# Employee Administration

The CSEE department plans to develop a new software for the administration of its employees and to calculate salaries. Your company shall develop a first prototype performing a complete engineering cycle.

## Customer Requirement Specification

### General topics:

The employee administration software shall store employees' data relevant for the company using it. It should allow changing and deleting data. Additionally, the software shall manage salary data for all employees, perform the monthly salary calculation for all salary classes under consideration of the working time, print the salary sheets for all employees and transfer the salaries to bank accounts. The program shall distinguish between 3 salary levels (lecturer, associate professor, and professor). The salaries are calculated in different ways depending on the salary level.

### Detailed requirements:

1. Data shall be stored for new employees.
2. Employee data shall be changeable.
3. Employee data shall be erasable.
4. The application should be accessible to but administration and employees (with limited access).
5. The following data of an employee shall be stored:
   * First name, last name
   * Gender
   * Date of birth
   * Address
   * Salary class (lecturers, associate professors and executive. employees)
   * Monthly basic salary
   * for associate professors: bonus for success
   * for executives: bonus for success and number of shares
   * Tax class
   * Religion
   * Working time (full / part time)
   * Bank account
6. The access to employee data shall be comfortable.
7. The software shall manage enough employees.
8. Lecturers get a monthly basic salary and additionally two additional salaries as vacation and Christmas bonus. The vacation bonus is paid together with the June salary, the Christmas bonus is paid with the November salary.
9. Associate professors get a monthly basic salary and a success bonus that is paid in August.
10. Professors get a monthly basic salary, a success bonus and shares that are paid in August.
11. It shall be possible to calculate salaries each month.
12. It shall be possible to transfer salaries each month to bank accounts.
13. It shall be possible to print out salary sheets for the employees each month with the following content:
    * Personal data
    * Basic salary
    * If applicable, additional salary components (e.g. bonus, shares)
    * Total salary before taxes (under consideration of full time / part time)
    * Taxes
    * Social charges
    * Salary after taxes and social charges
14. Additional extra bonuses shall be possible for each employee.
15. Full time is 40 hours per week, part time can be between 39 and 10 hours per week.
16. Part time reduces the salary corresponding to part time / full time share.
17. A simplified tax system shall be used for the first version of the program:
    * Tax rate of 20% of the salary for lecturers
    * Tax rate of 25% of the salary for associate professors
    * Tax rate of 30% of the salary for professors
18. A simplified social charge of 10% of the basic salary shall be used for the first version of the program.

# Detailed task description for practical exercise:

1. Organize your team, distribute the tasks, take care that all team members are involved in all tasks.

2. Write a project application using the available template. Get the agreement of your customer.

3. Plan your project based on the task description

* Time planning of work packages and their dependencies
* Resource planning (who does what)
* Configuration management (where is what stored, definition of access)

4. Requirements

* Analyze the requirements for
  + Understandability
  + Completeness
  + Feasibility
* Clarify and complete requirements with your customer (where necessary). Document the requirements in Word document. Start with the requirements from the exercise description. Ensure that the requirements are unique and testable.

5. Perform an object-oriented analysis of the tasks (technology agnostic). Define and document the most important use cases. Create the program structure from analysis point of view. Document your results using an appropriate tool.

6. Create the SW design (architecture and detailed design) using UML (at least some important parts of the detailed design).

* Identify and describe the program components including interfaces and relations among each other
* Identify and describe the classes including interfaces and relations among each other
* Identify and describe program states (where necessary) using state diagrams
* Identify and describe important execution sequences using sequence diagrams. Make sure to include System Sequence Diagrams

7. Implement the software according to the design created.

8. Do code reviews for the pull requests for all the features added.

9. Create automated tests for your business logic classes.

10. Perform a validation of the implemented software

* Create a test specification (Word document) for the validation.
  + using appropriate test strategies
  + testing all requirements
* Make sure that the test cases can be traced to requirements.
* Implement the test cases and execute the test.