

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 20, 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).

X Jiamin Yuan

Jiamin Yuan
Team Member

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).

X Dinal Patel

Dinal Patel
Team Member

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).

X Craig Justin Balibalos

Craig Justin Balibalos
Team Member

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).

X Alihan Djamankulov

Alihan Djamankulov
Team Leader

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).

X Ibrahim Awad

Ibrahim Awad
Team Member

Table of contents

Statement	-----	4
Executive Overview	-----	4
Summary Desc. of Client	-----	4
Business Problem	-----	4
Narrative Description	-----	5
Appendix 1	-----	6
Appendix 2	-----	6-7
Appendix 3	-----	8
Bibliography	-----	9

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 & 2

Dinal had a meeting with our client, Simon, to ask about the requirements and flow of the application. He gave us an excel sheet as an example and Dinal drew a rough sketch of the application GUI, with him.

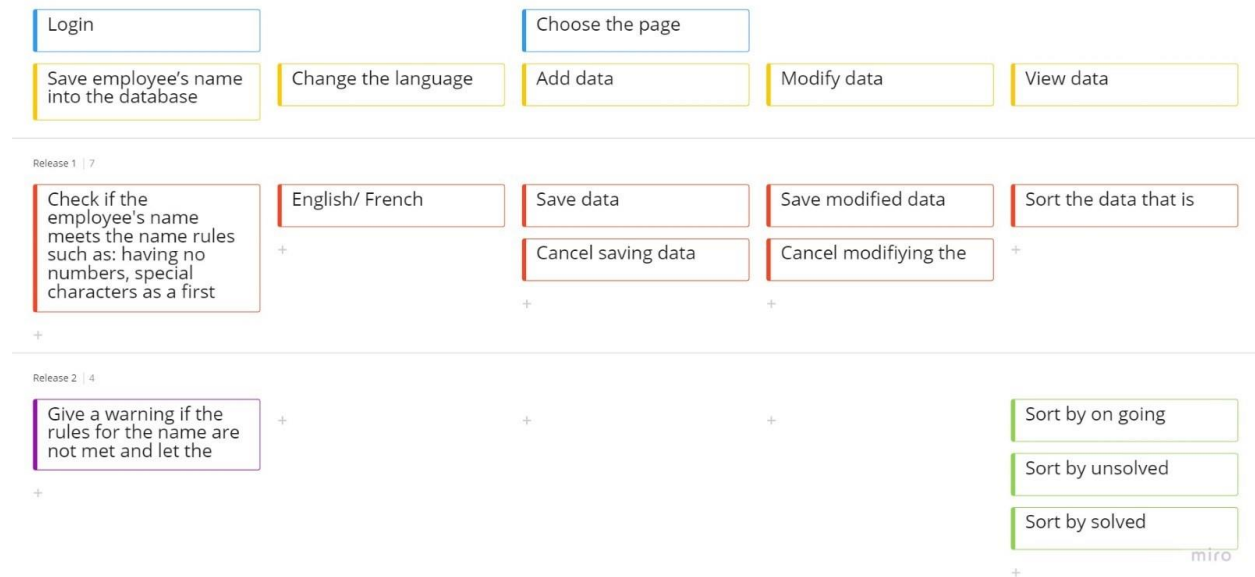
User Stories				
	As a	I want to	So that	Test criteria
	An employee	save employee's name into the database	It can be recorded and viewed later.	Run a "Select name from table" query to check if the name is saved
	An employee	Choose the language	It will be easier to navigate through the application. Choose 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 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.	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	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 that are marked as solved.	Run a "Select * 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 that are marked as ongoing.	Run a "Select * 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 that are marked as unsolved.	Run a "Select * from table where status = unsolved" query to check if the output data from the application matches

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



Bibliography

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