tuition post and view them whereas the students will be able to apply to all the active tuition post and view the post on which they have applied. Students can also upload and download a certificate. Finally, after the end of using the website all three admin, tutor and student can logout of their account.

### **Sequence diagram**

A sequence diagram is an UML diagram that shows the flow of logic within the different phases of project in a visual manner and identifying the behavior within the system. This will be useful in both documenting and validating the logic of our work.



The above sequence diagram shows the registration process of my website. First the user i.e. teacher and student will fill up the registration form. The system will forward the filled up form to the admin where the admin will check the genuineness of the candidate who has filled up the form. Finally, if the admin will approve the form then the registration process will end and the user will be able to login to the system.

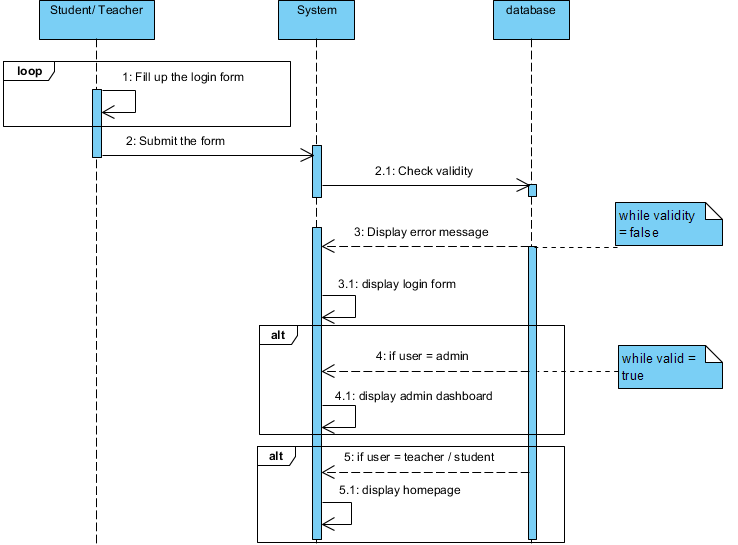


Illustration 11: sequence diagram for login process.

Now the approved user will be able to fill up the login form by entering their username and password and submit it. The system then checks the validity of the users by checking through their database and if it’s not valid the system will show and error message and the user is sent back to the login form but if the user is valid the system will distinguish them to two categories i.e. admin and tutor/student. In case of the user being admin, the admin dashboard will open and if the user is teacher or student their homepage will open respectively.

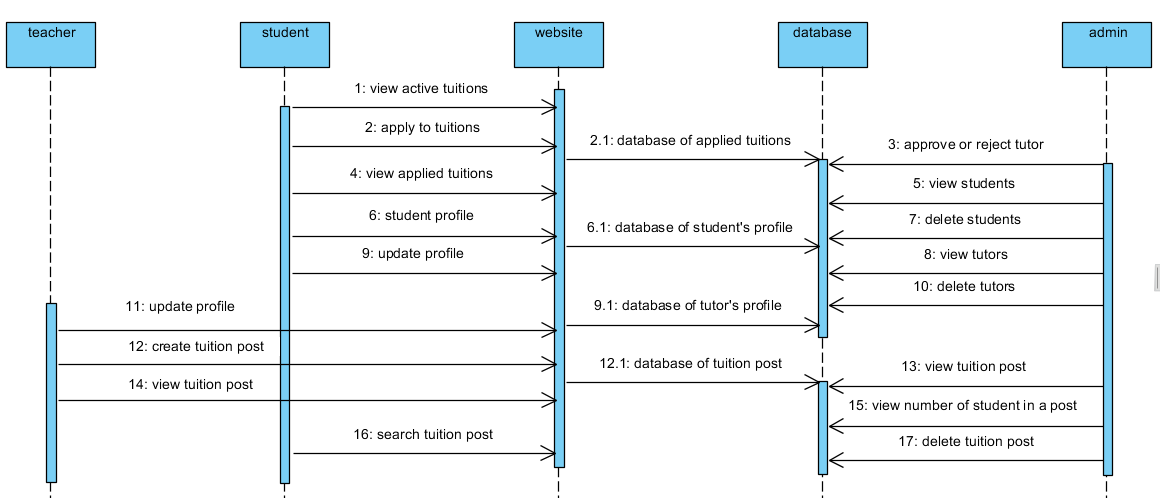


Illustration 12: sequence diagram for the main system process.

The student and the teacher will make their profiles respectively where the teacher will post their resume, qualification and what subject they are able to teach with their contact information similarly the student will post about in which subject they are in need of tuition. The profiles data will be stored in the system database from where the admin will be able to view it. The admin will also be able to update and delete the profile based on its activities. Now the student can view the teacher’s profile and vice versa from where they will be able to check qualification and needs and if satisfied can contact each other and schedule a meeting where they will come to a final decision.

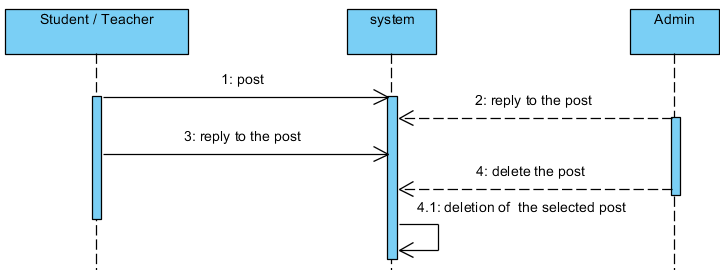


Illustration 13: sequence diagram of the FAQ page.

In the FAQ page the student, teacher and admin will be able to post and comment on the posts. This will be beneficial for any problems the users are facing and for review, feedback and remarks for how useful they are finding this website. The admin will have the power to delete the posts if found inappropriate.

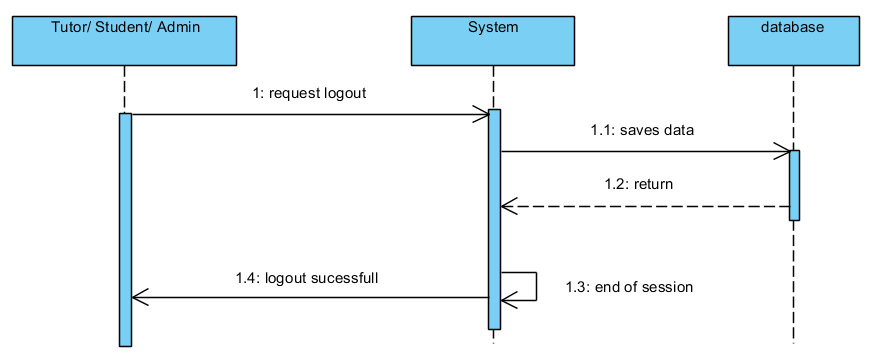


Illustration 14: sequence diagram for logout.

Here the tutor, admin and student can request logout by clicking the logout button this will trigger a process where the system will save the active data into a database and end the session by logging the user out of their profile.

## **Database design**

A database in simple language is the collection of information that is kept in an organized manner so that it can be easily accessed, managed and updated.

a database design includes logical (entity relationship) and physical (table, column and key) designs tools for data. This is shown with the help of data dictionary and entity relationship diagram.

### **Data dictionary**

A data dictionary are the set of files that will show a database’s metadata. It will contain record about the objects that have been used in the database. It is the collection of descriptions of data model for the benefit of professionals who will need to refer to them.

The data dictionary of tutor student portal has been shown below with column name, data type, PK/FK and constraint

|  |
| --- |
| **Database Tables** |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  | | | |
|  | **Name** |  | **Description** |
|  | | | |
|  | [user\_post : Entity](file:///F:\\content\\DBTable_NBmUqSaGAqESAQiK.html) |  |  |
|  | [post : Entity](file:///F:\content\DBTable_b2BEqSaGAqESAQeI.html) |  |  |
|  | [profile : Entity](file:///F:\content\DBTable_Li6PT8aGAqESAQ9B.html) |  |  |
|  | [user\_type : Entity](file:///F:\\content\\DBTable_eTdnT8aGAqESAQ23.html) |  |  |
|  | [database : Entity](file:///F:\content\DBTable_a5O7T8aGAqESAQxg.html) |  |  |
|  | [login : Entity](file:///F:\content\DBTable_YUubT8aGAqESAQwW.html) |  |  |
|  | [user : Entity](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html) |  |  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **user\_post Table** |
|  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Column Name** |  | **Data Type** |  | **PK / FK** |  | **Nullable** |  | **Description** |
|  | | | | | | | | | |
|  | [user\_id](file:///F:\content\DBTable_NBmUqSaGAqESAQiK.html#ZxmUqSaGAqESAQiM) |  | int(10) |  | PK / FK |  | No |  |  |
|  | [post\_id](file:///F:\content\DBTable_NBmUqSaGAqESAQiK.html#.pmUqSaGAqESAQiS) |  | int(11) |  | PK / FK |  | No |  |  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **post Table** |
|  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Column Name** |  | **Data Type** |  | **PK / FK** |  | **Nullable** |  | **Description** |
|  | | | | | | | | | |
|  | [post\_id](file:///F:\content\DBTable_b2BEqSaGAqESAQeI.html#npFEqSaGAqESAQeo) |  | int(11) |  | PK |  | No |  |  |
|  | [tuition\_post](file:///F:\content\DBTable_b2BEqSaGAqESAQeI.html#sRVEqSaGAqESAQew) |  | varchar(255) |  |  |  | Yes |  |  |
|  | [apply\_tuition\_post](file:///F:\content\DBTable_b2BEqSaGAqESAQeI.html#211EqSaGAqESAQez) |  | varchar(255) |  |  |  | Yes |  |  |
|  | [search\_tuition\_post](file:///F:\content\DBTable_b2BEqSaGAqESAQeI.html#8BnkqSaGAqESAQhU) |  | varchar(255) |  |  |  | Yes |  |  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **profile Table** |
|  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Column Name** |  | **Data Type** |  | **PK / FK** |  | **Nullable** |  | **Description** |
|  | | | | | | | | | |
|  | [profile\_id](file:///F:\content\DBTable_Li6PT8aGAqESAQ9B.html#4U0vT8aGAqESAQ.U) |  | int(10) |  | PK |  | No |  |  |
|  | [qualification](file:///F:\content\DBTable_Li6PT8aGAqESAQ9B.html#8NvPT8aGAqESAQ9U) |  | varchar(255) |  |  |  | Yes |  |  |
|  | [needs](file:///F:\content\DBTable_Li6PT8aGAqESAQ9B.html#85gvT8aGAqESAQ9u) |  | varchar(255) |  |  |  | Yes |  |  |
|  | [user\_id](file:///F:\content\DBTable_Li6PT8aGAqESAQ9B.html#r1mvT8aGAqESAQ_L) |  | int(10) |  | FK |  | No |  |  |
|  | [useruser\_name](file:///F:\content\DBTable_Li6PT8aGAqESAQ9B.html#k9r.r8aGAqESAQfC) |  | varchar(255) |  |  |  | No |  |  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **user\_type Table** |
|  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Column Name** |  | **Data Type** |  | **PK / FK** |  | **Nullable** |  | **Description** |
|  | | | | | | | | | |
|  | [user\_type\_id](file:///F:\content\DBTable_eTdnT8aGAqESAQ23.html#kMbnT8aGAqESAQ3J) |  | varchar(255) |  | PK |  | No |  |  |
|  | [user\_type](file:///F:\content\DBTable_eTdnT8aGAqESAQ23.html#.AHnT8aGAqESAQ3M) |  | int(10) |  |  |  | Yes |  |  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **database Table** |
|  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Column Name** |  | **Data Type** |  | **PK / FK** |  | **Nullable** |  | **Description** |
|  | | | | | | | | | |
|  | [database\_id](file:///F:\content\DBTable_a5O7T8aGAqESAQxg.html#UVR7T8aGAqESAQxq) |  | int(11) |  | PK |  | No |  |  |
|  | [database](file:///F:\content\DBTable_a5O7T8aGAqESAQxg.html#KL4HT8aGAqESAQza) |  | varchar(255) |  |  |  | No |  |  |
|  | [post\_id](file:///F:\content\DBTable_a5O7T8aGAqESAQxg.html#VIJkqSaGAqESAQhH) |  | int(11) |  | FK |  | No |  |  |
|  | [user\_id](file:///F:\content\DBTable_a5O7T8aGAqESAQxg.html#D0rUqSaGAqESAQjh) |  | int(10) |  | FK |  | No |  |  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **login Table** |
|  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Column Name** |  | **Data Type** |  | **PK / FK** |  | **Nullable** |  | **Description** |
|  | | | | | | | | | |
|  | [useruser\_name](file:///F:\content\DBTable_YUubT8aGAqESAQwW.html#udr.r8aGAqESAQe1) |  | varchar(255) |  |  |  | No |  |  |
|  | [password](file:///F:\content\DBTable_YUubT8aGAqESAQwW.html#4thbT8aGAqESAQwd) |  | varchar(255) |  |  |  | Yes |  |  |
|  | [user\_id](file:///F:\content\DBTable_YUubT8aGAqESAQwW.html#KbOnT8aGAqESAQ2a) |  | int(10) |  | FK |  | No |  |  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| **user Table** |
|  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | **Column Name** |  | **Data Type** |  | **PK / FK** |  | **Nullable** |  | **Description** |
|  | | | | | | | | | |
|  | [user\_id](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html#Soj7T8aGAqESAQyZ) |  | int(10) |  | PK |  | No |  |  |
|  | [first\_name](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html#80HLT8aGAqESAQsu) |  | varchar(255) |  |  |  | Yes |  |  |
|  | [middle\_name](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html#j1QrT8aGAqESAQte) |  | varchar(255) |  |  |  | No |  |  |
|  | [last\_name](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html#NjArT8aGAqESAQtb) |  | varchar(255) |  |  |  | Yes |  |  |
|  | [gender](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html#8VYrT8aGAqESAQtj) |  | varchar(255) |  |  |  | Yes |  |  |
|  | [phn\_number](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html#AHBrT8aGAqESAQuI) |  | varchar(255) |  |  |  | No |  |  |
|  | [email](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html#4pRrT8aGAqESAQuL) |  | varchar(255) |  |  |  | No |  |  |
|  | [address](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html#DSxrT8aGAqESAQuO) |  | varchar(255) |  |  |  | No |  |  |
|  | [user\_name](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html#Sntiz8aGAqESARFF) |  | varchar(255) |  |  |  | Yes |  |  |
|  | [user\_type\_id](file:///F:\content\DBTable_z9JzT8aGAqESAQsZ.html#mvjXT8aGAqESAQ5y) |  | varchar(255) |  | FK |  | No |  |  |

### **Entity relationship diagram**

An entity relationship diagram is a data modeling technique that graphically represents an information system’s entities and the relationships between them. It is a conceptual and representational model of data used to represent the entity framework infrastructure.

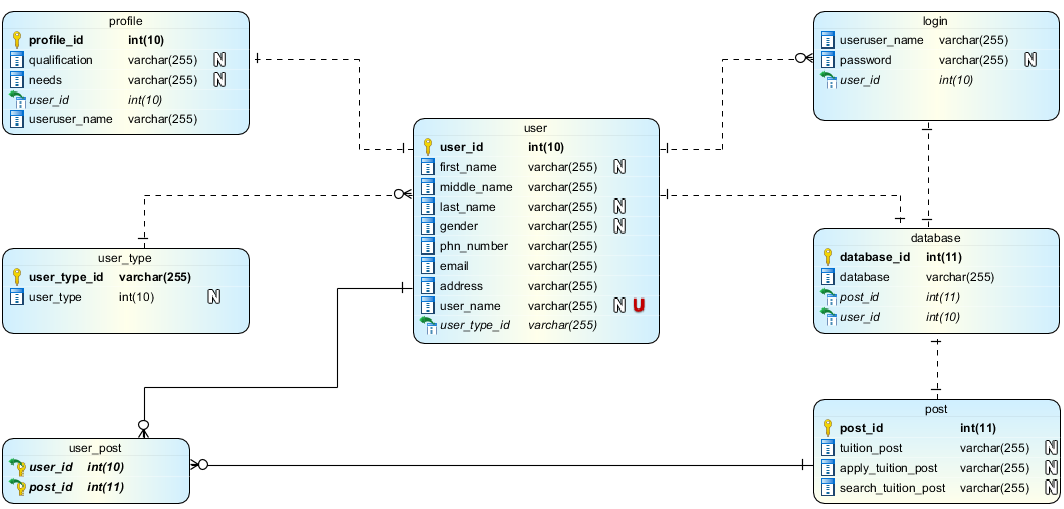


Illustration 15: entity relationship diagram for tutor student portal.

### **User interface design**

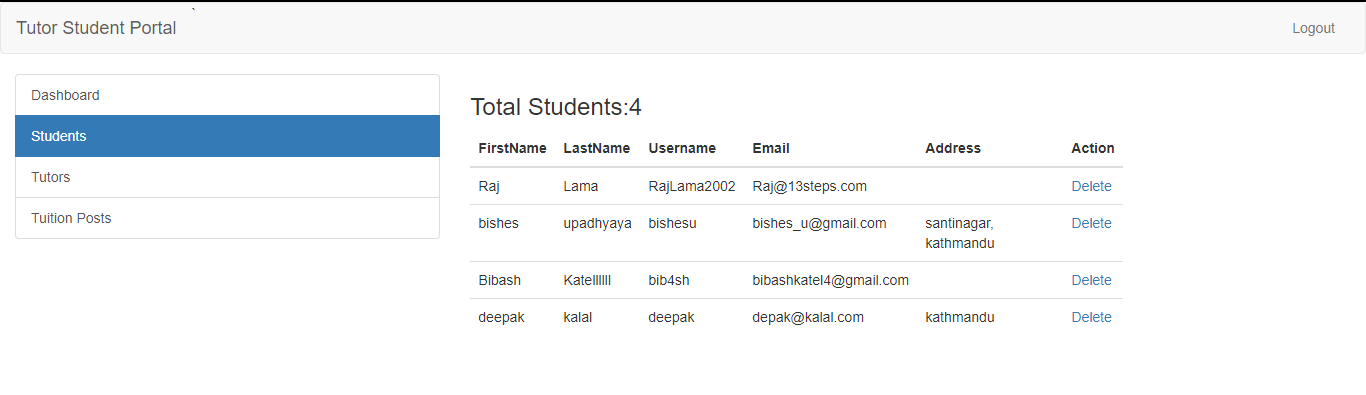


Illustration 16: User interface design.