

Consultation 1 (Interview):

Date: April 28, 2024

Purpose: To gather information about the problem the client faces and the features they require the solution to contain.

(this transcription is part of the interview)

Me:

All right. Hi, what is your name? And what do you currently do?

Client:

I am [REDACTED], I manage our household accounts, including the payroll for the staff, for the employees in this household.

Me:

All right, and what is the task that you currently require help with?

Client:

I, I run the payroll once a month, once or twice a month, and I manage the different balances of the employees, their advances, I have to compute for their salaries, incentives, additional pay deductions, and all of that before the 15th and or the end of the month 30th, or 31st.

Me:

So how do you currently manage all of those things at the moment?

Client:

I use an Excel sheet, an Excel file with different tabs for each pay period. I also have a tab for their employment history, and also for the 13th month or 13th month pay for the year, for the year end, and then have several tabs for their cash advance balances. Basically, that's it.

Me:

What are sort of the problems with the current system, like anything that you'd like improved, anything that's sort of like making it inefficient?

Client:

Well, because sometimes there are updates that we've done with the current file, Current Pay Period, for example, let's say, I want to maintain a cash advance file, I'm maintaining a cash advance file for the employees. And for that pay period, it will reflect the deduction to be made for the employee salary, but then I need to maintain another file that would reflect those deductions, as well as those balances for that period, so sometimes, I end up not updating that. And then I have to go back to it, when I'm done. So there it's more of the updating and linking of those tables, or tabs. And then also, when there are changes in the whenever there are

changes in your salaries, let's say increases. I also maintain a file for their employment history, like when they started. What salary did they receive? When was the last increase? How much was the increase? Who were the employees that resigned? And there are some also who resigned and came back like that. So it's more of to compute tenure. There.

Me:

Okay, so sort of the main problem with the current system is like sort of interconnectedness, right? It leads to discrepancies because there's no like, master file. There's no automatic updates between the sheets, right?

Client:

Yes.

Me:

So you sort of need that kind of interconnectedness for those auto updates. So there's like no problem.

Client:

Yes.

Me:

Okay. That makes a lot of sense. And then the same thing also with calculating tenure. And what was it salary increases based on that?

Client:

Yes.

Me:

So that is not automatic. That's all manual?

Client:

Yes.

Me:

Okay. So from what you've been saying, you've already suggested quite a few features for the app. So what if you would mind like what features do you think it will require?

Client:

Well, it should have the basic employee maintenance feature, there should be an employee file master file, which contains their basic information. That would be nice. So it would be an immediate look up like, where's the provincial address of this person? When did he start like that? basic information about the employee

Me:

So searching, searching and sorting basically?

Client:

Your employee list, yes, an employee list. So the profile employee profile and the next there should be a cash advances table as well, where in all their advances and deductions are, are reflected. And then there should be a, also an employment history, like when they quit he started when he left, did he come back? Did he ever get a salary increase in when how much like that. And then, and then there's the periodic transaction generation of the payroll, for payroll. So that's where the payroll would be grabbing these information from, from the list of employees to let's say, their salaries, and then the deductions like SSS, Philhealth and HDMF or Pag-Ibig. And then the cash advances, it has to take note of that. And also the absences if any, or overtime, if any additional incentives, if any, or and maybe other deductions such as policies that we've set up, for example, if they their bills, phone bills consume more than the allocated amount, it's going to be billed to them. So that's where it all gets deducted in the payroll.

Me:

So you wouldn't mind for example, if deductions were inputted was it manually, but like all of the changes would be like reflected automatically, right?

Client:

Yes, it has to be that way. It has to be that way. Um, it varies on the computation, because for example, for the extra days, it will have to be computed per day. So if they came, if they work for an extra day, then it will be based on their birthday rate, based off the monthly pay rate. So it will be divided to X amount of days times how many days they came in for things like that.

Me:

Okay. All right. So that makes a lot of sense. So, all right. Um, okay. I would also like to suggest some features, like maybe a login system, would that be fine? To make sure that all of the information stays safe?

Client:

Yes, that's okay.

Me:

All right. Then, um, also for as for the kind of product, right? It - should it be like a web app, mobile app, single user, like...

Client:

single-user's fine. It's a very simple system, not more than 20 employees.

Me:

So like desktop applications, something like that?

Client:

Yes.

Me:

Okay. Okay. That makes a lot of sense. Then, um, all right. Are there any aesthetics that you'd like it to like conform to? You have a color scheme in mind or design? Or do you want me to just show you or does it not really matter?

Client:

It just needs to be pleasant and organized, easy to use, and not all over the place like that. So long as it looks cohesive, and not using multiple colors all at the same time, all loud colors, just something basic, not too boring, and, yeah, sorted out properly.

Me:

Okay, so clean and clear then?

Client:

Yes.

Me:

Okay. So do you have anything else you'd like to know about the application like ease of use future maintenance? And what do you think? Flexibility even?

Client:

The only maintenance that I foresee is probably, let's say I will, if there are new policies, incentives or regulations that we put in place, then that's when we will need maybe additional features to address that. But at the moment, I don't see anything. Of course, the system also has to take into account that the salaries are not fixed, and it could and it also changes so there should be a facility for that, like the change in rates for salary rate, benefit rate. All the things that involve values.

Me:

Okay. So, for now, I'd say you're just looking for that sort of basic salary calculator, then

Client:

Yeah payroll app.

Me:

Your payroll app just splits up all the values. Make sure to list all of your current and past employees also those who have worked before who have returned?

Client:

Yes.

Me:

Um, and how long they have worked here, sort of like in total, right?

Client:

Yes.

Me:

Okay, so I think that sounds good. All in all, I will get back to you as soon as I can with some of the features that you requested. And like just a final list of all of the features that you wanted, and all of the problems that needed fixing, and we can just discuss that on a later date, and just confirm everything.

Client:

Yes, we can do that. Um, I think it's more of the transactions that you have to take note off. Basically, you should have, because there will be new transactions from time to time. And one of these is like for, for example, the cash advance, if the employee wants to apply for cash advance, then we'd have a transaction window for that. To say, you know, you want to create a cash advance, and that will be added into the cash advance table, which would update the current balances. And that will be picked up in the next payroll period when it's generated.

Me:

So is there sort of like a timing window for them to apply for these? Is it date dependent?

Client:

No, they just apply when they need and it gets approved also, and it has to take note, let's say, how many payments would that be? And how much would the deductions be, it will be there as a guide, or it could add to their current cash advance, and it would just carry on the deductions would just carry on. Sometimes they have to the staff cannot pay in a certain period, they will request to skip payment for that period. So it remains editable as well. When we generate the the, for that payroll period of

Me:

So the payments, the continuous payments for that,

Client:

There are values that need to be editable, some values don't have to be editable. Some, some values have to be, especially for the deductions for the cash advance.

Me:

Then in that case, yeah, I'll just make it editable for everything. But like, I'll have it such that. It sort of recurring. So like you don't have to manually re input it every time. But it can be edited with ease.

Client:

Yes, yes. Is it just the amount? There's a standard amount? A set in the

Me:

Yeah, but it can be changed at any time? Yeah, you're fine with that? Yes. Okay. Okay. That makes sense. So, just to summarize, you need a sort of employee payroll app. With the details of your current past and resigned employees?

Client:

Yes.

Me:

Those who have returned and how long they've been tenured here, basic information, add, edit, delete, that sort of thing.

Client:

Yes so basic information includes the let's say the height, weight initial height, weight

Me:

Oh that also, okay

Client:

Yes, according to their employee file, we have a physical file for that.

Me:

Okay so,

Client:

Your bio data,

Me:

okay, so in that case, I'll just get that from you then. Then, the next major part is the payroll generators. So that's their salary based on their tenure, which you will input manually and just be recurring. Same with their cash advance payments, so the loans and how much is deducted every time, then their contribution. So their SSS, their Philhealth, their Pag-Ibig fund, all of that, all of that stuff, recurring payments again. And then the bonuses, as you mentioned, also,

Client:

Yes, and the number of days worked because, sometimes they have absences. So maybe for per employee we put in to how many days work, but it can have a suggested amount, let's see a full, let's say a whole period like that. And then it's also editable in case the person is absent. So that can be adjusted. So for example, it's 15 days, it will have a suggested amount of 15 days. If if there are no changes, then you can just continue.

Me:

Yeah, so it so like I said recurring.

Client:

Yeah, it's just recurring also, because this is a recurring task.

Me:

Okay, fine, fine, that works for me. Okay. And then yeah, as I've mentioned, login and password. So user authorization, of course, for security. And then, of course, error checking for everything. So are those all of the sort of main features that you want? Yes. Okay. All right.

Client:

Basically, we can add as we go, if there are other functions that we've missed.

Me:

Um, okay, because I'll be sending you the features. Soon, like, a finalized list. And once it's been finalized and confirmed, it can't be changed anymore.

Client:

Okay.

Me:

Because again, the system design is going to be, you know, it's a pretty challenging thing.

Client:

Yes.

Me:

And I don't think I'd be able to change the system architecture like on the fly like that,

Client:

Yes.

Me:

So yeah, I'll be sending you a follow up email. If ever. Sorry. I'll be I will be sending you a follow up email. And then we just confirm then.

Client:

Okay.

Me:

All right. Thank you so much for the interview.

Client:

Thank you for doing this app.

Me:

All right. Thank you.

Consultation 2 (Email)

Date: May 12, 2024

Purpose: To finalize the success criteria and functionality of the desktop application.

The screenshot shows an email conversation between two parties, with redacted names. The first email, sent at 12:22PM (2 hours ago), contains the subject "Payroll Application Success Criteria", the recipient's name, and a message body detailing success criteria. The second email, sent at 2:37PM (7 minutes ago), is a response from the recipient, acknowledging the features and providing a closing statement. The interface includes standard email controls like reply, forward, and delete.

Payroll Application Success Criteria [Delete](#) [Inbox](#)

12:22PM (2 hours ago) [Star](#) [Green checkmark](#) [Forward](#) [Details](#)

To: [REDACTED]
Subject: Payroll Application Success Criteria

Dear [REDACTED]

Good day! I hope this email finds you well.

The success criteria for the payroll application can be found below for your approval.

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 "Year-to-Date" report based on deductions and social security allocations.
7. The program will automatically track all charges 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.

Thank you very much!

Kind regards,

[REDACTED]

✓ Sender notified by
Mailtrack

2:37PM (7 minutes ago) [Star](#) [Green checkmark](#) [Forward](#) [Details](#)

To: [REDACTED]
Subject: Re: Payroll Application Success Criteria

Hallo [REDACTED]

Yes, these features will indeed address our current needs in the payroll system. Thank you!

Best regards,

[REDACTED]

...

Noted with thanks. This is noted, thank you. Thank you for your response.

Reply Forward

Consultation 3 (Email)

Date: Aug 6 - 7, 2024

Purpose: To finalize the form designs for the desktop application.

8/9/24, 11:41 AM

[REDACTED] Initial Form Design for Single-user Employee Management Application



Initial Form Design for Single-user Employee Management Application

5 messages

To [REDACTED]

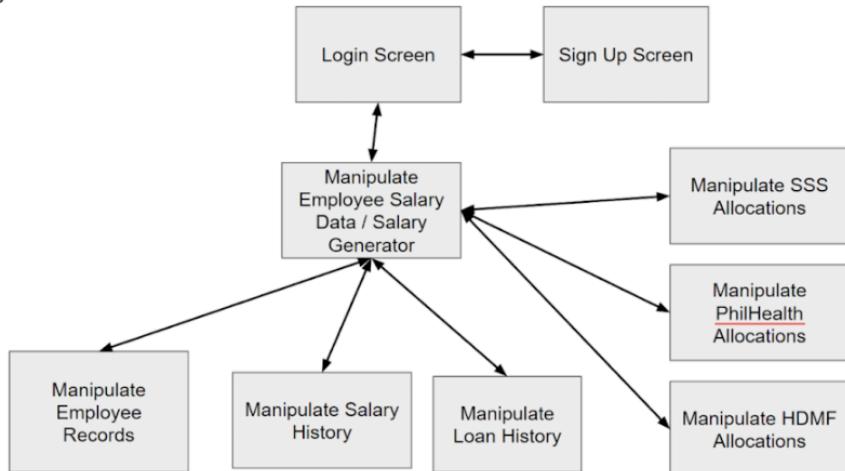
Tue, Aug 6, 2024 at 8:00 AM

Dear [REDACTED]

Good day! I hope this email finds you well.

Attached below are the initial form designs for the different pages of the application. If you have any questions or clarifications regarding the flow of the application or design of the pages, I'd be happy to help explain and clear up any confusion. The attached program structure will serve as a guide to understand the functions of the different pages.

Program Structure:



Login

Page:

Login Page

Components and Descriptions:

- JLabel:** Welcome Back
- JLabel:** Email:
- JLabel:** Password:
- JRadioButton:** Show Password
- JLabel:** Register
- JLabel:** Sign In
- JButton:** Takes the user to Sign Up Page
- JButton:** Takes the user to Manipulate Employee Salary Data / Salary Generator Page

Sign Up Page:

Sign Up Page

Components and Descriptions:

- JLabel:** Sign Up
- JLabel:** Email:
- JLabel:** Password:
- JLabel:** Show Password
- JLabel:** Re-enter Password:
- JLabel:** Show Password
- JRadioButton:** Show Password
- JLabel:** Back
- JLabel:** Register
- JButton:** Takes the user to Sign Up Page
- JButton:** Takes the user to Sign Up Page after validation.

Manipulate Employee Salary Data / Salary Generator

8/9/24, 11:41 AM

Initial Form Design for Single-user Employee Management Application

Manipulate Employee Records

This UML Class Diagram illustrates the components and their interactions on the Employee Records Page. The diagram shows various UI elements like JLabel, JTextField, JButton, JComboBox, and JTable, along with their associated JavaDoc-style descriptions.

- EmployeeRecordsPage** (represented by a blue rounded rectangle):
 - Attributes:** First Name, Middle Name, Last Name, Birth Date, Age (years), Address, Phone Number, Email, Height, Weight, Notes.
 - Components:**
 - Header: Edit Personal Records, Salary Generator, Hired / Not Hired, Sort by, Save, Reset.
 - Form Fields: First Name, Middle Name, Last Name, Birth Date, Age (years), Address, Phone Number, Email, Height, Weight, Notes.
 - Buttons: Clear Fields, Delete, Add, Update, Delete All, Delete.
 - Table: A large table showing employee details (First Name, Middle Name, Last Name, Birth Date, Age, Address, Phone Number, Email, Height, Weight, Notes).
 - Labels: JLabel pointing to various fields and buttons.
 - Validators: JTextBox (read-only, automatically generated), JTextBox (number only).
 - Search: JTextFields for First Name, Middle Name, Last Name, Birth Date, Age, Address, Phone Number, Email, Height, Weight, Notes.
 - Sort: JComboBox for sorting columns.
 - Reset: JButton for resetting table order.
- Annotations:**
 - JLabel: Takes user to Salary Generator page.
 - JTextFields: Takes search parameters input.
 - JComboBox: List of columns to search in.
 - JComboBox: List of columns to sort by.
 - JComboBox: Sort by ASC or DESC.
 - JButton: Confirms Sort Parameters.
 - JButton: Deletes all the records.
 - JButton: Deselects the item.
 - JButton: Adds a record to the table. Has required field validation. Automatically creates the SALARY_ALLOCATIONS with default values table.
 - JButton: Deletes the selected record from both the (USERID)INFO table and the SALARY_ALLOCATIONS table.
 - JButton: Updates the selected values in the table based on the textbox values. Contains error checking.

Manipulate Salary History

<https://mail.google.com/mail/u/1/?ik=6d9a14a275&view=pt&search=all&permthid=thread-a:r-6765859077180280078&simpl=msg-a:r7341527975222...> 3/16

8/9/24, 11:41 AM

Initial Form Design for Single-user Employee Management Application

This UML Class Diagram illustrates the components and their interactions on the Salary History page.

Components:

- Salary History**: The main container window.
- Salary Generator**: A button within the window.
- Search Bar**: Contains a search icon, input field, and dropdown.
- Sort By**: A section for sorting data.
- Table**: Displays salary history data with columns: EID, Date, Wage, SSS EE, SSS SR, PH EE, PH SR, HDMF EE, HDMF SR, Bonus, Deduction, Final Wage, Final ER.
- Buttons**: Clear Fields, Deselect, Add, Delete.
- Labels**: Various labels for fields like EID, Date, Wage, etc.
- Text Boxes**: Input fields for EID, Date, Wage, etc.
- Calendars**: A calendar component.
- Checkboxes**: Checkboxes for SSS EE, SSS SR, PH EE, PH SR, HDMF EE, HDMF SR.
- Spinners**: Spinners for Bonus and Deduction.
- Scrollers**: A vertical scroll bar for the table.
- Reset Buttons**: A button to reset table order.

Relationships:

- Labels (JLabel) point to various input fields and checkboxes.
- Text boxes (JTextBox) are associated with fields like EID, Date, Wage, etc.
- Checkboxes (JCheckBox) are associated with SSS EE, SSS SR, PH EE, PH SR, HDMF EE, HDMF SR.
- Spinners (JSpinner) are associated with Bonus and Deduction.
- Buttons (JButton) include Salary Generator, Sort by, and various table manipulation buttons.
- The Search Bar (JTextFields and JComboBoxes) interacts with the Table component.
- The Sort By section (JLabel, JComboBoxes, and JButton) interacts with the Table component.
- The Table component (JTable) is connected to all input fields and buttons, indicating bidirectional data flow.
- Scrollers (JScrollBar) are connected to the Table component.
- Reset buttons (JButton) are connected to the Table component.

Manipulate Loan History

Manipulate Loan History Page

Loan History Panel:

- EID: *JTextFields*
- Date: *JTextFields*
- Pay Prd.: *JTextFields*
- Loan: *JTextFields*
- Addition: *JTextFields*
- Subtraction: *JTextFields*
- Final Loan: *JTextFields*
- Buttons: Clear Fields, Deselect, Add, Delete

Salary Generator Panel:

- EID: *JTextFields*
- Pay Prd.: *JTextFields*
- Date: *JTextFields*
- Sort by: *JComboBox*
- Sort Asc: *JComboBox*
- Sort Desc: *JComboBox*
- Buttons: Sort, Reset

Annotations:

- JLabel:** takes user to Salary Generator page
- JTextFields:** takes search parameter input
- JComboBox:** List of columns to search in
- JComboBox:** List of columns to sort by
- JComboBox:** Sort by ASC or DESC
- JButton:** Confirms Sort Parameters
- JButton:** Resets Table order
- JTextFields:** readonly automatically generated
- JCalendar:**
- JTextFields:** number only
- Note:** Editing a value in Loan History doesn't retroactively change the loan value of a current loan
- JButton:** Clears all fields
- JButton:** Deselects the item
- JButton:** Deletes the selected record
- JButton:** Adds a record to the table. Has required field validation
- JScrollBar:**
- JTable:**

Manipulate SSS Allocations

<https://mail.google.com/mail/u/1/?ik=6d9a14a275&view=pt&search=all&permthid=thread-a:r-6765859077180280078&simpl=msg-a:r7341527975222...> 4/16

Manipulate SSS Allocations Page

This diagram shows the initial form design for the Manipulate SSS Allocations Page. It features a header with a search bar and sort options, followed by a table for managing SSS allocations. The table includes columns for BID, Lower Bound, Upper Bound, Employer's Share (ER), and Employee's Share (EE). Below the table are several buttons for managing records: Clear Fields, Deselect, Add, Update, Delete All, and Delete. A detailed callout box provides descriptions for each button.

- BID:** JTextField: takes user to Salary Generator page
- Bracket lower bound:** JTextField: takes search parameter input
- Bracket upper bound:** JTextField: takes search parameter input
- Employer's Share (ER):** JTextField: numbers only
- Employee's Share (EE):** JTextField: numbers only
- Sort by:** JComboBox: List of columns to search in
- Sort by:** JComboBox: List of columns to sort by
- Sort by:** JComboBox: Sort by ASC or DESC
- Sort:** JButton: Confirm Sort Parameters
- Reset:** JButton: Resets Table order
- Clear Fields:** JButton: Clears all fields
- Deselect:** JButton: Deletes all the records.
- Add:** JButton: Adds a record to the table. Has required field validation.
- Update:** JButton: Updates the bracket's values in the table based on the textbox values. Contains error checking.
- Delete All:** JButton: Deletes the item
- Delete:** JButton: Deletes the selected record

Manipulate PhilHealth Allocations

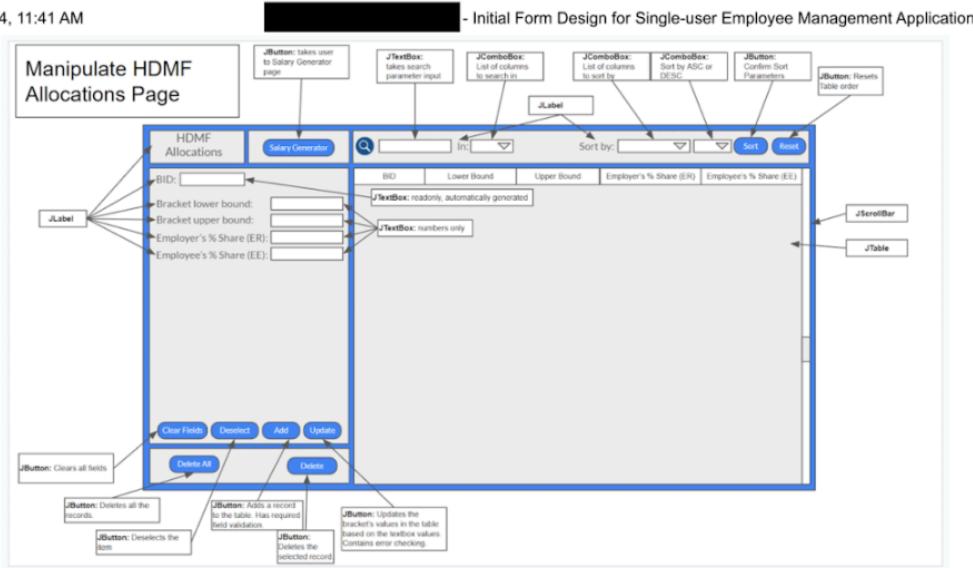
Manipulate PhilHealth Allocations Page

This diagram shows the initial form design for the Manipulate PhilHealth Allocations Page. It has a similar structure to the SSS allocations page, with a header, a table for managing PhilHealth allocations, and a set of buttons below for clearing fields, deselecting items, adding new records, updating existing ones, and deleting all or specific records. A detailed callout box describes the functions of each button.

- BID:** JTextField: takes user to Salary Generator page
- Bracket lower bound:** JTextField: takes search parameter input
- Bracket upper bound:** JTextField: takes search parameter input
- Employer's % Share (ER):** JTextField: numbers only
- Employee's % Share (EE):** JTextField: numbers only
- Sort by:** JComboBox: List of columns to search in
- Sort by:** JComboBox: List of columns to sort by
- Sort by:** JComboBox: Sort by ASC or DESC
- Sort:** JButton: Confirm Sort Parameters
- Reset:** JButton: Resets Table order
- Clear Fields:** JButton: Clears all fields
- Deselect:** JButton: Deletes all the records.
- Add:** JButton: Adds a record to the table. Has required field validation.
- Update:** JButton: Updates the bracket's values in the table based on the textbox values. Contains error checking.
- Delete All:** JButton: Deletes the item
- Delete:** JButton: Deletes the selected record

Manipulate HDMF Allocations

8/9/24, 11:41 AM



Kind regards,

Sent with Mailsuite · Unsubscribe

Tue, Aug 6, 2024 at 3:14 PM

To: [REDACTED]

Dear [REDACTED]

Thank you for sending me the initial draft of your systems design.

With regards to the forms, it would be helpful if some of the labels are organized according to function and topic. Likewise, each page or window should differ in design from each other in order for the user to associate a particular function with that window. We want to avoid the user mistaking one from the other as he/she navigates around the system.

Additionally, could you add an export to pdf feature on the following pages: Salary Generator, Salary History, and Loan History?

Sincerely,

[Quoted text hidden]

Wed, Aug 7, 2024 at 12:34 PM

Dear [REDACTED]

Good day! I have revised the form designs to your specifications.

Will these designs do?

Login Page:

<https://mail.google.com/mail/u/1/?ik=6d9a14a275&view=pt&search=all&permthid=thread-a:r-6765859077180280078&simpl=msg-a:r7341527975222...> 6/16

Login Page

Components:

- JLabel:** Welcome Back
- JLabel:** Email:
- JTextField:** Allows the user to input the email.
- JLabel:** Password:
- JPasswordField:** Allows the user to input password. Characters are displayed as "•".
- JRadioButton:** To toggle showing and not showing password.
- JLabel:** Register
- JLabel:** Sign In
- JButton:** Takes the user to Sign Up Page
- JButton:** Takes the user to Manipulate Employee Salary Data / Salary Generator Page

Sign Up Page:

Components:

- JLabel:** Sign Up
- JLabel:** Email:
- JTextField:** Allows the user to input the email.
- JLabel:** Password:
- JPasswordField:** Allows the user to input password. Characters are displayed as "•".
- JRadioButton:** To toggle showing and not showing password.
- JLabel:** Re-enter Password:
- JPasswordField:** Allows the user to reenter password. Characters are displayed as "•".
- JRadioButton:** To toggle showing and not showing password.
- JLabel:** Back
- JLabel:** Register
- JButton:** Takes the user to Sign Up Page
- JButton:** Takes the user to Sign Up Page after validation.

Manipulate Employee Salary Data / Salary Generator

8/9/24, 11:41 AM

Initial Form Design for Single-user Employee Management Application

Manipulate Employee Records

This diagram illustrates the components and their interactions on the 'Manipulate Employee Records Page'. It shows various UI elements like labels, text boxes, buttons, and a calendar, along with their associated Java component types and descriptions.

- Labels:**
 - JLabel: All text in this panel is a JLabel.
- Text Boxes:**
 - JTextBox: readonly automatically generated
 - JTextBox: number only
 - JTextBox: Deletes all the records
 - JTextBox: Deselects the item
 - JTextBox: takes user to Salary Generator
 - JTextBox: takes search parameter input
 - JTextBox: Sort by ASC or DESC
 - JTextBox: Resets Table order
- Buttons:**
 - JButton: Clears all fields
 - JButton: Deletes all the records
 - JButton: Adds a record to the table. Has required field validation. Automatically creates a new row in SALARY_ALLOCATION LOCATIONS with default values.
 - JButton: Deletes the selected record from both the [USERINFO]INFO table and SALARY_ALLOCATION table.
 - JButton: Updates the selected row's values in the table based on the textbox values. Contains error checking.
 - JButton: Confirm Sort Parameters
 - JButton: Salary Generator
 - JButton: Radio button: To select if need or not
 - JButton: JRadioButton: To show only hired employees
 - JButton: JCheckBox: List of columns to sort in
 - JButton: JCheckBox: List of columns to sort by
 - JButton: Sort
 - JButton: Search
 - JButton: In
 - JButton: Out
 - JButton: Hired
 - JButton: Sort by
 - JButton: Delete All
 - JButton: Delete
 - JButton: Clear Fields
 - JButton: Deselect
 - JButton: Add
 - JButton: Update
- Calendars:**
 - JCalendar: Birthdate
- Tables:**
 - JTable: Displays employee information with columns: ID, Hired, First Name, Middle Name, Last Name, Birthdate, Age, Address, Phone Number, Email, Height, Weight, Notes.
- Other:**
 - JScrollBar: Associated with the JTable.

Manipulate Salary History

<https://mail.google.com/mail/u/1/?ik=6d9a14a275&view=pt&search=all&permthid=thread-a:r-6765859077180280078&simpl=msg-a:r7341527975222...> 8/16

8/9/24, 11:41 AM

Initial Form Design for Single-user Employee Management Application

Manipulate Salary History Page

This form allows users to search, sort, and print salary history data. It includes fields for EID, Date, Wage, SSS ER, SSS EH, PH EH, PH ER, HDMF EH, HDMF ER, Bonus, Deduction, Final Wage, and Final ER. A note states: "Note: Adding a value in Loan History doesn't retroactively change the loan value of a current loan." Buttons include Sort, Reset, PRINT, and various manipulate buttons (Clear Fields, Delete, Add) for the right panel.

Salary Generator

This panel contains fields for Pay Prd., Date, SSS ER, PH ER, HDMF EH, Bonus, Deduction, Final Wage, and Final ER. It includes a JCalendar for Date and a note: "All text in this panel is a JLabel". Buttons include Deselect, Clear Fields, Delete, Add, and a JButton that prints the table to a PDF.

Salary History

This panel contains fields for EID, Pay Prd., Date, SSS ER, PH ER, HDMF EH, Bonus, Deduction, Final Wage, and Final ER. It includes a JCalendar for Date and a note: "JTextField: readonly, automatically generated". Buttons include Deselect, Clear Fields, Delete, Add, and a JButton that prints the table to a PDF.

Manipulate Loan History

Manipulate Loan History Page

This form allows users to search, sort, and print loan history data. It includes fields for EID, Pay Prd., Date, Loan, Addition, Subtraction, and Final Loan. A note states: "Note: Adding a value in Loan History doesn't retroactively change the loan value of a current loan." Buttons include Sort, Reset, PRINT, and various manipulate buttons (Clear Fields, Delete, Add) for the right panel.

Loan History

This panel contains fields for Pay Prd., Date, Loan, Addition, Subtraction, and Final Loan. It includes a JCalendar for Date and a note: "JTextField: readonly, automatically generated". Buttons include Deselect, Clear Fields, Delete, Add, and a JButton that prints the table to a PDF.

Manipulate SSS Allocations

<https://mail.google.com/mail/u/1/?ik=6d9a14a275&view=pt&search=all&permthid=thread-a:r-6765859077180280078&simpl=msg-a:r7341527975222...> 9/16

8/9/24, 11:41 AM

Initial Form Design for Single-user Employee Management Application

Manipulate SSS Allocations Page

Manipulate PhilHealth Allocations Page

Manipulate HDMF Allocations

<https://mail.google.com/mail/u/1/?ik=6d9a14a275&view=pt&search=all&permthid=thread-a:r-6765859077180280078&simpl=msg-a:r734152797522...> 10/16

8/9/24, 11:41 AM

Initial Form Design for Single-user Employee Management Application

The diagram illustrates the initial form design for a single-user employee management application. The main window title is "Manipulate HDMF Allocations Page". It features a header with a "Salary Generator" button and a search bar. Below the header is a table with columns: BID, Lower Bound, Upper Bound, Employer's % Share (ER), and Employee's % Share (EL). The table includes sorting and filtering options. On the left side, there is a panel titled "HDMF Allocations" containing fields for BID, Bracket lower bound, Bracket upper bound, Employer's % Share (ER), and Employee's % Share (EE). It also has buttons for Clear Fields, Delete, Add, and Update. A central logo features a house with hands holding it, labeled "Pag-IBIG FUND". At the bottom, there are buttons for Delete All, Delete, and Delete. A scroll bar is located on the right side of the table area.

Thank you very much!

Kind regards,

[Quoted text hidden]

To: [REDACTED]

Wed, Aug 7, 2024 at 2:48 PM

Dear [REDACTED]

Good day! Please disregard my previous email as there were some elements missing. Here is the revised version:

Login Page:

<https://mail.google.com/mail/u/1/?ik=6d9a14a275&view=pt&search=all&permthid=thread-a:r-6765859077180280078&simpl=msg-a:r734152797522...> 11/16

Login Page

Initial Form Design for Single-user Employee Management Application

Login Page

Welcome Back

Email:

Password:

Show Password

Register

Sign In

JLabel

JTextField: Allows the user to input the email.

JPasswordField: Allows the user to input password. Characters are displayed as "•".

JRadioButton: To toggle showing and not showing password.

JLabel

JButton: Takes the user to Sign Up Page

JButton: Takes the user to Manipulate Employee Salary Data / Salary Generator Page

Sign Up Page:

Sign Up Page

Sign Up

Email:

Password:

Show Password

Re-enter Password:

Show Password

Back

Register

JLabel

JTextField: Allows the user to input the email.

JPasswordField: Allows the user to input password. Characters are displayed as "•".

JRadioButton: To toggle showing and not showing password.

JLabel

JPasswordField: Allows the user to reenter password. Characters are displayed as "•".

JRadioButton: To toggle showing and not showing password.

JLabel

JButton: Takes the user to Sign Up Page

JButton: Takes the user to Sign Up Page after validation.

Manipulate Employee Salary Data / Salary Generator

Initial Form Design for Single-user Employee Management Application

Salary Page

This form allows users to manage employee salaries. It includes fields for SSS Brackets, PhilHealth Brackets, HDMF Brackets, and various wage components like Wage, SSS_EE, SSS_ER, PH_EE, PH_ER, HDMF_EE, HDMF_ER, Loan, Add, Sub, Bonus, Deduction, and Final Wage. A table displays Active Employees with columns for EID, Wage, SSS_EE, SSS_ER, PH_EE, PH_ER, HDMF_EE, HDMF_ER, Loan, Add, Sub, Bonus, Deduction, and Final Wage. Buttons include Edit SSS, Edit PH, Log Out, Save, PRINT, Clear Fields, Deselect, Update, Salary History, Loan History, and Edit Employee List.

Manipulate Employee Records

This form allows users to edit personal records for employees. It includes fields for First Name, Middle Name, Last Name, Birthdate, Address, Phone Number, and optional fields like Email, Height, Weight, and Notes. A table lists employees with columns for EID, First Name, Middle Name, Last Name, Birthdate, Age, Address, Phone Number, Email, Height, Weight, and Notes. Buttons include Edit Personal Records, Salary Generator, Hired/Not Hired, Search, Sort, Reset, Clear Fields, Deselect, Add, Update, Delete All, Delete, and Delete Item.

Manipulate Employee Records

Manipulate Employee Records Page

This form allows users to edit personal records for employees. It includes fields for First Name, Middle Name, Last Name, Birthdate, Address, Phone Number, and optional fields like Email, Height, Weight, and Notes. A table lists employees with columns for EID, First Name, Middle Name, Last Name, Birthdate, Age, Address, Phone Number, Email, Height, Weight, and Notes. Buttons include Edit Personal Records, Salary Generator, Hired/Not Hired, Search, Sort, Reset, Clear Fields, Deselect, Add, Update, Delete All, Delete, and Delete Item.

Manipulate Salary History

8/9/24, 11:41 AM

Initial Form Design for Single-user Employee Management Application

Manipulate Salary History Page

This form allows users to search, sort, and print salary history data. It includes fields for EID, Date, Wage, SSS ER, PH ER, HDMF ER, Bonus, Deduction, Final Wage, and Final ER. A note states: "Note: Adding a value in Loan History doesn't retroactively change the loan value of a current loan." Buttons include Sort, Reset, PRINT, and various actions for Salary Generator and Salary History panels.

Salary Generator Panel:

- EID: Pay Prd.: [JTextField]
- Date: [JTextField]
- SSS ER: [JTextField]
- PH ER: [JTextField]
- HDMF ER: [JTextField]
- Bonus: [JTextField]
- Deduction: [JTextField]
- Final Wage: [JTextField]
- Final ER: [JTextField]

Salary History Panel:

- JLabel: All text in this panel is a JLabel.
- JButton: Deletes the item.
- JButton: Clears all fields.
- JButton: Adds a record to the table. Has required field validation.
- JButton: Deletes the selected record.

Manipulate Loan History

Manipulate Loan History Page

This form allows users to search, sort, and print loan history data. It includes fields for EID, Pay Prd., Date, Loan, Addition, Subtraction, and Final Loan. A note states: "Note: Adding a value in Loan History doesn't retroactively change the loan value of a current loan." Buttons include Sort, Reset, PRINT, and various actions for Salary Generator and Loan History panels.

Loan History Panel:

- EID: Pay Prd.: [JTextField]
- Date: [JTextField]
- Loan: [JTextField]
- Addition: [JTextField]
- Subtraction: [JTextField]
- Final Loan: [JTextField]

Manipulate SSS Allocations

<https://mail.google.com/mail/u/1/?ik=6d9a14a275&view=pt&search=all&permthid=thread-a:r-6765859077180280078&simpl=msg-a:r734152797522...> 14/16

Initial Form Design for Single-user Employee Management Application

Manipulate SSS Allocations Page

SSS Allocations					Salary Generator
BID:	Bracket lower bound:	Lower Bound	Upper Bound	Employer's Share (ER):	Employee's Share (EE):
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Clear Fields"/>	<input type="button" value="Deselect"/>	<input type="button" value="Add"/>	<input type="button" value="Update"/>	<input type="button" value="Delete All"/> <input type="button" value="Delete"/>	
					

Manipulate PhilHealth Allocations Page

PhilHealth Allocations					Salary Generator
BID:	Bracket lower bound:	Lower Bound	Upper Bound	Employer's % Share (ER):	Employee's % Share (EE):
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Clear Fields"/>	<input type="button" value="Deselect"/>	<input type="button" value="Add"/>	<input type="button" value="Update"/>	<input type="button" value="Delete All"/> <input type="button" value="Delete"/>	
					

Manipulate HDMF Allocations Page

HDMF Allocations					Salary Generator
BID:	Bracket lower bound:	Lower Bound	Upper Bound	Employer's % Share (ER):	Employee's % Share (EE):
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="button" value="Clear Fields"/>	<input type="button" value="Deselect"/>	<input type="button" value="Add"/>	<input type="button" value="Update"/>	<input type="button" value="Delete All"/> <input type="button" value="Delete"/>	
					

8/9/24, 11:41 AM

Initial Form Design for Single-user Employee Management Application

Manipulate HDMF Allocations Page

This diagram illustrates the initial form design for a single-user employee management application, specifically for manipulating HDMF allocations. The form includes a title bar, a toolbar with a 'Salary Generator' button, and a main content area divided into two sections: 'HDMF Allocations' and 'Salary Generator'.

HDMF Allocations Section:

- BID:** A text input field.
- Bracket lower bound:** A dropdown menu.
- Bracket upper bound:** A dropdown menu.
- Employer's % Share (ER):** A text input field.
- Employee's % Share (EE):** A text input field.
- Buttons:** 'Clear Fields', 'Deselect', 'Add', 'Update', 'DeleteAll', and 'Delete'.
- Records:** A table with columns: BID, Lower Bound, Upper Bound, Employer's % Share (ER), and Employee's % Share (EE). It includes a search bar ('Int'), a sort dropdown ('Sort by'), and buttons ('Save', 'Reset').
- Icons:** A logo for 'Pag IBIG FUND'.

Salary Generator Section:

- JLabel:** Labels for various fields.
- JTextFields:** Input fields for 'Lower Bound', 'Upper Bound', 'Employer's % Share (ER)', and 'Employee's % Share (EE)'.
- JComboBoxes:** Selection dropdowns for 'List of columns to search in', 'Sort by ASC or DESC', and 'Sort by'.
- JButtons:** Buttons for 'Prints first Parameters' and 'Resets Table order'.
- JTable:** A table component.
- JScrollBar:** A vertical scroll bar.

Thank you very much!

Kind regards,

[Quoted text hidden]

Wed, Aug 7, 2024 at 10:38 PM

To: [REDACTED]

Dear [REDACTED]

Thank you for these updates! The new forms have reflected the necessary functionalities of our payroll system. I look forward to using the finished product.

Sincerely,

[Quoted text hidden]

Consultation 4 (Evaluation):

Date: Feb 17, 2025

Person: Mrs. Ng

Purpose: Answers to a Google form and a transcript of the fourth consultation with Mrs. Ng to evaluate the application.

General Feedback:

Criteria	Met?	Comments
Does this application meet your needs and expectations?	Yes	The functions were all met. The system was able to take into account the list of all employees. The benefits table keeps track of all salary brackets as defined by the government. The add, update, and delete buttons allow me to change any of the records within the database. I also receive emails depending on when I need to fill in the salaries, and I'm also able to print salary reports.
Was your feedback taken into account by the student?	Yes	My feedback was taken into account by the student. During each of our interactions, the student made sure to account for my specific needs, my suggested modifications, functionality issues, and working within the limitations of the programming language and skillset. The student made sure to implement these well, such that I deem the functionality complete and acceptable for my needs.
Was the app intuitive to navigate and easy to use?	Partially	The functionality worked well. There were no major crashes or errors that occurred while I used it. All the buttons were thoroughly tested, with some showing helpful error messages that told me what I did wrong or explained the character limits of specific text fields.
Do you see yourself using this application in the future?	Yes	Yes, because all of the functions are there. Although it may take a while for myself to get used to navigating through all the functions, I believe I'll be able to use all the functionality included to its fullest extent, streamlining my payroll process. Additionally, I'm sure that with a few more GUI and user experience tweaks, I see myself using the app regularly to carry out bi-monthly payrolls.
Was the design of the app visually appealing?	Yes	The colors used are comfortable to the eyes and quite neutral. It fits the color scheme that I envisioned, as I wanted it to be quite simple and nondescript. Additionally, none of the colors are too jarring to the eyes. The contrast between the panels, buttons, textboxes, and more, make it clear where one function ends and another begins.
Did you encounter problems when using	No	The functionality worked well. There were no major crashes or errors that occurred while I used it. All the buttons were thoroughly tested, with some showing helpful error messages that told me what I did wrong or explained the character limits of specific text

the application?		fields.
What improvements could be implemented?	Yes	<p>Some improvements that can be implemented, in addition to what I've mentioned beforehand, can include placing more focus on the user experience. For me, it was quite hard to use the application at first, as the Salary Generator main page has too much functionality on it. I was confused when I first saw all the buttons. Additionally, given that it's the main menu of the app, it shouldn't have too much information on it. It should be distilled down into something simpler, and the more technical adding, update, and modify functions for different specific categories, such as loans, bonuses, etc. should be moved to a different page.</p> <p>Another improvement could be auto-assignment of social security benefits to an employee depending on their salary level. Currently, I have to manually assign the social security benefits to every employee. It's highly repetitive and a waste of time. If the system were to do it automatically, it would save much more time.</p> <p>There is also no "forgot password" feature. If I forget my password, I don't want to lose access to the payrolls.</p>

Feedback on 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	I was able to sign up using my email and input a password using the login and register screens of the application. As with any email and password system, I was denied access if I entered the wrong email or password.
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	The payroll system allowed me to add, edit, update, delete, search, and sort through the personal records of all the employees I've added into the database. They also worked together with one another. I was able to update or delete certain records when I had a search term in the search bar or sorted it by a certain column.
The program will automatically assign a unique ID number to each employee.	Yes	The system assigned a unique employee ID number to every employee I added to the system. In fact, this employee ID number was also used on the salary generator page, linking that salary to the employee.
The program will allow the client to select and deselect employees	Yes	There was a checkbox on the employee list page that allowed me to easily select and deselect employees who are hired or inactive. Additionally, only the ones who were

who are currently active or who have resigned.		hired appeared on the salary generator page. I could then edit their salaries and edit specific allocations towards loans or social security.
The program will allow the client to manually change deduction amounts and edit percentages allocated to social security of each employee.	Yes	I was able to manually change the SSS, HDMF, and PhilHealth social security amounts for each employee using the values I added on their respective pages. This included a helpful upper and lower bracket with an attached employee and employer share for social security. Additionally, I was able to add, edit, and update the brackets on their respective pages.
The program will automatically generate a “suggested salary” based on the deductions and social security allocations.	Partially	This is only partially met because the suggested salary wasn't explicitly specified, but the final wage was generated based on the social security allocations, the bonus and deduction amounts, and loan amounts that I entered. A suggested salary was technically generated, but it wasn't called that. I still put in the appropriate wage, and it was factored into the final wage calculation.
The program will automatically track all changes to deductions, such as cash advance payments, number of off-days, or extra charges.	Yes	The save button on the Salary Generator page tracked all these changes. The changes to the bonus and deduction columns were saved to the Salary History page. The same is tracked for the loans in the Loan History page. By filtering for a certain employee ID, I can track how a loan, bonus, or deduction changed over time.
The program will automatically allocate a certain percentage of each employee's income towards different social security programs.	Yes	The final wage calculations factors in the social security allocations I've placed in SSS, HDMF, and PhilHealth. This includes removing a certain amount or percentage from the wage as the employer's share or employee's share.
The program can automatically generate a PDF salary report or summary.	Yes	The print function allowed me to export the contents of the table on the Salary Generator page as a pdf. I was able to specify the location in which to save the pdf. The formatting of the pdf was a bit simple, however, as it just copied over the contents of the table. All the information for a complete report is there, but it's not formatted nor shown in a very user-friendly way.
The program will automatically send email reminders to the client near payday dates, such as the 15th, and end of each	Yes	I received an email once per day when I was didn't accomplish the payroll for the pay period. It's quite useful, as it reminds me to save the values after I export the salary summary to prepare for the next pay period. It only works however, when I log into the application. So, there's a possibility I may forget if it doesn't run "in the

month.		background", even while the application itself isn't running.
The program will display proper error messages for all features of the application.	Yes	All the features of the application had proper error handling. Whenever any text values were deemed invalid, or an operation didn't execute properly, there was a helpful error message that pointed me in the right direction. They described the error that I was experiencing concisely.

(this transcription is part of the interview)

Me:

Alright, I just like to have this sort of final consultation with you about the application now that you've reviewed it and I guess used it for a little a little while. Now, I want to start off by asking you, does this application meet your needs and expectations?

Client:

Yes, it does meet my needs and expectations, because all the functions were there and it was able to take into account the list of all employees and the benefits table to keep track of all the salary brackets as defined by the Philippine government. And there are different functionalities that I specified earlier when the system was being created and there were requirements gathering, which is specifically the add, modify and delete functions for all the records in that database.

Me:

All right? Um, now talking about, for example, our interactions with one another. Do you believe that your feedback was taken into account by me, and was I able to sort of take that into consideration when designing the app and throughout the process of talking to you, getting to know your needs, etc.?

Client:

Yes, definitely, definitely, because there was a sample document that was used as a design for the project, as a basis for the design, all the information gathering, and the needs analysis. And there was a communication back and forth with regards to feedback and if the functionalities were met, all of that was done in the course of this development.

Me:

Alright. Thank you. Now, talking about the app and its design, do you believe that it was intuitive to navigate and easy to use?

Client:

With regards to the ease of use, I would say there could be improvements to be made, specifically with the design.

Me:

All right. Sorry about that. I'll just continue.

Client:

With regards to the design there, the dashboard could be made simpler, because right now, all the information is on the main page, on the main menu, and maybe clustering of information could be done so that it's easier for the user to navigate at one glance. So let's say for the generation of reports or queries or the master files and the transactional tables, they could all be segregated into those categories. Yeah.

Me:

All right. Um, now looking ahead, uh, having used the application so far as part of the, you know, testing period, do you see yourself using this in the future?

Client:

Yes, because this was, this program was mainly created from my needs and requirements for generating a payroll bi-monthly, although it may take a while for me to get used to navigating through all the functions, because that could use some improvement. But I think with enough time, I will be able to go around it very easily.

Me:

Okay, um, all right. Now, from a subjective standpoint, would you say the design, the design of the app, was visually appealing?

Client:

It is visually appealing, because the colors used are quite easy on the eyes, and they're neutral, it's a color that is comfortable and the size of the text are pretty well-designed. So in the contrast between the panels, the buttons, text boxes, they're quite clear with regards to the functions.

Me:

Okay, now, um, did you encounter any major problems when using the application?

Client:

Um, there were no major crashes or errors that occur, that occurred while I was using it. There were no major problems so far,

Me:

So far...

Client:

Hmm. And whenever there are functions that should not proceed, there are also error messages that pop up for that.

Me:

So, so like, for example, like some of the data validation, that kind of stuff for like, character limits.

Client:

Yes, they were in place.

Me:

Yeah, yeah. Alright. So finally, what other improvements do you think could be implemented for the system?

Client:

Well, what I noticed that was lacking is the automatic assignment of the Social Security benefits or the bracket to the employee. So since the table was defined and brackets were, brackets were defined and the contributions, corresponding contributions were also there on the table. I think it should automatically pull up those values when the employee is tagged that particular bracket, salary bracket. And there's also no forgot password feature in this case, because there's login and there's register, but we all know that it this can very well happen to any one of us, and in case a user does forget the password, there is no feature for that. So those are improvements that we could work on.

Me:

Okay, okay, okay, okay, all right. I think that basically it when it comes to what is it when it comes to the application. Thank you so much for what was it consulting with me and like allowing me to build this app for you to address your needs, because it's been, I guess, quite an experience for me also, to build an app from scratch, collaborate with a client and, like, really, I guess, deliver the final product. So with that, yeah, thank you so much for all of your feedback. Your helpful was it your helpful consultations and yeah, thank you.

Client:

You're welcome.