UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

SCHOOL MANAGEMENT SYSTEM

MIHAI – IONUȚ CURAC

CONTENTS

SHORT DESCRIPTION

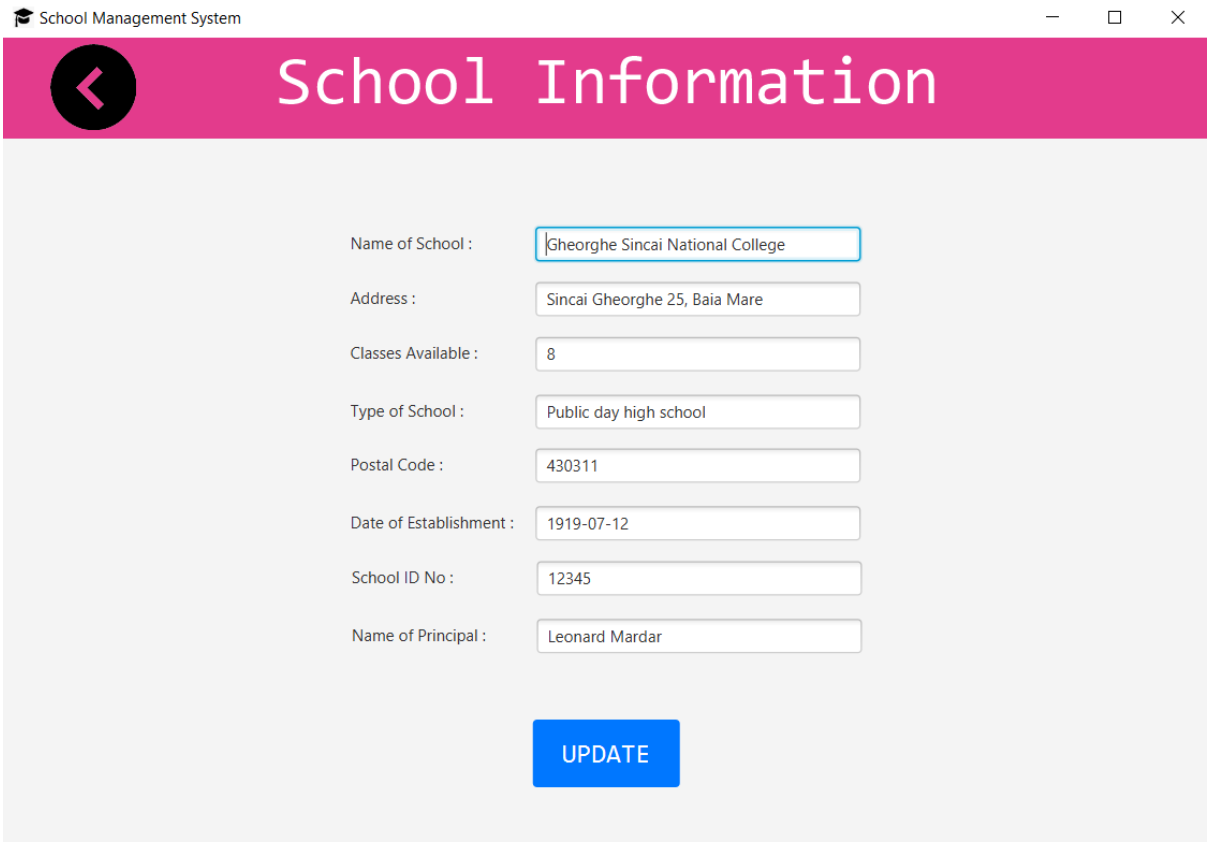
My project is a School Management System application that uses a MySQL database to process information about a school, its students and its staff. It is capable of adding, updating, deleting and showing all of this information that is connected to the database. I made the graphical user interface of the project using JavaFX and a bit of JFoenix (open source java library that implements the Google Material Design). I also used the MySQL Connector/J, that provides connectivity to the database in Java.

There are three standard accounts from which users can log into the application: the ‘admin’, ‘teacher’ and ‘student’ accounts. The first has control over everything and can edit the records in the database, like the school information, teachers’ data and students’ data. The ‘teacher’ and ‘student’ accounts can only view a table with all the school’s students organized by their year of study.

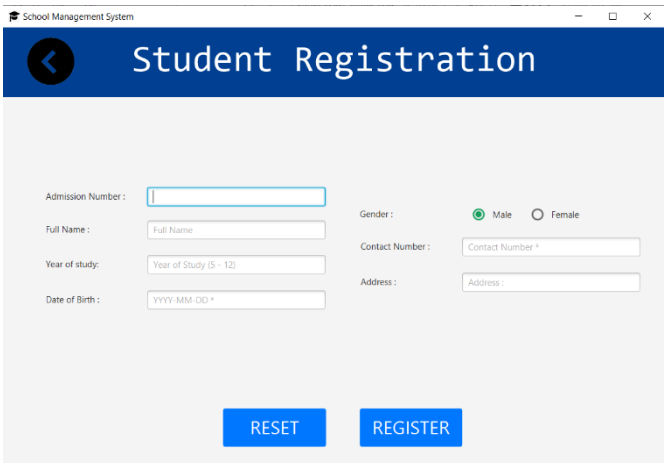
PRESENTATION OF USE-CASES

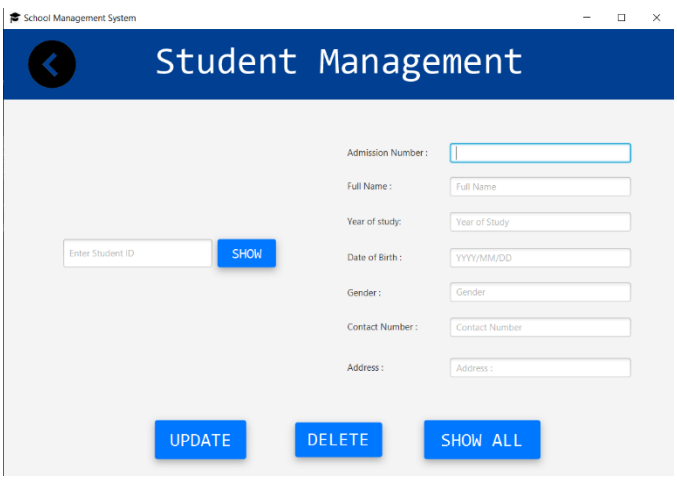
After logging in as ‘admin’, the user sees the dashboard of the application, with three buttons named “School Information”, “Student Management” and “Staff Management

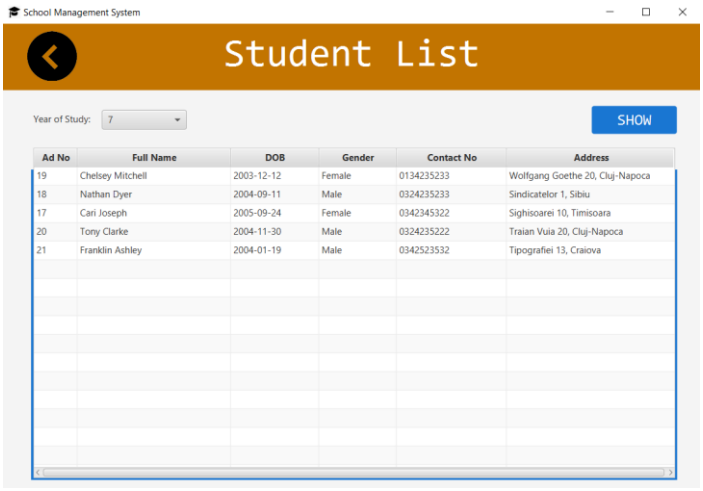
In the “School Information” section, the administrator can enter new info about the school in the 8 fields of text and then click the “Update” button. This will update the information about the school in the database. The black button with the arrow goes back to the dashboard window.



In the "Student Management" section, the user can register new students ("Student Registration" button) or modify/view the data of the already registered students ("Student Information" button). In the registration section, 6 fields of text and two Radio buttons show up. These can be filled with the new student's data and then the user can click the "Register" button to send everything to the database. In the "Student Information" section, the user can enter the ID of a student and then click the "Show" button, which will display the data of that particular student. Then, the admin can update the student's information by modifying the text fields or he can delete the student from the database entirely, by clicking the "Delete" button. The user can also see a table with all of the students enrolled in the school by clicking the "Show all" button. Here, the year of study can be selected from a drop-down list and then the "Show" button can be clicked, which will display the table with the students in that year of study. This section is the only one that can be viewed by all the three main accounts.

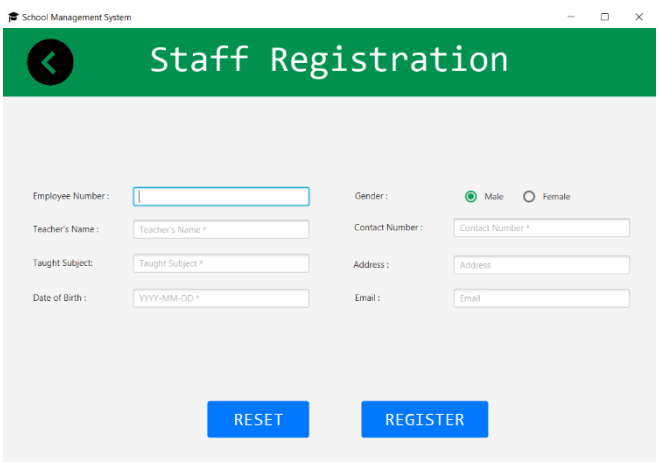


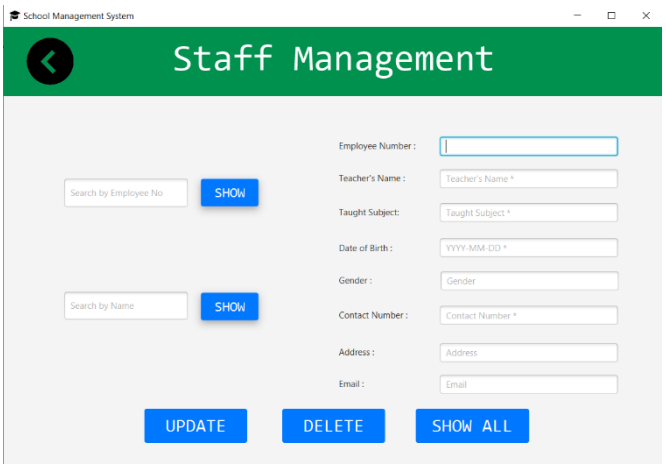


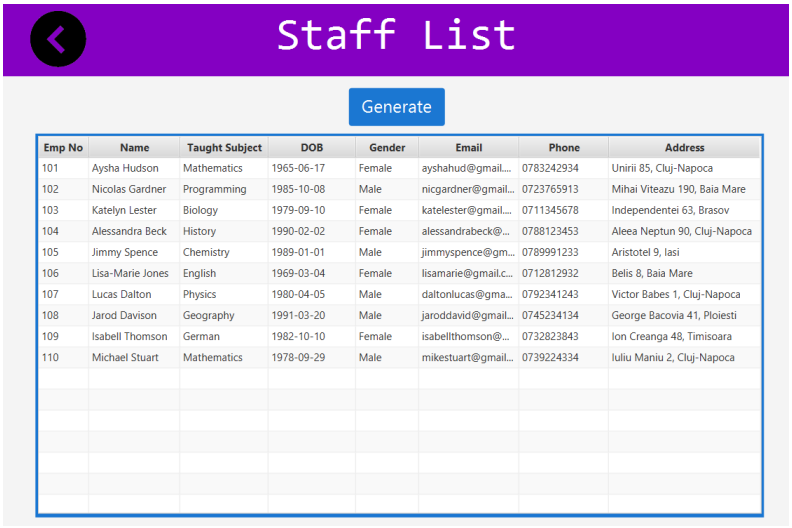


Users can sort the data in the table by clicking the column name by which they want to sort (for example, sort alphabetically by clicking the “Full Name” column).

In the “Staff Management” section, the administrator can similarly register new teachers or modify/view data about them. Here, the user can search for staff members by employee number or by name, and then update information about them or delete teachers from the database entirely. A “Show all” button is present here as well, which leads to a window that generates a table with all the staff member’s information.







SOLUTION PRESENTATION

For this project I used the MVC design pattern. I separated the “model” layer into the main model one and the table model one. I did this as well with “controller” – the database controller layer and the main controller layer. “View”, the presentation of the application, contains the .fxml files corresponding to JavaFX. I also used Scene Builder to place the elements of the interface in an easier, more friendly manner.

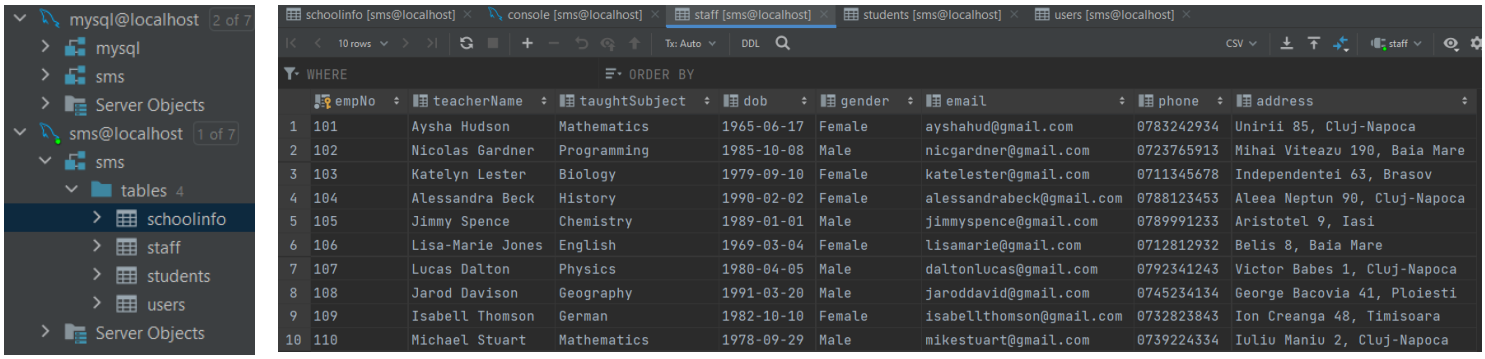
The “DBConnection” class tries to connect to the MySQL database, while the classes from “dbController” execute queries and INSERT, UPDATE or DELETE operations on the “sms” database. In all of these, we use Getters from the “model” classes to access the information of the school, staff and students and modify the database. “getYears()” in “YearController” also returns an ArrayList of the distinct years of study from the database (the school may not have students at all in a certain year, or it may have multiple students in all years of study). This will be used in a ComboBox in the table of students.

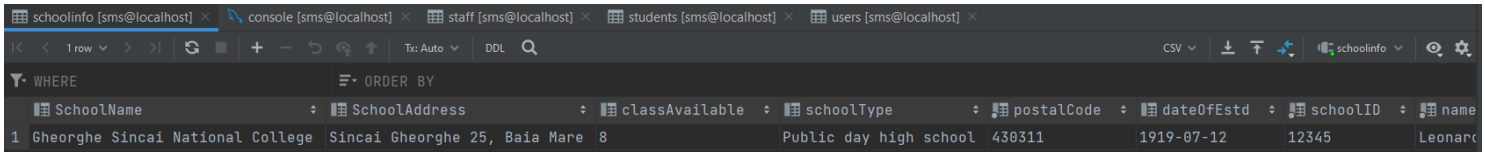
The ”School”, “Staff”, “Student”, “StaffTableModel” and “StudentTableModel classes have as fields the relevant information for each entity, as well as Constructors, Getters and Setters.

The “LoginFormController” class makes sure than only the admin account gets access to the main dashboard of the application, while the teacher and student accounts are directed to the section with the table of students. The last two can only choose a year of study and show the corresponding student data, not being able to press the “Back” button that is available to the admin account. There are also errors for invalid username or password.

The rest of the controller classes describe the functionality of the various buttons in the project. For every add/register, search, delete and update operations I also made alerts that display possible errors. In “ValidationController”, I added some more check methods: one to verify that the text fields in Student and Staff Registration are not left empty, one to verify that only numbers were input (using regex), one to validate phone numbers and one to validate dates.

The MySQL database that I created and connected to the Java application contains 4 tables, with information about the school, staff, students and the user accounts (username and password).





schoolinfo [sms@localhost] | console [sms@localhost] | staff [sms@localhost] | students [sms@localhost] | users [sms@localhost]

25 rows | Tx: Auto | DDL | CSV | students

WHERE | ORDER BY

	adNo	fullName	yearofstudy	dob	gender	phone	address
1	10	Jason Medrano	5	2001-05-11	Male	0747345923	Iuliu Maniu 1, Cluj-Napoca
2	11	Claire Kim	5	2002-07-03	Female	0793425245	Independentei 8, Brasov
3	12	Jarrod Savage	5	2000-10-28	Male	0724534524	Vasile Lucaciu 99, Baia Mare
4	19	Chelsey Mitchell	7	2003-12-12	Female	0134235233	Wolfgang Goethe 20, Cluj-Napoca
5	18	Nathan Dyer	7	2004-09-11	Male	0324235233	Sindicatelor 1, Sibiu
6	17	Cari Joseph	7	2005-09-24	Female	0342345322	Sighisoarei 10, Timisoara
7	16	Sophie Rodgers	6	2005-07-20	Female	0423523523	Robert Schumann 91, Baia Mare
8	15	Edgar Cousins	6	2007-02-03	Male	0324324523	Oltului 19, Pitesti
9	14	Timothy Archer	6	2006-04-09	Male	0732423553	Mihail Eminescu 87, Constanta
10	13	Riccardo Solomon	6	2005-08-10	Male	0372846235	Liviu Rebreanu 33, Cluj-Napoca
11	20	Tony Clarke	7	2004-11-30	Male	0324235222	Traian Vuia 20, Cluj-Napoca
12	21	Franklin Ashley	7	2004-01-19	Male	0342523532	Tipografiei 13, Craiova
13	22	Thiago Flynn	8	2002-05-05	Male	0231242353	Somesului 2, Galati
14	23	Joanna Ashton	8	2002-10-01	Female	0542345232	Soporului 24, Cluj-Napoca
15	24	Molly Lane	8	2003-09-04	Female	0432525233	Radu Tudoran 3, Oradea
16	25	Clayton Cooper	9	2001-05-28	Male	0432523523	Rahovei 5, Braila
17	26	Chanel Ryder	9	2001-07-18	Female	0342352452	Gheorghe Marinescu 18, Arad
18	27	Tyrell Woodward	9	2001-04-11	Male	0852352355	Napoca 90, Bacau
19	28	Jillian Ramsey	10	2000-03-20	Female	0342452352	Mihail Kogalniceanu 13, Baia Mare
20	29	Ho Bridges	10	2000-09-10	Female	0742343243	Luncii 28, Targu Mures
21	30	Dora Bradshaw	10	2001-08-14	Female	0243124233	Jiului 1, Botosani
22	31	Tobi Sweeney	11	1999-04-30	Male	0852353253	Henri Coanda 30, Cluj-Napoca
23	32	Luther Lister	11	1998-10-23	Male	0423423532	Gheorghe Dima 18, Satu Mare
24	33	Daniella Peterson	12	1997-09-01	Female	0842343252	Emil Racovita 77, Suceava
25	34	Angela Atkins	12	1996-11-07	Female	0723425322	Cezar Petrescu 21, Cluj-Napoca

CONCLUSION

All in all, this is a simple School Management System application that lets us register, modify, delete and view data about a school, its students and staff. It can be accessed with multiple accounts with different “permissions”, for safety reasons (only an administrator should edit or delete information from the school database). It also has a friendly user interface that was made using JavaFX and is connected to the MySQL database.

FURTHER IMPROVEMENTS

The application could definitely be further improved by adding a grade book (record of the students’ grades at various subjects), an average grade system for exams, attendance lists, medical records and student schedules. It could also benefit from adding more functionality to the student and staff accounts and expanding some permissions, like teachers being able to modify students’ data, but not staff’s data.