Vanier College

Deliverable 5

  Client: Opeq, Simon

System Development Section 01

Team Orange:

Jiamin Yuan

Dinal Patel

Craig Justin Balibalos

Alihan Djamankulov

Ibrahim Awad

Submission Date:

I, (Jiamin Yuan), student ID# (2055624), certify that I have contributed to this deliverable, (signature – this can be a scanned image, or an electronic signature).



I, (Dinal Patel), student ID# (2042827), certify that I have contributed to this deliverable, (signature – this can be a scanned image, or an electronic signature).



I, (Craig Justin Balibalos), student ID# (2069192), certify that I have contributed to this deliverable, (signature – this can be a scanned image, or an electronic signature).



I, (Alihan Djamankulov), student ID# (2033628), certify that I have contributed to this deliverable, (signature – this can be a scanned image, or an electronic signature).



I, (Ibrahim Awad), student ID# (2032818), certify that I have contributed to this deliverable, (signature – this can be a scanned image, or an electronic signature).



**Table of contents**

Statement ------------------------------------------------------ **4**

Executive Overview ------------------------------------------------------ **4**

Summary Desc. of Client -------------------------------------------------- **4**

Business Problem ------------------------------------------------------ **4**

Summary of Project Narrative -------------------------------------------- **5**

10 Usability Guidelines ------------------------------------------------------ **6-9**

Prototype Interface ------------------------------------------------------ **10**

Client Interactions ------------------------------------------------------ **10**

Hand-drawn Prototype ------------------------------------------------------ **10**

Computer-drawn Prototype ---------------------------------------------- **11-13**

Prototype Changes ------------------------------------------------------ **14**

Appendix 1 ------------------------------------------------------ **15-16**

Appendix 2 ------------------------------------------------------ **17**

Bibliography ------------------------------------------------------ **18**

**Previous Work Statement**

Our team will focus on creating the application from scratch using C# language. Some requirements for the application are that it must be downloadable on PC, and it must be able to connect to a web database. We will use the ideas that we learned from Application Development 1 in the previous semester. We will not use any previous code, but we will use the knowledge that we learned from before.

**Executive Overview**

The problem that the company OPEQ has is storing caller information on an excel sheet. They want us to make an application that will make it more efficient to store caller information. In this deliverable, we will do the UI/UX part of the application where we go over usability and design. We made prototypes for the application that shows the approximate look and feel of the app which will include all the functions and pages of the application. The first computer drawn prototype is based on the hand drawn prototype and the second prototype is based on the previous computer drawn prototype. We showed the first prototype to the client and then based on the client’s review and suggestions; we made the second prototype. This way, we have the application that our client will like using.

**Summary** **Description of Client**

The client’s full name is Simon Provencher; he is a supervisor at a company named OPEQ, Dinal and Ibrahim’s former manager. He deals with computer components like hard drives, motherboards, RAM etc. The client is remarkably familiar with computers, so he is very skilled and has decent literacy, for example, he is quite familiar with Microsoft Office 365 apps like word, excel, etc. He is also familiar with databases as we used databases for items during the internship.

**Business Problem**

The problem that our client told us is that they are having a tough time recording the information of the customers that call them. They use an excel sheet to record the information but it takes a while to do it so some information might get mixed up with the others or get forgotten completely. The solution that our client proposed is to make a desktop application that will make recording customer information fast and easy and it will also make the viewing of the records more organized. We did the prototype on Figma, the login page with language selection, the view page, etc. It's going to help us by giving ourselves an idea of how exactly the application will look like and it also shows the flow of the application.  **Summary of the Project Narrative**

Upon opening the application, the employee will be asked to “login”. The employee needs to enter their name in order to access the application. Their name will be saved into the database. On the same page the employee will be asked which language they prefer, French or English. The employee will be directed to the options page after logging in. which will offer the choice of adding, changing, or viewing data.

The application will be redirected to the add data page when the employee decides to add data. The employee will enter data and will have the option to save or cancel it. An alert box will appear once the employee clicks the save button to ask if they really want to save the data. If so, the information will be saved into the database and the page will be refreshed. If the answer is no, it will remain on the same page. An alert box will appear once the employee clicks the cancel button to ask if they really want to cancel. If the answer is yes, the data won't be saved, and the employee will be taken back to the choices page. It will stay on the same page if not.

The application will redirect to the modify data page when the employee chooses to modify the data. By entering a valid MAT, the employee can update the current data. If the MAT is invalid, a warning message will be displayed in an alert window. The employee has the option to save or cancel it after confirming it. An alert window will appear once the employee clicks the save button to ask if they really want to upload the data. If so, the information will be saved into the database and the page will be refreshed. If the answer is no, it will remain on the same page. An alert box will appear once the employee clicks the cancel button to ask if they really want to cancel. If the answer is yes, the data won't be saved, and the employee will be taken back to the choices page. It will stay on the same page if not.

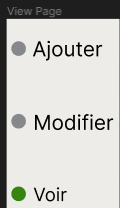
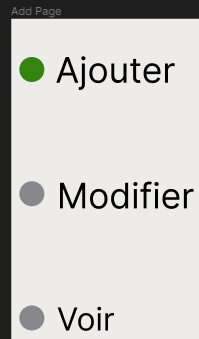
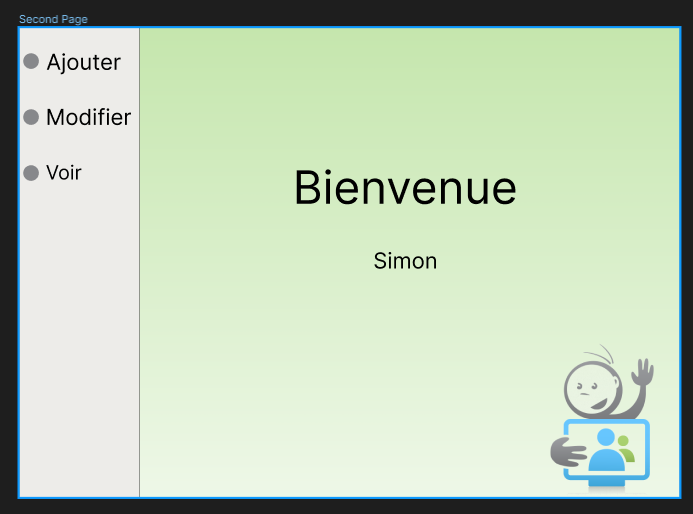
The application will be redirected to the view data page when the employee chooses to view the data. The employee has the choice to filter ongoing, unresolved, and solved issues. Three columns will be output: MAT, Client Name, and Order Number. The employee has the option of looking up a certain customer, MAT, or order number. Selecting a row in the result table will link the employee to another page that displays that row's particular detail.

**10 Usability Guidelines**

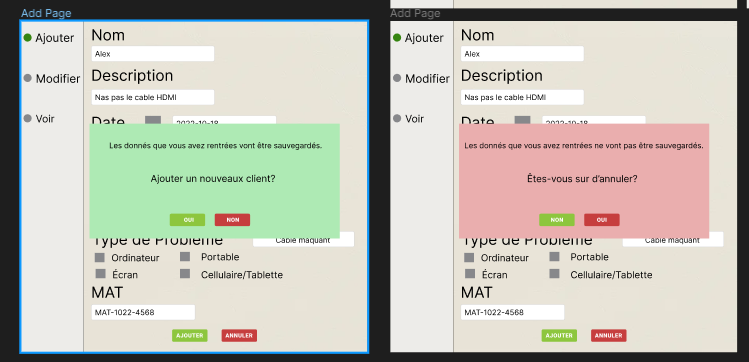
1. On the login page the user can choose the language. This will help the user navigate through the application in their preferred language.



1. All the pages have a panel on the left side with access to radio buttons to switch between pages. This makes it easier for the user to navigate through the application without getting lost. The buttons will be highlighted to indicate what page the user is on.



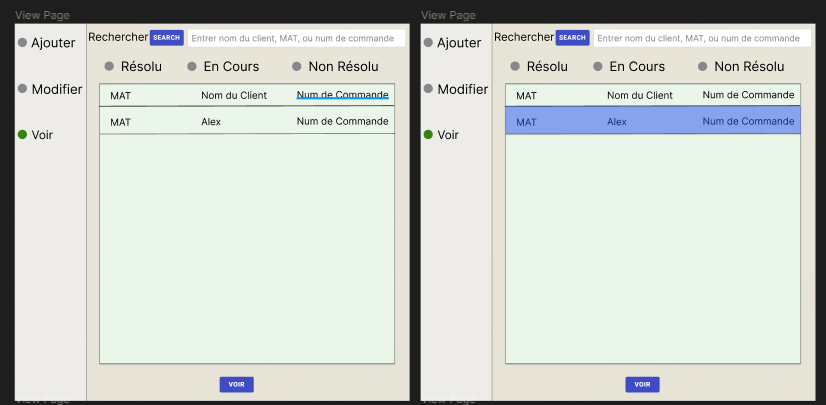
1. Every time the user clicks the save or the cancel buttons a pop-up will appear to confirm if the user wants to cancel or if they want to add another client. This will prevent accidental cancelation of the data.



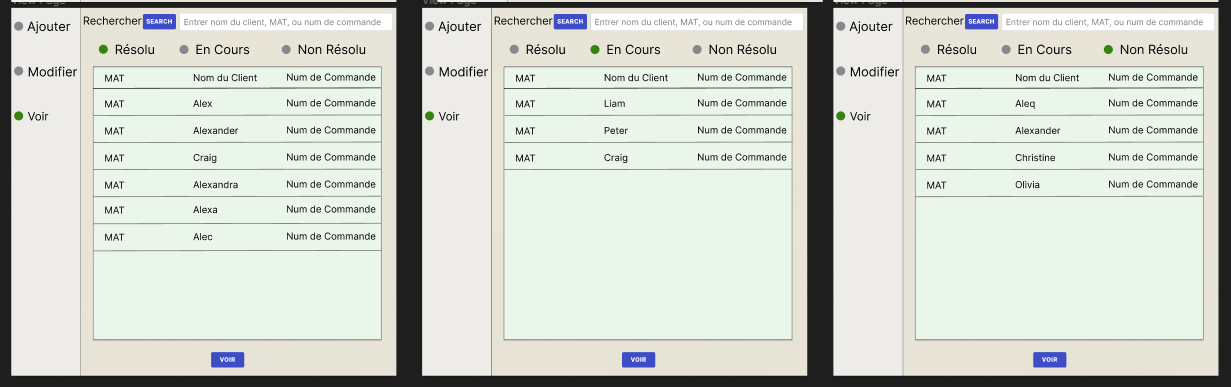
1. Error pop-ups will appear in the case of an error. This would make the user aware if something went wrong.



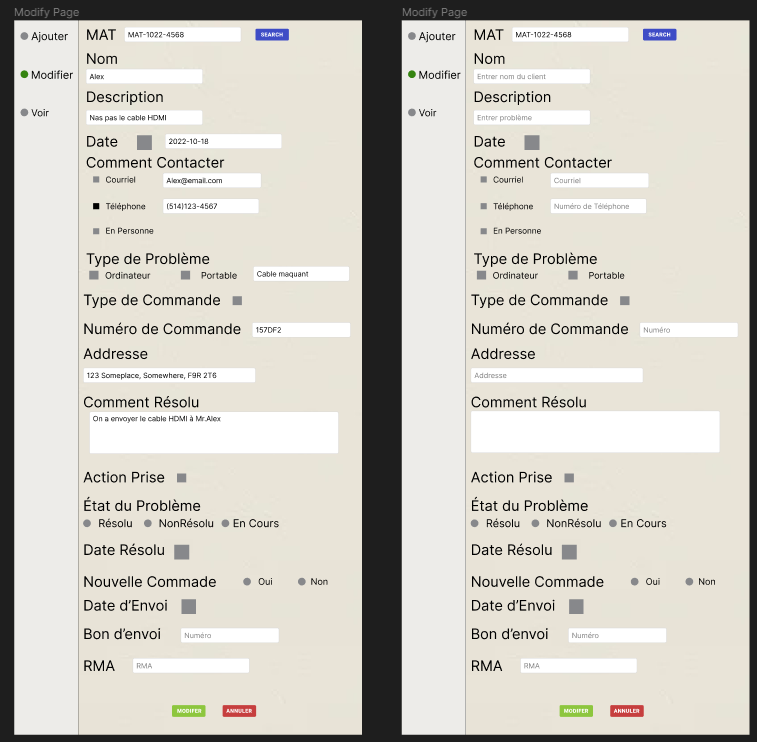
1. The data in the view page will be shown in a table so that the data is not too crowded and makes the data look clearer. It also makes it easier to click between the rows. The rows will only show client name, MAT (barcode), and order number because those are the options the user can use to search for a specific record.



1. The filter buttons in the view page are located on the same line so that the user can easily be accessed. The data is shown in the table and not in another page, which prevents the user from losing track of which filter they selected.



1. In the modify page, the user can input the MAT to load previously added record inside the database. This make modifying the data that has been stored and prevent modifying the wrong record.



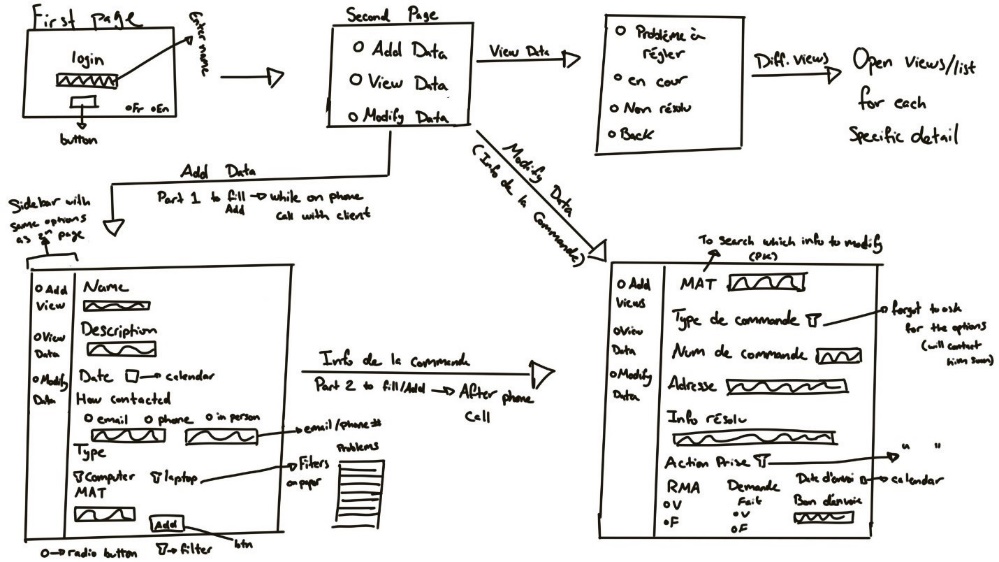
**Prototype interfaces**

[https://www.figmacom/file/zzQ6TjwvEuIe6QvEjTFrTe/ProjectPrototype?node-id=0%3A1](https://www.figma.com/file/zzQ6TjwvEuIe6QvEjTFrTe/ProjectPrototype?node-id=0%3A1)

**Client interactions**

The way we interacted with the client was often by phone number to ask a few questions quickly here and there or to schedule a meeting, sometimes by email but rarely, of course, real life meetings at the workshop. About the prototype, he wants us to add a search bar, change view page, add options button to see if the client's problems were solved and the date of when the problem was solved.

**Hand-drawn prototype**

  
In the filters:

Computer (in add page): Ordinateur Défectueux, Écran Défectueux, Windows non-installé, Windows corrompu, Wifi ne fonctionne pas, Matériels brise au transport, Cable Maquant(s), Accessoire(s) manquant(s), Erreur lieu de livraison, Matériel non reçu, Client non satisfait, Question général, Accessoire(s) brise, Mauvais matériel reçu, Cable brise(s), Non livre, disque non détecter

Laptop (in add page): Ordinateur Défectueux, Écran Défectueux, Windows non-installé, Windows corrompu, Wifi ne fonctionne pas, Matériels brise au transport, Cable Maquant(s), Accessoire(s) manquant(s), Erreur lieu de livraison, Matériel non reçu, Client non satisfait, Question général, Accessoire(s) brise, Mauvais matériel reçu, Cable brise(s), Non livre, Faute du client, webcam défectueux, Aucun problème, Problème touchpad, disque dur, finger print, problème clavier, penture brisée, power du portable scrap, mémoire, disque non détecter

Écran (in add page): Briser, affichage, maque des cables

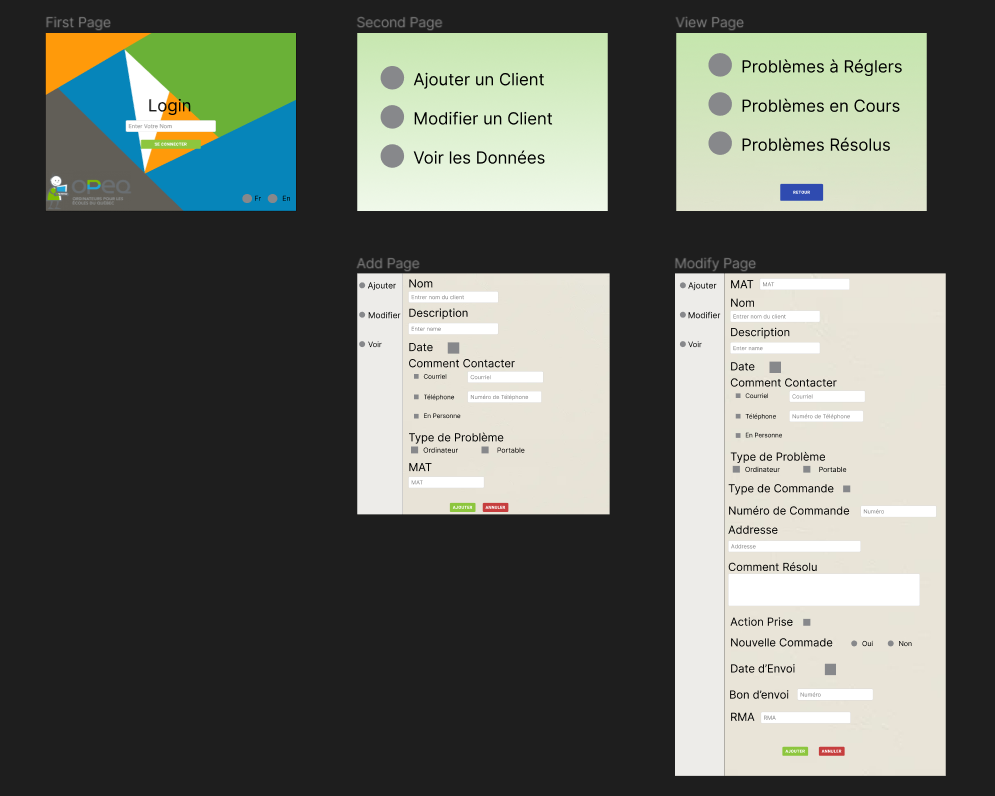
Cellulaire/Tablette (in add page): Batterie, affichage, accessoires manquants, ne convient pas

Type of order (in modify page): CLIC, Famille branchée, OBNL, École, Bibliothèque

Action took (in modify page): Création RMA, envoient accessoire(s) sans RMA, Résolution au téléphone/Courriel, Passage à l'atelier, Transfert vers un autre centre OPEQ, Mise en attente

**Computer-drawn prototype**

Computer-Drawn #1

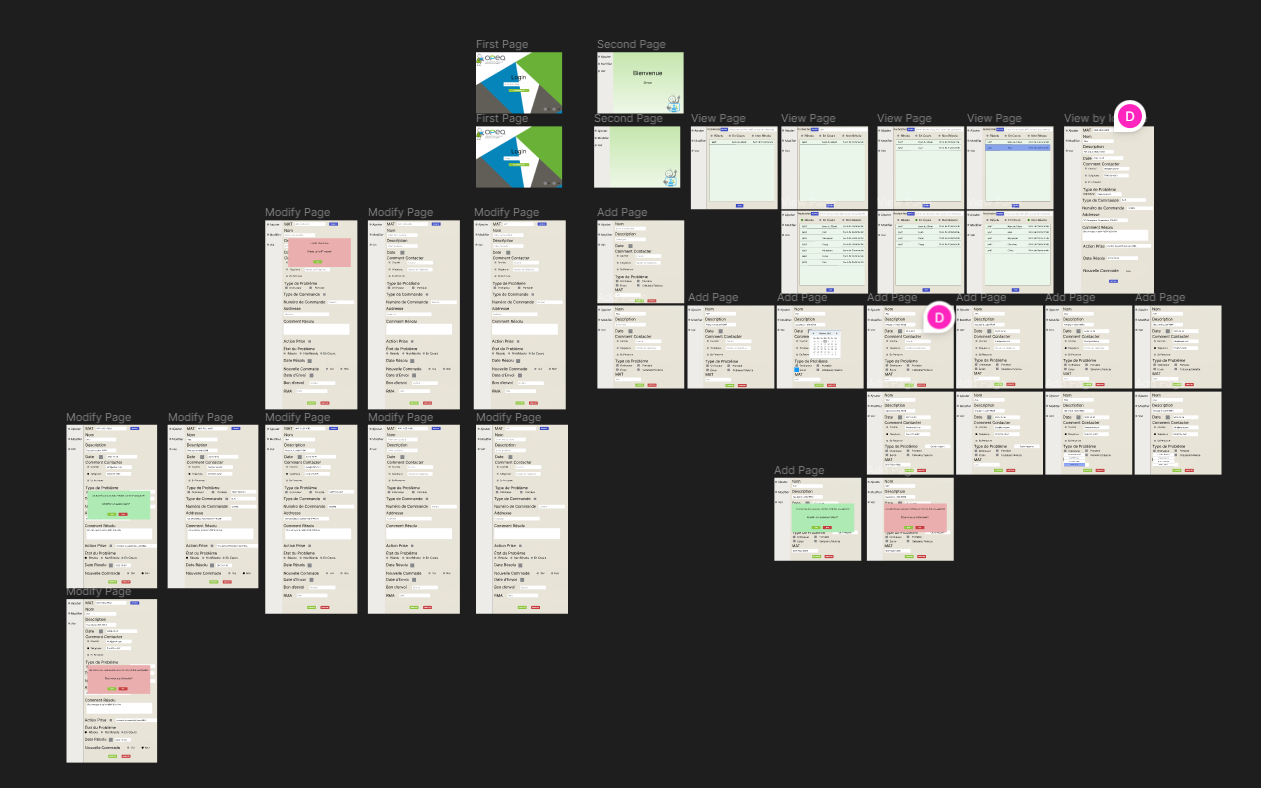


Changes to be made:

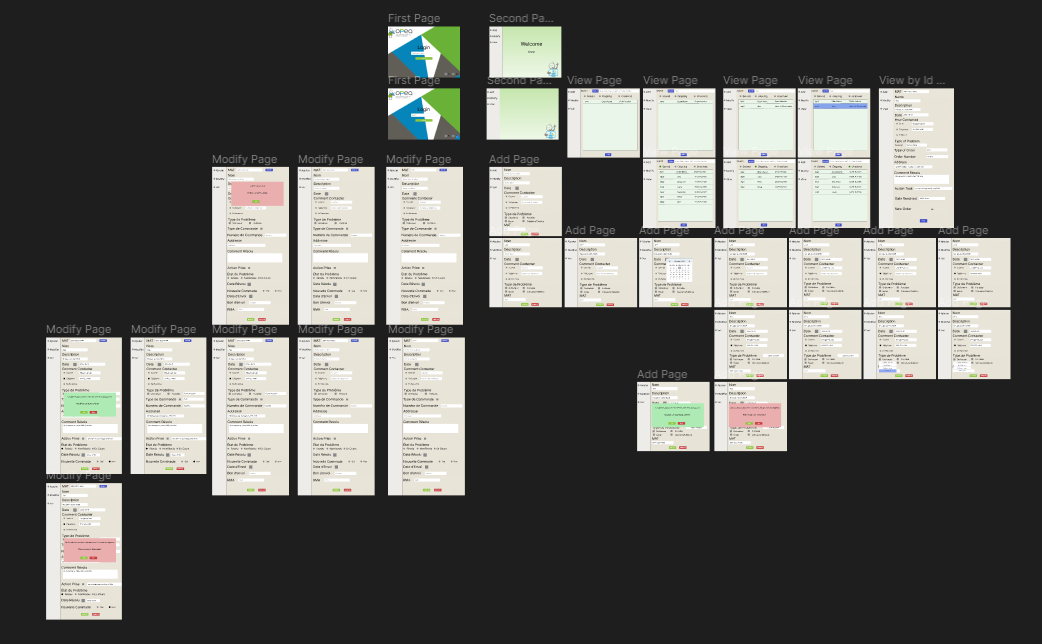
* Show data in a table in the view page
* Search is in the same page as view
* Filters are on the same page as the view
* Search by: nom, MAT, num de commande

Computer-Drawn #2 (Dinal + Justin)

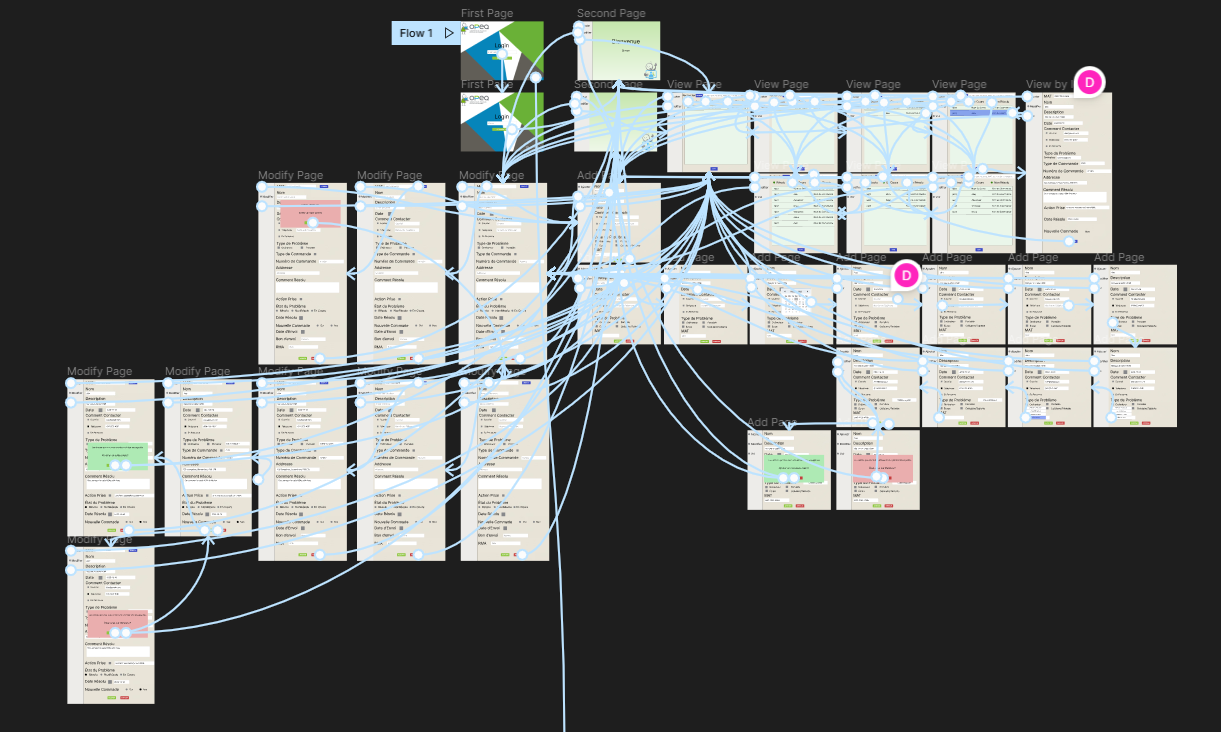
French ver.



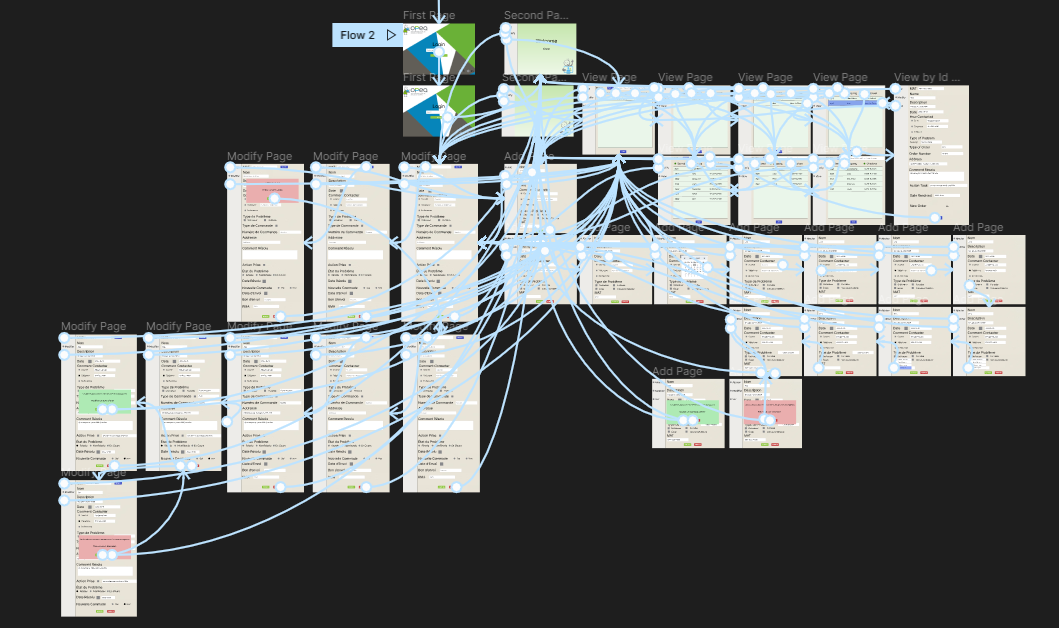
English ver.



French ver. with Connections



English ver. with Connections



**Prototype changes**

There are no changes between the hand-drawn prototype and the first ver. Of the computer-drawn prototype.

In the first ver. of the computer-drawn prototype, we forgot to include 2 other options, Écran and Cellulaire/Tablette, for the problem type options in the add page and our client wanted to add another field that tells what date the problem was solved on the modify page.

In the second ver. of the computer-drawn prototype, we changed the background of the login page because our client wanted us to remove the color orange. We created new pages such as a page that welcomes the user once they have entered their names, a page to list the records depending on the filter selected or if the user searched for a specific record and we also created pop-ups to confirm if the data was added successfully, if an error occurred or to ask confirmation from the user if they want to continue with the action they selected. We also did the connections between the pages together to show the flow of our application and we also added the missing options in the add page and the new field to be added in the modify page.

**Appendix 1**

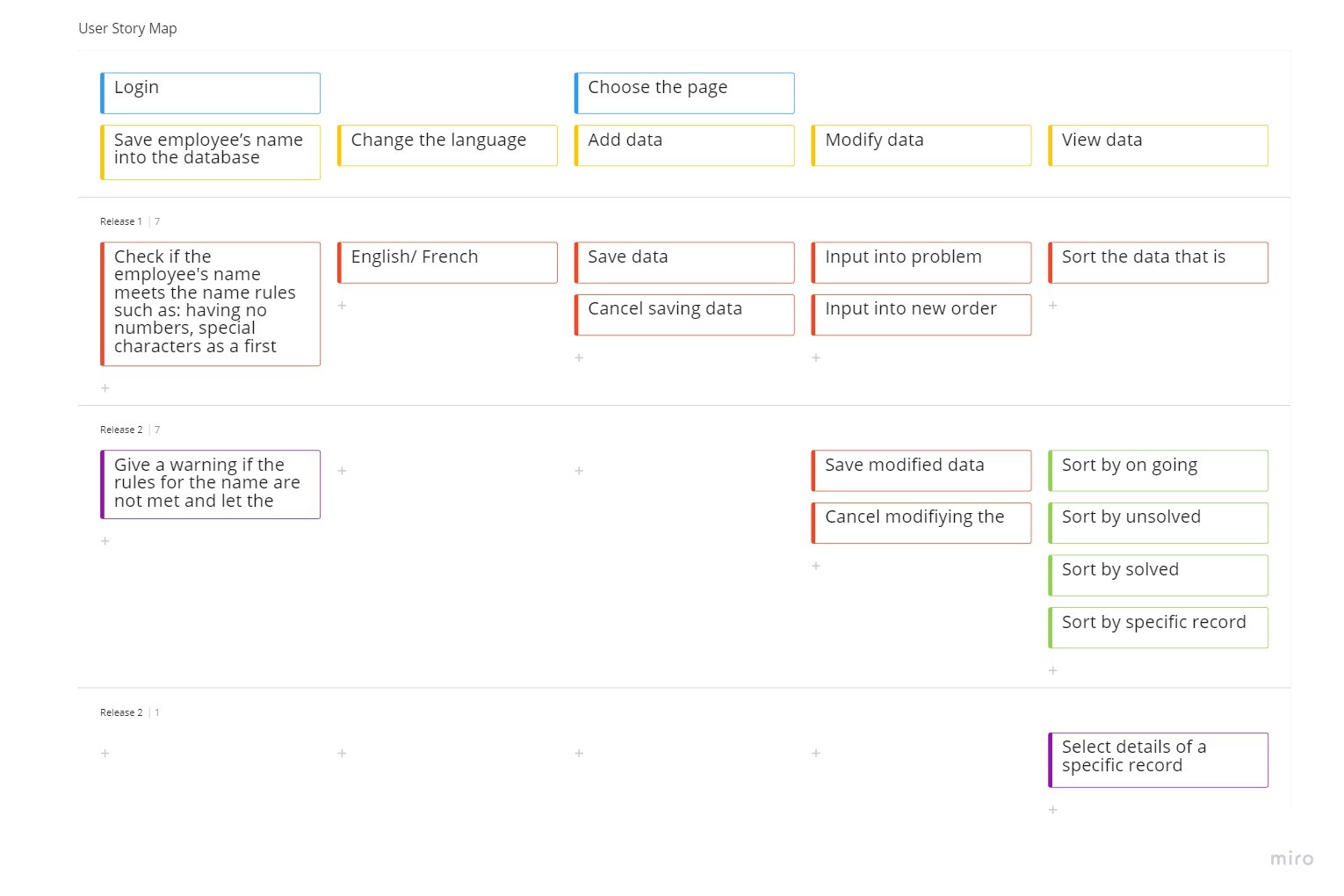
Revised user stories and User story tests

We choose the three columns—MAT, Client Name, and Order Number—that will be displayed on the view data page. A search feature has been added that allows users to look up a certain record by entering a valid MAT, Client Name, and Order Number. The employee will be able to view a record's detail. We added two new user stories (specific record search & detail of a specific record searched). And we have updated the select statements of the three-filter data function (solved, unsolved, ongoing) in the test criteria.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| User Stories | | | | |
|  | **As a** | **I want to** | **So that** | **Test criteria** |
|  | An employee | save employee’s name into the database | It can be recorded and viewed later. | Run a “Select name from table” query to check if the name is saved |
|  | An employee | Choose the language | It will be easier to navigate through the application. Choose between English and French. | Check if language on the page is changed in the application |
|  | An employee | Choose the page | The employee can choose between the add, view, or modify pages. If there is no data stored in in the database, employee will only have the option to go to the add data page. | Check if the page was changed in the application |
|  | An employee | Add data | The data will be stored in the database to be viewed later. The data that must be saved is client name, description, date, how contacted, contact info, type of problem, barcode. | Run a “Select \* from table” query to check if the record is saved |
|  | An employee | Save data | The data, from the add data form, is saved into the database. Current page will be refreshed. | Run a “Select \* from table” query to check if the record is saved |
|  | An employee | Cancel saving the data | The data, from the add data form, is not saved into the database. | Run a “Select \* from table” query to check if the record is not saved |
|  | An employee | Modify data | The existing data can be updated/modified based on new information. The existing data will be loaded using the barcode, which is the primary key, and new data will have to be saved: Type of order, order number, address, problem solved, action took, new order. | Run a “Select \* from table” query to check if the record is saved |
|  | An employee | Input into problem solved field | The employee can choose whether the problem is solved or not. If the problem is solved, the solution taken will be described and if it is not solved, it will be blank. | Run a “Select problem solved from table” query to check if the field has a solution or null |
|  | An employee | Input into new order field | The employee can choose whether to add a new order or not. If a new order is required, new fields will become visible and the employee will input new data: Data sent, return number and new barcode and if it is not required then no new data will be inputted. | Run a “Select \* from new order table” query to check if the new order record is saved |
|  | An employee | Save modified data | The data, from the modified form, is saved in the database. | Run a “Select barcode from table” query to check if the record was modified |
|  | An employee | Cancel modifying the data | If the employee changes their mind, the data will not be modified and will remain the same. | Run a “Select barcode from table” query to check if the record was not modified |
|  | An employee | View data | The employee can see the saved data. | Check if the output data matches the data in the database. |
|  | An employee | Specific record search | The employee has the option of conducting a specific record search for a MAT, Order number, or client name. | Run a “Select MAT, ClientName, Order\_Number from table where MAT= xxx (search bar input) or Order\_Number= xxx or ClientName= xxx” query to check if the output data from the application matches |
|  | An employee | Details of a specific record searched | The employee can select a record from the result table to see its detailed information. | Run a “Select \* from table where MAT= xxx (search bar input) or Order\_Number= xxx or ClientName= xxx” query to check if the output data from the application matches |
|  | An employee | Choose how to filter the data | The employee can see the entries that are solved, unsolved and ongoing | Check if the records that are being output matches the example view |
|  | An employee | Filter data, that are going to be viewed, by ones that are solved. | The employee can see the records’ MAT, Client Name and Order Number that are marked as solved. | Run a “Select MAT, ClientName, Order\_Number from table where status = solved” query to check if the output data from the application matches |
|  | An employee | Filter data, that are going to be viewed, by ones that are ongoing. | The employee can see the records’ MAT, Client Name and Order Number that are marked as ongoing. | Run a “Select MAT, ClientName, Order\_Number from table where status = ongoing” query to check if the output data from the application matches |
|  | An employee | Filter data, that are going to be viewed, by ones that are unsolved. | The employee can see the records’ MAT, Client Name and Order Number that are marked as unsolved. | Run a ““Select MAT, ClientName, Order\_Number from table where status = unsolved” query to check if the output data from the application matches |

**Appendix 2**

Revised story map



**Bibliography**

Khusid, A., & Shardin, O. (2011). Miro. <https://miro.com/app/board/uXjVPNABqzc=/>

Field, D. (2016, September 27). *Figma: the collaborative interface design tool.*

Figma. <https://www.figma.com/>