Adamson University
College of Engineering
Computer Engineering Department

A Case Study on
Designing and Developing a Database with a User Interface for a Café Company

In Partial Fulfillment
of the Requirement of the Subject
Database Design and Development

Submitted by:

Angelo, Jamaica Joy S.
Borejon, Jene Reiner N.
Escobar, Angela Shanine G.
Guzman, Nicole Anne R.

Schedule:
Tue/10:30 AM – 1:30 PM/OZ212

Submitted to:

Engr. Jordan Vhane D. Sardalla
Instructor

Abstract

This case study was conducted to examine and test simple connections between a database and front-end application. The researchers designed a user interface that reflects a café business in which the user will use to transact with customers. Given with its relevance to café business, the system aims to acquire the order of a customer and display the transaction with the necessary information related to operations. This study is focused on establishing connections between the database and the used front-end application, which is Visual Studio 2019, specifically C# programming language.

This case study followed the SDLC Waterfall Model. The researchers focused on each phase of this model to properly perform the required progression of a certain stage. Following the model, the researchers developed the project with planning the design, implementation of the design, and integration of the developed system. Later phases are recommended to future follow-up studies. Verification of results from the test conducted are evaluated based on the objectives the researchers aims to achieve.

Results from given test shows a user interface reflecting a café cashier system. One error was observed where a certain table wasn't reflected in the database, but the objectives were accomplished.

Certain commands should be applied in systems to connect both back-end and front-end applications. The front-end application can be used to establish the connection and the back-end syntax once connected to the database.

Acknowledgment

The researchers would like to extend their gratitude to all the people who helped them to make this research study successful.

First is to God whom always guided us, protected us and taught us how to turn challenges into opportunity and turn our weaknesses into strength for us to succeed this kind of research.

The researchers are also grateful to Engr. Jordan Vhane D. Sardalla, our professor. We are extremely thankful to him for sharing expertise, and sincere and valuable guidance and encouragement extended to us.

Last, but certainly not the least, the researchers would also like to thank our parents for giving us strength and for supporting us all throughout.

We researchers always remember that for us to be a responsible researcher we should first need to be a good follower and do what we can, with what we have and where we are because God wants us to become successful and overcome this wonderful opportunity we had.

Table of Contents

List of Tables

List of Figures

Chapter I
The Problem and Its Background

This case study was designed as a point of sale (POS) system for a café business, wherein transactions are being done will be reflecting with the café's database. Aside from a loyal customer, a developed system is also an essential especially with a start-up business. Having a database is a great way to organize information from the customers and also the inventory of the products, similarly, it tracks sales, expenses, and other financial information that matters with the business

A. Background of the Study

According to The Business Dairy, one of the most popular type of business in the Philippines are the café industry, this case study is relevant among aspiring entrepreneurs in starting their café business particularly in maintaining the customer's satisfaction with the café's service and in keeping track with the sales, having a system that allows the café to operate while considering these measures are quite helpful and eases the workload of the employees. This study also offers a simple yet substantial tool which is a good start for business.

B. Objectives of the Study

The main objective of this case study is to allow a user to access a system that would record the collected data from a customer referring to the café's menu. Other specific objectives include the following:

1. To allow a log-in attempt in accessing the system using information from the database.
2. To compute the total amount of the products bought by the customer.
3. To generate an invoice with the information carried out in the transaction for every customer.
4. To develop a database using SQL queries.

C. Significance of the Study

To the students – creating a system like this could tremendously help the students since it allows to maximize the use of databases and an introduction with the Sql Commands and Queries that would enable them to learn and apply with their future careers.

To the community – this case study would provide a preview of what start-up entrepreneurs could expect and be prepared for their upcoming business.

To the future researchers – as the system is created with simple transactions and operations, it offers a huge opportunity to be developed and turn into a platform where it offers a great interaction.

D. Scope and Delimitation

This case study is limited with using Visual Studio and Microsoft SQL Database only, it is also limited within the customer and cashier interaction but could possibly modified with the manager's involvement for deleting orders.

This case study considers every of aspect of each transaction between the customer and the cashier-in-charge with the system, this includes the selection of orders along of its quantity, the computation of the subtotal and total amount with the display of change and whether the amount receive is exact or insufficient, aside from that, the system also update the stocks of the ordered items from the database and fill it with the data gathered from the transaction.

E. Operational Definition of Terms

- Customer – involved during transaction with the user.
- ERD (Entity Relationship Diagram) – it represents the connection or relationship of the tables to another table.
- Error – it is the faults in the program that the researchers encounter
- Events – it is the main process the system will handle depending of what the user will do.
- Front End – The main system the user interacts at.
- Manipulation – is the statement in order to change the values of data whether the system has transaction and to retrieve specific data.
- SQL Database – Database is the one that has the data where the products are stored. It easily to locate the data with the use of database.
- System – is the one that the user uses to manage the menu in which the customer can order.
- User – The user can access or operate the system.

Chapter II
Review of Related Literatures and Studies

This chapter reviews the related literature which support the details of this research. The researcher to different viewpoints and perspectives to present a thorough treatment of the problem. Concepts, findings, theories, and notions, from scholarly researches and articles related to leadership development are presented to provide an extensive background of the study and to justify the researcher's objective in undertaking it.

What is a Database?

A database is an organized collection of structured records, or data, usually stored in a computer system. A database is normally managed by a DBMS (Database Management System). Together, the data and the DBMS are referred to as a database system along with the software associated with them, often shortened to just a database.

Data within the most common types of databases currently in use is usually represented in rows and columns in a series of tables to make data processing and querying efficient. The data can then be easily accessed, managed, modified, updated, monitored and organized. Most databases use structured query language (SQL) to write and query data.

What is Structured Query Language (SQL)?

SQL (Structured Query Language) is a standardized programming language that's used to manage relational databases and perform various operations on the data in them. Originally developed in the 1970s, SQL is used frequently not only by database managers, but also by developers writing data integration scripts and data analysts looking to set up and run analytical queries.

SQL uses include modifying database table and index structures; adding, updating and deleting data rows; and retrieving information subsets from the database for transaction processing and analytics applications. Queries and other SQL operations take the form of commands written as statements — the widely used SQL statements include select, add, insert, update, delete, create, and etc.

SQL became the de facto standard programming language for relational databases after it emerged in the late 1970s and early 1980s. Often known as SQL databases, relational structures consist of a series of tables containing data in rows and columns. Each column in a table corresponds to a data category—for example, the name of the client or the address of the client—while each row contains a data value for the intersecting column.

What’s the Difference between Database and Spreadsheets?

Databases and spreadsheets (such as Microsoft Excel) are both convenient ways to store information. The primary differences between the two are:

- How the data is stored and manipulated
- Who can access the data?
- How much data can be stored?

The spreadsheets were initially designed for one person, and their characteristics reflect that. They 're perfect for a single user or a limited number of users who don't need to do a lot of extremely complex data manipulation. Databases, on the other hand, are designed to hold a much larger collection of organized information — huge amounts, quite often. Databases allow many users to easily and safely access and query data utilizing highly complex logic and language at the same time.

Types of Databases

There are many different types of databases. The best database for a specific organization depends on how the organization intends to use the data.

- Relative repositories. Relational databases became dominant in the 1980's. Items in the relational database are organized as a set of column and row tables. Relational database technology offers the most effective and efficient way of accessing structured information.
- Object-based databases. Information in an object-oriented database is expressed in the form of objects, such as object-oriented programming.
- Distributed databases. A distributed database consists of two or more files located in different locations. The database can be housed on several devices, located at the same physical location, or distributed over various networks.

What is a Database Management System?

A database typically includes a robust database software program known as a database management system (DBMS). The DBMS acts as an interface between the database and its end users or applications, enabling users to access, update and monitor how the information is structured and configured. DBMS also facilitates the supervision and control of databases, enabling a variety of administrative operations, such as performance monitoring, tuning, and backup and recovery.

Examples of popular database applications or DBMS include MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database, and dBASE.

Front End

Database systems are composed of a front-end and back-end. The term front-end refers to the user interface (UI) which consists of the screen forms. It is everything that the user sees including the system's design and languages like C#. It is to provide an interface that a developer designed to fetch and store data that is entered by the system's user that is stored in the database. A front-end developer or designer may create a website without a back-end development. Any information that is going to be stored in a database is not that necessary. Unless the time comes for a redesign, the site will stay the same. The main front-end development languages are HTML, CSS, and Javascript. This is a very vital part on the researchers' case study as it serves as the main interface of the system. The user would then be able to view and access the computing features and integrant of the interface made by the developer.

At the process of coding the syntax, the developer is responsible to guarantee that the database front-end looks exactly as designed prior to coding. He is also responsible for verifying if the system is free from errors. The researchers used Visual Studio and Microsoft SQL Server Studio in creating the system's syntax as well as connecting the database to C#. The syntax was created with the use of the C# language in order to connect it to the SQL server.

Back End

The back-end has the tables that store the data, including the relationships between the tables, data queries and such. It is also where the creation of more logical back-end and computational database logic. The features and components are made by the developer in which are obliquely accessed and viewed by the user through the front-end. To ensure that the data requested by the front-end are transmitted by any programmatic means is the vital role of the back-end. The back-end is the "server side" or the portion of the system or website that people don't see. The back-end system and its entirety which contains the core of databases, application logic, and integration was also created and maintained by the back-end developers. These developers also perform in the executing, testing, and debugging of the back-end application system.

Meanwhile, Structured Query Language or SQL is used to communicate with the database. This scripting language is being used at the back-end of the database in order to create tables, and are also used for querying the information from the database itself. Database software like MongoDB, MySQL, Oracle 12c, and Microsoft SQL Server are the ones powering both big and small companies like banks, hospitals, businesses, et.al.

Online Food Order System for Restaurants

The study "Online Food Order System for Restaurants" states that, "It is known globally that, in today's market, it is extremely difficult to start a new small-scale business and live-through the competition from the well-established and settled owners. In fast-paced time of today, when everyone is squeezed for time, the majority of people are finicky when it comes to placing a food order. The advantage of the system is to simplify the ordering process of the customers and to the restaurant. On this system it will help the restaurant's end to lighten the load for the entire process of taking orders online." This study is related to our case study since the flow of the system that the author did is similar to this study. The said study also included a database that has logical components where it is used on the system such as ordering system, management system and the order retrieval. The order system is where the customer could place their order. Management system controls the customer's order. The order retrieval section allows the café management to keep on track to the placed order. However, this study is accessible online. Future recommendations for this study regarding online access is discussed in chapter 5.

The Waterfall Model

It's really easy to grasp and use. In a waterfall model, each step must be completed before the next phase can begin and there is no overlap in the phases. The Waterfall model is the first SDLC technique used to build apps.

The Waterfall Model displays the software development process in a linear sequential flow. This means that every step of the development cycle can only begin if the previous phase is completed. In this model of the waterfall, the phases do not overlap.

Waterfall approach was the first SDLC model to be commonly used in software engineering to ensure the performance of the project. The entire process of software development is split into different phases in the "Waterfall" approach. Usually, the result of one step in this Waterfall model serves as a sequential reference for the next process.

Waterfall Model — Design

The sequential phases of the Waterfall model are −

- Requirement Gathering and Analysis − All potential system specifications to be created are gathered at this point and recorded in a configuration specification document.
- System Design − The specifications of the first phase are analyzed at this level and the design of the system is prepared. This system design helps to determine hardware and device specifications and helps to define the overall system architecture.
- Implementation − With system design inputs, the system is first developed in small programs called units, which are implemented in the next process. Each device is designed and tested for its functionality, which is referred to as *unit testing*.
- Integration and Testing − All systems created during the implementation process are incorporated into the system after testing of each unit. After installation, the entire system is tested for any defects or failures.
- Deployment of the system − Once functional and non-functional testing has been carried out; the product is deployed in the customer environment or released on the market.

- Maintenance − There are certain problems that emerge in the company climate. Patches are published to correct these issues. Some improved models are also released to boost the product. Maintenance is done to deliver these changes to the customer environment.

All these phases are cascaded to each other, through which progress is seen to flow slowly downward (like a waterfall) through the phases. The next step is only started after the specified goal set has been achieved for the previous phase and is signed off, so the name "Waterfall Design" is used. The phases in this model do not overlap.

Waterfall Model — Application

Each software produced is different and needs an effective SDLC approach to be followed on the basis of internal and external factors. Several circumstances in which the use of the Waterfall model is most fitting are:

- Requirements are well known, simple and set.
- Production definition is stable.
- Technology is known and not complex.
- There are no uncertainty criteria.
- Ample resources with the requisite skills are available to support the company.
- The project is short.

Advantages of Using the Waterfall Model

The advantages of waterfall development are that it enables departmentalization and control. The timeline can be set with time limits for each stage of production and the product can progress one by one through the stages of the model development cycle.

Design shifts from concept to design, development, testing, deployment, troubleshooting, and ends up in service and maintenance. Each stage of development proceeds in strict order.

Several of the key benefits of the Waterfall System are as follows:

- Simple and quick to understand and to use
- Simple to manage due to the structure of the platform. Each step involves unique goals and objectives and a review process.
- The phases will be processed and completed one at a time.
- Works well for smaller projects where requirements are well known.
- Stages clearly defined.
- Well-understood milestones.
- Easy to set up tasks.
- System and effects are well documented.

Disadvantages of Using the Waterfall Model

The downside of waterfall construction is that it does not require much analysis or revision. If an application is in the development stage, it is very difficult to go back and modify something that was not well established or thought about at the design level. The major disadvantages of the Waterfall Model are as follows:

- No working software is developed until late in the life cycle.
- High levels of risk and uncertainty
- Not a good model for complex, object-oriented projects.
- Bad model for long-term and ongoing programs.
- Not suitable for projects where there is a moderate to high risk of change in requirements. As a consequence, the risk and complexity of this process model is high.
- Progress is difficult to measure in stages.
- Cannot meet changing requirements
- A project can be terminated by changing the reach during the life cycle.
- Integration is done as a 'big-bang.' at the very end, which does not allow the early identification of any technological or business bottleneck or challenges.

What is a Unified Modeling Language (UML) Diagram?

UML is an acronym for the Unified Modeling Language. Simply put, UML is a modern approach to software modeling and recording. In reality, this is one of the most common business process modeling techniques.

What is the Use of the UML?

UML was mainly used as a general-purpose modeling language in the field of software engineering. However, a number of business processes or workflows have now been documented. For instance, activity diagrams, a form of UML diagram, can be used to replace flowcharts. They provide both a more streamlined way of modeling workflows and a wider range of features to improve readability and performance. UML itself finds different uses in software development and business process documentation:

Sketch

In this case, UML diagrams are used to explain the various aspects and characteristics of the device. However, this is just a top-level view of the program and will most likely not provide all the information required to operate the project until the very end of the project.

- Forward Design – The layout of the sketch is done before the application is encoded. This is done to get a clearer view of the program or workflow you are trying to build. Many design problems or defects may be detected, thereby improving overall safety and well-being of the project.

- Backward Design – UML diagrams are drawn after writing the code as a means of documentation for the various tasks, roles, actors and workflows.

Blueprint

In such a scenario, the UML diagram acts as a complete design that only includes the actual implementation of the program or software. It is mostly done using CASE (Computer Aided Software Engineering Tools) software. The biggest downside to using CASE tools is that they require a certain degree of experience, user training as well as management and dedication to staff.

Pseudo Programming Languages

UML is not a stand-alone programming language like Java , C++ or Python, but with the necessary equipment it can turn into a pseudo programming language. To do this, the whole framework has to be written in various UML diagrams and the illustrations can be directly converted into code using the appropriate software. This approach can only be effective if it takes less time to draw the diagrams than to write the actual code.

Despite the creation of UML for the modeling of software systems, several adoptions have been found in business fields or non-software systems.

What is an Entity Relationship Diagram (ERD)?

The Database is an integral part of the software system. In order to fully use ER Diagram in database engineering, you are required to produce high-quality database design for use in database development, management and maintenance. The ER model also offers a means of communication.

Entity Relationship Diagram, also known as ERD, ER Diagram or ER model is a type of structural diagram to be used in database design. The ERD includes various symbols and connections that represent two essential information: the main entities within the framework of the program, and the inter-relationships between those entities.

ERD entities, also refer to business objects such as people/ roles (e.g. students), tangible business objects (e.g. products), intangible business objects (e.g. logs), etc. *Relationship* is about how these entities relate to each other within the program.

When to Draw ER Diagrams?

While ER models are mostly created for the design of relational databases in terms of concept visualization and physical database design, there are still other situations where ER diagrams can help. Here are some common cases of use:

- Database Design - depending on the scale of the change, it may be risky to modify the database structure directly in the DBMS. In order to prevent the destruction of the data in the database system, it is necessary to plan the changes carefully. The ERD is a tool that helps you. Through drawing ER diagrams to imagine database design concepts, you have a chance to detect mistakes and design defects and make adjustments before you make improvements to the database.
- Database Debugging - debug database issues can be difficult, particularly when the database includes several tables that require extensive SQL writing to get the information you need. You have a clear image of the entire database schema by visualizing the data model with the ERD. You can easily locate entities, view their attributes, and identify relationships with others. All of these make it easier for you to analyze the existing database and discover problems with the database.
- Database creation and patching - Visual Paradigm, an ERD tool, supports a database generation tool that can optimize database creation and patching using ER diagrams. So, with this ER Diagram method, the ER template is no longer just a static diagram, but a mirror that actually represents the physical database structure.
- Aid in gathering specifications - Determine the requirements of the information process by drawing up a conceptual ERD that represents the high-level business artifacts of the system. Such an initial prototype can also be transformed into a physical database model that facilitates the creation of a relational database or aids in the creation of process maps and data flow modes.

ERD Notations

Entity

An ERD entity is a definable thing or concept within a system, such as a person/role (e.g. student), object (e.g. invoice), concept (e.g. profile) or event (e.g. transaction) (note: in ERD, the term "entity" is often used instead of "table" but it is the same). Think of them as nouns when deciding entities. In ER models, the entity is shown as a rounded rectangle, with its name at the top and attributes in the body of the entity shape. The following ERD example provides an example of an ER object.

Entity Attributes

Also known as a column, an attribute is a property or characteristic of the entity that holds it. An attribute has a name that describes the property and a type that describes the kind of attribute it is, such as varchar for a string, and int for integer. When an ERD is drawn for physical database development, it is important to ensure the use of types that are supported by the target RDBMS.

Primary Key

Also known as PK, a primary key is a special kind of entity attribute that uniquely defines a record in a database table. In other words, there must not be two (or more) records that share the same value for the primary key attribute.

Foreign Key

Also known as FK, a foreign key is a reference to a primary key in a table. It is used to identify the relationships between entities. Note that foreign keys need not be unique. Multiple records can share the same values.

Relationship

A connection between two entities means that the two entities are somehow connected with each other. For instance, a student may be enrolled in a course. The entity Student is therefore related to Course, and the relationship is presented as a connecting link between them.

Cardinality

Cardinality defines the possible number of occurrences in one entity that is affiliated with the number of occurrences in another entity. For reference, one team has a lot of players. When present in the ERD, the Team member and the Player are inter-connected with a one-to-many relationship.

The cardinality is represented in the ER diagram as the crow 's foot at the ends of the connector. The three common cardinal relations are one-to-one, one-to-many, and many-to-many.

- One-to-one Cardinality - A one-to-one relationship is mostly used to split an entity in two to provide information concisely and make it more understandable.
- One-to-many Cardinality - A one-to-many relationship refers to the relationship between two entities X and Y in which an instance of X may be linked to many instances of Y, but an instance of Y is linked to only one instance of X.
- Many-to-many Cardinality - A many-to-many relationship refers to the relationship between two entities X and Y in which X may be linked to many instances of Y and vice versa. Note that a many-to-many relationship is split into a pair of one-to-many relationships in a physical ERD

Chapter III
Methodology

This chapter contains the development of the design and arrangement of the system for this case study. The researchers observed the Waterfall Model for the flow of the creation of the system. The following will cover the progress of the students on this case study.

A. General Method Used

The researchers used the Waterfall Model for the development of the system. This model is simple and easy to understand since the system designed by the researchers can be used by young entrepreneurs or individuals who wants to start a simple business, specifically a café business. Mainly, this model is used to focus on each phase the researchers were currently working on as they progress to finish the system. Each phase of this model has its own purpose and it can be worked on efficiently. Other alternative models can also be considered with this type of system since it can be advance for future improvements. Alternative model like the Agile Model can be used where the program may progress continuously and can be tested for following weeks.

B. Procedure

The procedure that follows contains the phases included in the Waterfall Model. The researchers focused on each phase of the model used for this case study to better perceptualize the output the design was expected to be produced.

1. Design Phase

The first phase was focused with the gathering of the requirements for the system to run and analyzing its contents. The main point of view for the design is from the cashier of the café business. The researchers visualized the user interface that was used for the system. The cashier in-charge of the system will login with the correct credentials to gain access inside the system. After logging in the correct credentials, a form of menu selection will appear where the cashier will indicate or select the choice of the customer and will be displayed from it the total price from the quantity they will input. If the choice is final, the user can now confirm the order and will display the receipt.

The flow of the system mentioned above was finalized by the researchers. The next step from this phase was to create a UML diagram to further perceptualize the required data needed by the system. The credentials needed to login to the system was the first part of the created diagram to limit the access only for the staffs. Data required to such credentials mainly includes the cashier name and status. This was followed by the products which serves as the main entity for the system. The products are consisted of coffee, doughnuts and miscellaneous items that are part of the selection menu. Such products include the stock available inside the database, or the researchers called it inventory, and their price equivalent. The next part of the diagram was consisted of the order preview which will display the selected products by the user and its subtotal amount. This part of the diagram has its relationship with the product part since the subtotal amount of the selected items will be coming inside the data provided in the product table. This part will also be containing the amount the customer will pay and automatically compute for the change. The final part of the diagram was the receipt part where the final decision from the user will display the final order and its total amount. Necessary information will be also included in this part since it acts as a receipt and can be directly printed.

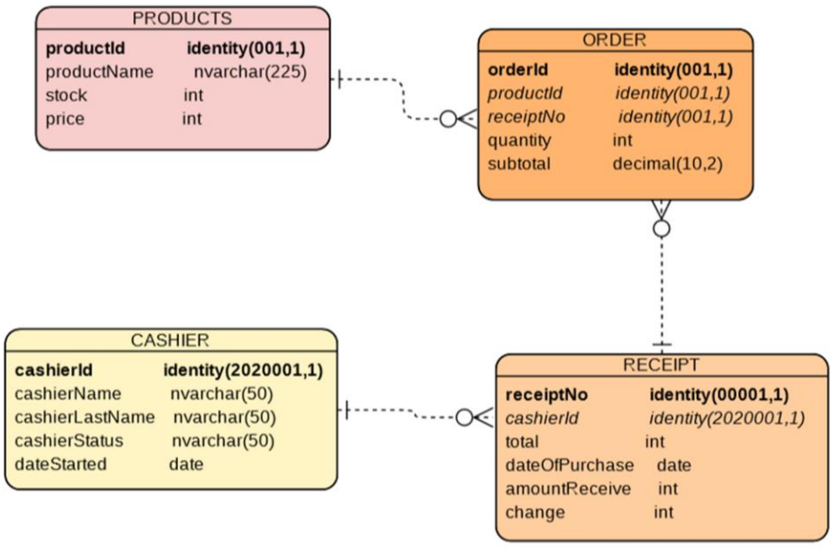

PRODUCTS
productId identity(001,1)
productName nvarchar(225)
stock int
price int
ORDER
orderId identity(001,1)
productId identity(001,1)
receiptNo identity(001,1)
quantity int
subtotal decimal(10,2)
CASHIER
cashierId identity(2020001,1)
cashierName nvarchar(50)
cashierLastName nvarchar(50)
cashierStatus nvarchar(50)
dateStarted date
RECEIPT
receiptNo identity(00001,1)
cashierId identity(2020001,1)
total int
dateOfPurchase date
amountReceive int
change int

Figure 1: UML Diagram (ERD)

The researchers came up with this kind of design based on their own observation in café shops. The user interface for the system is reflected to any café shops based also on their observations.

2. Implementation Phase

This phase was focused in developing the system using two programming languages. SQL (Structured Query Language) for the backend and C# for the frontend. The backend was developed to create a database containing the essential data needed by the system. Microsoft SQL Server Management Studio 2018 was the application used by the researchers in developing the backend. After executing the query file, the database was created for the system including primary and foreign keys to form a relationship between the tables.

The frontend was created using Microsoft Visual Studio 2019. The researchers implemented a project using windows form application which consisted of two forms. Inside the forms are several controls that is capable in handling events as the user wishes what to do. The first form was for the login of a staff. From the created database, the system will check if the credentials inputted are acceptable based on the correct information the database holds. Before establishing this part, the researchers observed how to connect the database to the frontend. Using certain commands like SqlCommand and SqlDataAdapter, the researchers were able to establish the connection between the backend and frontend. SQL statements were used inside the controls to select the required data for the login form. The second form was for the selection menu and receipt. This form contains the products the user will select based on the customer's order and the preview of the receipt. This form is also consisted of controls that allows the user to select commands depending on their choice or the transaction. A connection was also established for this form to connect the frontend to the database allowing the system to read or update the stocks inside the product table. A dropdown list is consisted of the products that can be selected. After selecting the product, its price will be displayed from a textbox. The user can now input the quantity the customer wishes to have. The subtotal of the product is automatically calculated by clicking a button control to preview its price before adding it to the invoice. The user can now confirm the order and may select another product before the payment. The user may also delete one of the orders from the invoice by inputting the correct string inside the specific text box and clicking the specific button. If orders are finalized, the user may click the button that will confirm the request of the customer which automatically computes its total price. The user may now input the payment the user will provide and automatically computes for the change. The orders and payment will now be reflected in the invoice portion where the transaction can be finalized by clicking the final button control. The update of stocks can now be reflected inside the database.

The researchers faced several errors regarding the syntax. While achieving to correct the errors, changes were also applied in the system including the user interface and transaction process.

3. Testing and Evaluation Phase

Testing the system follows simple instructions to see if the main objective can be achieved. The students started to input the correct cashier ID in the text box inside the first form. They also tested if it will not accept any values other than the content of the cashier table in the database. Then, the students started to select three products from the dropdown box and inputted the sample quantity to display the subtotal price. By clicking the button "order checkout" which reflects the order in the invoice, the students started to test if the order can be deleted from the invoice. After deleting one product, the students inputted sample payment to check if it can automatically compute for the change. The previous steps were reflected in the invoice and can be finished by simply clicking the "finish transaction" button.

Below is the table that served as checklist to the evaluation phase of the developed system. The test is considered as acceptable if the test satisfies the objectives and non-acceptable if not.

Test	Acceptable	Non-acceptable
Used username and password for logging in the system		✓

Used cashier ID for logging in the system	✓	
Message box appeared if cashier status is “Off Duty”	✓	
Message box appeared if inputted value doesn’t exist in database	✓	
Selected product displayed its respective price	✓	
Subtotal automatically computed and displayed based on quantity value	✓	
Selected product is displayed in invoice panel	✓	
System allows additional selection of products from drop-down menu	✓	
Selected order in invoice panel can be deleted	✓	
Inputted string “dddcafedabest” deletes selected order	✓	
“Order checkout” button displayed total price in text box	✓	
Sample payment automatically computed the change	✓	
Order is displayed in invoice panel with total amount and change	✓	
Data in database reflected the stocks’ quantity	✓	
Final transaction finishes the order and displayed message box	✓	
Exit button exits the application/system	✓	

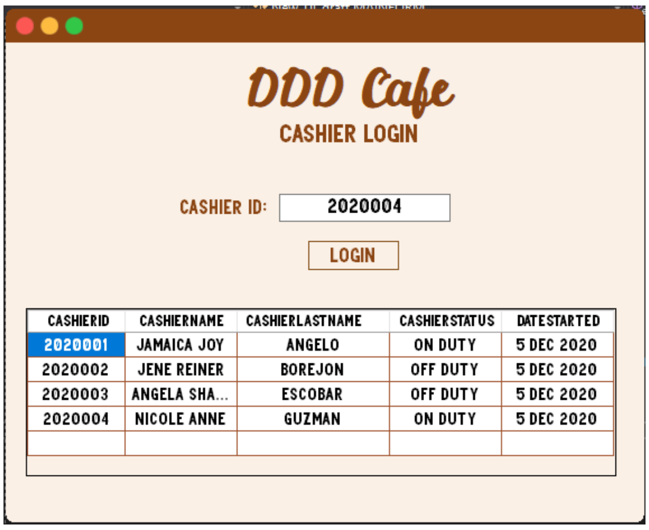
Table 1: Evaluation checklist

Chapter IV
Results and Discussions

This chapter will discuss the developed system and the expected results discussed from the previous chapter. This chapter will also present the visual representation of the developed system. The previous checklist for verifying the system is also discussed in this chapter with display output.

A. The Developed System

As discussed in the previous chapter, the user interface will be coming from the point of view of the cashier. Below is the actual developed output consisted of two forms which are the login form and the menu selection form.

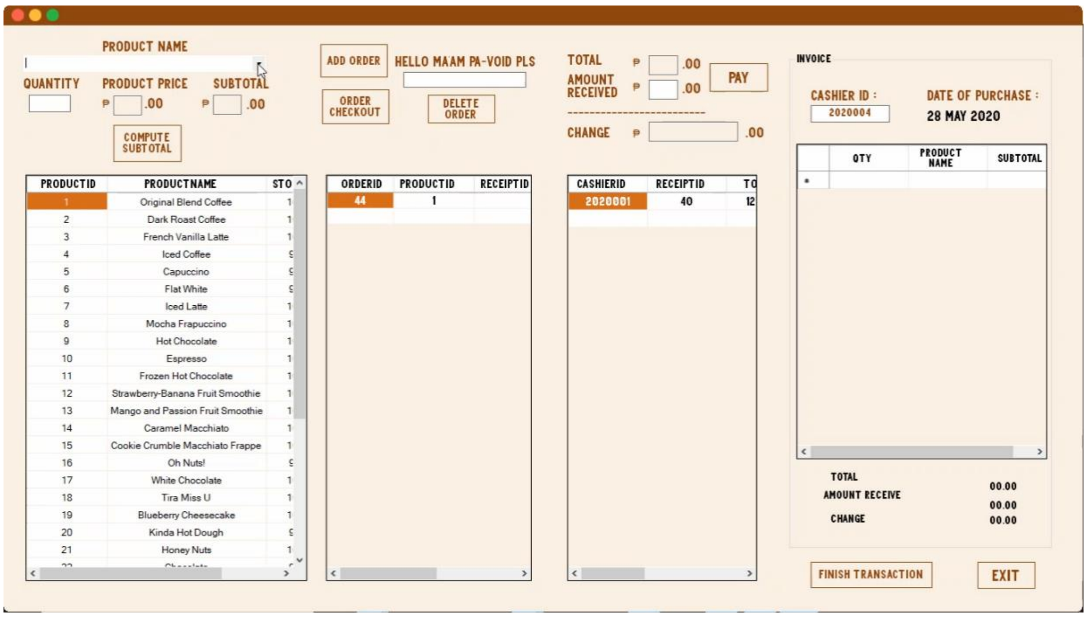


CASHIER ID: 2020004

CASHIERID	CASHIERNAME	CASHIERLASTNAME	CASHIERSTATUS	DATESTARTED
2020001	JAMAICA JOY	ANGELO	ON DUTY	5 DEC 2020
2020002	JENE REINER	BOREJON	OFF DUTY	5 DEC 2020
2020003	ANGELA SHA...	ESCOBAR	OFF DUTY	5 DEC 2020
2020004	NICOLE ANNE	GUZMAN	ON DUTY	5 DEC 2020

Figure 2: Log-in form

The text box in the login form should be inputted by the correct cashier ID shown in the data grid table below the “login” button. The user can gain access if the correct credential is entered. The system will not accept other values other than stored values inside the database.



PRODUCT PRICE ₱ .00 SUBTOTAL ₱ .00

TOTAL ₱ .00

AMOUNT RECEIVED ₱ .00

CHANGE ₱ .00

PRODUCTID	PRODUCTNAME	STO
1	Original Blend Coffee	1
2	Dark Roast Coffee	1
3	French Vanilla Latte	1
4	Iced Coffee	9
5	Capuccino	9
6	Flat White	9
7	Iced Latte	1
8	Mocha Frapuccino	1
9	Hot Chocolate	1
10	Espresso	1
11	Frozen Hot Chocolate	1
12	Strawberry-Banana Fruit Smoothie	1
13	Mango and Passion Fruit Smoothie	1
14	Caramel Macchiato	1
15	Cookie Crumble Macchiato Frappe	1
16	Oh Nuts!	9
17	White Chocolate	1
18	Tira Miss U	1
19	Blueberry Cheesecake	1
20	Kinda Hot Dough	9
21	Honey Nuts	1

ORDERID	PRODUCTID	RECEIPTID
44	1	

CASHIERID	RECEIPTID	TO
2020001	40	12

CASHIER ID : 2020004

DATE OF PURCHASE : 28 MAY 2020

	QTY	PRODUCT NAME	SUBTOTAL
*			

TOTAL 00.00

AMOUNT RECEIVE 00.00

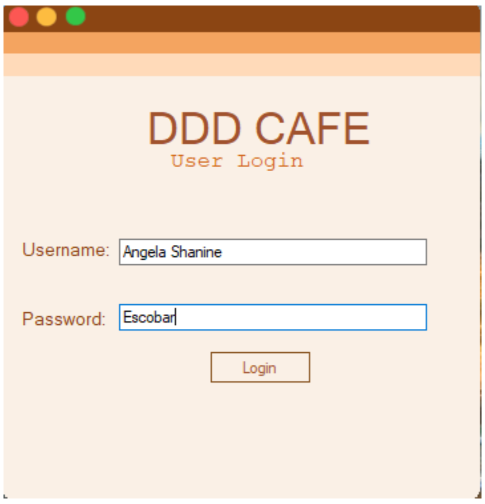
CHANGE 00.00

Figure 3: Main form (menu selection)

Once accepted, the menu selection form will appear consisted of several controls. Each control has its own purpose depending on its label.

B. Verification and Testing Result

The first test in the system was to validate the correct credentials to gain access in the menu selection form. The researchers first tried to gain access by inputting the cashier first name and last name to validate if they're recognized by the system.

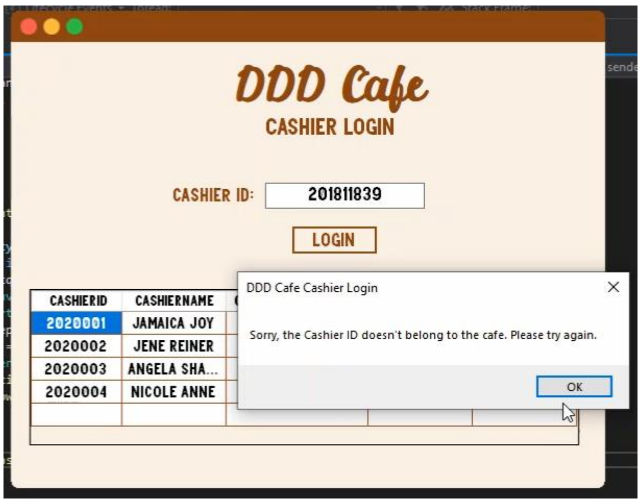

DDD CAFE
User Login

Username: Angela Shanine

Password: Escobar

Figure 4: First log-in form draft using cashier name and last name

Then after, the researchers then changed the login access into inputting the cashier ID instead of the name. The system shows a message box whether the inputted value was not inside the database or the cashier is off duty.


DDD Cafe
CASHIER LOGIN

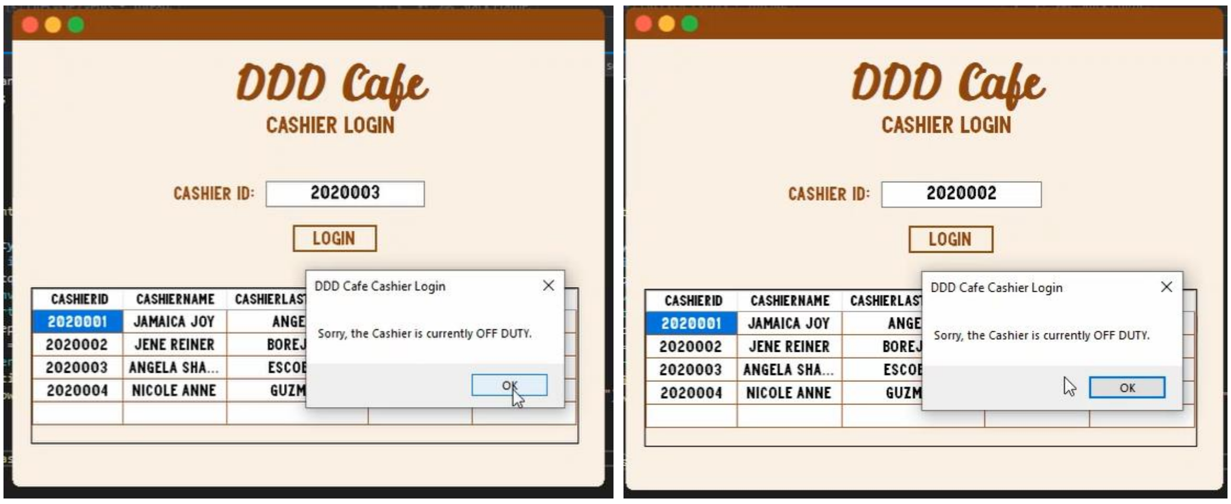
CASHIER ID: 201811839

CASHIERID	CASHIERNAME
2020001	JAMAICA JOY
2020002	JENE REINER
2020003	ANGELA SHA...
2020004	NICOLE ANNE

DDD Cafe Cashier Login

Sorry, the Cashier ID doesn't belong to the cafe. Please try again.

Figure 5: Cashier Id entered is not in the database


DDD Cafe
CASHIER LOGIN

CASHIER ID: 2020003

CASHIERID	CASHIERNAME	CASHIERLAS
2020001	JAMAICA JOY	ANGE
2020002	JENE REINER	BOREJ
2020003	ANGELA SHA...	ESCOB
2020004	NICOLE ANNE	GUZM

DDD Cafe Cashier Login

Sorry, the Cashier is currently OFF DUTY.

DDD Cafe
CASHIER LOGIN

CASHIER ID: 2020002

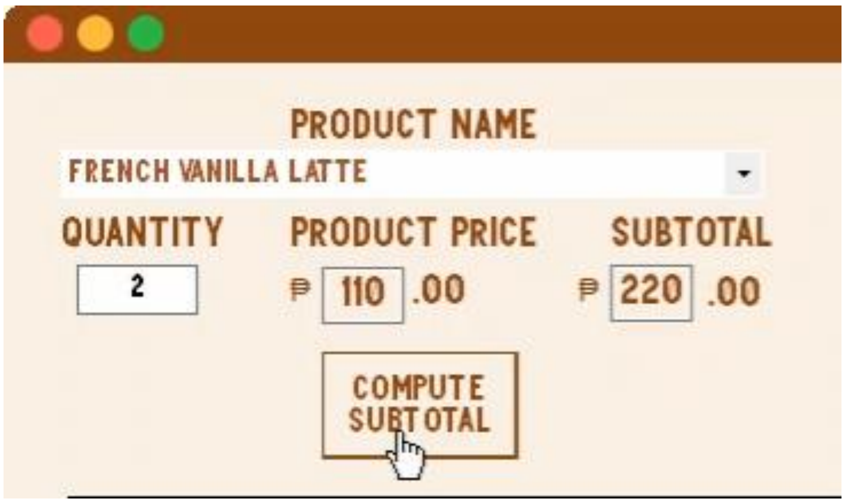
CASHIERID	CASHIERNAME	CASHIERLAS
2020001	JAMAICA JOY	ANGE
2020002	JENE REINER	BOREJ
2020003	ANGELA SHA...	ESCOE
2020004	NICOLE ANNE	GUZM

DDD Cafe Cashier Login

Sorry, the Cashier is currently OFF DUTY.

Figure 6: Cashier Id entered has an 'Off Duty' cashier status

After gaining access in the system, the second test in the system was to select products inside the drop-down box. The price should appear depending on the product selected inside the indicated text box. After selecting the product, the researchers tested if it will automatcally compute for the subtotal depending on the quantity the user will input.



PRODUCT NAME: FRENCH VANILLA LATTE

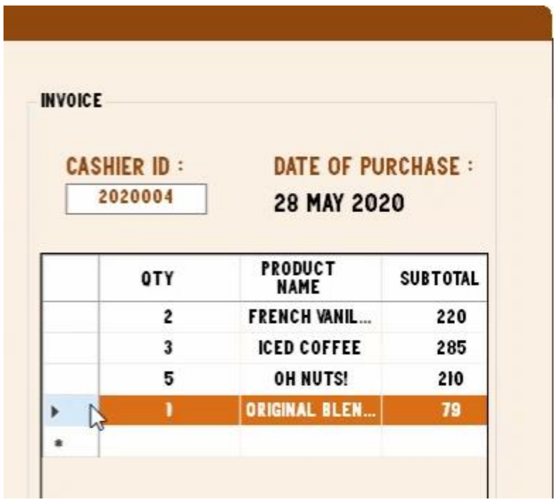
QUANTITY: 2

PRODUCT PRICE: ₱ 110 .00

SUBTOTAL: ₱ 220 .00

Figure 7: Display of product price and computation of subtotal

The next test was to check if orders can be deleted from the receipt preview by clicking it and indicating the correct string in the text box.



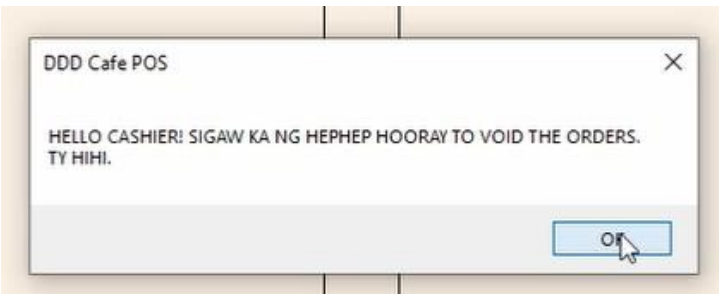
INVOICE

CASHIER ID : 2020004

DATE OF PURCHASE : 28 MAY 2020

	QTY	PRODUCT NAME	SUBTOTAL
	2	FRENCH VANIL...	220
	3	ICED COFFEE	285
	5	OH NUTS!	210
▸	1	ORIGINAL BLEN...	79
*			

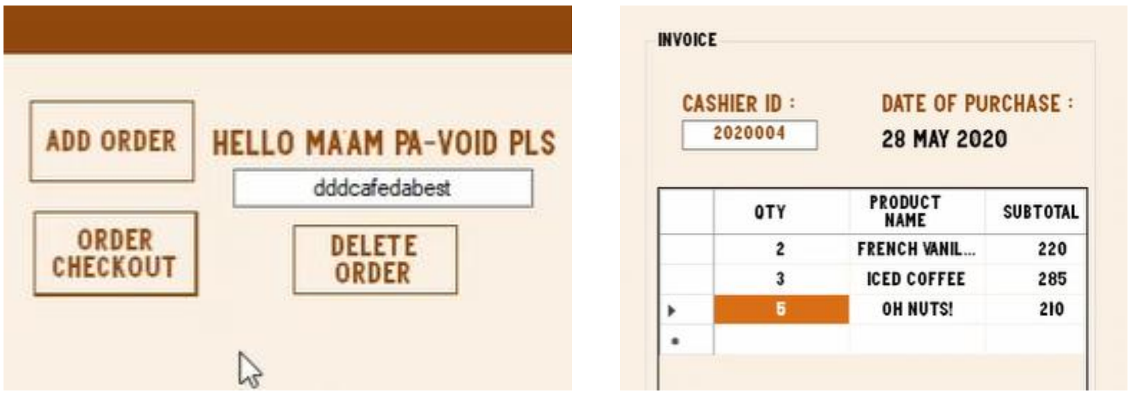
Figure 8: Selection of the product to be deleted



DDD Cafe POS

HELLO CASHIER! SIGAW KA NG HEPHEP HOORAY TO VOID THE ORDERS.
TY HIHI.

Figure 9: Message box shown when incorrect string is entered



HELLO MA'AM PA-VOID PLS: dddcafedabest

INVOICE

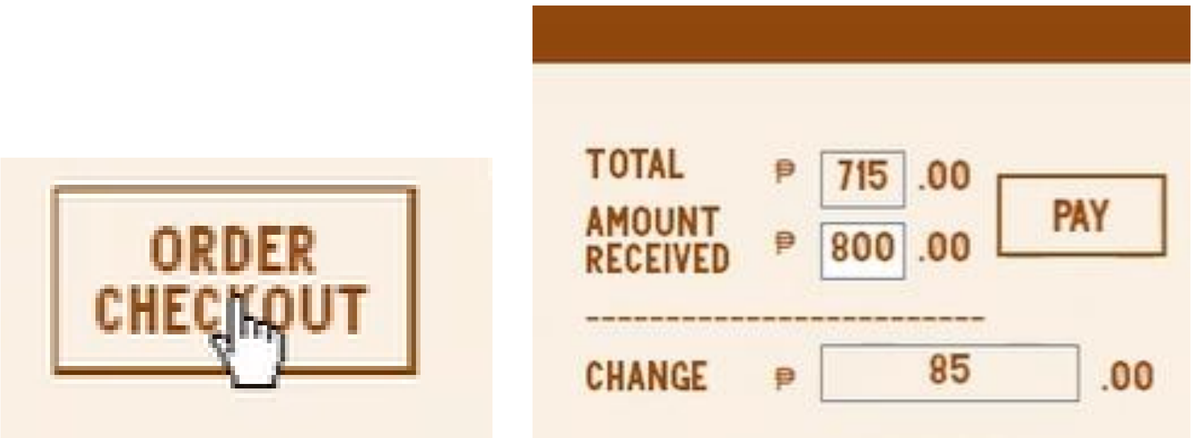
CASHIER ID : 2020004

DATE OF PURCHASE : 28 MAY 2020

	QTY	PRODUCT NAME	SUBTOTAL
	2	FRENCH VANIL...	220
	3	ICED COFFEE	285
▸	5	OH NUTS!	210
*			

Figure 10: Order deleted upon entering the correct string

The “order checkout” button will now compute the total amount of the selected products. The researchers then inputted a sample payment and checked if the change will automatically be computed.



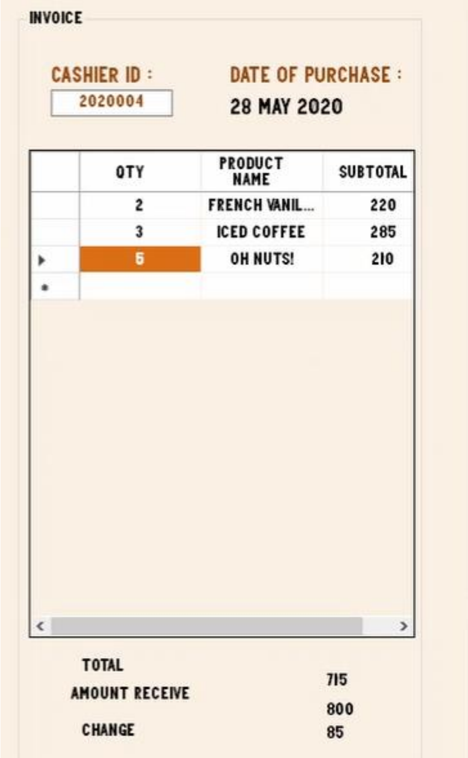
TOTAL ₱ 715 .00

AMOUNT RECEIVED ₱ 800 .00

CHANGE ₱ 85 .00

Figure 11: Total amount displayed through order checkout button with the payment and change

The researchers then observed that all event has been handled by the system and was reflected in the receipt panel.



INVOICE

CASHIER ID : 2020004

DATE OF PURCHASE : 28 MAY 2020

	QTY	PRODUCT NAME	SUBTOTAL
	2	FRENCH VANIL...	220
	3	ICED COFFEE	285
▸	6	OH NUTS!	210
*			

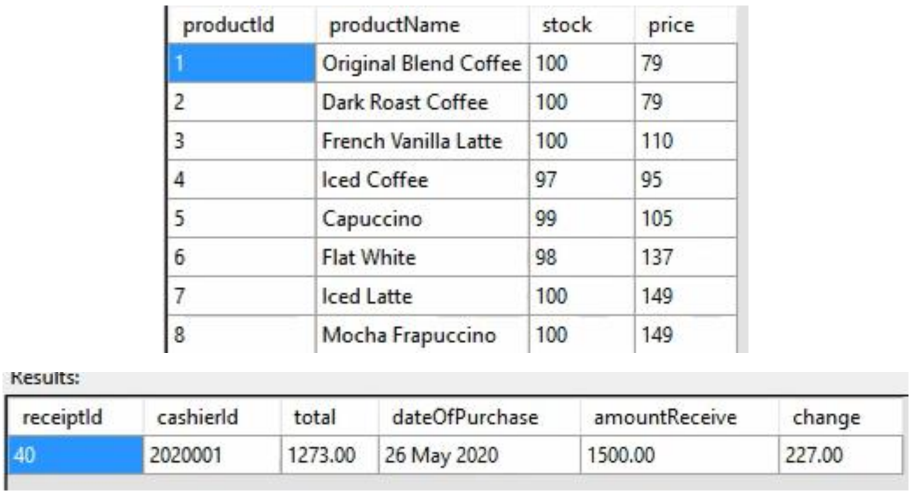
TOTAL 715

AMOUNT RECEIVE 800

CHANGE 85

Figure 12: Invoice panel

Data inside the database were reflected based on the event handled by the system. The only exception for this is the dbo.Orders table.



productId	productName	stock	price
1	Original Blend Coffee	100	79
2	Dark Roast Coffee	100	79
3	French Vanilla Latte	100	110
4	Iced Coffee	97	95
5	Capuccino	99	105
6	Flat White	98	137
7	Iced Latte	100	149
8	Mocha Frapuccino	100	149

Results:

receiptId	cashierId	total	dateOfPurchase	amountReceive	change
40	2020001	1273.00	26 May 2020	1500.00	227.00

Figure 13: Reflected table in database

Chapter V
Summary of Findings, Conclusions and Recommendations

This chapter will conclude the researchers case study and further discuss recommendations on how to update or improve the system.

A. Summary of Findings

The researchers observed that SQL statements can be directly assessed in the frontend application besides stored procedures. Such statements should be read in certain commands of frontend applications inside strings. In this case, SqlDataAdapter and SqlCommand were the commonly used commands. Events in the running system can update and read the data inside the database if used correctly. The objectives of this study were also achieved since the researchers have established the target system.

B. Conclusions

The main purpose of this case study is to allow the user to access the system that would record the data from the customer. The researchers conclude that accessing data from the database is can be done in frontend applications. In this process, the researcher used the SQL to create a database that been used to access the system by the user. The user can access the system by using the correct statements. Using the queries, the researchers validate the user where the user is the only one who can access the system. Queries are the ones that calculate the customer's purchase when they used insert, deletion and update on the database. The connection between the database and the frontend application will also depend on the server of the personal computer is using. It is important to observe that first to avoid connection errors.

C. Recommendations

Based on the results of this study, the researchers can recommend the following to future researchers:

1. Add additional options inside the system including the update of stocks when it is nearly empty.
2. Manage the data inside the database using the system instead of manual manipulation.
3. Include queue portion that indicates the time the customer can get their products.
4. Support online access for those who will use this system in online businesses or transactions.

Appendices

Appendix A
References

[1] 2019. *How To Get Data From Sql Database To Textbox Directly In C# Using Connectionstring*. [online] Available at: <https://www.youtube.com/watch?v=Zgu_PF0tFf0&feature=youtu.be>.

[2] 2016. *C# Fill Combobox From SQL Database, With Database Creation, Data Insert In The Table*. [online] Available at: <https://www.youtube.com/watch?v=ndIFlqd7X_M&feature=youtu.be>.

[3] 2016. *Add Subtract Multiply Division Two Textbox Values Show In Third Textbox Simple C#*. [online] Available at: <https://www.youtube.com/watch?v=6z6MI3eYcXo&feature=youtu.be>.

[4] 2013. *C# Tutorial 11:Database Values In Textbox If Select Combobox*. [online] Available at: <https://www.youtube.com/watch?v=dwG6eg5UofI&feature=youtu.be>.

[5] C-sharpcorner.com. 2017. *Add And Multiply Two Textbox Values And Automatically Display The Result In Third Textbox In C#*. [online] Available at: <https://www.c-sharpcorner.com/blogs/add-and-multiply-the-two-textbox-values-and-automaticallydisplay-the-third-textbox-in-c-sharp>.

[6] Waguih, M., 2019. *How To Display Data From Sql Database To Textbox Using C#.Net - Codeproject*. [online] Codeproject.com. Available at: <https://www.codeproject.com/Questions/155444/how-to-display-data-from-sql-database-to-textbox-u> [Accessed 27 May 2020].

[7] Khan, M., 2018. *How To Minus Stock From Inventory Tables - Codeproject*. [online] Codeproject.com. Available at: <https://www.codeproject.com/Questions/782466/how-to-minus-stock-from-inventory-tables>.

[8] Simplylearnprogramming.com. 2017. *Learn Programming*. [online] Available at: <http://www.simplylearnprogramming.com/insert-update-delete-select-data-sqlserver-csharp/41> [Accessed 27 May 2020].

[9] Gale, B., Bychenko, D. and S, N., 2014. *Sum Two Textboxes Whenever Button Is Clicked*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/21303512/sum-two-textboxes-whenever-button-is-clicked>.

[10] 2015. *How To Add Textbox Value Into Gridview*. [online] Available at: <https://www.youtube.com/watch?v=maE0_gPxQ-w&feature=youtu.be>.

[11] 2016. *Datagridview Column Total Show In Textbox*. [online] Available at: <https://www.youtube.com/watch?v=2BRJfKkuW7Q>.

[12] 2016. *Add Subtract Multiply Division Two Textbox Values Show In Third Textbox Simple C#*. [online] Available at: <https://www.youtube.com/watch?v=6z6MI3eYcXo&feature=youtu.be>.

[13] Farhadi, N., 2010. *How To Delete A Selected Datagridviewrow And Update A Connected Database Table?*. [online] Stack Overflow. Available at: <https://stackoverflow.com/questions/2084346/how-to-delete-a-selected-datagridviewrow-and-update-a-connected-database-table>.

[14] Codeproject.com. 2012. *[Solved] Auto Increment Number In Textbox Using C# - Codeproject*. [online] Available at: <https://www.codeproject.com/Questions/505381/AutoplusIncrementplusnumberplusinplustextboxplusus>.

[15] 2016. *Passing Textbox Text To Another Form In Visual C# (Csharp)*. [online] Available at: <https://www.youtube.com/watch?v=-4sedzHtsqs&feature=youtu.be>.

[16] 2020. *SDLC Waterfall Model*. [online] Available at: <https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm>.

[17] Amit. (2020). *All You Need to Know About UML Diagrams*. [online] Available at: <https://tallyfy.com/uml-diagram/> [Accessed on May 30, 2020]

[18] Bouwkamp, K. (2016). *The 9 Most In-Demand Programming Languages of 2016.* [online] Available at: <https://www.codingdojo.com/blog/9-most-in-demand-programming-languages-of-2016> [Accessed May 28, 2020]

[19] Ferguson, N. (2020). *What's The Difference Between Frontend And Backend Web Development?* [online] Available at: <https://careerfoundry.com/en/blog/web-development/whats-the-difference-between-frontend-and-backend/> [Accessed May 28, 2020]

[20] Oracle. (n.d.), *What is a Database?* [online] Available at: <https://www.oracle.com/ph/database/what-is-database.html> [Accessed May 30, 2020]

[21] Patel, M. (2015). *Online Food Order System for Restaurants.* [online] Available at: < https://scholarworks.gvsu.edu/cgi/viewcontent.cgi?article=1222&context=cistechlib> [Accessed May 29, 2020]

[22] Plural Sight, LLC. (2019). *Difference Between Front-End & Back-End Developer*. [online] Available at: < https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end?fbclid=IwAR0or7x-VrAHPE0h0fYnrwLUCAPJZNDQM2u0a-X32rUVxbPQnsWRvrTgsYc> [Accessed May 28, 2020]

[23] Rouse, M. (2016). *What is SQL (Structured Query Language)?* [online] Available at: <https://searchsqlserver.techtarget.com/definition/SQL>

[24] Shu, L., (2017). *Contact Information Manager Prototype*. [online] Available at: https://www.theseus.fi/bitstream/handle/10024/139084/IT_COMPLETED_thesis.pdf?sequence=1&isAllowed=y&fbclid=IwAR2nYQnuP4WxE70ZZlwG6zQUL-t9lP1kQ7b-vJ2S5eekM5_SMvWmc_I2Lx4 [Accessed May 28, 2020]

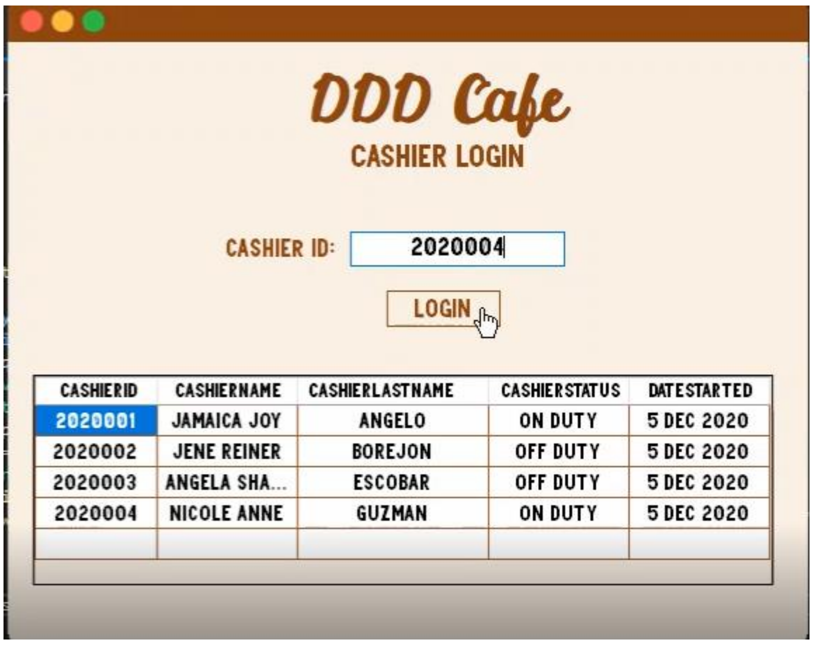
[25] Technopedia Inc. (n.d.). *What is Database Front End?* [online] Available at: <https://www.techopedia.com/definition/6761/database-front-end> [Accessed May 28, 2020]

[26] Visual Paradigm. (n.d.). *What is Entity Relationship Diagram (ERD)?* [online] Available at: <https://www.visual-paradigm.com/guide/data-modeling/what-is-entity-relationship-diagram/> [Accessed May 30, 2020]

[27] Winning Solutions, Inc. (n.d.). *Microsoft Access Front End: Back End: Explained*. [online] Available at: < https://www.microsoftaccessexpert.com/Microsoft-Access-Front-End-Back-End.aspx?fbclid=IwAR02n--oGC__CFwa-rMnguGzkrAZ1TzScTiEQ4D58KSUE9swVdxlj3kBln4> [Accessed May 28, 2020]

Appendix B
User Manual

Step 1: Type on the textbox the cashier ID that you will see on the data grid table below. The cashier ID that the system will only accept are those cashier's status who is "On Duty".

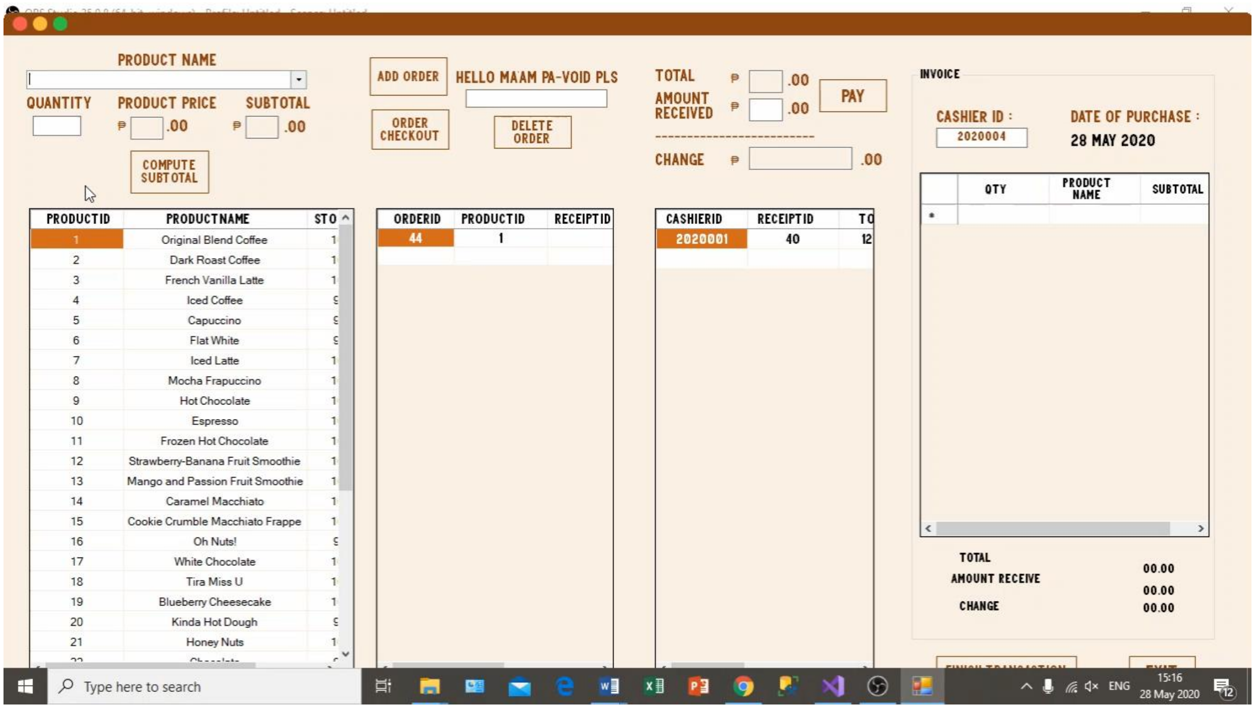


CASHIER ID: 2020004

CASHIERID	CASHIERNAME	CASHIERLASTNAME	CASHIERSTATUS	DATESTARTED
2020001	JAMAICA JOY	ANGELO	ON DUTY	5 DEC 2020
2020002	JENE REINER	BOREJON	OFF DUTY	5 DEC 2020
2020003	ANGELA SHA...	ESCOBAR	OFF DUTY	5 DEC 2020
2020004	NICOLE ANNE	GUZMAN	ON DUTY	5 DEC 2020

Figure 14: User-Manual Step 1_Log-in

Step 2: As the user enter the cashier ID, the Main Form would display.



PRODUCTID	PRODUCTNAME	STO
1	Original Blend Coffee	1
2	Dark Roast Coffee	1
3	French Vanilla Latte	1
4	Iced Coffee	9
5	Capuccino	9
6	Flat White	9
7	Iced Latte	1
8	Mocha Frapuccino	1
9	Hot Chocolate	1
10	Espresso	1
11	Frozen Hot Chocolate	1
12	Strawberry-Banana Fruit Smoothie	1
13	Mango and Passion Fruit Smoothie	1
14	Caramel Macchiato	1
15	Cookie Crumble Macchiato Frappe	1
16	Oh Nuts!	9
17	White Chocolate	1
18	Tira Miss U	1
19	Blueberry Cheesecake	1
20	Kinda Hot Dough	9
21	Honey Nuts	1

ORDERID	PRODUCTID	RECEIPTID
44	1	

CASHIERID	RECEIPTID	TO
2020001	40	12

CASHIER ID : 2020004

DATE OF PURCHASE : 28 MAY 2020

	QTY	PRODUCT NAME	SUBTOTAL
*			

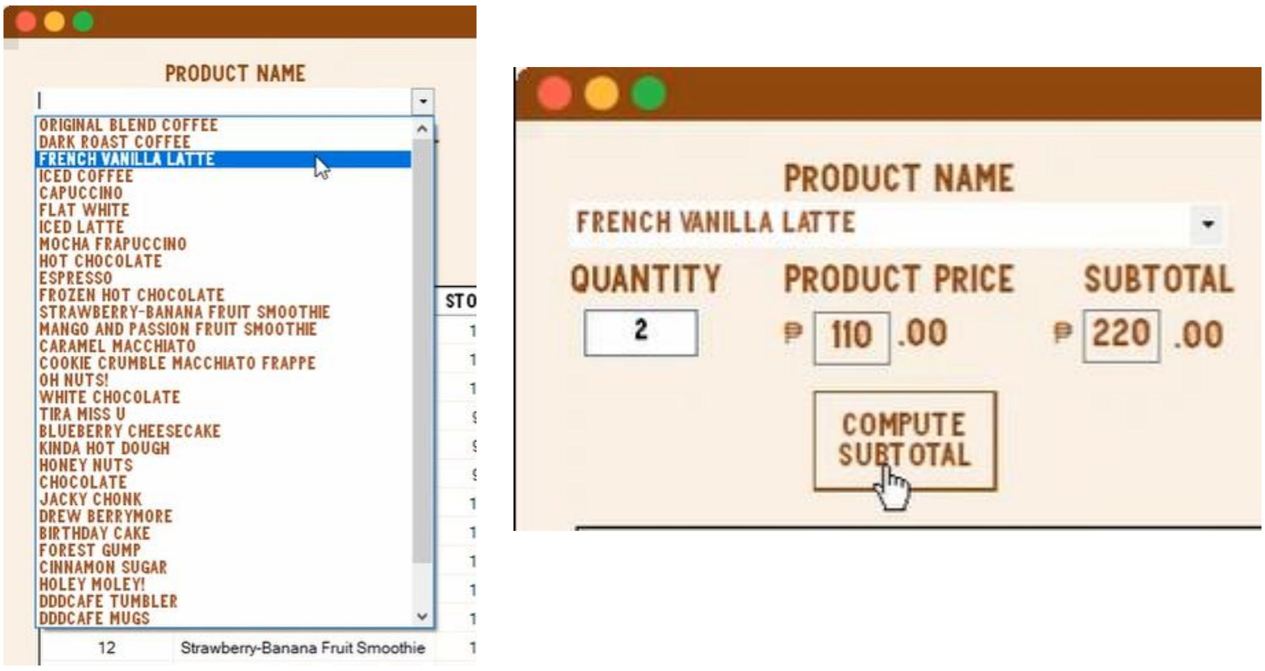
TOTAL 00.00

AMOUNT RECEIVE 00.00

CHANGE 00.00

Figure 15: User-Manual Step 2_Displayed Mainform

Step 3: The cashier will choose the product on the drop-down box according to the customers' choice. If the customer wants to order more of that specific product, the cashier will input the quantity in the textbox, after that, click the "compute subtotal" button.



PRODUCT NAME: FRENCH VANILLA LATTE

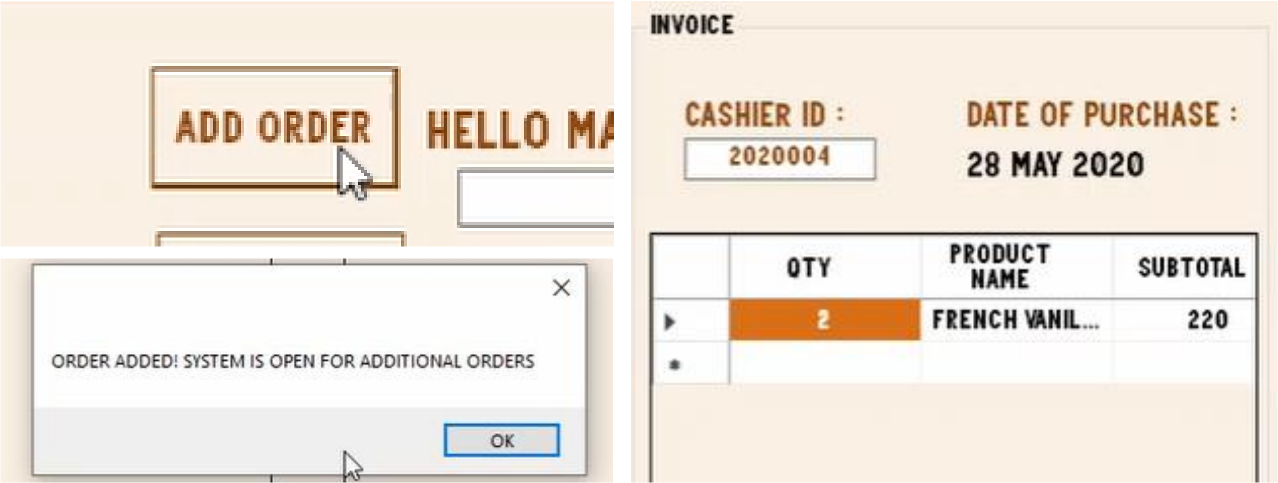
QUANTITY: 2

PRODUCT PRICE: ₱ 110 .00

SUBTOTAL: ₱ 220 .00

Figure 16: User-Manual Step 3_Choosing order, input quantity, and subtotal computation

Step 4: If the customer is satisfied on their order, the user can click the "add order" button and product that the customer chose will appear on the invoice display.



ORDER ADDED! SYSTEM IS OPEN FOR ADDITIONAL ORDERS

CASHIER ID : 2020004

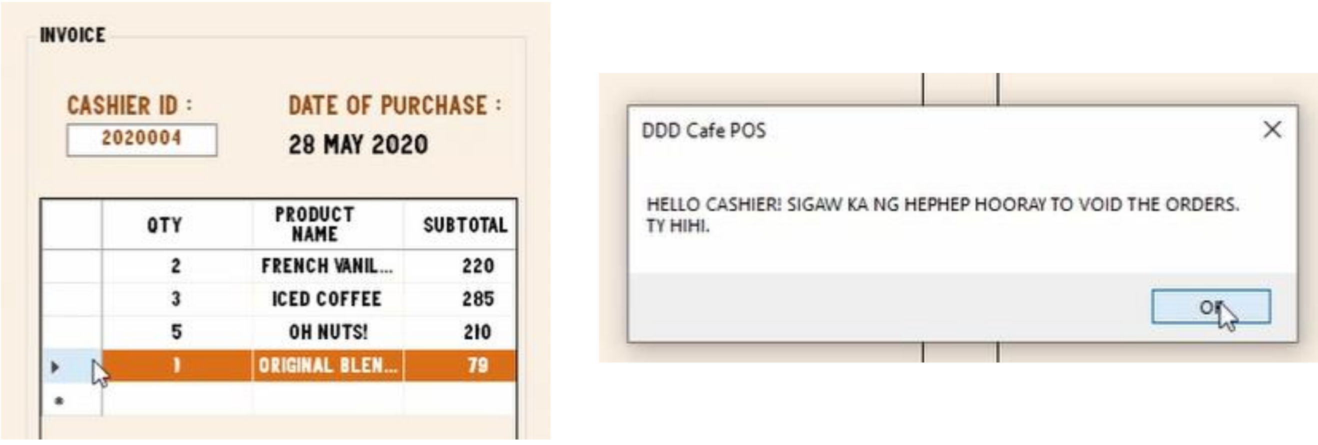
DATE OF PURCHASE : 28 MAY 2020

QTY	PRODUCT NAME	SUBTOTAL
2	FRENCH VANIL...	220

Figure 17: User-Manual Step 4_Add order button for additional orders

Step 5: If the customer wants to order another product, the user could repeat steps 3 to 4 since the system will allow additional orders after clicking it.

Step 6: If the customer wants to delete/void a certain product, click that product in the invoice display and click the delete button, however, a password is needed to do this otherwise a message box will be shown, therefore typing the password "dddcafedabest" then clicking the delete order button will delete the order.



CASHIER ID : 2020004

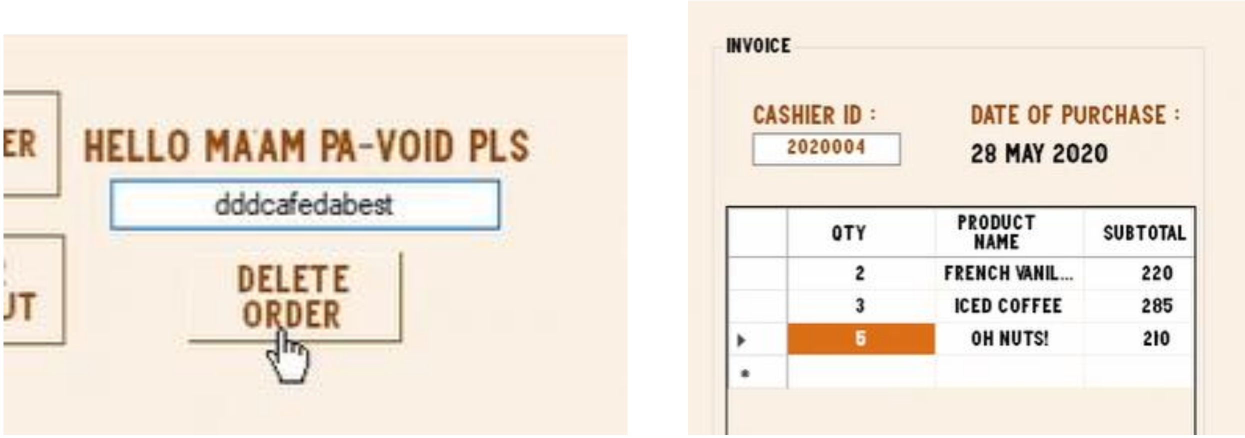
DATE OF PURCHASE : 28 MAY 2020

QTY	PRODUCT NAME	SUBTOTAL
2	FRENCH VANIL...	220
3	ICED COFFEE	285
5	OH NUTS!	210
1	ORIGINAL BLEN...	79

DDD Cafe POS

HELLO CASHIER! SIGAW KA NG HEPHEP HOORAY TO VOID THE ORDERS. TY HIHI.

Figure 18.1: User-Manual Step 6_Deleting the order without the correct password



HELLO MA'AM PA-VOID PLS

dddcafedabest

INVOICE

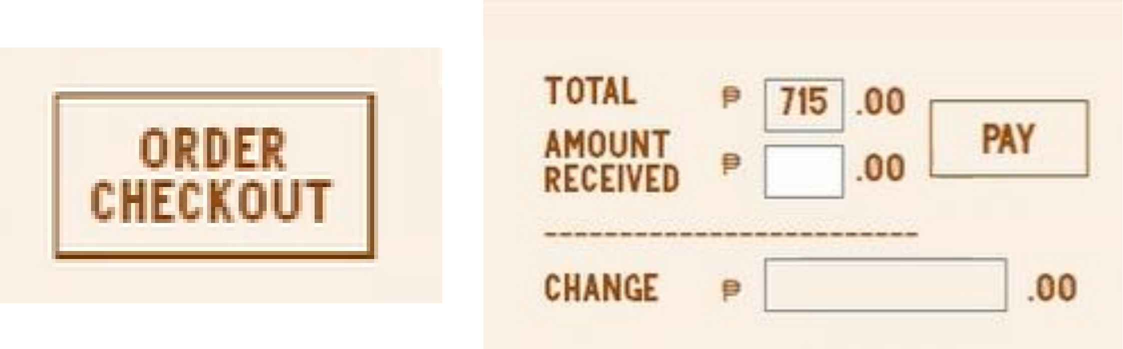
CASHIER ID : 2020004

DATE OF PURCHASE : 28 MAY 2020

	QTY	PRODUCT NAME	SUBTOTAL
	2	FRENCH VANIL...	220
	3	ICED COFFEE	285
▸	5	OH NUTS!	210
*			

Figure 18.2: User-Manual Step 6_Deleting the order with the correct password

Step 7: If the customer is completed the ordering process, the cashier can click the order checkout button for the total computation of the products bought.



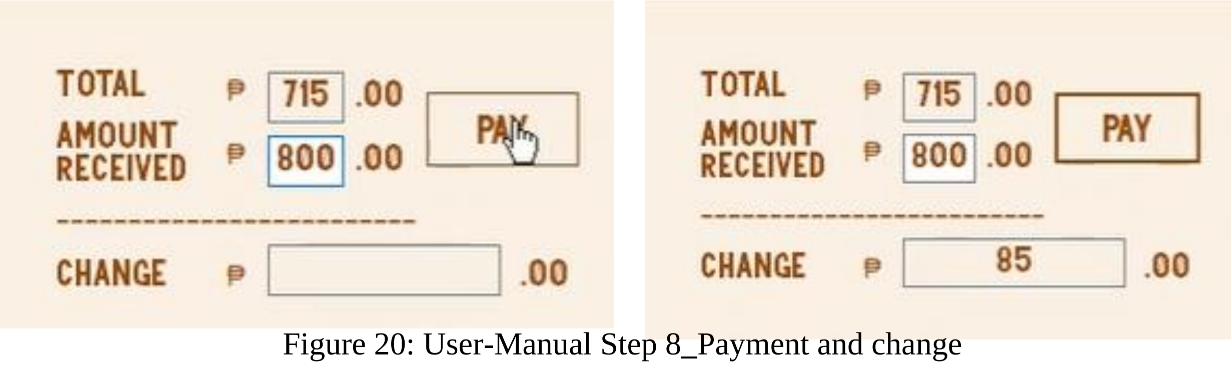
TOTAL ₱ 715 .00

AMOUNT RECEIVED ₱ .00

CHANGE ₱ .00

Figure 19: User-Manual Step 7_Order checkout button to display the total of the products bought

Step 8: After the computation for the total cost, the cashier will input the amount received from the customer the click the pay button to display the change.



TOTAL ₱ 715 .00

AMOUNT RECEIVED ₱ 800 .00

CHANGE ₱ .00

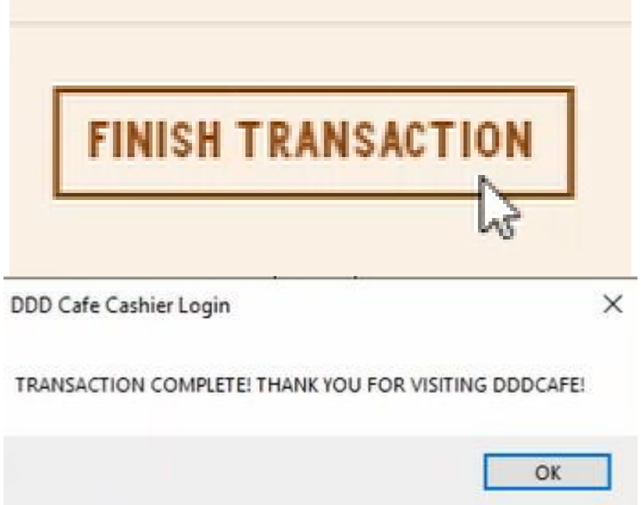
TOTAL ₱ 715 .00

AMOUNT RECEIVED ₱ 800 .00

CHANGE ₱ 85 .00

Figure 20: User-Manual Step 8_Payment and change

Step 9: Click the finish transaction button and a message will be shown.

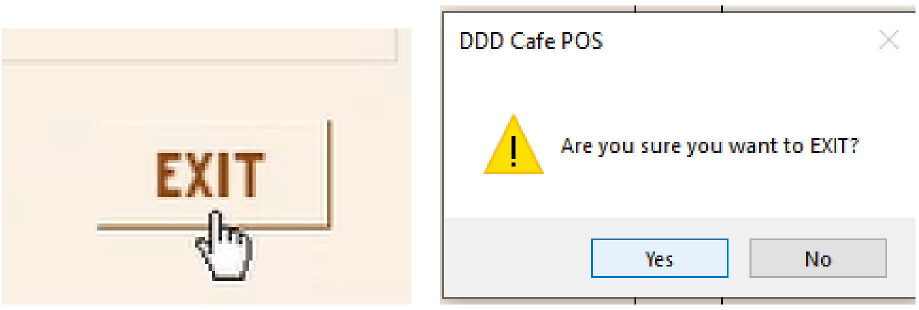


DDD Cafe Cashier Login

TRANSACTION COMPLETE! THANK YOU FOR VISITING DDDCAFE!

Figure 21: User-Manual Step 9_Finish transaction

Step 10: Lastly, click the exit button to exit the program.



Are you sure you want to EXIT?

Figure 22: User-Manual Step 10_Exit program

Appendix C
Source Code

CASHIERLOGIN.cs

```
using System;
using System.Data;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace New_UI_draft
{
    public partial class CASHIERLOGIN : Form
    {
        public static string passingText;

        SqlConnection con = new SqlConnection(@"Data Source=ASPIRE-ES1-132;Initial
Catalog=CAFE;Integrated Security=True");

        public CASHIERLOGIN()
        {
            InitializeComponent();
        }

        private void Bttnlogin_Click_1(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand cmd = con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "SELECT * FROM dbo.Cashier WHERE cashierId = '" +
txtcashierid.Text + "'";
            cmd.ExecuteNonQuery();
            DataTable dt = new DataTable();
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            da.Fill(dt);

            if (txtcashierid.Text != "")
            {
                if (txtcashierid.Text != "")
                {
                    if (dt.Rows.Count == 1)
                    {
                        if (dt.Rows[0][0].ToString() == "2020001" || dt.Rows[0][0].ToString() ==
"2020004")
                        {
                            passingText = txtcashierid.Text;
                            MAINFORM mf = new MAINFORM();
                            mf.Show();
                            this.Hide();
                        }
                        else if (dt.Rows[0][0].ToString() == "2020002" || dt.Rows[0][0].ToString() ==
"2020003")
                        {
                            MessageBox.Show("Sorry, the Cashier is currently OFF DUTY.", "DDD Cafe
Cashier Login");
                        }
                    }
                    else
                    {
                        MessageBox.Show("Sorry, the Cashier ID doesn't belong to the cafe. Please try
again.", "DDD Cafe Cashier Login");
```

```
                    }
                }
            }
            else
            {
                MessageBox.Show("Please input cashier ID");
            }
            con.Close();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            this.cashierTableAdapter.Fill(this.cAFEDataSet.Cashier);
        }

        //==================== TITLE BAR =====================
        private void close_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void maxi_Click(object sender, EventArgs e)
        {
            if (this.WindowState == FormWindowState.Normal)
            {
                this.WindowState = FormWindowState.Maximized;
            }
            else
            {
                this.WindowState = FormWindowState.Normal;
            }
        }

        private void Mini_Click(object sender, EventArgs e)
        {
            this.WindowState = FormWindowState.Minimized;
        }
        //======================================================
    }
}
```

MAINFORM.cs

```
using System;
using System.Data;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace New_UI_draft
{
    public partial class MAINFORM : Form
    {
        SqlConnection con = new SqlConnection(@"Data Source=ASPIRE-ES1-132;Initial
Catalog=CAFE;Integrated Security=True");
        SqlCommand cmd;
        SqlDataReader dr;

        public MAINFORM()
        {
            InitializeComponent();
        }
        private void MAINFORM_Load(object sender, EventArgs e)
        {
            this.receiptTableAdapter1.Fill(this.cAFEDataSet.Receipt);
            this.ordersTableAdapter1.Fill(this.cAFEDataSet.Orders);
            this.productTableAdapter1.Fill(this.cAFEDataSet.Product);

            lbldatepurchase.Text = DateTime.Today.ToShortDateString();
            txtcashier.Text = CASHIERLOGIN.passingText;

            comboboxproducts.Items.Clear();
            con.Open();
            cmd = con.CreateCommand();
            cmd.CommandType = CommandType.Text;
            cmd.CommandText = "SELECT productName FROM dbo.Product ORDER by
productId ASC";
            cmd.ExecuteNonQuery();
            DataTable dt = new DataTable();
            SqlDataAdapter da = new SqlDataAdapter(cmd);
            da.Fill(dt);

            foreach (DataRow dr in dt.Rows)
            {
                comboboxproducts.Items.Add(dr["productName"].ToString());
            }
            con.Close();
        }

        // EXIT BUTTON
        private void Bnexit_Click(object sender, EventArgs e)
        {
            DialogResult iExit;

            iExit = MessageBox.Show("Are you sure you want to EXIT?", "DDD Cafe POS",
MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
            if (iExit == DialogResult.Yes)
            {
                Application.Exit();
            }
        }

        private void Txtquantity_TextChanged(object sender, EventArgs e)
        {
            int quantity;
            con.Open();
            if (int.TryParse(txtquantity.Text, out quantity))
            {
                SqlCommand stockcontrol = new SqlCommand("UPDATE dbo.Product SET stock =
stock - @qty WHERE productName = @itn", con);
                stockcontrol.Parameters.AddWithValue("@qty", txtquantity.Text);
                stockcontrol.Parameters.AddWithValue("@itn", comboboxproducts.Text);
                stockcontrol.ExecuteNonQuery();
            }
            con.Close();
        }

        private void Comboboxproducts_SelectedIndexChanged(object sender, EventArgs e)
        {
            con.Open();
            SqlCommand pricecheck = new SqlCommand("SELECT price FROM dbo.Product
WHERE productName = '" + comboboxproducts.Text + "'", con);
```

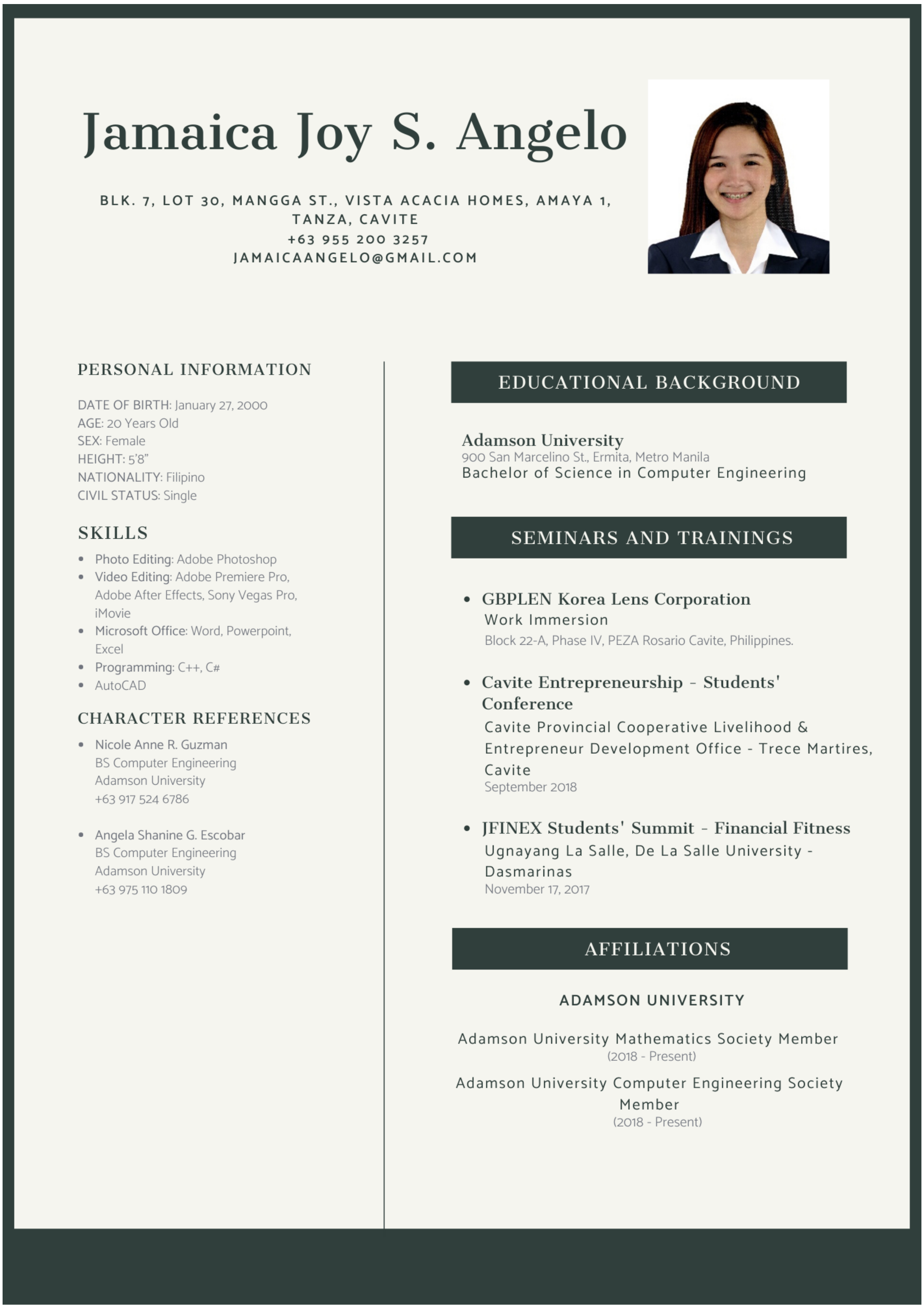
```
            dr = pricecheck.ExecuteReader();

            while (dr.Read())
            {
                txtproductprice.Text = dr.GetValue(0).ToString();
            }
            con.Close();
        }

        private void Bnaddorder_Click(object sender, EventArgs e)
        {
            int row = 0;
            dataGridView1.Rows.Add();
            row = dataGridView1.Rows.Count - 2;
            dataGridView1["qty", row].Value = txtquantity.Text;
            dataGridView1["productname", row].Value =
comboboxproducts.SelectedItem.ToString();
            dataGridView1["subtotal", row].Value = txtsubtotal.Text;

            comboboxproducts.Text = "";
            txtproductprice.Text = "";
            txtquantity.Text = "";

            MessageBox.Show("ORDER ADDED! SYSTEM IS OPEN FOR ADDITIONAL
ORDERS");
        }

        private void bncomputesubtotal_Click(object sender, EventArgs e)
        {
            con.Open();
            if (txtquantity.TextLength > 0 && txtproductprice.TextLength > 0)
            {
                txtsubtotal.Text = Convert.ToString(Convert.ToInt32(txtquantity.Text) *
Convert.ToInt32(txtproductprice.Text));
            }
            con.Close();
        }

        private void NumbersOnly(object sender, KeyPressEventArgs e)
        {
            char ch = e.KeyChar;
            if (!char.IsDigit(ch) && ch != 8)
            {
                e.Handled = true;
            }
        }

        private void Bnpay_Click(object sender, EventArgs e)
        {
            double tl, ar, cg;
            double.TryParse(txttotal.Text, out tl);
            double.TryParse(txtamountreceive.Text, out ar);
            cg = ar - tl;

            if (cg > 0)
            {
                txtchange.Text = cg.ToString();
            }
            else if (cg == 0)
            {
                txtchange.Text = "Exact Amount Received.";
            }
            else if (cg < 0)
            {
                txtchange.Text = "Insufficient Amount.";
            }
            lblreceiptar.Text = txtamountreceive.Text;
            lblamountreceivechange.Text = txtchange.Text;
        }

        private void Bndeleteorder_Click(object sender, EventArgs e)
        {
            string i;
            i = Convert.ToString(dataGridView1.SelectedRows[0]);
            if (txtvoid.Text == "dddcafedabest")
            {
                foreach (DataGridViewRow item in this.dataGridView1.SelectedRows)
                {
                    dataGridView1.Rows.RemoveAt(item.Index);
                }
            }
            else
            {
                MessageBox.Show("HELLO CASHIER! SIGAW KA NG HEPHEP HOORAY TO
VOID THE ORDERS. TY HIHI.", "DDD Cafe POS");
            }
        }

        private void Bnordercheckout_Click(object sender, EventArgs e)
        {
            int sum = 0;
            for (int i = 0; i < dataGridView1.Rows.Count; ++i)
            {
                sum += Convert.ToInt32(dataGridView1.Rows[i].Cells[2].Value);
            }
            txttotal.Text = sum.ToString();

            lblreceipttotal.Text = txttotal.Text;
        }

        private void Close_Click(object sender, EventArgs e)
        {
            Close();
        }

        private void Maxi_Click(object sender, EventArgs e)
        {
            if (this.WindowState == FormWindowState.Normal)
            {
                this.WindowState = FormWindowState.Maximized;
            }
            else
            {
                this.WindowState = FormWindowState.Normal;
            }
        }

        private void Mini_Click(object sender, EventArgs e)
        {
            this.WindowState = FormWindowState.Minimized;
        }
        public void finishtransaction(string quantity, int subtotal, int total, string date, string
amountreceive, int change)
        {
            con.Open();
            int productId = 0;

            String getdata, insertorder, insertreceipt;
            getdata = ("SELECT TOP 1 * FROM dbo.Product ORDER BY productId ASC");

            cmd = new SqlCommand(getdata, con);
            SqlDataReader dz = cmd.ExecuteReader();
            while (dz.Read())
            {
                productId = Convert.ToInt32(dz.GetValue(0));
            }
            dz.Close();
            cmd.Dispose();

            SqlDataAdapter or = new SqlDataAdapter();
            insertorder = "INSERT INTO dbo.Orders (productId, quantity, subtotal) VALUES ('" +
Convert.ToString(productId) + "', '" + quantity + "', '" + Convert.ToString(subtotal) + "')";
            cmd = new SqlCommand(insertorder, con);
            or.InsertCommand = new SqlCommand(insertorder, con);
            or.InsertCommand.ExecuteNonQuery();

            SqlDataAdapter re = new SqlDataAdapter();
            insertreceipt = "INSERT INTO dbo.Receipt (cashierId, total, dateOfPurchase,
amountReceive, change) VALUES ('" + Convert.ToString(txtcashier.Text) + "', '" +
Convert.ToString(total) + "', '" + date + "', '" + Convert.ToString(amountreceive) + "', '" +
Convert.ToString(change) + "')";
            cmd = new SqlCommand(insertreceipt, con);
            re.InsertCommand = new SqlCommand(insertreceipt, con);
            re.InsertCommand.ExecuteNonQuery();
            cmd.Dispose();
            con.Close();
        }

        private void Bnfintransaction_Click(object sender, EventArgs e)
        {
            string quantity;
            int subtotal; int total; string date; string amountreceive; int change;
            quantity = txtquantity.Text;
            subtotal = Convert.ToInt32(txtsubtotal.Text);
            total = Convert.ToInt32(txttotal.Text);
            date = lbldatepurchase.Text;
            amountreceive = txtamountreceive.Text;
            change = Convert.ToInt32(txtchange.Text);
            finishtransaction(quantity, subtotal, total, date, amountreceive, change);
            MessageBox.Show("TRANSACTION COMPLETE! THANK YOU FOR VISITING
DDDCAFE!", "DDD Cafe Cashier Login");
        }
    }
}
```

CAFÉ.sql

```
CREATE DATABASE [CAFE]
GO

USE [CAFE]
GO
```

```
CREATE TABLE dbo.Product
        ([productId] [bigint] IDENTITY (001,1),
        [productName] [nvarchar] (100),
        [stock] [int],
        [price] [int]
        PRIMARY KEY (productId))
ON [PRIMARY]
GO

INSERT INTO dbo.Product
        ([productName],
         [stock],
         [price])


VALUES
        /*CAFE DRINKS*/
        ('Original Blend Coffee','100','79'),
        ('Dark Roast Coffee','100','79'),
        ('French Vanilla Latte','100','110'),
        ('Iced Coffee','100','95'),
        ('Capuccino','100','105'),
        ('Flat White','100','137'),
        ('Iced Latte','100','149'),
        ('Mocha Frapuccino','100','149'),
        ('Hot Chocolate','100','129'),
        ('Espresso','100','149'),
        ('Frozen Hot Chocolate','100','149'),
        ('Strawberry-Banana Fruit Smoothie','100','134'),
        ('Mango and Passion Fruit Smoothie','100','134'),
        ('Caramel Macchiato','100','159'),
        ('Cookie Crumble Macchiato Frappe','100','159'),

        /*DOUGHNUTS*/
        ('Oh Nuts!','100','42'),
        ('White Chocolate','100','45'),
        ('Tira Miss U','100','43'),
        ('Blueberry Cheesecake','100','43'),
        ('Kinda Hot Dough','100','38'),
        ('Honey Nuts','100','42'),
        ('Chocolate','100','38'),
        ('Jacky Chonk','100','42'),
        ('Drew Berrymore','100','10'),
        ('Birthday Cake','100','10'),
        ('Forest Gump','100','10'),
        ('Cinnamon Sugar','100','10'),
        ('Holey Moley!','100','10'),

        /*MISC ITEMS*/
        ('DDDCafe Tumbler','500','1500'),
        ('DDDCafe Mugs','500','500'),
        ('DDDCafe Planner','300','700'),
        ('DDDCafe Tote Bags','500','1000'),
        ('DDDCafe Reusable Cups','500','600')

GO

CREATE TABLE dbo.Cashier
        ([cashierId] [bigint] IDENTITY (2020001,1) NOT NULL,
        [cashierName] [nvarchar] (50),
	[cashierLastName] [nvarchar] (50),
	[cashierStatus] [nvarchar] (50),
	[dateStarted] [date]
	PRIMARY KEY (cashierId))
	ON [PRIMARY]
GO

INSERT INTO dbo.Cashier
	([cashierName],
	[cashierLastName],
	[cashierStatus],
	[dateStarted])

VALUES
	('Jamaica Joy', 'ANGELO','On Duty', '12/05/2020'),
	('Jene Reiner', 'BOREJON','Off Duty', '12/05/2020'),
	('Angela Shanine', 'ESCOBAR','Off Duty', '12/05/2020'),
	('Nicole Anne', 'GUZMAN','On Duty', '12/05/2020')
GO

CREATE TABLE dbo.Receipt
	([receiptId] [bigint] IDENTITY (00001,1),
	[cashierId] [bigint] FOREIGN KEY REFERENCES dbo.Cashier(cashierId),
	[total] [decimal] (10,2),
	[dateOfPurchase] [date],
	[amountReceive] [decimal] (10,2),
	[change] [decimal] (10,2),
	PRIMARY KEY (receiptId))
	ON [PRIMARY]
GO

CREATE TABLE dbo.Orders
	([orderId] [bigint] IDENTITY (001,1),
	[productId] [bigint] FOREIGN KEY REFERENCES dbo.Product(productId),
	[receiptId] [bigint] FOREIGN KEY REFERENCES dbo.Receipt(receiptId),
	[quantity] [int],
	[subtotal] [decimal] (10,2)
	PRIMARY KEY (orderId))
	ON [PRIMARY]
GO
```

Appendix D
Researcher’s Profile



Jamaica Joy S. Angelo

BLK. 7, LOT 30, MANGGA ST., VISTA ACACIA HOMES, AMAYA 1,
TANZA, CAVITE
+63 955 200 3257
JAMAICAANGELO@GMAIL.COM

PERSONAL INFORMATION

DATE OF BIRTH: January 27, 2000
AGE: 20 Years Old
SEX: Female
HEIGHT: 5’8”
NATIONALITY: Filipino
CIVIL STATUS: Single

SKILLS

- Photo Editing: Adobe Photoshop
- Video Editing: Adobe Premiere Pro, Adobe After Effects, Sony Vegas Pro, iMovie
- Microsoft Office: Word, Powerpoint, Excel
- Programming: C++, C#
- AutoCAD

CHARACTER REFERENCES

- Nicole Anne R. Guzman
  BS Computer Engineering
  Adamson University
  +63 917 524 6786

- Angela Shanine G. Escobar
  BS Computer Engineering
  Adamson University
  +63 975 110 1809

EDUCATIONAL BACKGROUND

Adamson University
900 San Marcelino St., Ermita, Metro Manila
Bachelor of Science in Computer Engineering

SEMINARS AND TRAININGS

- **GBPLEN Korea Lens Corporation**
  Work Immersion
  Block 22-A, Phase IV, PEZA Rosario Cavite, Philippines.

- **Cavite Entrepreneurship - Students' Conference**
  Cavite Provincial Cooperative Livelihood & Entrepreneur Development Office - Trece Martires, Cavite
  September 2018

- **JFINEX Students' Summit - Financial Fitness**
  Ugnayang La Salle, De La Salle University - Dasmarinas
  November 17, 2017

AFFILIATIONS

ADAMSON UNIVERSITY

Adamson University Mathematics Society Member
(2018 - Present)

Adamson University Computer Engineering Society Member
(2018 - Present)

Jene Reiner N. Borejon

Address: Blk 4 L 10 Arveemar Homes Julugan VI, Tanza, Cavite
Email Address: jenereinernosaborejon@gmail.com
Contact No.: 0906-670-9023

EDUCATIONAL TIMELINE

Year	**School**
2018-Present	Adamson University 900 San Marcelino St, Ermita, Manila, 1000 Metro Manila
2016-2018	Saint Augustine School – Senior High School Daang Amaya 3, Tanza, Cavite, Philippines 4108
2011-2016	Saint Augustine School 4108, Tanza - Trece Martires Rd, Tanza, Cavite

SKILLS

- Strategic Planning
- Programming Skills (C++, C#)
- Knowledgeable in MS Office, Photoshop and AutoCAD

ACHIEVEMENTS ATTAINED

- Augustinian staffer
- Church service awardee
- Outstanding student primary level
- Included in the Top 15 Honor’s in junior high school
- Bronze medalist in senior high school
- Included in the Top 10 students taking BS CPE in Adamson University

SEMINAR AND ORGANIZATION ATTENDED

- 2018 – Red Cross First Aid Seminar
- 2017 – Disaster Readiness (CPR)
- 2016 – Reader’s Society
- 2015-2016 – Campus Journalism

I hereby certify that the information contained above of my own knowledge is true and correct.

Jene Reiner N. Borejon

Angela Shanine G. Escobar

PERSONAL INFORMATION

AGE :19 years old
DATE OF BIRTH : March 02 ,2000
PLACE OF BIRTH : Metro Manila
Height : 5'4
Civil Status : Single
Religion : Roman Catholic

CONTACT DETAILS

Phone:
0975-110-1809

Address:
158 Diamond st. Shapell Homes,
Habay 1, Bacoor City, Cavite

CHARACTER REFERENCE

- Danica Roa
  +63 927 462 6348
- Nicole Anne Guzman
  +63 917 524 6786

EDUCATION BACKGROUND

THERESIAN SCHOOL OF CAVITE

SCIENCE,TECHNOLOGY,ENGINEERING and MATHEMATICS
S.Y 2016-2018

ADAMSON UNIVERSITY

Bachelor of Science in Computer Engineering

AFFILIATIONS

- Adamson Univesity Mathematics Society Member (2018-2020)
- Adamson Univesity Computer Engineering Member

SKILLS

- AutoCAD
- C# programming language

I hereby certify that the information above is true.

Angela Shanine G. Escobar
Applicant's Signature

NICOLE ANNE R. GUZMAN

6-0076, Jarger Homes, Lot 6 Block 7 Phase 1, San Guillermo St.
Bayanan, Muntinlupa City
+63 917 524 6786
guzmannicoleanne@gmail.com

OBJECTIVE

To obtain a Computer Engineering job that will maximize huge engineering training and experience to design, integrate, develop, and test both hardware and software solutions

EDUCATION BACKGROUND

TERTIARY

2018-PRESENT	**ADAMSON UNIVERSITY** 900 San Marcelino St., Ermita, Manila 1000 Philippines **BACHELOR OF SCIENCE IN COMPUTER ENGINEERING**

SECONDARY

2016 – 2018	**Asia Pacific College** 3 Humabon Place, Magallanes, Makati City
2012 – 2016	**San Roque Catholic School** Mendiola St. Alabang, Muntinlupa City

EXTRA-CURRICULAR ACTIVITIES

Member	**Adamson Computer Engineering Society** *2018 – present*
Member	**Adamson University Mathematics Society** *2019*

SEMINARS/ACTIVITIES ATTENDED

September, 2018	**CPENTELLIHENTE 3** Adamson Computer Engineering Society
October, 2017	**Microsoft Workshop** Hour of Code Asia Pacific College

SKILLS AND QUALIFICATIONS

- Knowledgeable in constructing programs using C++ and C#
- Knowledgeable in basic programming, Visual Basic
- Basic knowledge in 3D CAD Software, Solidworks and AutoCAD
- Basic knowledge in JAVA Programming, Greenfoot
- Basic knowledge in Circuit Development and Design using Multisim
- Knowledgeable in Microsoft word, Powerpoint and Excel
- Proficient in English and Filipino communication skills
- Willingness to learn and work hard to the assigned tasks
- Easily adaptive to any working environment

I hereby certify that the above information is true and correct based on my knowledge.

Nicole Anne R. Guzman