The scenario:

My client is Mrs. Ng, and she currently manages the payroll for the staff within her household. After consulting with her **(refer to Appendix A, Consultation 1)** the system she uses is a set of spreadsheets on Microsoft Excel that contains all the data she needs to calculate payroll for each employee for each pay period.

The main problem of this system is interconnectedness between the sheets due to a lack of a master spreadsheet and no automatic updates between spreadsheets. For example, if an employee has increased their cash advance, or loan, it isn't automatically deducted from their salary during that pay period. Similarly, the length of the employee's tenure based on their employment history isn't automatically calculated for an employee. Furthermore, allocations towards social security funds aren't automatically calculated.

To solve these problems, she requires a single-user desktop application with a clean and clear graphical user interface (GUI) or design that can add, edit, update, delete, search, and sort the personal records and employment histories of both current and resigned employees. It should be able to generate payrolls for all current employees with the correct deductions based on their cash advances, days off, and corresponding allocations towards SSS, Philhealth, and Pag-Ibig Funds.

Rationale to the solution:

I will create a single-user desktop application using Java Swing for the Graphical User Interface (GUI) frontend, Java for handling the programming backend, and an embedded SQL jderby database for storing data. This will be done using Apache Netbeans, an Integrated Development Environment (IDE).

A different solution to addressing the client's problem would be Microsoft Access **(refer to Appendix B)**, a database solution Microsoft offers as a part of its 365 or Office subscription. However, it is aimed towards large-scale and long-term data storage and analysis applications for firms or teams of data analysts. As such, the app is resource intensive, requiring four gigabytes of disk space. Although it contains all the functionality necessary for the payroll application, it comes with unnecessary extra features. As a result, it has a high learning curve that the client would find difficult to overcome. It isn't tailored to the client's specific needs.

The aforementioned tools **(refer to Appendix B)** will be used to solve the client's problem due to their many pros which outweigh their cons. Java Swing will be used for the GUI due to its highly customizable design, platform independence, and its focus on desktop applications. Java will be used for the application's back-end because of its platform independence and my prior knowledge in it. For the database, I will use Apache Jderby due to its flexibility, reliability, and optimization of smaller Java applications. Finally, I will use Apache Netbeans as my IDE as it can support Java Swing, Java, and Apache Derby and allows them to

work together very easily. It also contains an extensive import library and has a simple yet effective layout.

Total Word Count: 460

Success Criteria:

1. The program will allow the client to log in and or sign up using an email and a password for information protection.
2. The program will allow the client to add, edit, update, delete, search, and sort the personal records of current and resigned employees into a database.
3. The program will automatically assign a unique ID number to each employee.
4. The program will allow the client to select and deselect employees who are currently active or who have resigned.
5. The program will allow the client to manually change deduction amounts and edit percentages allocated to social security of each employee.
6. The program will automatically generate a “suggested salary” based on the deductions and social security allocations.
7. The program will automatically track all changes to deductions, such as cash advance payments, number of off-days, or extra charges.
8. The program will automatically allocate a certain percentage of each employee’s income towards different social security programs.
9. The program can automatically generate a PDF salary report or summary.
10. The program will automatically send email reminders to the client near payday dates, such as the 15th, and end of each month.
11. The program will display proper error messages for all features of the application.