**Odd Semester (2019/2020)**



**BINUS UNIVERSITY**



**BINUS INTERNATIONAL**



**Assignment Cover Letter**

**(Teamwork)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | |  |  |  |  |
| **Student Information**: | **First Name** | | |  |  | **Last Name**  **Rasendriya Rasyid**  **Hisyam**  **Michael Halim**  **Anderson** | **Student ID Number**  **2201798295**  **2201797430**  **2201798761**  **2201796636** |
| 1.  2.  3.  4. | **Alifio**  **Muchsin**  **Nicholas**  **Ricky** | |
|  |  |  |  |  |  |  |  |
| **Course Code** | **: ISYS6169** |  |  |  |  | **Course Name** | **: Database Systems** |
| **Class** | **: L3AC** |  |  |  |  | **Name of Lecturer(s)** | **:** **Nurul Qomariyah** |
|  |  |  |  |  |  |  |  |
| **Major** | **: CS** |  |  |  |  |  |  |
| **Title of Assignment**  (if any) | : **Little Miss Order Management** |  |  |  |  |  |  |
| **Type of Assignment**    **Submission Pattern** | **: Final Project** | |  |  |  |  |  |
| **Due Date** | **: 08-01-2020** | |  |  |  | **Submission Date** | **: 08-01-2020** |

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

# Plagiarism/Cheating

BiNus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

# Declaration of Originality

By signing this assignment, I understand, accept and consent to BiNus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student:

1. Alifio Rasendriya Rasyid 2. Muchsin Hisyam 3. Nicholas Michael Halim 4. Ricky Anderson

# Introduction

In this project, we aim to create a database program for the owner of *Little Miss* bakery store. This program is created with the goal of helping the owner in keeping track of the bakery order history and revenue based on what they sell.

# Background

A small business may find it difficult to keep track of their orders; which ones are pending, done, need to be delivered, etc. That’s why we set out to make this program. Bakery owners usually sell their items without keeping track of their revenue daily, this makes it hard for them to see whether they are gaining profit or loss. They don’t know their exact sales and revenue numbers. They are busy enough working on the orders and would have a hard time going through hand-written bills and calculate their earnings.

Each business have their own needs and wants. We shouldn’t make a sophisticated and complex application with features that the business will clearly not use; this may even make them feel burdened (by the application). That’s why we interviewed the owner directly and asked for the specific things she needs from our program. With the interview, we came up with several features to work on:

* Customer status become members when the customer already have 5 or more orders history.
* Customer who has members status get a specific discount amount.
* Revenue graph or bar chart based on weekly, monthly and yearly sales depends on what the owner wants to see.
* Generate invoice details in PDF file.
* And other ordinary features.

# Proposed Alternative Solution

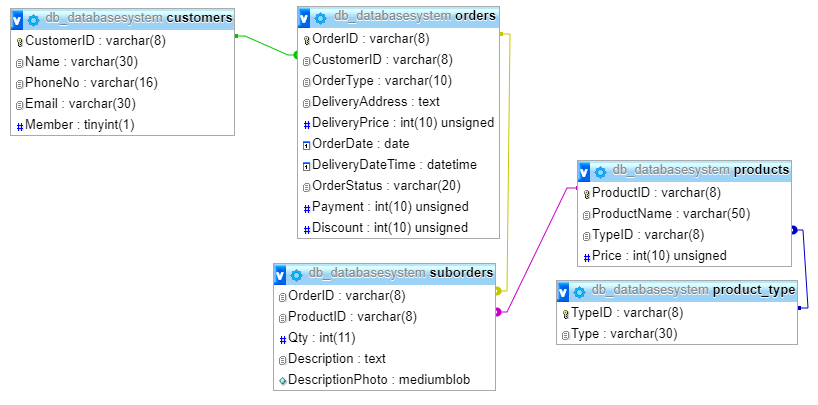
Our proposed application is affixed on keeping track of all sorts of information regarding orders that have been taken by the bakery. This helps the owner to keep track when an order comes in; when the order needs to be completed; what’s inside of the order; is the order completed or not; and how much have the customer paid in advance and how much left they need to pay. Not only does it keep track of the orders, but also the customer table which will help the owner to know whether the customer is a member or not. This is important in determining whether the customer is eligible for discounted prices or not. The discount system is also automated, further reducing work needed to manage the orders and more time for the owner to focus on working with her expertise instead.

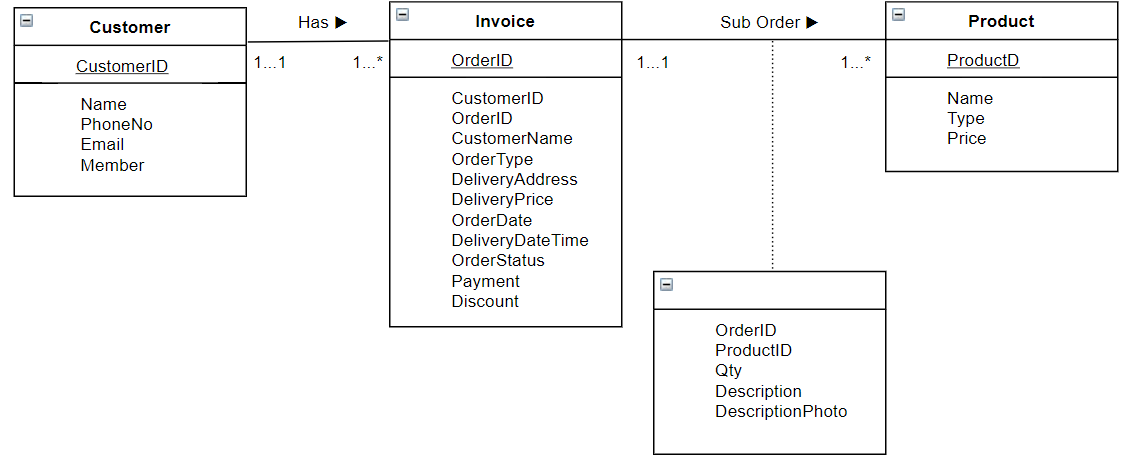
In addition to that, we also implemented features such as invoice generation, photo storage, and a simple insight on how much revenue the bakery has generated over a period of time. These are nice-to-have features that further improve the experience in using our application.

# Team Member Contribution

* **Nicholas Michael Halim :** Adding the UI effect, Make a graph/chart revenue on home menu, Back-end programming, Add convert PDF invoice code, Handle complex query, and final project report.
* **Alifio Rasendriya Rasyid :** Adding the UI effect, Designing UI, Whole Back-end programming, and complex back-end, Making Database on PhpMyAdmin, Make Database relation, error handling & validation, and final project report.
* **Muchsin Hisyam :** Most of UI Designs, Back-end programming, Making Database on PhpMyAdmin, Make Database relation, error handling & validation, and final project report.
* **Ricky Anderson :** Make proposal, and final project report.

# Database Design Entity-Relationship (ER)

*Relation View*



*Entity Relationship Diagram*

# 

# 

# Normalization

*Normalization Process*

# 

# Sample Query

These are some of the queries that use to manipulate the data that of what the owner wanted.

|  |
| --- |
| INSERT INTO products(ProductID, ProductName, TypeID, Price) VALUE('%s', '%s', '%s', '%d') |

*Add Customer Query*

|  |
| --- |
| *SELECT SUM(Payment) AS revenue,CONCAT( STR\_TO\_DATE(CONCAT(YEARWEEK(OrderDate, 2), ' Sunday'), '%X%V %W'), ' to ', STR\_TO\_DATE(CONCAT(YEARWEEK(OrderDate, 2), ' Sunday'), '%X%V %W') + INTERVAL 6 DAY ) AS t FROM orders WHERE OrderDate BETWEEN dateStart and dateEnd GROUP BY YEARWEEK(OrderDate, 2) ORDER BY YEARWEEK(OrderDate, 2);* |

*Getting Weekly Sales Query*

|  |
| --- |
| *SELECT orders.\*, (tv.total + orders.DeliveryPrice - orders.discount) as OrderTotal, (tv.total + orders.DeliveryPrice - orders.Payment - orders.discount) as RemainingPayment FROM orders INNER JOIN (SELECT suborders.OrderID, SUM(products.Price \* suborders.Qty) as total FROM suborders INNER JOIN products on suborders.ProductID = products.ProductID GROUP BY suborders.OrderID) as tv on orders.OrderID = tv.OrderID;* |

*Getting Total Price per Order and its Balance Due*

|  |
| --- |
| SELECT MAX(TypeID) FROM product\_type |

*Getting Last Type ID from Database Table*

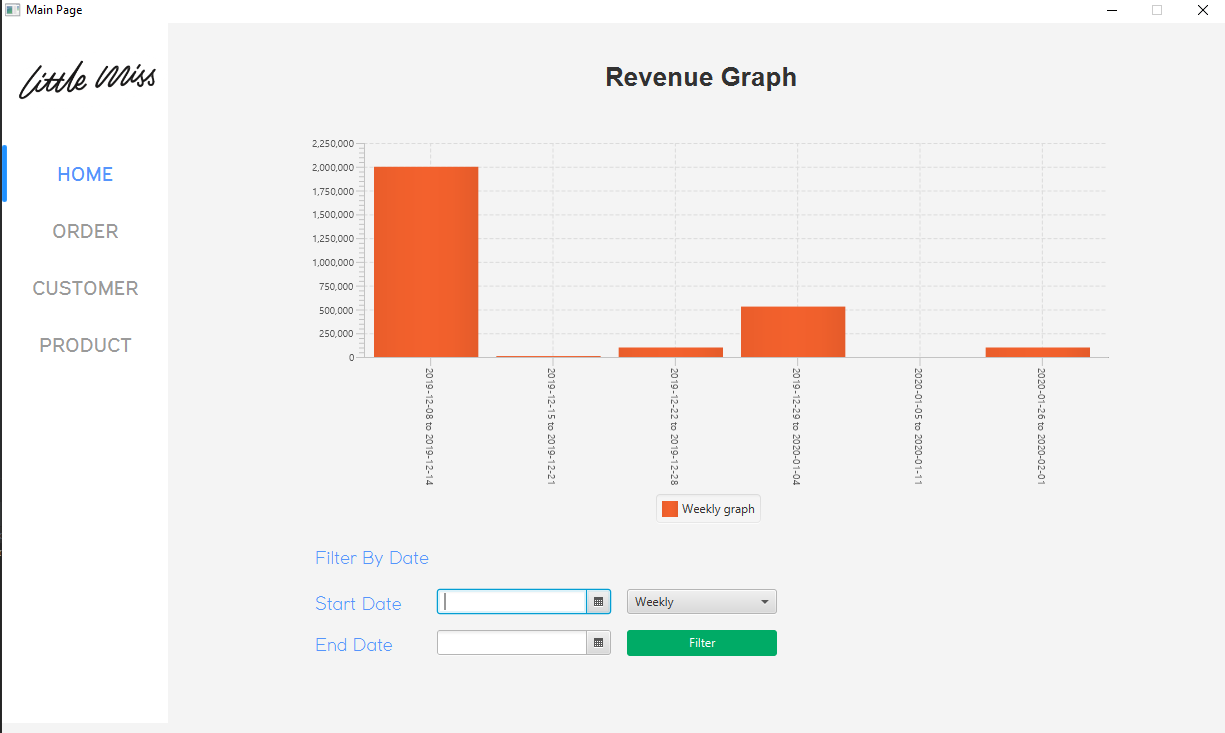
|  |
| --- |
| SELECT OrderDate from orders ORDER BY OrderDate ASC LIMIT 1 |

*Getting First Sale from Database Table by Specific Date Filter*

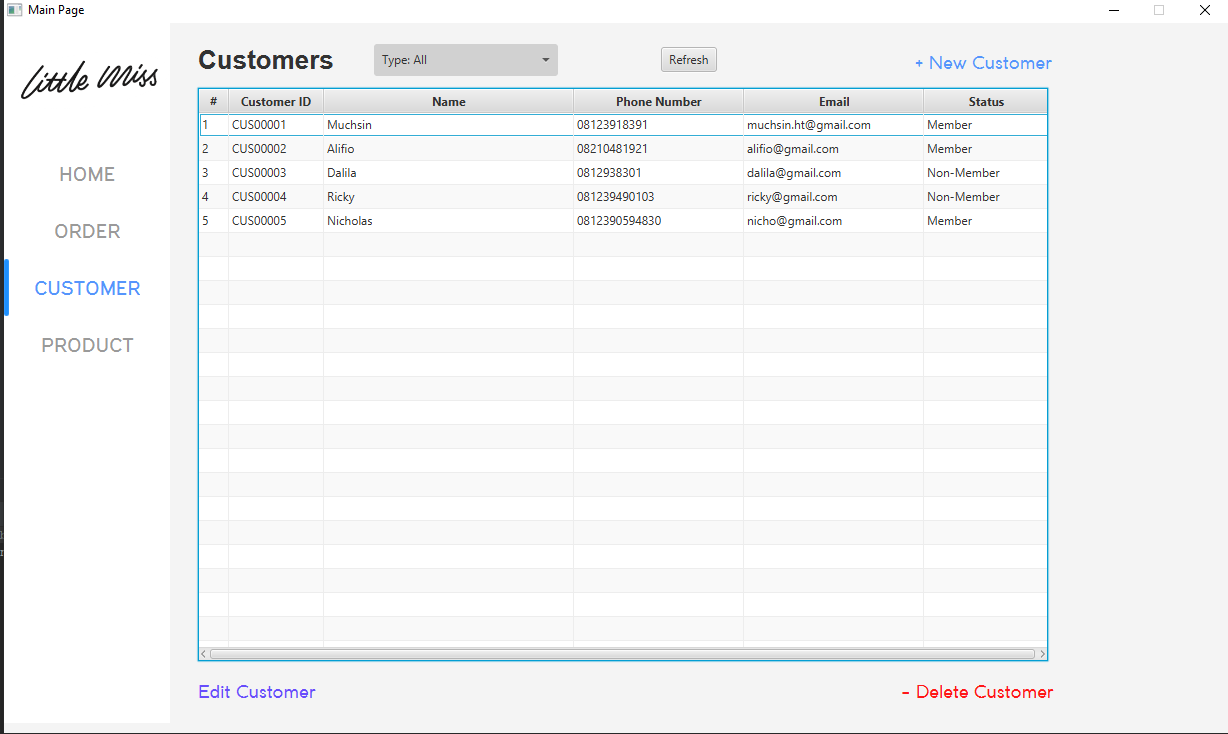
# User Interface (Program Manual)

Our program helps the owner to see their bakery revenue chart based on weekly, monthly, and yearly, and also track bakery order history. The program also store the customer’s, and product’s data to make the owner easier to search and checking the data. The bakery itself have a rules of customer about how to be a member on their bakery. The rules are that you must have 5 or more order history. That’s why we implement the automatic member when the certain person reach the minimum amount of the member requirement.

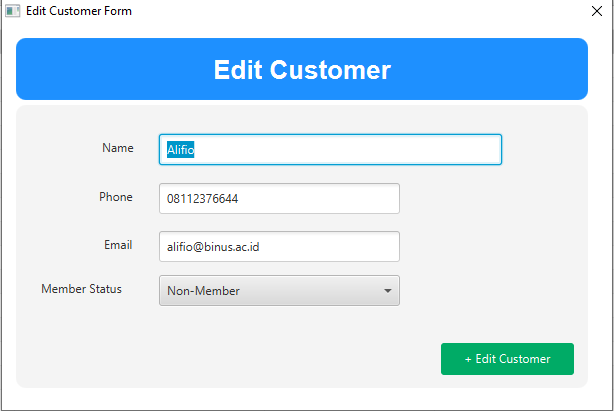
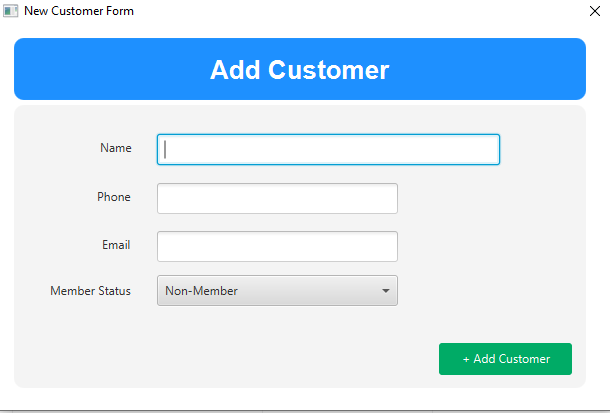
When the program starts it shows the home menu which contains revenue graph or bar chart of the bakery selling revenue. The owner can filter the revenue graph based on what specific data she want to see (e.g. weekly sales, monthly sales, yearly sales by setting the certain start and end date). Our program also helps the owner to add their specific data such as their customer data, product data, and order data from the application form that we already provided.

*Home Page Photo (Revenue Graph)*

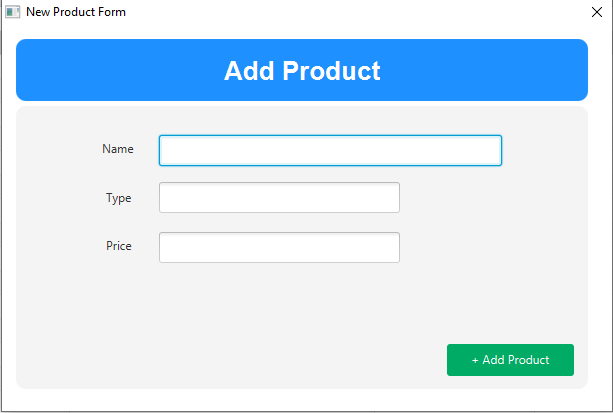
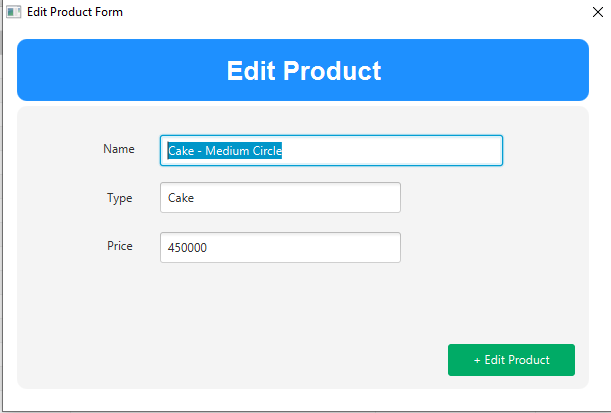
The Customer page shows the customer table and all the columns. Before adding the order data, the owner must add some customers first to the database because our order menu can only get the customer data from the database, otherwise the order form can’t get the customer data. To add a new customer, the user can click on the “*+ New Customer”* button. This will prompt a new windows on which the user can input the new customer’s data. Editing a row of data can be done by selecting the customer on the table and clicking on the “*+ Edit Customer”* button. A deletion of data can be done in a similar way.

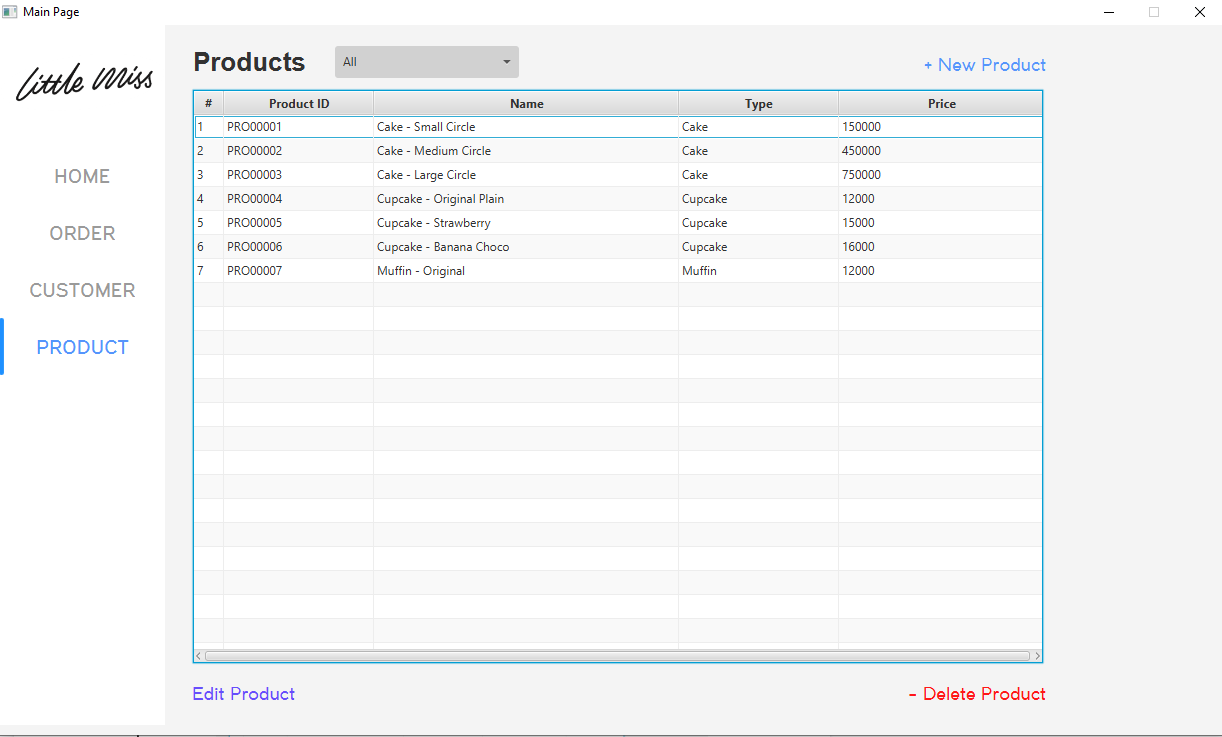
**

*Customer Page Photo*

*Add and Edit Customer Forms Photo*

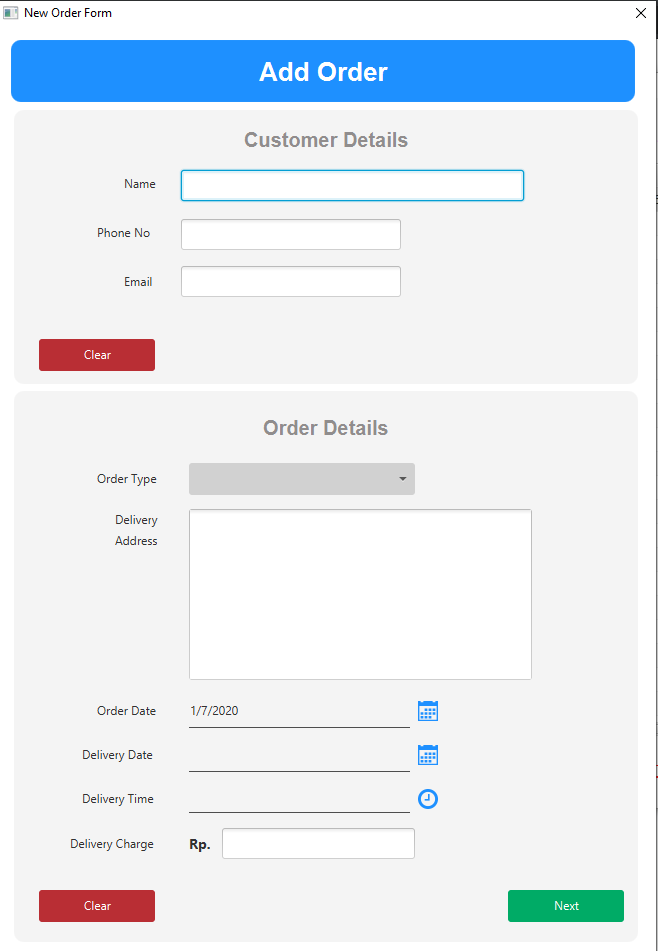
The Product page shows a list of purchasable products. This list is used when creating sub-orders that is a part of the bill, thus the user is expected to add into the Product list first. Here, the user can see all columns on the Product table. The user can add into the table by adding on “*+ New Product”* and inserting the appropriate data. They can also select a product from the table and click on the “*Edit Product”* button. A product can also deleted in a similar way by pressing the “*Delete Product”* button.

*Add and Edit Product Forms*

**

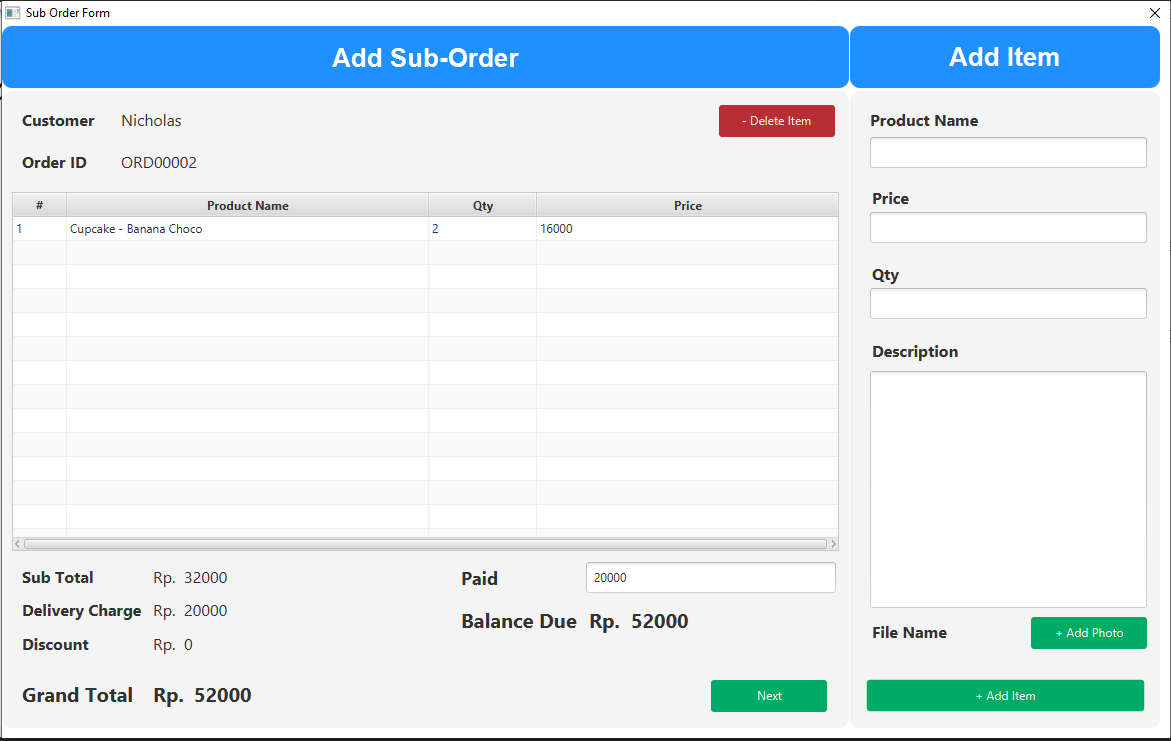
*Product Table Photo*

When the order form opens the owner must fill the specific order data before opens the sub order form.

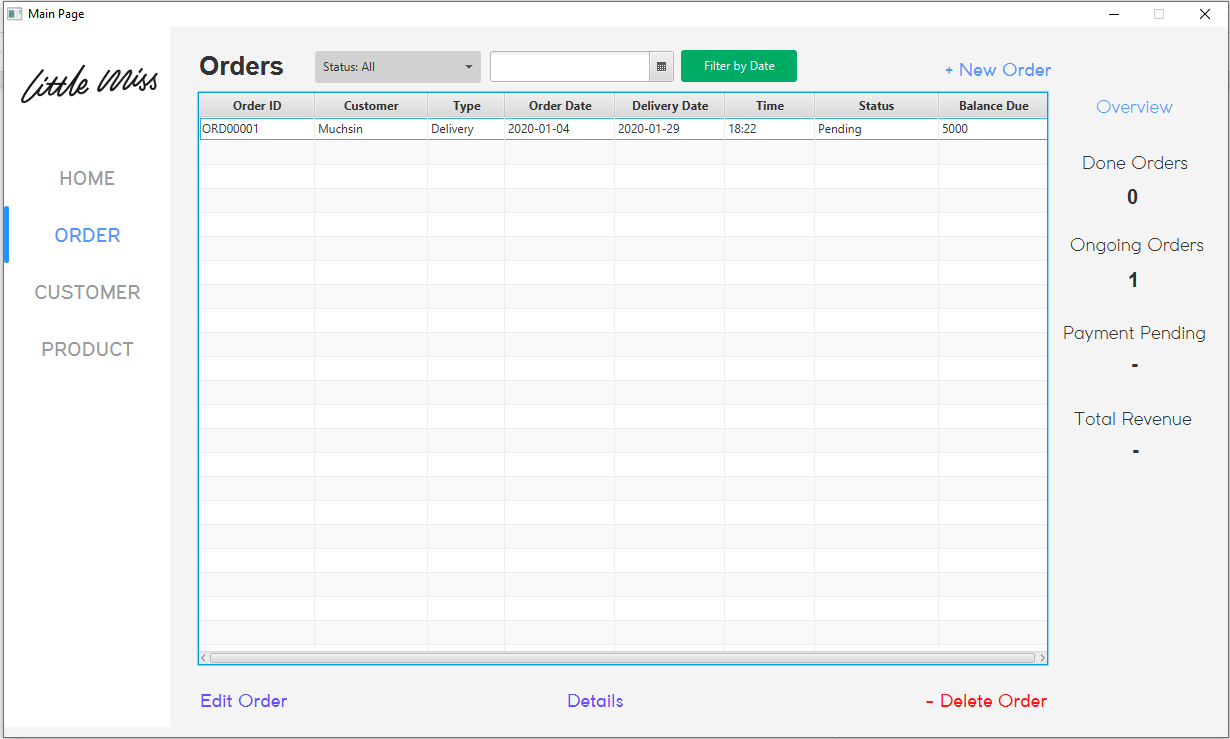
**

*Adding a New Order Forms Photo*

Same like the order form, the owner must add some products first because the sub orders form get the product data from the database. There are also description and description photo for the products if the bakery’s customers want some specific add-ons on their cake. We also provide what the owner needs like the paid text field for tracking how much the certain amount that the customer paid, and vice versa for the balance due.



*Adding Sub Order Forms Photo*

**

*Order Table Photo*

# 

# Libraries/APIs Used

* **Java SDK 8**

We used this particular SDK since it has the JavaFX & Scenebuilder built-in already; this is important as we’ll be using JavaFX as our main UI libraries. Using newer SDKs would mean that we would have to install JavaFX separately.

* **JDBC Driver**

The JDBC Driver is an API that enables us to connect to our mysql database, send queries and update statements as well as retrieving the results and processing it.

* **AnimateFX**

It’s a library with various ready-to-use animations for JavaFX. We used this to animate some of the buttons and transitions in our program.

* **ControlsFX**

A library providing high quality UI control for JavaFX. We used this library as it has it helps us to autobind (autocomplete) textfields easily.

* **JFoenix**

A library providing Google Material Designs using java components. This library was used as it provides excellent looking date & time picker; JavaFX has them built-in, but it wasn’t as pleasing.

# 

# Database Security

# For now, the owner of the business will be the only one operating this application. This is why we only apply security in the form of a local database which is password protected with the help of PhpMyAdmin.

In the future, when the business grows and other employees may be tasked to create new orders, it is possible to apply a login system and disabling the Products and disabling all deletion buttons. This prevents accidental deletion of important rows of data, be it a customer, product, or even an order data. Only the owner’s login information or an admin should have all privileges.

# 

# References

* <https://www.youtube.com/watch?v=PUFNtreMHv8&t=312s>
* <https://gist.github.com/mahimahi42/0f8dc95296e8915c18ae>
* <https://www.youtube.com/watch?v=pI4VHBgFSTc&t=598s>
* <https://www.youtube.com/watch?v=HXEpl7fM7e4>
* <https://www.geeksforgeeks.org/jdbc-drivers/>
* <https://github.com/Typhon0/AnimateFX>
* <https://github.com/controlsfx/controlsfx>
* <https://github.com/jfoenixadmin/JFoenix>

**Link**

* **Git :** [**https://github.com/alibanana/ISYS6169-Final-Project**](https://github.com/alibanana/ISYS6169-Final-Project)
* **Youtube :**
* **Blog,**
  + **Alifio Rasendriya Rasyid :** [**http://2201798295alifio.blog.binusian.org/**](http://2201798295alifio.blog.binusian.org/)
  + **Nicholas Michael Halim :** [**http://2201798761nicholas.blog.binusian.org/**](http://2201798761nicholas.blog.binusian.org/)
  + **Muchsin Hisyam :** [**http://2201797430muchsin.blog.binusian.org/**](http://2201797430muchsin.blog.binusian.org/)
  + **Ricky Anderson :** [**http://2201796636ricky.blog.binusian.org/**](http://2201796636ricky.blog.binusian.org/)