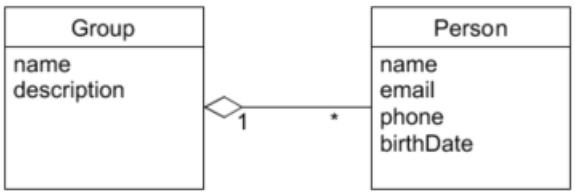
Microproject - Full DAO architecture with GUI

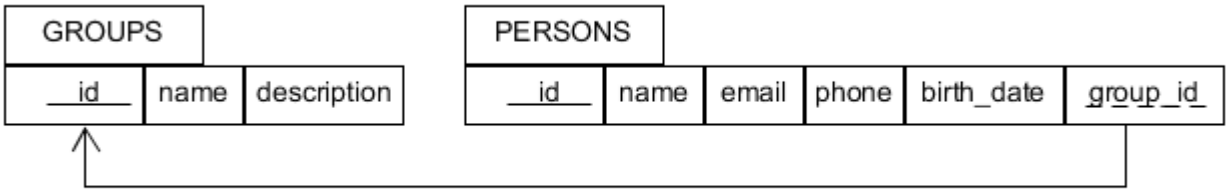
In this exercise, we create a program to manage contact persons in groups. Most of the cursory documentation is in place, your task is to implement the system.

Note that the "most complex" use case is to move a person from one group to another, which is something of a stretch to call a "complex" use case. Nevertheless, it's got its own controller.

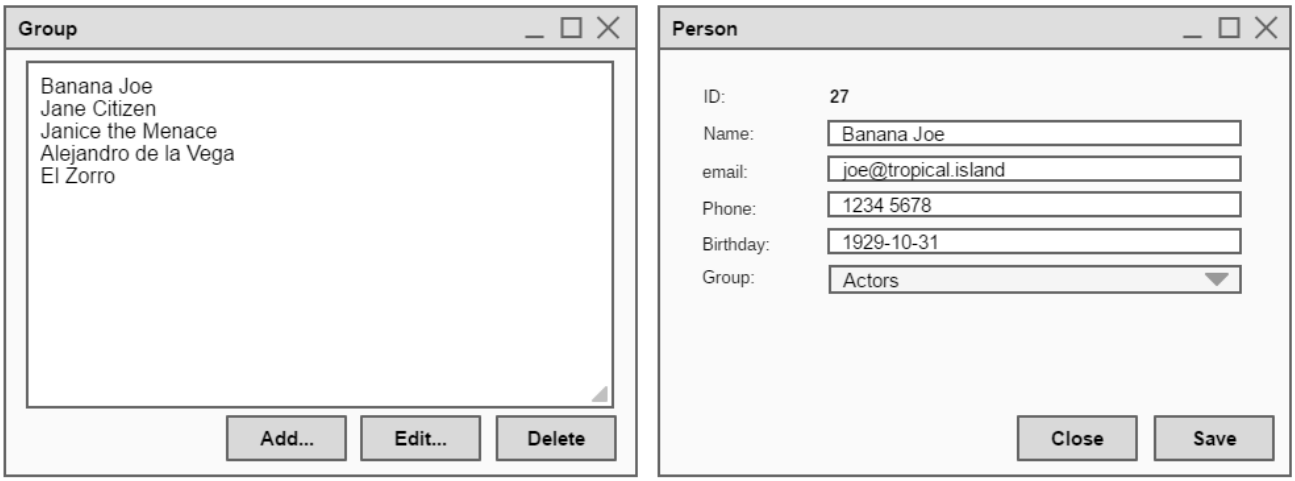
Domain model:



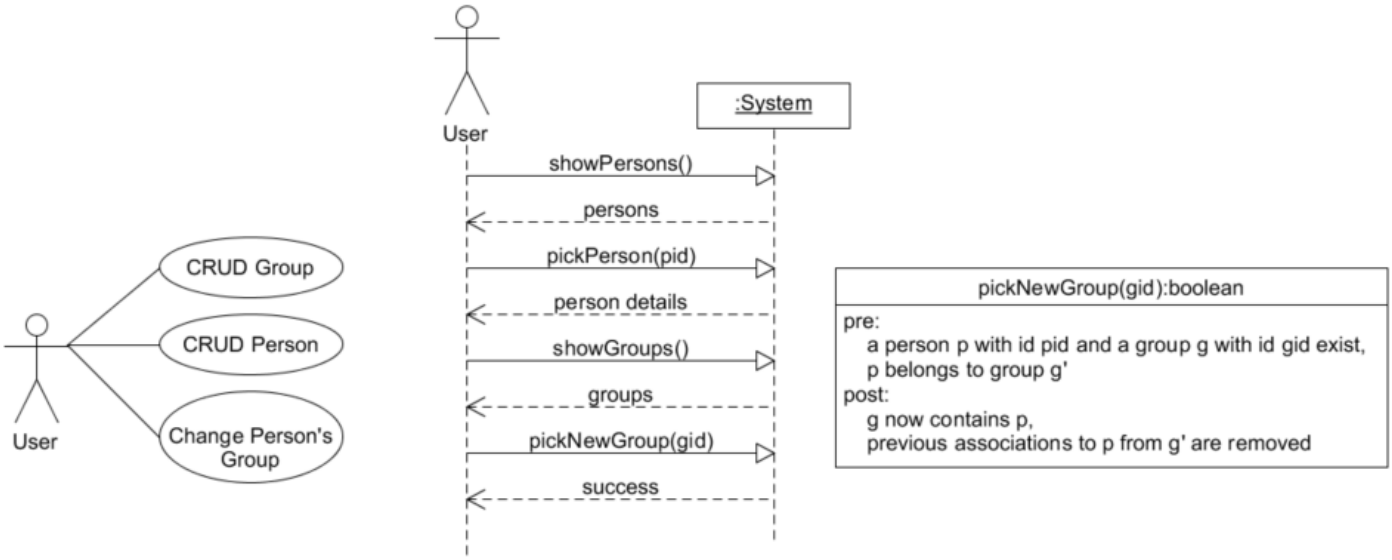
Derived, normalized relational model:



The idea is that you want to pick a person from a list of persons, change his group membership from an object view, and save the changes. The mock up is as shown:

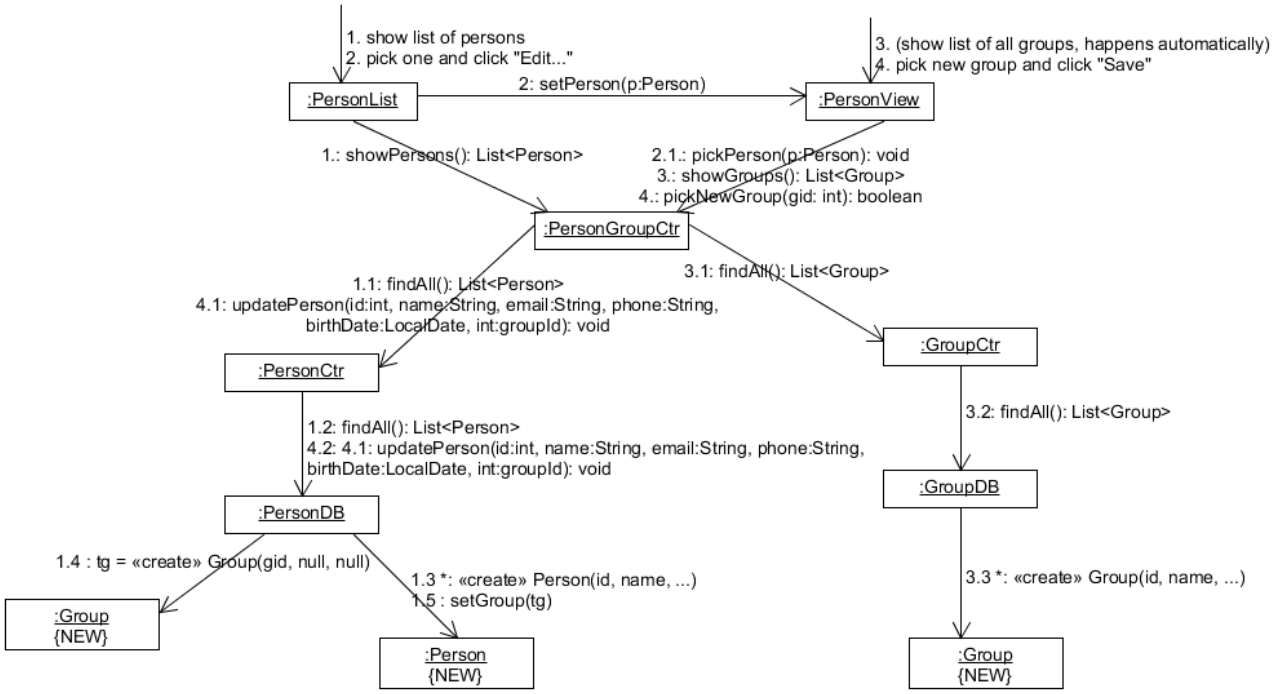


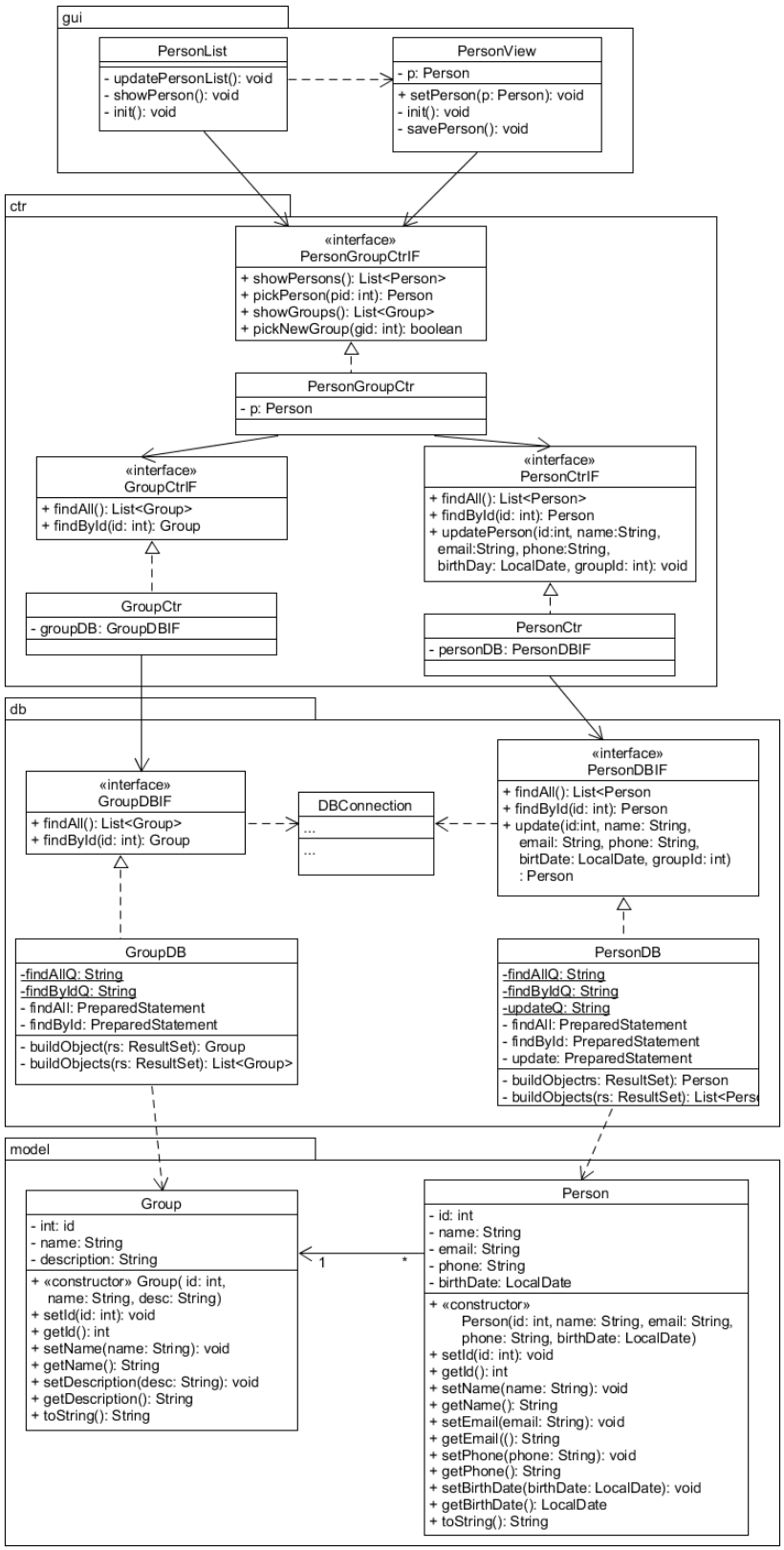
The use case diagram and SSD of the system are shown below:



Notice the only message that changes the system state is the **pickNewGroup** call. For this, an operational contract has been provided.

The communication diagram for the "complex" use case is shown below:





And we derive a design class diagram (←)

You should take the implementation details of the DBConnection class from the previous exercises.

To solve the exercise, a top-down approach is advised, use the S.T.R.E.A.M. method from the 1st semester and start implementing the GUI, while writing stubs for the methods. Note that the design class diagram may contain intentional and unintentional omissions, it is your task to fix and complete the design and code the system.

If you find the exercise too easy, you should implement additional use cases and a complete test suite.

All working files can be found in the exercise folder (mockups, design diagrams). To open the mockups, you must use the Firefox based "Pencil" project:

https://pencil.evolus.vn/Downloads.html

Solution strategies:

1. **I am good at this**, I'll take the diagrams and code it all. For the ambitious and skilled.
2. **I am uncertain about some of this,** I'll take the partial solution and write the missing classes. For the average students.
3. **I have no idea where to begin,** I'll look at the solution and copy (by hand!) the interaction no. 1 and see if I can solve interactions 2 – 4 as in solution strategy 2.
4. **I have zero idea where to begin,** I'll try to understand some basic Java and SQL before I start catching up with all the stuff I missed out on. ***Also, I'll contact my programming teacher ASAP for guidance***.