

Vanier College

Deliverable 7

Client: Opeq, Simon

System Development Section 01

Team Orange:

Jiamin Yuan

Dinal Patel


Craig Justin Balibalos

Alihan Djamankulov

Ibrahim Awad


Submission Date: 12 December 2022

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).

 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).

 Dinal Patel

Dinal Patel
Team Leader

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).

 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).

 Alihan Djamankulov

Alihan Djamankulov
Team Member

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
Business Problem	-----	4
RACI	-----	5
Revised Description of the System	-----	6
Client's comment	-----	6
Design and implementation	-----	7
Security measures	-----	7
Future work		
User Interface improvements	-----	8
Unimplemented user stories and functions	-----	8
Future Security Recommendations	-----	8
Unit and Integration Test Strategies	-----	8
Appendix 1		
Revised User Interface	-----	9
Appendix 2		
User Guide	-----	9
Appendix 3		
Implemented User Story Screenshot	-----	10-12

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.

Using all the past deliverables we started coding and are at the final part. We used Click Up, RACI, and the Gantt chart to divide the work. We also showed the client and got some comments for finalizing the project. We finished the project and are ready to send it to our client.

Business Problem

Our client Simon Provencher from OPEQ has a problem storing caller information by saving it into an excel sheet but as the number of callers increases, the time to search for a specific caller also increases. Therefore, our client proposed making a desktop application which can store caller information into a database, modify saved caller information and view them. In this deliverable, we will be finishing the final demo of the application and present it to our client, Simon, and get his opinion/comments on the application.

RACI

R – Responsible

I – Inform

C – Consult

A – Accountable

Project Team	D I N A L	J I A M I N	C J	I B R A H I M	A I H A N	A L E X	N A G A T	S Y E D	Weekly Meeting	Monthly Steering
Login Page	I		R	I				C		
Welcome page front end	R		A	I				C		
Welcome page back end	I		A	I	R	C				
Add page front end	A	R		I				C		
Add page back end	I		R	I	A	C				
Modify page front end	R	A		I				C		
Modify page back end	I		A	I	R	C				
View page front end	A	R		I				C		
View page back end	I			R	A	C				
View searched page front end	I	R		I				C		
View searched page back end	I		R	A		C				
Language change	I			R				C		

Revised Summary Narrative Description of the System

The user can login as an employee by entering their name on the login page. Their name will be stored into the database with a unique employee id. They will be able to choose from add data, modify data, or view data. If they choose to add data, they need to input the client's name, client description, date, client email. They can add the client phone number and client address if it is available (nullable). They can select how they contacted the client using radio buttons. If it is by email, telephone or in person. An MAT (item barcode) is required when adding a product. Moreover, they can select the problem from dropdown. Whether it is computer problem, laptop problem, screen problem or phone tablet problem, it will provide different category under type of problems. Once the user login as an admin using name and password, they have the power to add or delete a record of the problem tables. The dropdown field for problem type on the add data page will change following the modifications made by the admin.

When the employee chooses to modify data, they need to enter a valid MAT to access the matching record. It will show the data they have entered on add data page. They need provide some other information in addition to that. Describe in words that the order type and the action they took to solve to problem. The specific date that the problem got solved. Choose between solved, unresolved, and ongoing radio buttons to indicate the status of the present problem (nullable). If the client's phone number is currently available, they can add it (nullable). If it is a new order, it needs the send date and return voucher.

The employee can filter the data by problem status when they decide to view it (solved, unsolved, ongoing). It will show the MAT, client name, and order number for the records that match. By providing the client's name, MAT, or order number, it can locate a particular record.

Client Comments

The client wanted us to add some error handling so the user could identify what the error is. It will also prevent the application from crashing. The client also asked us to see if they can add the application to a server, so they can distribute it to the other OPEQ branches. The client also asked us if we can work on/improve the application in the future if there are any concerns.

Design & Implementation Decisions

For the design of the application, we focused on the OPEQ colors, because they are the colors that represent the company. Blue representing recycling, green representing the environment, and grey representing social awareness. The client did not focus on the design and focused more on the functionality, but his request was to add the colors and the OPEQ mascot. Our mock-up and final project are similar but in our final demo we made it more functional and realistic. We changed some of the design, so the flow is smoother.

For the software design we made it so that there are radio buttons on the side of each form to navigate through the application. We added buttons to save and modify at the bottom of the form so that the user will have to go through the whole form to get to the button, which ensures that the user will fill out the form. We also added message boxes to notify the user of changes made or for errors.

Security measure description

For the security measures, only admin account is protected by a password and username authentication. An additional form is only accessible to the admin. Error handling is also implemented to prevent the application from crashing and disrupting the flow

Future work

User Interface Improvements:

By standardizing the line spacing, text box and button sizes, and making each form the same size, we can increase the consistency of the user interfaces.

Unimplemented User Stories and Functions:

In our project, every feature from user stories has been implemented.

Future Security Recommendations:

For future security recommendation, we can include an authentication step at the login interface, such as a captcha. and the option to create an account, if necessary. An administrator should have access to a forget password function in case they need to reset their password or modify it. Additionally, make sure that the application in its entirety is well covered in error handling.

Test Recommendations:

The application we tested has a normal flow, but we can test error handling in the future by entering alternative data types.

Appendix 1 -Revised User Interface

We mostly kept the same UI as the one we had in our prototype on Figma. The changes we made are for the admin where they can add data to the problem tables, order type and action took and the view form. For the admin add, we made a separate form to make refreshing the dataset easier and it is only visible and accessible by the admin. For the view form, we created a textbox where the user can input the MAT (barcode) because it was the only solution we could come up with and the only one that worked.

Appendix 2 – User Guide

In the User_Guide.docx

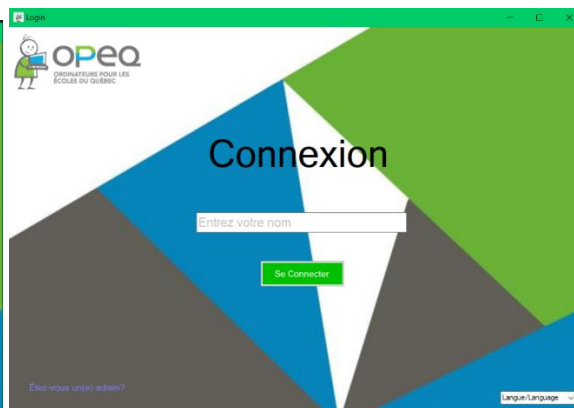
Appendix 3

Screenshots of implemented user stories:

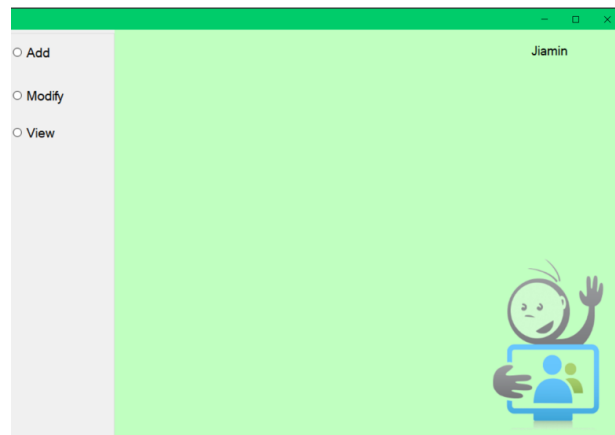
User Stories			
As a	I want to	So that	Test criteria
An employee	save employee's name into the database	It can be recorded and viewed later.	Run a "Select name from table" query to check if the name is saved



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

The image displays two screenshots of a web application interface, likely for managing equipment issues.

Left Screenshot (Ajouter Form):

- Navigation:** A sidebar on the left contains three links: "Ajouter" (highlighted), "Modifier", and "Voir".
- Form Fields:**
 - Nom:** A text input field with a placeholder "Ajouter votre Nom".
 - Description:** A text input field with a placeholder "Entrer problème".
 - Date Entrée:** A date picker showing "Monday, December 05, 2022".
 - Comment Contacter:** A section with three radio buttons: "Courriel", "Téléphone", and "En Personne". Below these are corresponding text input fields for email, phone number, and address.

Right Screenshot (Ajouter Form):

- Form Fields:**
 - Type de Problème:** A section with four dropdown menus: "Ordinateur", "Écran", "Portable", and "Cellulaire/Tablette".
 - MAT:** A text input field for the material number.
- Buttons:** At the bottom right, there are two buttons: "Ajouter" (green) and "Annuler" (red).

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

The image shows three screenshots of a web application interface, likely for managing orders and problems. The interface is divided into several sections:

- Left Sidebar:** Contains navigation links: "Ajouter", "Modifier", and "Voir".
- Main Content Area:**
 - Top Section:** A form for adding or modifying data, including fields for "Nom", "Description", "Date Entrée", and "Comment Contacter".
 - Middle Section:** A form for adding or modifying data, including fields for "Type de Problème", "Type de Commande", "Numero de Commande", and "Comment Résolu".
 - Bottom Section:** A form for adding or modifying data, including fields for "Numero de Commande", "Comment Résolu", "Action Prise", "État du Problème", "Date Résolu", "Nouvelle Commande", "Date D'envoi", "Bon D'envoi", and "RMA".

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

Jiamin

☐ Resolved
☐ Processing
☒ Unresolved

MAT	Nom du Client	Numero de Commande
MAT-010	Alex	54321
MAT-010	Alex	54321
MAT-9657	Simon	5236

Jiamin

☐ Resolved
☐ Processing
☒ Unresolved

MAT	Nom du Client	Numero de Commande
MAT-ger	ger	Numero de Commande

Jiamin

MAT

Name

Description

Entry Date

How to Contact

☐ Email
☐ Phone
☐ In Person

How Resolved

Action Taken

Date Solved

Problem State

☒ Resolved
☐ Unresolved
☐ Processing

New Order

Date Sent

Return Voucher

RMA