Vanier College

Deliverable 7

  Client: Opeq, Simon

System Development Section 01

Team Orange:

Jiamin Yuan

Dinal Patel

Craig Justin Balibalos

Alihan Djamankulov

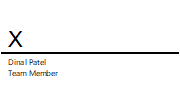
Ibrahim Awad

Submission Date:

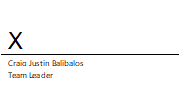
I, (Jiamin Yuan), student ID# (2055624), certify that I have contributed to this deliverable, (signature – this can be a scanned image, or an electronic signature).



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).



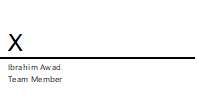
I, (Craig Justin Balibalos), student ID# (2069192), certify that I have contributed to this deliverable, (signature – this can be a scanned image, or an electronic signature).



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

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

Revised Description of the System ----------------------------------------- 4

Client’s comment ---------------------------------------------- 4

Design and implementation ---------------------------------------------- P

Security measures ---------------------------------------------- P

**Future work**

User Interface improvements ---------------------------------------------- P

Unimplemented user stories and functions ------------------------------- P

Future Security Recommendations --------------------------------------- P

Unit and Integration Test Strategies --------------------------------------- P

**Appendix 1**

Revised User Interface --------------------------------------- P

**Appendix 2**

User Guide --------------------------------------- P

**Appendix 3**

Implemented User Story Screenshot --------------------------------------- P

Bibliography --------------------------------------- P

**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**

**Business Problem**

**Revised Summary Narrative Description of the System**

**Client Comments**

**Design & Implementation Decisions**

**Security measure description**

For the security measures, for the client’s contact information, they can have one of them saved in the database or both so, for example a client can only put their phone number but not their email address as contact information and vice versa. When it comes to the employee's passwords...

**Future work**

**User Interface Improvements:**

**Unimplemented User Stories and Functions:**

**Future Security Recommendations:**

**Test Recommendations:**

**Appendix 1 -Revised User Interface**

**Appendix 2 – User Guide**

**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 |  |
|  | 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 |  |

**Bibliography**