**CS4013 Software Development Project**

**University of Limerick**

[**Michael.English@ul.ie**](mailto:Michael.English@ul.ie)

**Objectives:**

* Apply an object-oriented approach to the design and development of a software application in Java.
* Apply the concepts and techniques introduced in lectures, tutorials and labs to the design and development of a small software application.

**UL Payroll System**

**System Requirements:**

You are required to build a Payroll system for UL Staff.

A link to the UL salary scales for full-time employees is given here (<https://www.ul.ie/hr/current-staff/pay-benefits/salary-information>).

There are similar salary scales for hourly paid employees. For example, there may be different scales such as Lecture, Tutorial, Lab, Demo, Exam Invigilator, Exam Supervisor etc... Each of these scales would also have several points.

Part-time employees must submit a pay claim form by the second Friday of a month in order to be paid that month. The payroll system should generate pay slips for all full-time staff and hourly paid staff (with current claims) on the 25th day of each month.

When calculating net pay for a pay slip, certain deductions must be made. These include deductions for Health Insurance, Union Fees, and for the various taxes (e.g. PRSI, USC, Income Tax). See [www.revenue.ie](http://www.revenue.ie/) for more information.

In October each year, full-time staff are moved to the next point on their salary scale (if they are not already at the top of that scale. From time to time an employee may be promoted to the next salary scale within their professional category (Academic/Administrative/IT/Technical etc..). The point to which a staff member is promoted to the new salary scale is dependent on the number of years spent at the top of the previous scale.

Other information about the payroll system in UL is available here: [Payroll | University of Limerick (ul.ie)](https://www.ul.ie/finance/our-services/payroll).

All relevant data associated with the payroll system (such as employee, payroll, payslip details, including historic payslips) and any other files you deem appropriate for this application should be stored as csv files.

The application has three user types.

An employee can log in to the system, see their details, and view their most recent or historical pay slips.

An admin user can log in and can add a new employee to the system.

A human resources user can log in to implement the promotion functionality for full-time staff.

In such cases the staff member should be asked to confirm the promotional changes being applied.

A command line interface (CLI) should be included to facilitate this interaction between users and the application. The CLI should be included with the expectation that a Graphical User Interface (GUI) may be required in the future. In other words the separation between the text-based user interface should be well-defined to allow a graphical user interface to be substituted easily.

**Suggestions**

* In terms of managing the project, consider adopting some of the ideas from an Agile approach such as Scrum…for example such an approach advocates daily meetings. In the context of this project and your work on other modules, maybe you could agree to meet every 2-3 days where it is made clear what each team member will work on. See the contributions file where details of such meetings should be recorded.

**Project Deliverables**

This project must be implemented in the Java programming language. As can be seen from the deliverables below this project is not just an implementation project, it also involves design and documentation.

The following are the deliverables for the project:

* A document outlining the Class Responsibility Collaboration (CRC) cards. These can be created in a word document.
* A UML class diagram showing the relationships between the classes. This should include the class names (no details of attributes/constructors/methods required). It must show the relationships between the classes (class inheritance, interface inheritance, aggregation, dependency as appropriate) and utilising the notation presented in lecture slides (please refer to the slides). This UML diagram must be generated manually and drawn using a tool like draw.io.
* Documentation for the software generated using the javadoc utility and a help file describing briefly how to run the application.
* The source code for the system where each Java class/interface is stored in a separate file. Use packages to group files as appropriate.
* Any csv files that are required by the system should also be included. These files must be populated with data (e.g. sample data used to test the application).
* The use of a GitHub repository for source code version management.
* The completed contributions file which will be available on the Brightspace Content Project tab.

**Rules and Regulations**

* You must undertake this assignment on a group basis. Strictly groups with max size 4.
* You must add your details (irrespective of whether you have a group of four/a smaller group/no group) to the project group spreadsheet by **Sunday Nov. 3rd.** Failure to do so will result in zero marks being awarded for the project. Here is the link: [cs4013ProjectGroups2425.xlsx](https://ulcampus-my.sharepoint.com/:x:/g/personal/michael_english_ul_ie/ES72TCVV7FZKnjaEl0boYqcBprC5rJiORFkOz4JYiHMeoA?e=5789Qa) Please follow the instructions in this file. I will merge those of you that are not part of a group of four into groups on Monday November 4th.
* *You will be required to conduct an interview where each team member will be requested to give a code/project walkthrough and demo of the code that you have written at a selected time from the start of week 13 to the end of the examination period.* It is your responsibility to prepare adequately for such a demonstration and to be knowledgeable about the design, implementation and documentation submitted. Failure to satisfactorily demonstrate your project will result in an F grade for the assignment irrespective of the contents of the submitted documentation. An unsatisfactory demonstration is also likely to be considered plagiarism. It is also your responsibility to ensure that other students do not submit any part of your work as part of their project. This will also be deemed as plagiarism. The use of GenAI tools will also be considered plagiarism where not adequately acknowledged. Plagiarism will result in an F-grade being awarded for all in-term assessments and the University Code of Conduct will be applied.
* An F-grade is awarded if the contributions file is not included and complete.
* **Only one team member should submit the project**. However, that team member should communicate with the rest of the team to confirm submission. It is up to each team to manage submission correctly. The project must be submitted as a single compressed file to Brightspace by the stated deadline. Submission deadline is **12h00 Monday Week 13.** A **penalty** of 10% for each day thereafter will be imposed for late submissions. Submissions will not be accepted after 12h00 Thursday Week 13. Note: **There will be no further extensions to the dates above. Please manage your time accordingly.**
* Lab interviews may be undertaken before the deadline to establish your progress towards completing this assignment. Marks may be awarded for this purpose. Non-attendance at your specified lab may result in you forfeiting these marks.
* Accidental loss of work will not be accepted as an excuse.

**Version History:**

1. released on 30/10/2024

to make a pay role system for UL staff:

-make different pay scales

--can move up

-submit pay claim

-make professional category

-user types

- a cli

pay role system class

data

class

add employee

add pay claim form

Pay claim:

* Type of employee

Classes:

* Parttime: must submit pay claim form by the second Friday of the month
* The payroll system should generate pay slips for all full-time staff and hourly paid staff (with current claims) on the 25th day of each month

pay scale:

* job
* Scale Point
* Annual Rate

methods:

* array list for Lecture, Tutorial, Lab, Demo, Exam Invigilator, Exam Supervisor with points made up of pay scale object
* Array list for category.
* if point of PayScale is bigger than length of array list doesn't go up
* Full time staff Moved up each October
* Ability to promote to a different category (dependent of years served)

Pay slip:

* Health insurances
* Union fees
* Taxes

Employee

* Scale Point
* job
* Years in job

List:

Meeting 13:00-14:00

Attendance all

-looked over project

Day