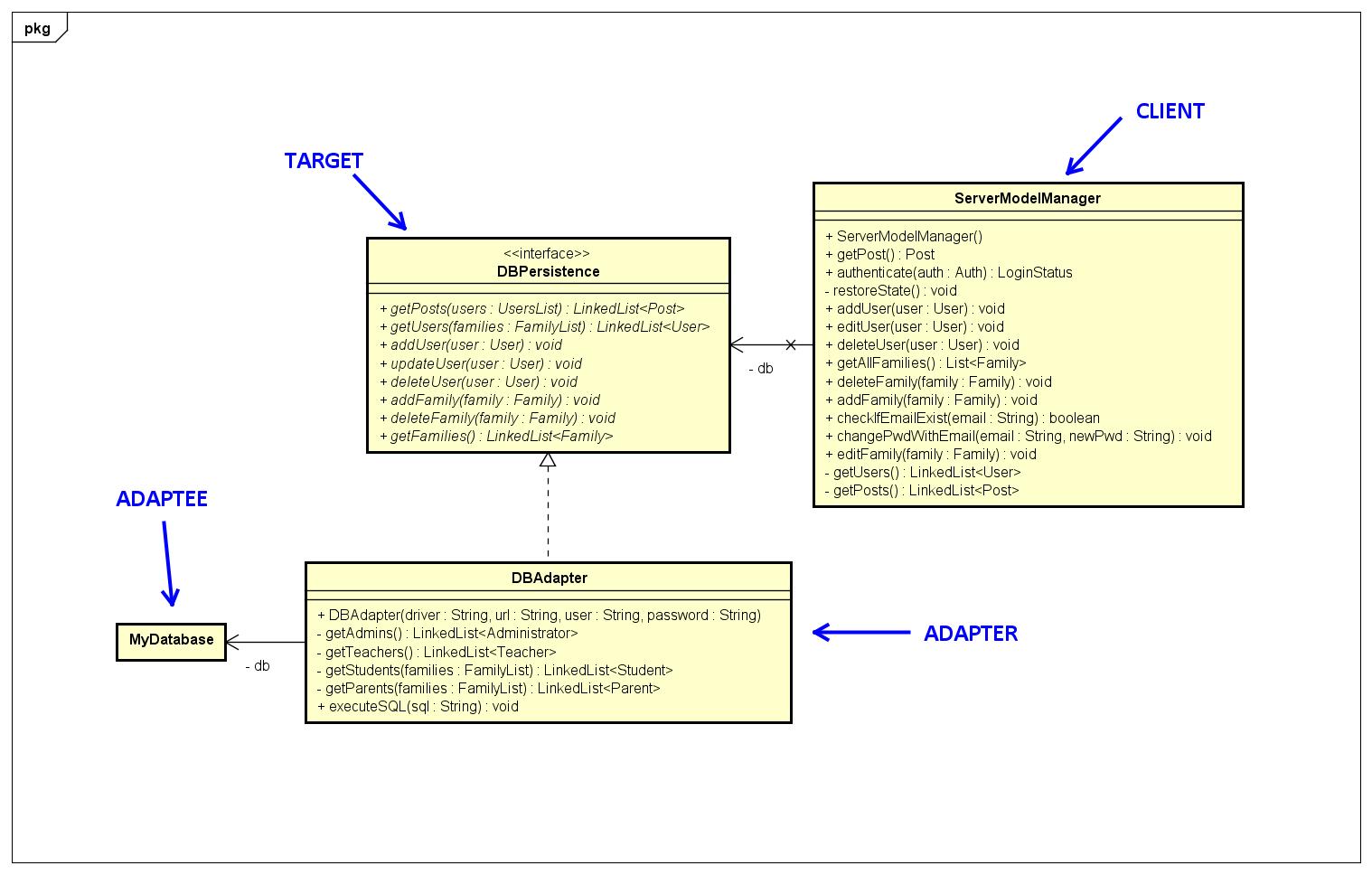
# Connection to the database

The connection between Java and the database is established by using standard Java API - JDBC (Java Database Connectivity). The latest version of JDBC Driver (PostgreSQL JDBC 4.2 Driver, 42.2.2) has been imported in order to access to the database’s data that is placed on eNTe’s private server with IP address - 207.154.237.196.

The adapter design pattern has been chosen for transforming data to and from the database. The main reason for using this design pattern is that it makes the system clearer, divides responsibility and provides the possibility to change the database without having to make any changes in other parts of the system.

It is implemented in the following way. The adapter class (DBAdapter) implements the interface DBPersistence that is like a target that holds methods needed for the model on the server’s side. Therefore the ServerModelManager has not any direct relations to the database (SQL language). The SQL strings are created by methods in DBAdapter class, which is responsible for calling these SQL statements on MyDatabase class, which is represented by the Adaptee in the adapter design pattern. MyDatabase class is made to be general in order to be able to access data from any kind of database.

NOTE: DBAdapter method called “executeSQL(String sql)” was created only for testing purposes to be able to delete all data after each particular test in DBAdapterTest class.

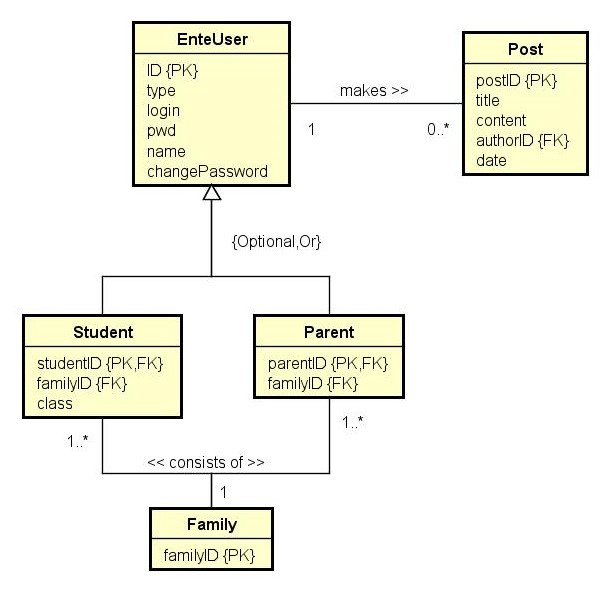


# Documentation DB design for managing users

There was a need to create a database for storing data about users from the system. Exact tasks for the database were following:

1. For every of 4 types of users (Administrator, Teacher, Student and Parent) eNTe wants to keep track of the user’s name, login and password.
2. For every student and parent it is needed to keep track of family that he/she belongs to. Student or parent can belong only to one family and a family can consist of multiple students (children) and parents.
3. For every student eNTe needs to hold student’s current class. Therefore, a student can only be a part of one class and a class can consist of multiple students.

The EER diagram below presents a possible solution for tasks mentioned above.



At the moment, there is no need to make Teacher or Administrator as an entity as they do not hold any specific data in comparison with enteuser. Therefore only Student and Parent are made as separate entities that hold specific data about family and also about class in the case of a student.

Following logical database design shows the primary and foreign keys for the given entities.

**EnteUser** (ID,type,login,pwd,name,changePassword)

PK: ID

**Post** (postID,title,content,authorID,date)

PK: postID

FK: authorID REFERENCES EnteUser (ID)

**Student** (studentID,familyID,class)

PK: studentID

FK: studentID REFERENCES EnteUser (ID)

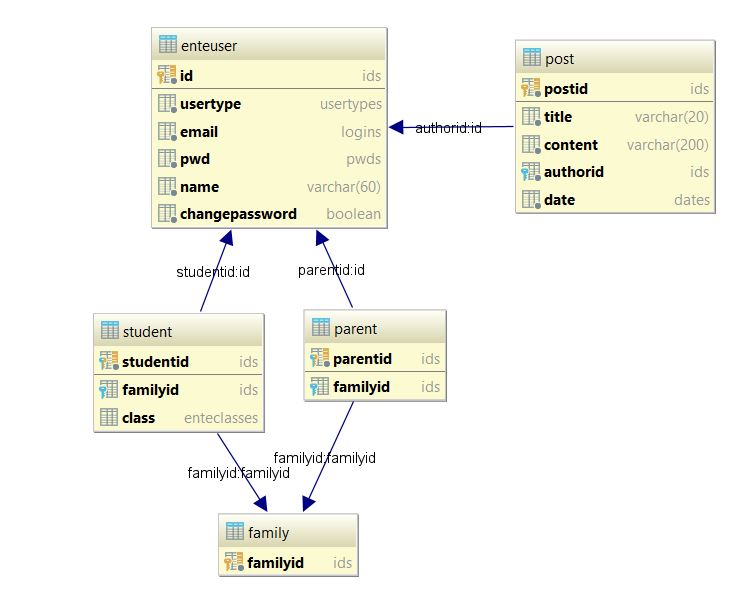
FK: familyID REFERENCES Family (familyID)

**Parent** (parentID,familyID)

PK: parentID

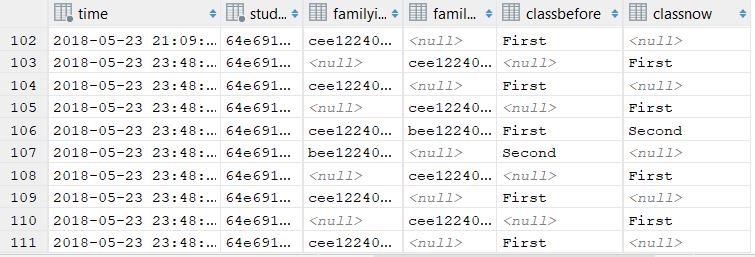
FK: familyID REFERENCES Family (familyID)

Physical database was based on the previous analysis (EER diagram, logical database design). It can be seen in the database visualization below.



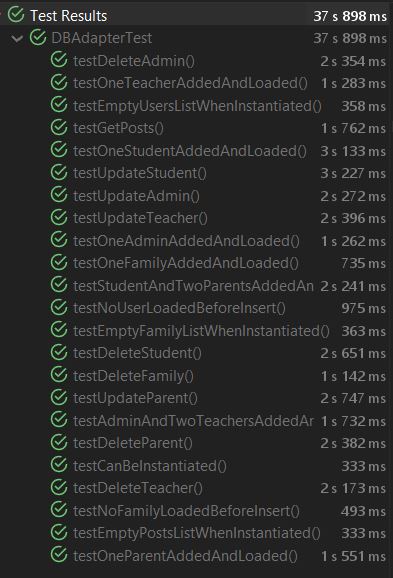
# JUnit Tests for managing users – DBAdapter

DBAdapter class was tested using unit testing framework for Java called Junit. In order to see data flow in the database, the structure (tables, domains) of the main schema were cloned into the testing schema. Moreover, log tables for all already existing tables were created in the testing schema with the intention of controlling data manipulation done by the DBAdapterTest class.



Log\_student table created in test schema of eNTe’s database

The reason of such a long execution of this test is that after each test all data from the database is dropped. That is because it is needed to have no data in any table while a particular test is being executed. Eventually, all tests passed.



Test list:

* Add user
* Delete user
* Update user
* Get users
* Add family
* Delete family
* Get family
* None data retrieved
* Connection
* Get post

After few sprints the system already the same extent as SEP1

Problems with database .. bad feeling .. but we helped each other

Even though we were not totally satisfied with the final version of our SEP1, we decided to stay together and continue as a group also for a SEP2. To my mind, this was a good decision as we did not need to go through the stage of getting to know each other. We already knew our own personalities, habits and roles in the team and that is why we could directly enter the stage when all team members are trying to achieve team’s goals. In my opinion, just these team’s goals were the stumbling block of the current project. The reason why I see it like this is that customer needs did not meet the needs of SEP2. As we wanted to stick to both, sometimes it was difficult to find the best way to satisfy both sides. For instance, on the hand the customer required some feature that we were not able to implement at that moment and it would have taken us too much time to learn it and implement it, so after the dialogue with customer it was decided to move this feature to the future of the project. On the other hand, we could do some stuff that would be interesting for SEP2, but as the customer did not need it, there was no reason to implement it.

Regarding the process of our work, I am sure that every team member has made every effort to build this project. The SCRUM way of controlling the development process helped us a lot. We did not have problems with work organization as we had had during SEP1. I had a feeling that during this project we had it more under control. We knew who is doing what and what is more, thanks to a lot of meetings (review, retrospective and daily meetings) we were totally confident about the progress of our group. In my case, I was responsible mostly for handling the database, the connection to the database and the data transformation to and from the system. As I had not any experience with databases before, during the project I found some difficulties in the realization of my tasks that a few times leaded to the postponements of their finalization to another sprint.

To sum up, as it was already the second project that we worked together, as a group we were more confident what semester projects are about, more consistent and mostly more efficient. I do believe that it was definitely a right decision that we worked on this project for eNTe, because it gave us even more clear vision of how are projects going in the real world than we would have acquired if we had decided to make project for an imaginary company.