University database management system report

- **Author:** Smeyanov Maxim, 211-1
- **Submission date:** 13.06.2022
- **Supervisor:** George Piatsky

Problem statement

Creating Qt application which would manage university database of a given example with possibility to upload it, view, edit fields, add profiles, delete profiles, sort listings by any of the fields and finally to save new file.

Individual project specification

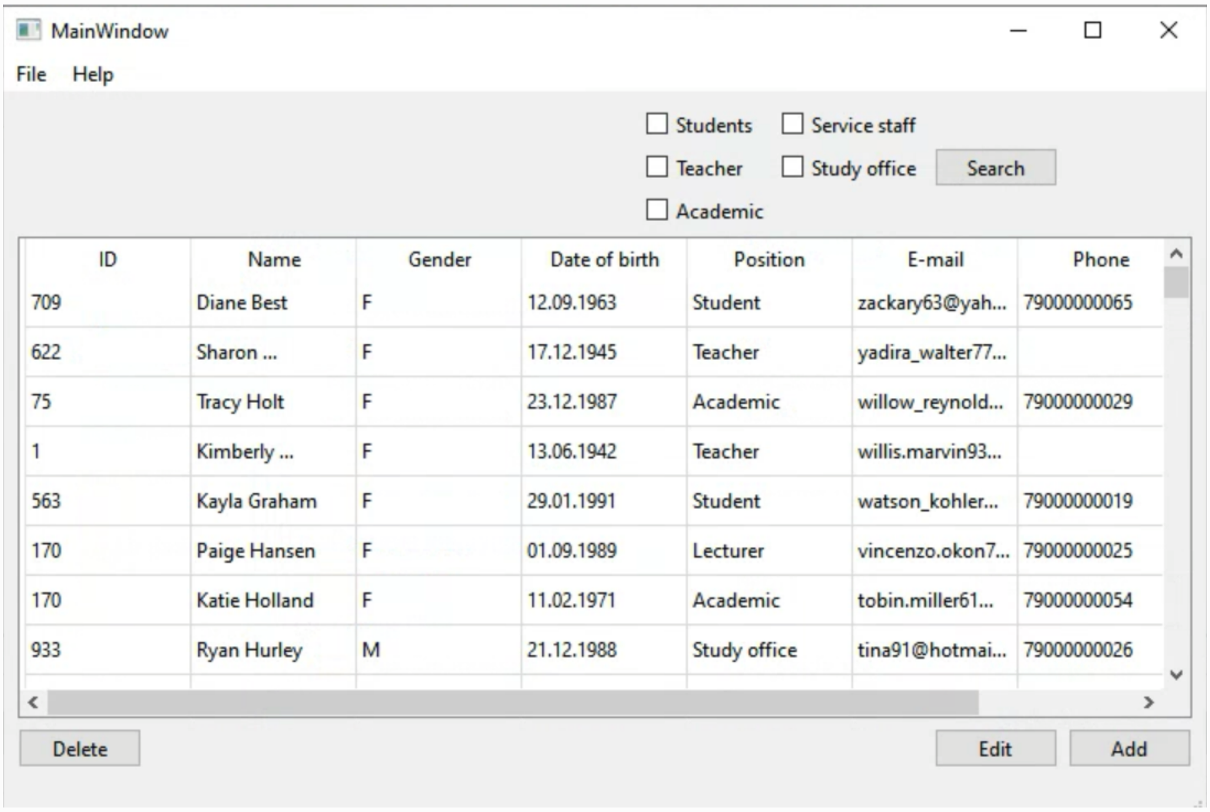
spec_110

Implementation details section

Generally I sticked to seminar approach of implementing most of the frontend and backend logic.

The worst issue I met was setting up Qt for Mac M1 machine, loss of the time after which I moved to Windows machine.

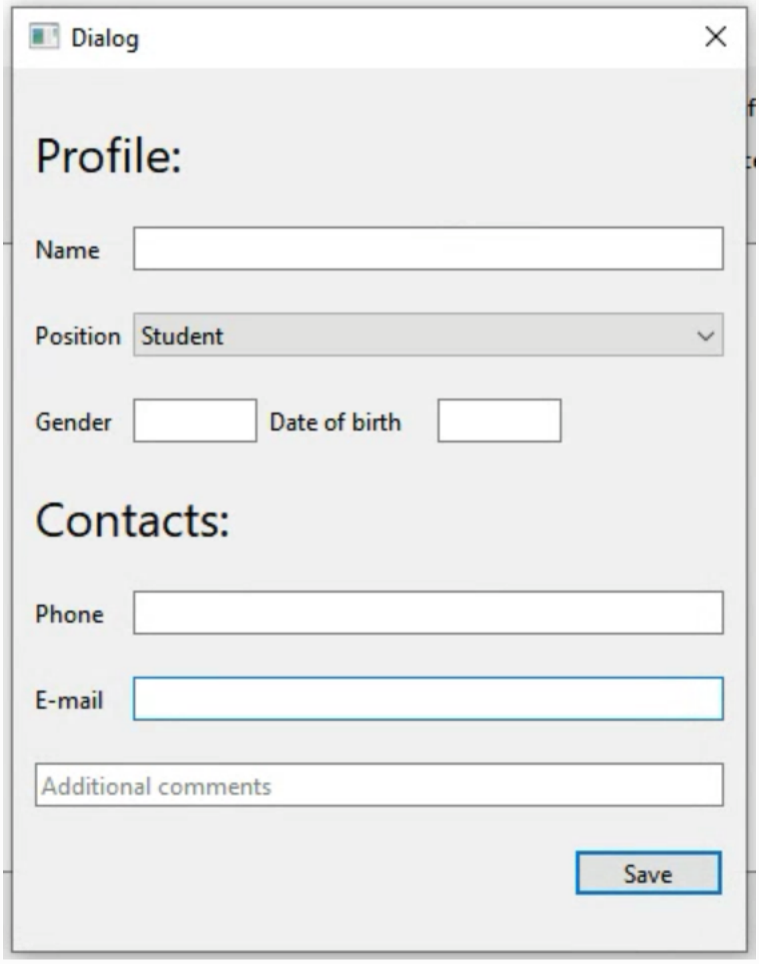
Main Window:



ID	Name	Gender	Date of birth	Position	E-mail	Phone
709	Diane Best	F	12.09.1963	Student	zackary63@yah...	79000000065
622	Sharon ...	F	17.12.1945	Teacher	yadira_walter77...	
75	Tracy Holt	F	23.12.1987	Academic	willow_reynold...	79000000029
1	Kimberly ...	F	13.06.1942	Teacher	willis.marvin93...	
563	Kayla Graham	F	29.01.1991	Student	watson_kohler...	79000000019
170	Paige Hansen	F	01.09.1989	Lecturer	vincenzo.okon7...	79000000025
170	Katie Holland	F	11.02.1971	Academic	tobin.miller61...	79000000054
933	Ryan Hurley	M	21.12.1988	Study office	tina91@hotmai...	79000000026

Made using Grid Layout for window size adjustment, QTableView for displaying the table, QPushButtons for some actions, QCheckBoxes for filtering and QActions for all buttons in menu.

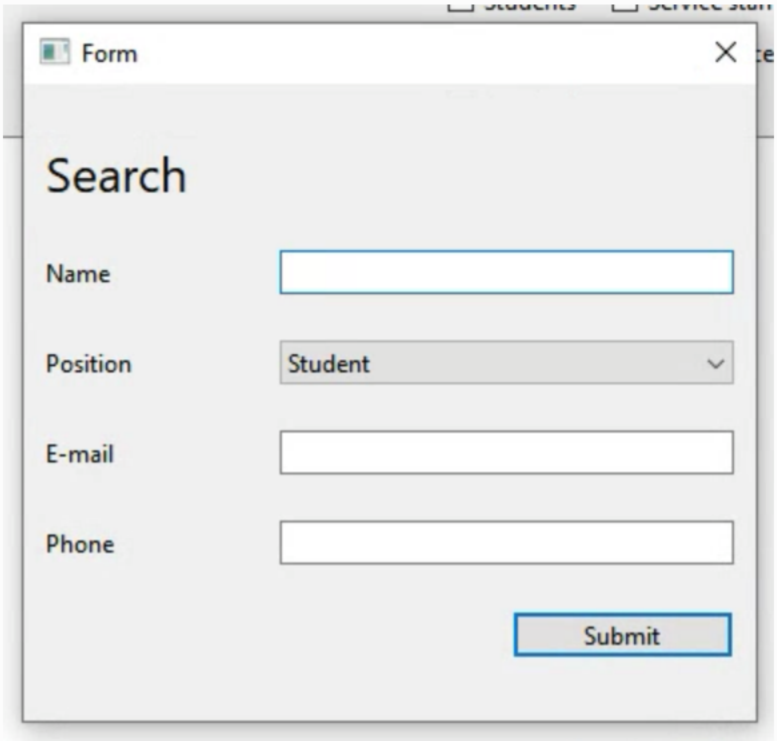
Adding profile filling window (same for editing):



Position: Student

Made using Grid Layout, combobox to choose out of limited number of options for person’s position.

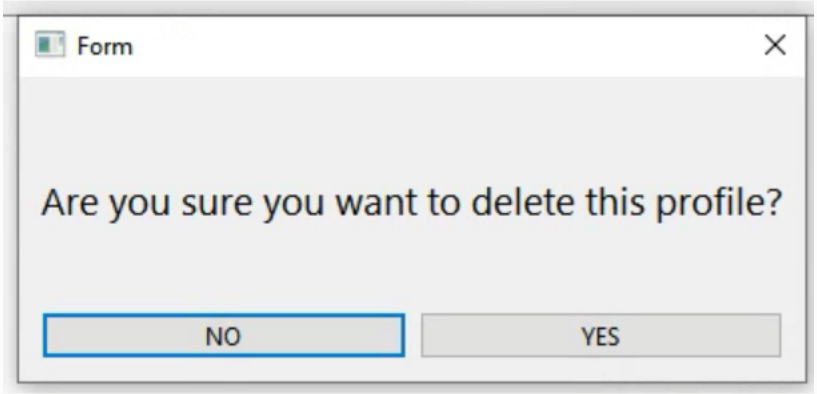
Search window:



Position: Student

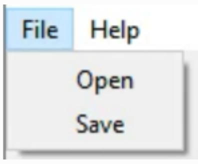
Made using Grid Layout, combobox to choose out of limited number of options for person’s position.

Delete confirmation window:

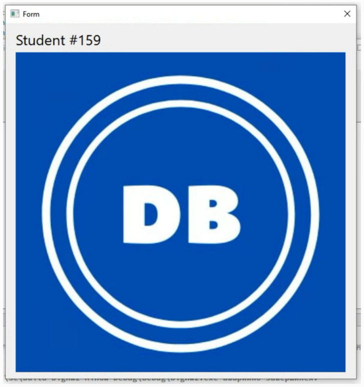


Are you sure you want to delete this profile?

File section dropdown:



About widget:



All the model logic is stored in UniversityModel with all the necessary overrides over QAbstractTableModel class.

Editing of the listings could be done both by clicking right on fields in table and by special form.

Results and discussion

Basic requirements for project were fulfilled as well as the window markups, unfortunately not all the advanced requirements like sorting and search were implemented. There is a slight change to ui of the main window where the input window was supposed to be, as there is no sense in it since search function should be created by clicking ‘search’ button and getting special dialog window to input information there, so I got rid of that field.

Conclusion

Firtstly, it is not finished project, where sorting and search functions are still to implement.

When new row is added id is now taken from name but it is better to make check for existing id’s and make it adjustable to that.

Possible further development was suggested by author of specification including photo implementation to existing system, for which I would use cloud system. Export to different formats could appear useful in practice.