

Self Evaluation of Success Criteria:

Criteria:	Met?	Comments:
The program will allow the client to log in and or sign up using an email and a password for information protection.	Yes	<p>The register page allows the user to sign up with a new email, with proper email checks to verify that it's a valid email. It's then stored in a table alongside a password that is retyped once for confirmation, to make sure that the user knows their password.</p> <p>Then, by using a SELECT statement on the "users" table, where the email and password are stored, the entered email and password are compared. If they match, the user can now access the main page of the application (Salary Generator). If they don't match, the user is asked to reenter the email and password. This system allows for sufficient information protection from bad actors who may want to gain access to the sensitive personal information of employees.</p>
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.	Yes	<p>Using the built-in SQL syntax, such as INSERT, UPDATE, DELETE to add, edit (update and edit are the same thing in SQL), and delete records from the table, the user can directly interact with the tables within the database in a pre-programmed and pre-planned manner.</p> <p>Using Java's jTable built-in RowFilter and RegexFilter functionalities allow the personal records of current and resigned employees to be searched through quickly and efficiently. Additionally, SELECT queries with the SORT BY keyword along with bubble sorting through the ResultSet ensure that the jTable only displays the records in the sorted manner inputted by the user.</p>
The program will automatically assign a unique ID number to each employee.	Yes	<p>The maxfinder() function on the Employee Records page employs finds the maximum Employee ID in the database table, setting a global variable, this.max, to that value. Incrementing this by one guarantees that the Employee ID added to the database is unique.</p> <p>Whenever an INSERT query is performed, the above process is done to guarantee that the new entry has a unique Employee ID, and doesn't violate the primary key (unique value required) constraint set out by the database table.</p>
The program will allow the client to select and deselect employees who are currently active or	Yes	The employee records table has a boolean column called 'hired' that can only be one of two values: true or false. This decreases the size of the tables, and the size that each of the records contributes to the table.

who have resigned.		There is a checkbox within the GUI of the Employee Records page that allows the user to select it as only true or false. This, along with the update and insert buttons, allow the user to select and deselect employees who are currently active or who have resigned. The application also only shows the payrolls of employees who have are employed.
The program will allow the client to manually change deduction amounts and edit percentages allocated to social security of each employee.	Yes	<p>Within the Salary Generator main page, the client will be able to manually change wage, bonus, deduction, loan amounts, and social security allocations of each employee using the GUI. These values can only be updated, not added, as there must only be one salary record per employee.</p> <p>There is a combo box that automatically retrieves the preset values within the SSS, HDMF, and PhilHealth allocation pages. These are automatically saved to the database whenever the employee's salary values are updated.</p>
The program will automatically generate a "suggested salary" based on the deductions and social security allocations.	Yes	<p>Using the inputted wage, deduction, bonus, social security allocations, loan changes, and other amounts, a final wage is calculated using the following equation: $(\text{Final Wage} = \text{Wage} - \text{SSS_EE} - (\text{Wage} * \text{PH_EE}) - (\text{Wage} * \text{HDMF_EE}) - \text{Sub} - \text{Deduction} + \text{Bonus})$, found in (Criterion B - Form Design) and implemented in (Criterion C - Technique 9a).</p> <p>This final wage or "suggested salary" is automatically calculated anytime any of the above variables change, and is displayed in a textbox for the user automatically. This allows them to see how the final wage changes based on how the variables change.</p>
The program will automatically track all changes to deductions, such as cash advance payments, number of off-days, or extra charges.	Yes	<p>When the 'Save' button of the Salary Generator Page is clicked, all changes to Salary and Loans are saved to Salary History and Loan History, respectively. The final loan, which is saved to Loan History, is determined using the following equation: $(\text{Final Loan} = \text{Loan} + \text{Add} - \text{Sub})$, and Loan becomes Final Loan within the Salary_Allocations table.</p> <p>All changes to deductions, cash advance payments, and extra charges can be tracked using the bonus, deduction, loan sub, and loan add fields. Although the number of off-days wasn't tracked, it can still be tracked through changing the deductions.</p>
The program will automatically allocate a certain percentage of each employee's income towards different social security	Yes	<p>In the following Final Wage equation: $(\text{Final Wage} = \text{Wage} - \text{SSS_EE} - (\text{Wage} * \text{PH_EE}) - (\text{Wage} * \text{HDMF_EE}) - \text{Sub} - \text{Deduction} + \text{Bonus})$, a portion of the wage is allocated towards the social security benefits. This is done by subtracting the social security amount from the wage.</p> <p>This automatic allocation is extremely convenient for the user,</p>

programs.		and allows them to see how the final wage changes depending on the allocations.
The program can automatically generate a PDF salary report or summary.	Yes	<p>The Salary Generator Page is able to produce a PDF salary report or summary of all the employees currently employed by pressing the 'Print' button. By using the .print function of jTable along with the MessageFormat class to edit how the PDF report is finally printed, the user can generate a PDF salary or report.</p> <p>It also has a pop-up menu for where to save the file on the user's device, any extra print options the user may prefer (number of copies, page layout, etc.). It's extremely convenient.</p>
The program will automatically send email reminders to the client near payday dates, such as the 15th, and end of each month.	Yes	<p>The program automatically calculates if the salaries have been carried out for that half-month, by comparing the most recent date within the salary history table to the current date. If it determines that the salaries haven't been performed for the half-month (which is the same as the 'Save' button not being pressed), then an email is sent to the user's email once per day.</p> <p>This only works when the application is logged into, however, and doesn't work in the "background". However, given that it's a payroll system that handles employees, it's unlikely that the user will forget to do the payrolls for extended periods of time.</p>
The program will display proper error messages for all features of the application.	Yes	<p>The program has highly efficient error checking methods that are able to pinpoint the exact error of the user. This may include text within text fields that are too long, invalid inputs, such as numbers that are too large, or an invalid email, which can tell the user the exact mistake they made when inputting the email.</p> <p>This is extremely helpful for new users, as the specificity of the error-checking methods allow them to pinpoint the exact problem. There are also error checking methods on every page, making sure that the user is always guided when they make an error.</p>

Client Evaluation and Criteria:

(refer to Appendix A, Consultation 4)

Conclusion:

My client required a single-user desktop payroll application with a clean and clear GUI to process the salaries of around twenty employees. It would be a more efficient, tailored solution compared to Microsoft Excel, which is the current system she uses. The problem she encountered was the automatic addition and processing of data, which was often tedious when calculating salaries and accessing records across tables within Microsoft Excel.

The solution I've created to address this is a multi-window Java Swing Application that specifically implemented her needs. This included a login and sign up page that ensures security by requiring an email and password for logging into the system, a main Salary Generator page for processing everything related to salaries (with automatic calculation of values), an Employee Records page for adding the personal information of employees and changing if they're active or resigned, Social Security allocation pages for inputting preset values based on government-mandated allocation values, and history pages for tracking all past salary and loan amounts of hired employees.

In the interview, my client expressed her satisfaction with the system. She mentioned how it addressed her problem, how it's feature-complete with her requested functionality, and how its GUI is quite easy on the eyes, and how it can be improved in the future. The application should have a forgot password feature and better clustering of information (where the information in the Salary Generator Page isn't contained to just one page).

Recommendations for Future Development:

The client requested a forgot password feature. One feasible implementation would be to set a security question that the user can choose and determine the answer to upon registering their email. There could be a JComboBox with the selected security question, such as ("What was your first car", or "What is your pet's name?") and the expected answer. This can be saved in two extra columns in the "info" database table. If the security question is answered correctly, an email can be sent using javax.mail to the user's email address containing their password. This will ensure access to the application.

The client also requested improved clustering of data, as all information is concentrated in the Salary Allocations page. Perhaps multiple database tables can be created, each with a one-to-one employee ID foreign key constraint in order to handle the loans (loan, loan add, loan sub, final loan), wages (wage, bonus, deduction), social security allocations (SSS EE, SSS ER, PH EE, PH ER, HDMF EE, HDMF ER) separately. Each of these will require separate Swing windows to handle the updating of each of these values. This will make the separation of data a lot more clear and will decrease the learning curve for the user.

Word Count: 452