BINUS UNIVERSITY
BINUS INTERNATIONAL

Assignment Cover Letter

(Individual Work)

Student Information:		**Surname**	**Given Names**	**Student ID Number**
	1.	**Muchsin**	**Hisyam**	**2201797430**

Course Code	**: COMP6510**	**Course Name**	**: Programming Languages**
Class	**: L2AC**	**Name of Lecturer(s)**	: **Minaldi Loeis, Jude Martinez**
Major	**: CS**		
Title of Assignment (if any)	: **Café Cashier**		
Type of Assignment	**: Final Project**		
Submission Pattern			
Due Date	**: 02 - 07 - 2019**	**Submission Date**	**: 02 - 07 - 2019**

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:

(Name of Student)
Muchsin Hisyam

CONTENTS

1. Project Spesification

The development of technology such as software nowdays is getting faster and even more faster. Now, computers is a really important role in faciliting the completion of a job, increasing work efficiency. And also its increasing the creativity and activities of employees so they have good skills or abilities. Since my family owns a restaurant business, I decided to make a simple cashier program that is user-friendly to ease the current workload of the admin, and increase the work efficiency.

2. Solution Design

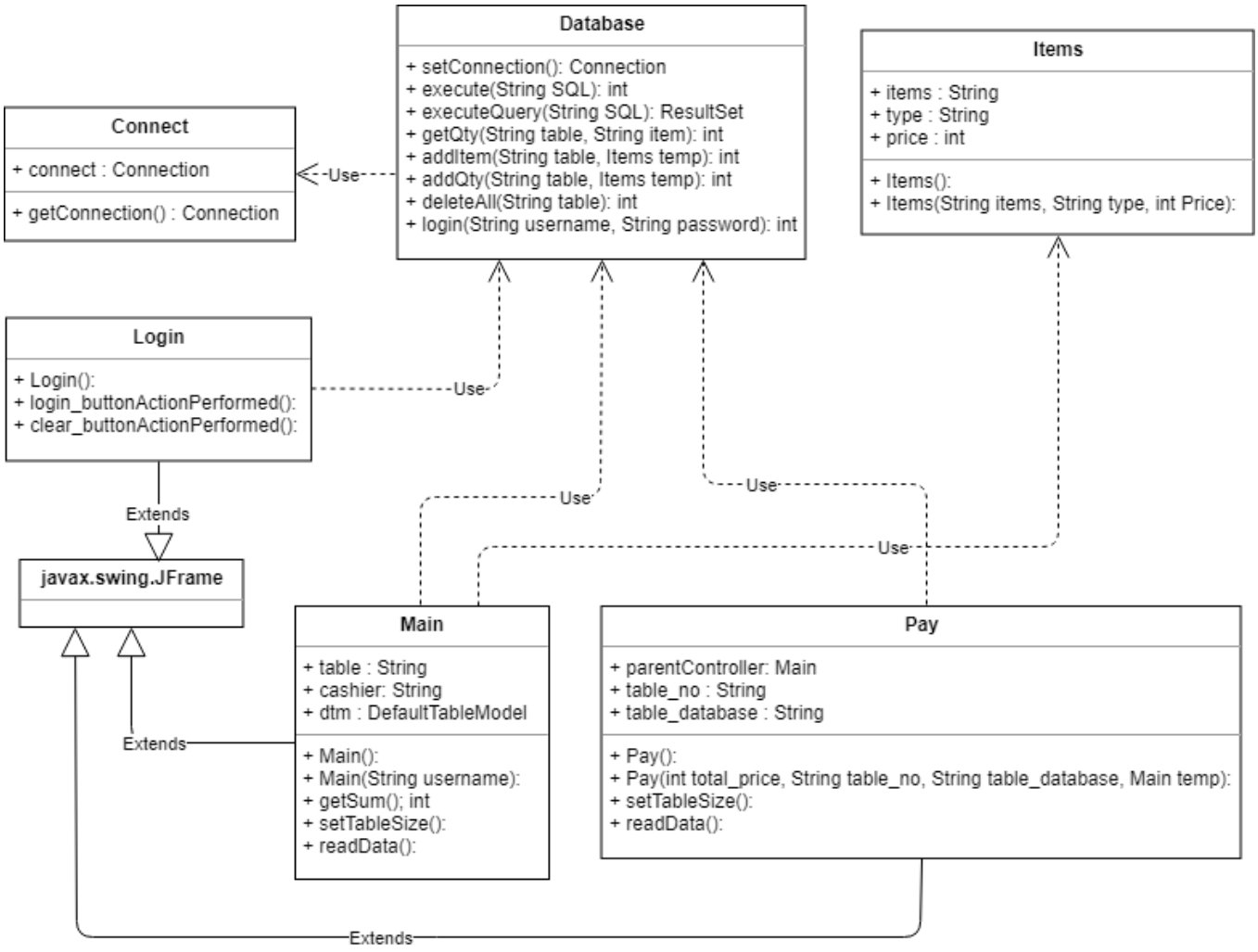

Image 2.1. Class diagram of the application

3. Code Implementation Discussion

3.1. Implementation

- **javax.swing,** is a set of classes that used to develop Graphic User Interface (GUI) based applications.
- **javax.swing.JFrame,** is one of the class of `javax.swing` that hold the interface of the program.
- **javax.swing.JOptionPane,** is one of the class of `javax.swing` that can print some message dialog or window, such as warning message, or success message.
- **javax.swing.JTable,** is one of the class of `javax.swing` for the interface of table.
- **javax.swing.table.DefaultTableModel,** is subclass of the `AbstractTableModel` that stores the data for the `JTable`.
- **java.sql,** is a package that contains all of the JDBC API.
- **java.sql.ResultSet,** is a package that hold the query statements and return the result of the executed query
- **java.sql.SQLException,** is an exception that provides information on a database errors.
- **com.mysql.jdbc.Driver,** is an API from `java.sql` and `javax.sql` packages for connecting from a database to a Java program.
- **java.util.logging.Level,** is a package that used for record all necessary information in the form of debuging or error messages.
- **java.util.logging.Logger,** is a package that control the logging logger output.
- **java.awt.print.PrinterException,** is an exception that provides information on printer or printing progress.
- **java.text.MessageFormat,** is a package that handle the message format for printing files.

3.2. How it works

So this application is a Graphic User Interface (GUI) based program using `javax.swing` library. The `javax.swing.JFrame` class is for hold the interface for the GUI, `javax.swing.JOptionPane` class for printing the error or success message, `javax.swing.JTable` class for stores all data from database table to this application table or `JTable` using `DefaultTableModel` class. I also use `java.sql` and `JDBC Driver` to connecting this application to `localhost` server, and stores this application data there. I used `java.util.logging` to record all necessary information in the form of debugging or error messages, so later when the I want to maintenance this application, I can know what kind of bug of that `java.util.logging` recorded. And last, I used `MessageFormat` for printing the bill or history of transanction.

4. Evidence of Work Program

4.1. Login page

When the program starts, the user (admin) will be directed to this page. Only registered users that can have access to use the program. This login page `username` and `password` is saved from database `tb_login` table, so if the user login with wrong `username` or `password` this page have a validation or error message through `JOptionPane`. Login button is for login progress, clear button is for clear the username and password textfield.

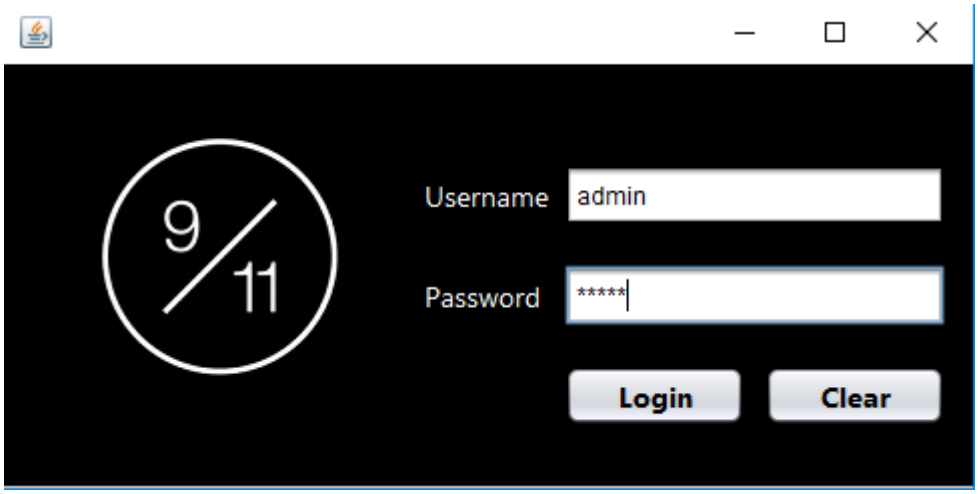

Image 4.1.1. Login page

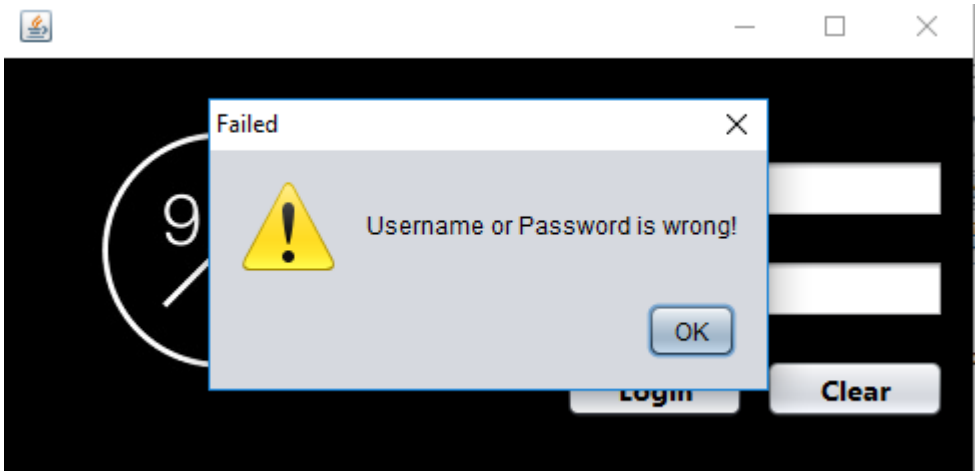

Image 4.1.2. Login page validation

4.2. Main page

This is the main page of the program. So, when the user (admin) is success to login, they will be directed to this page. Basically this page is hold all of the data manipulation such as, adding items (food or beverages) from certain restaurant's table to database table, deleting certains items from database table, etc. This page uses `JTabbedPane` for set the interface of table, menu of food, and menu of beverages. The admin must choose the certain table (customer's table) first before adding some items, otherwise it will print error message to `JOptionPane` such as "Please choose table first".

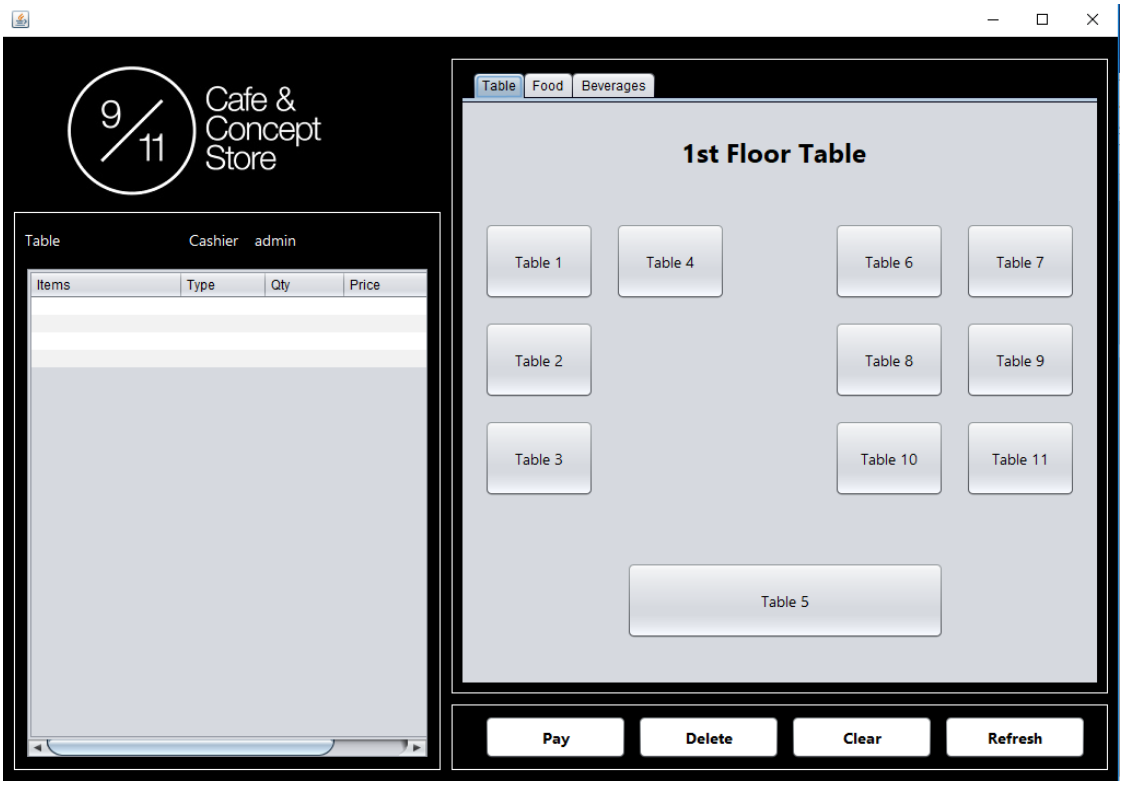

Image 4.2.1. Main page

4.3. Main page (Food panel)

This panel shows the food menu, so the user (admin) can clicked the certain menu button such as "Spaghetti Carbonara", etc. if the customer ordered that menu. This program have a features that allow the admin to add the quantity of the items automaticly when the admin clicked more than one time on the same button. For example, when the customer choose to order 2 pieces of fried rice, the admin can click the "Fried Rice" button twice, so the program will automaticly insert the fried rice 2 times (the quantity is increasing) and also the `total price` column will be updated on the database and also the `JTable`. The admin can also click the `delete` button to delete the certain items, but they must select the certain row that they want to deleted first, otherwise it will print error message to `JOptionPane` such as "Please

choose row first". And also the admin can click the `clear` button to delete all data from database's column. `Refresh` button is for refreshing the `JTable` by re-reading the database's table data.

Image 4.3.1. Main page (food panel)

4.4. Main page (Beverages panel)

Same like the food panel, this one also can manipulate the data of the beverages items, the difference is only this panel is for beverages menu.

Image 4.4.1. Main page (beverages panel)

4.5. Pay page

After already done the order, if the customer want to pay the bill or orders. The user (admin) can click the `Pay` button at the main page to finish the transaction. After clicked that button, the program will be directed to this `Pay` page. It will print the total amount of the bills, the table of the bills (ordered items), and the customer can choose wheter to pay with cash or with debit card. If the admin inputed lower value of the payment, the program will print error message to `JOptionPane` such as "Payment failed, money not enough", otherwise the payment is success with also printing the change of the payment. If the customer want to pay with debit, the admin can click the `Pay with debit card` button, and the payment will automaticly success without a change like the real payment with debit card.

Image 4.5.1. Pay page

4.6. Print result

After the payment success, the program will automaticly print the bills of the customer's orders. And the user (admin) can save the bills through ".oxps" file, and choose the place for save that file. After printing the bills, the all data of certain table that the customer occupied before in the database will automaticly deleted. So the table will not occupied again, which is if there some certain customer want to occupied that table, they can use that table.

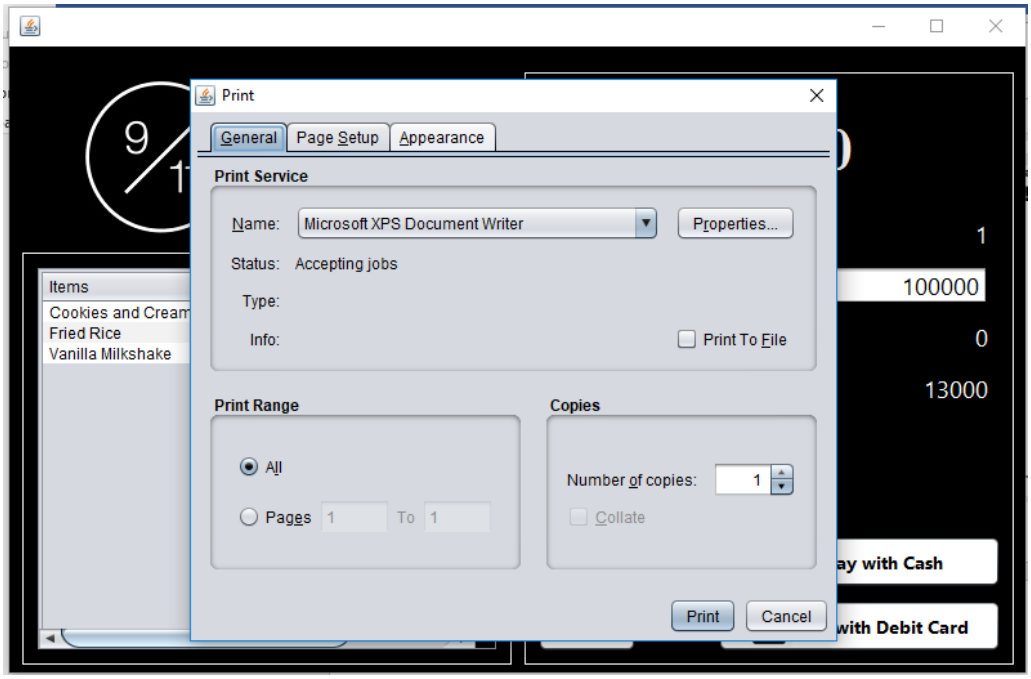

Image 4.6.1. Printing details option

Image 4.6.2. Printing in progress

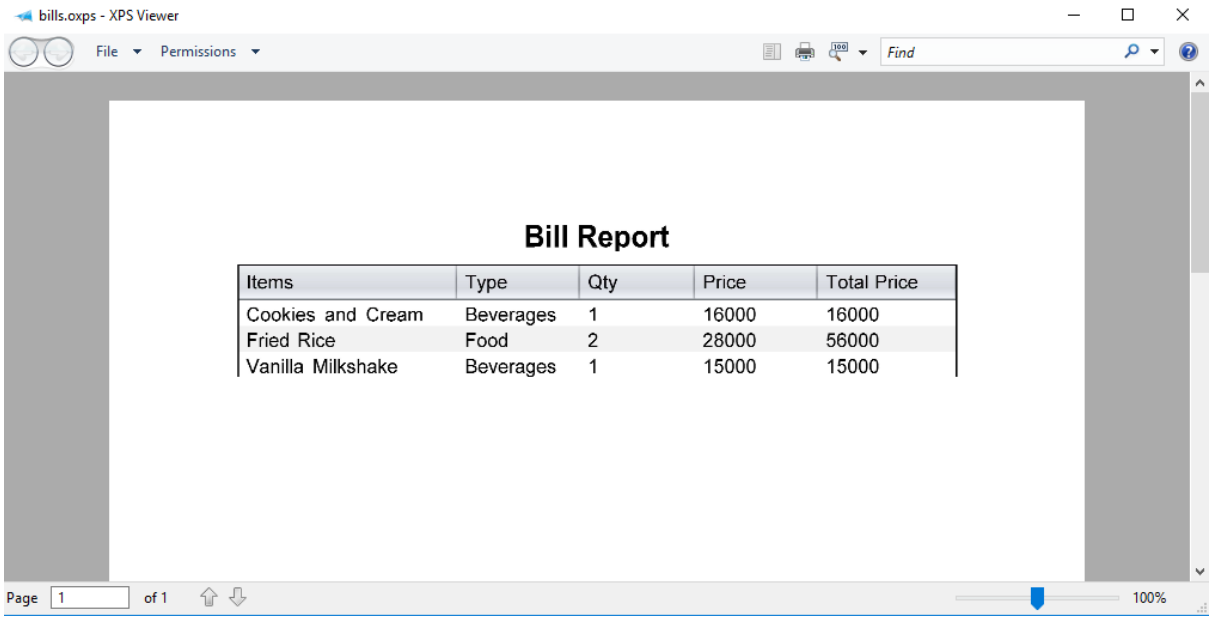

Image 4.6.3. Print result on .oxps file

5. References

- https://www.thoughtco.com/defaulttablemodel-overview-2033890
- https://azuharu.net/java/mengenal-jdbc/
- https://www.protechtraining.com/content/java_fundamentals_tutorial-jdbc
- https://docs.oracle.com/javase/7/docs/api/java/sql/SQLException.html

My final project github's link : https://github.com/muchsinhisyam/cashier-program