Vanier College

Deliverable 4

  Client: Opeq, Simon

System Development Section 01

Team Orange:

Jiamin Yuan

Dinal Patel

Craig Justin Balibalos

Alihan Djamankulov

Ibrahim Awad

Submission Date:

Wednesday October 17, 2021

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**

Narrative Description ------------------------------------------------------ **5**

Appendix 1 ------------------------------------------------------ **6**

Appendix 2 ------------------------------------------------------ **6**

Appendix 3 ------------------------------------------------------ **7**

**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 made a list of the user stories which are based on the client’s demands. We also created story tests and user story map. The user stories will help us keep track of the features that need to be added as well as the results and purpose of those specific features. Following the user stories lists, we created a user story map which will help us keep track of what exactly we must do for every part, for example the login part consists of saving the employee's name into the database and language choice.

**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 listed the user stories that an employee can do in the desktop application. Also, we have user story tests that correspond to the user stories and a user story map.

**Narrative description**

Upon opening the application, the employee will be asked to “login”. The employee will have to enter their name that will then be saved into the database. On the same page the employee will be asked which language they prefer, French or English. After logging in, the employee will be redirected to the option page. Which will have the option to add, modify or view data.

When the employee chooses to add data, the application will redirect to the add data page. The employee will enter data, after entering it, the employee can choose to save or cancel. By clicking the save button, the data will be saved into the database. By clicking the cancel button, the data will not be saved and redirected back to the options page.

When the employee chooses to modify data, the application will redirect to the modify data page. The employee can update the existing data based on the new information. After entering it, the employee can choose to save or cancel. By clicking the save button, the data will be saved into the database. By clicking the cancel button, the data will not be saved and redirected back to the options page.

When the employee chooses to view data, the application will redirect to the view data page. The employee has the option to filter solved, unsolved and ongoing problems.

**Appendix 1**

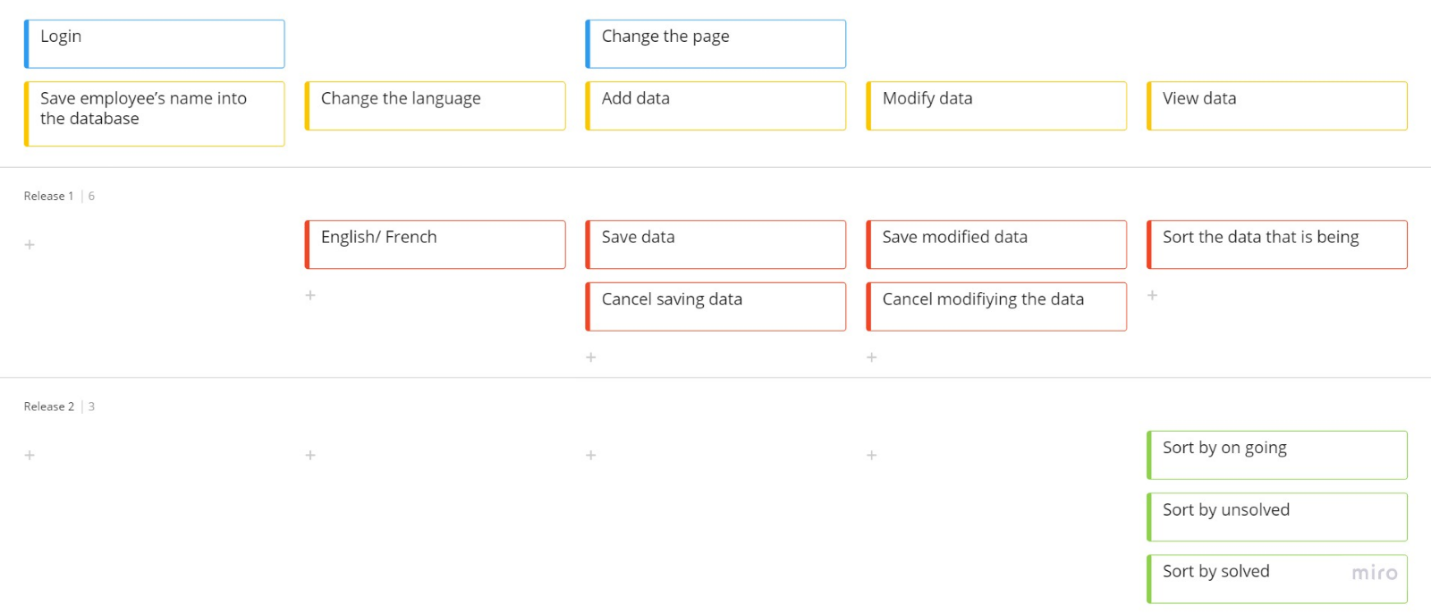
**Appendix 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| User Stories | | | | | |
|  | **As a** | **I want to** | **So that** | **Test criteriaa** | |
|  | An employee | save employee’s name into the database | It can be recorded and viewed later | View the Database to see if it was stored | |
|  | An employee | Change the language | It will be easier to navigate through the application | See if the language changes on all the pages |  |
|  | An employee | Change the page | The employee can either add, view, or modify data | See if the page changed |  |
|  | An employee | Add data | The data will be saved into the database to be viewed later | Check the database to see if the data was added |  |
|  | An employee | Save data | The data is stored into the database | Check the database to see if the data was saved |  |
|  | An employee | Cancel saving the data | If employee changes their mind the data will not be stored | Check the database to see if the data was not saved |  |
|  | An employee | Modify data | The existing data can be updated/modified based on new information | Check the database to see if the data was updated |  |
|  | An employee | Save modified data | The data is stored in the database | Check the database to see if the modified data was saved |  |
|  | An employee | Cancel modifying the data | If the employee changes their mind, the data will not be changed and will remain the same. | Check the database to see if the data was unchanged |  |
|  | An employee | View data | The employee can see the saved data. | Check if the outputted data matches the data in the database. |  |
|  | An employee | Sort the data that is being viewed | The employee can see the entries that are solved, unsolved and on going | Check if the data that are being outputted matches the example view |  |

**Appendix 3**

I chose Miro as the User Story Map tool because it is very beautiful in design and easy to use. I do not need to pay for the tool, and I can easily download the story map for free in different formats.

**Print User Story Map**



<https://miro.com/app/board/uXjVPNABqzc=/>