Dear Mr. Moniaga,

I would like to propose the following program for my Culminating Task:

**Program Name:** SimpliPay

**Language:** The program’s base language is going to be C# and it is going to be taking advantage of Windows Forms to provide the user with a graphical user interface (GUI).

**Functionality:** This program is going to be a simplified payroll software. The user will be able to manage a list of employees as well as keep track of employee salaries/wages, record deductions, and generate pay stubs automatically. Since this is a simplified version, it will lack features that are commonly found in industry-grade payroll software. Features such as automatically generating a general ledger, automatically generating T4/T4A forms and emailing them to the Canadian Revenue Agency, and more.

**User Inputs:**

* When the user first launches the program, they will be prompted with a dialogue box that asks them if they want to upload an existing Excel file of employees or if they want to create a new one. If they choose to upload an existing database, then the program will loop over each employee in the database and instantiate them as an Employee class object. If the user decides to create a new one, the program will automatically generate a new Excel file with the correct headings and save it in the root directory.
* The user will be able to create employee profiles and each profile will have the following fields that would need to be filled out upon its creation — the user will also be able to alter the fields after profile creation. In order to accomplish this, I will be taking advantage of classes to create an Employee object template. Whenever a new employee is added, an Employee object will be created and populated. The employee profiles will be saved in an Excel file.

*Fields that have an asterix (\*) next to them will be mandatory for a valid profile and the remaining fields are optional.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| First Name\* | Last Name\* | SIN\* | Date of Birth\* | Date of Hire\* |
| Employee Number\* | Email \* | Phone Number | Address Line 1\* | Address Line 2 |
| City\* | Postal Code\* | Province \* | Middle Name | Pay Period\* |
| Type of Employee (salaried or hourly)\* | Pay Rate (either hourly rate or salary per pay period)\* | Vacation Entitlement Year (date of hire by default but can be changed)\* | Vacation Pay Percentage\* (will be automatically filled to either 4% or 6% but will be left editable for user) |  |

* The user will also be able to keep track of common deductions such as Canadian Pension Plan Deductions (CPP), Employee Insurance Deductions (EI), and tax (federal and provincial) deductions. The program will parse the deduction charts provided by the Canadian government and automatically assign the deduction amount that applies to the given employee. In addition, there will be a button that will allow the user to add custom deductions such as dental or life insurance.
* Each employee profile will have a “Financial YTD” (year-to-date) section. This section will keep track of the employee’s total earnings for the given tax period, and it will also provide a breakdown on how much of the total earnings are regular hour earnings, overtime earnings, vacation earnings, and commission earnings. Furthermore, it will provide a detailed breakdown of the employee’s total deductions for the tax period. The program will be utilizing graphs to provide a visual breakdown, which in extension, would improve user experience. The year-to-date values will be reset at the beginning of each tax period.
* When doing a payroll, the user will be able to select the pay cycle that they are doing the payroll for (i.e. biweekly, yearly, monthly). The user will also be required to manually enter the “from” date which will represent the beginning of the pay period. After doing so, the program will automatically calculate the “to” date which will represent the end of the pay period (*this field will be editable in case the user wants to choose a different date).* The program will also record the processing date as that will be useful for keeping a record of payrolls.
* After filling out the aforementioned fields, the user will be taken to a screen where all the eligible employees (employees with the given pay cycle) will be listed. Each non-salaried listed employee will have fields that would correspond to the type of earning (regular, bonus, etc.), number of hours worked, their pay rate (pre-filled but will be editable), and their total pay. For salaried employees, all the previous fields will be listed except the number of hours field. From here, the user must select which employees are being paid and enter the number of hours (if applicable) the employees have worked. There will also be a button next to each employee that the user can click on to add additional wages such as vacation, overtime, sick day, and more.
* **­­­**Once finished, the user can click on the “Review” button and they will be taken to a screen that will list all the employees being paid and their overall net pay. The user will have the ability to edit the fields as required, and once done so, the user can click the “Finalize” button to finish the payroll.

**Program Output**

* After a payroll has been completed, the program will automatically generate the paystubs for the respective employees. The paystubs will be stored in Excel and PDF files and the program will email the files to the respective employee. It will also store a copy of the paystubs in the user’s records so the user can keep track of employee records.
* The program will also output an Excel file that will summarize the payroll. This is done so the user can keep a working log of the payrolls performed. This Excel file will outline which employees got paid, their net pay, their number of hours worked, pay date, as well as various other details that a user would deem necessary to keep track of.

Best Regards,

Muhammad