

Restaurant Delivery Software

Java Module 4 Project

Michael De Rubertis, Tajinderpal Singh, Mayur Patel

Programmer Analyst | January 21, 2019

Table of Contents

[Introduction 2](#_Toc299370)

[About this project 2](#_Toc299371)

[About this report 2](#_Toc299372)

[Analysis of the project 3](#_Toc299373)

[Class Diagram 3](#_Toc299374)

[Implementation of the project 4](#_Toc299375)

[Realization of the project 4](#_Toc299376)

[Screenshots 4](#_Toc299377)

[Project Discussion 6](#_Toc299378)

# Introduction

## About this project

This project is created for multiple restaurants to implement a basic food delivery network and keep track of deliverymen, restaurant managers, clients, system administrators, restaurants and menus.

This application will allow Administrators to add new Restaurant, edit them and delete them. Each restaurant can then have a menu, which can be edited and deleted. An administrator can also add a deliveryman, who can be edited and deleted.

The system will handle the clients and their account. Clients can edit and close their account. They can also order a meal from a given Restaurant and view their order history.

A Manager can view the orders accepted by any restaurant.

The Restaurateur is able to view the orders made for a given restaurant, has the option of accepting these orders, and must indicate when the orders are ready to be delivered.

A deliveryman can view all the Orders in his Postal Code area, and has the option of accepting the delivery of the orders and must indicate when the orders are delivered.

## About this report

This report will contain all information about how the program was created and implemented. This report will also cover our experiences throughout the project and how issues were encountered and solved in the process. In addition, a few sections of the application will be displayed and explained.

# Analysis of the project

The project required careful planning of how the “Use Cases” were to be implemented. We created our initial UML diagram, to be used as a guide. These contained the base classes, in essence the base of the application. We decided it would be best to use a base structure (Main Class) to control the data storage and handling of the flow of the application and build from there.

All the “view” classes were then created, along with the “controllers”. We then implemented the logic in the “controllers” and crosschecked it to the use cases provided in the PDFs.

## Class Diagram

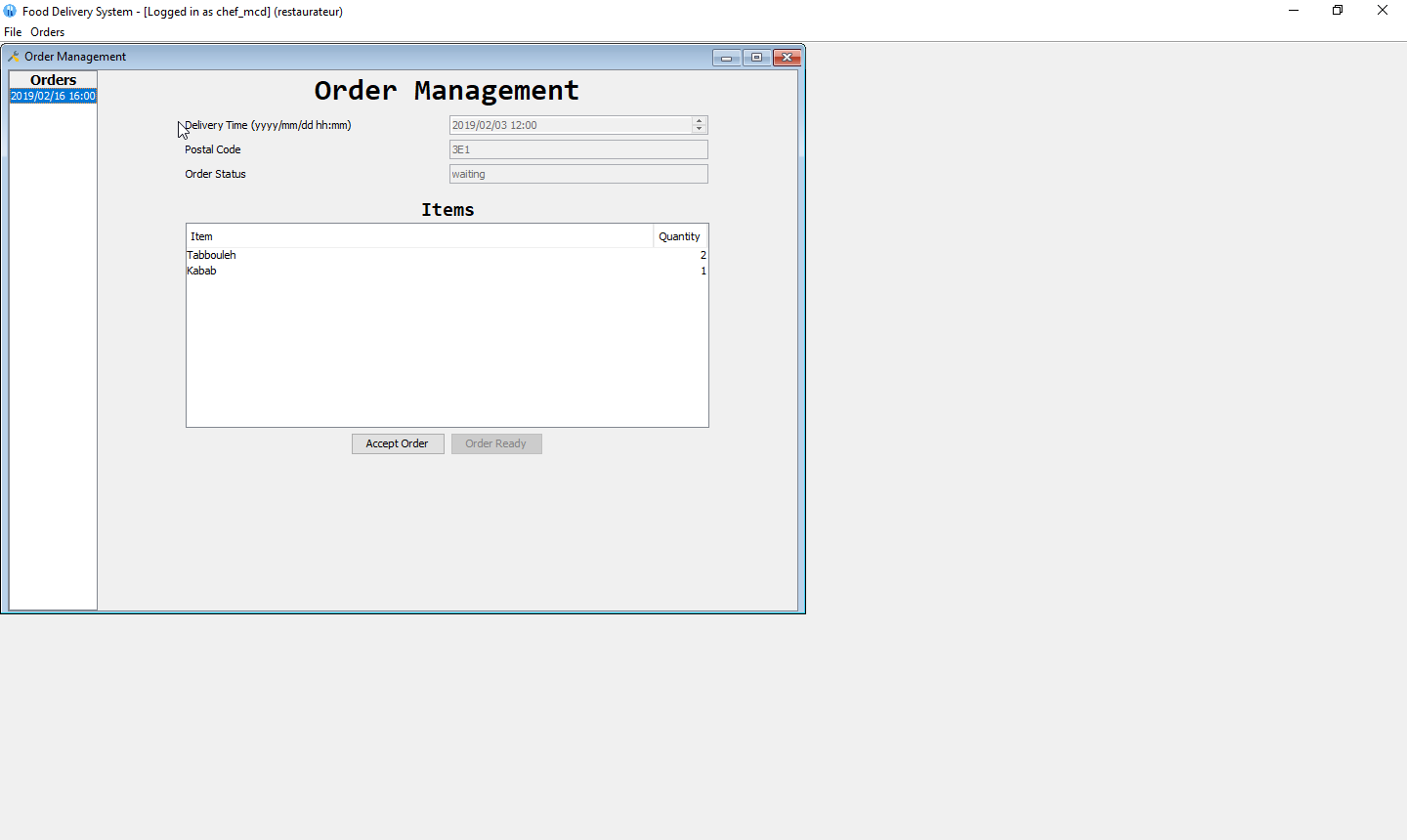
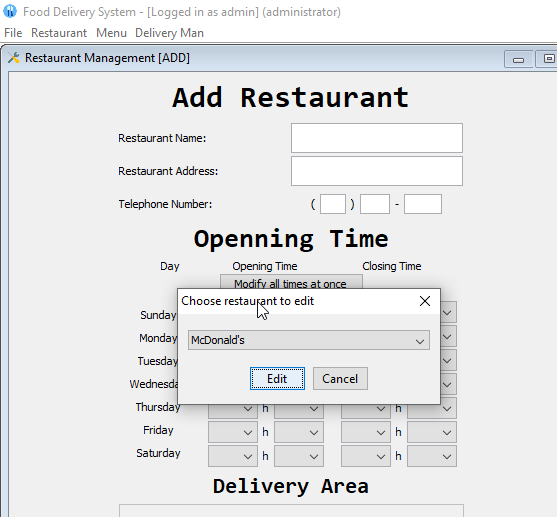
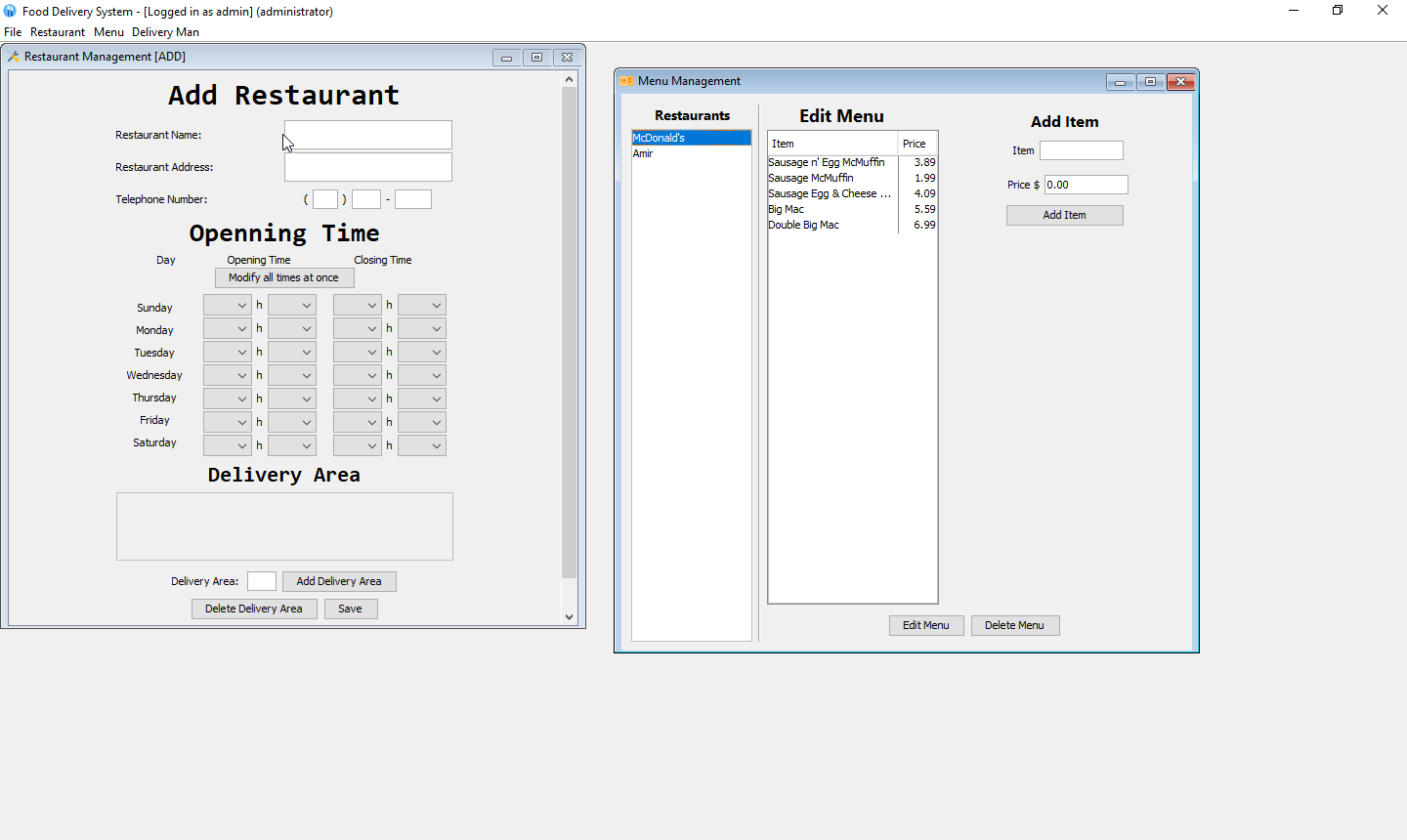
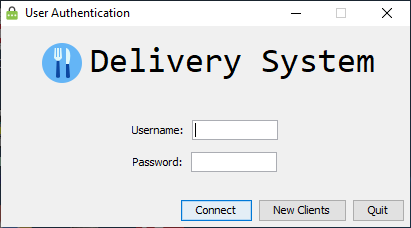
The class diagram has been appended to this document.

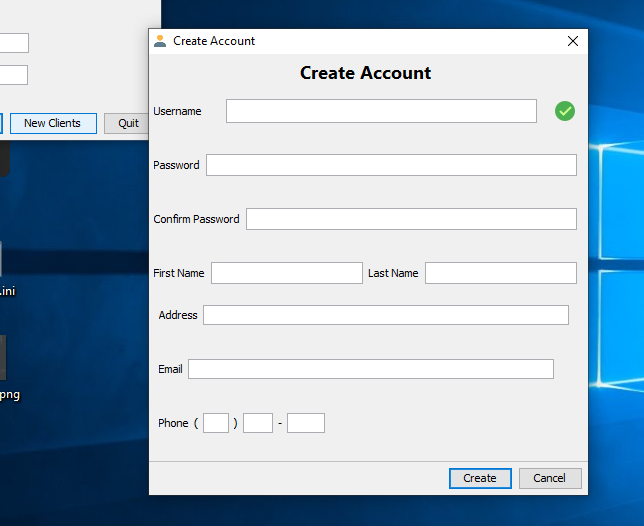
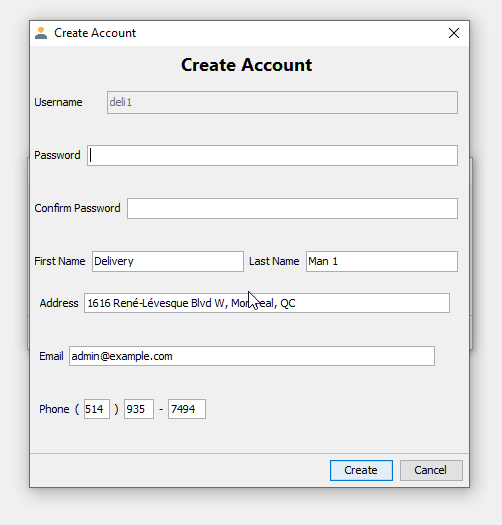
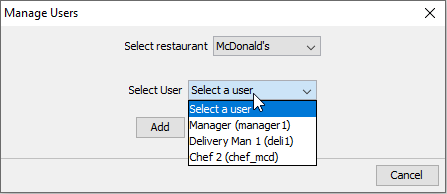
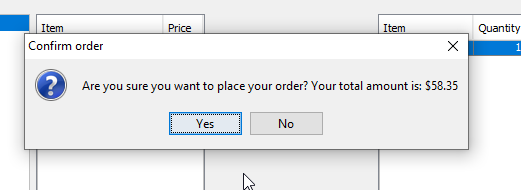
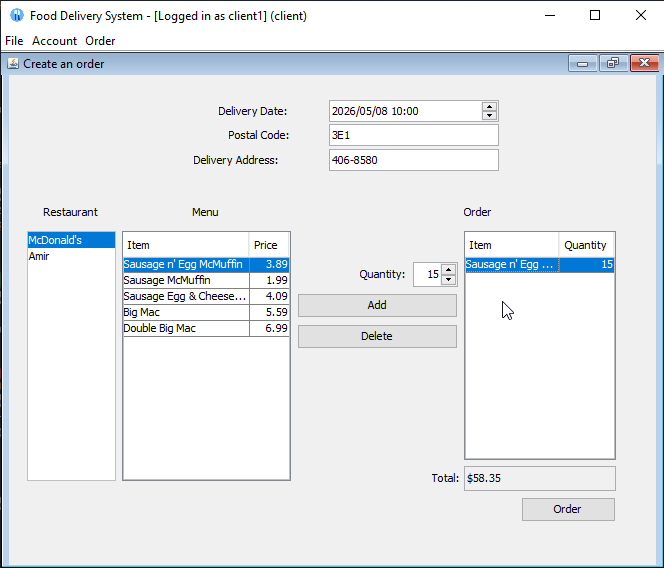
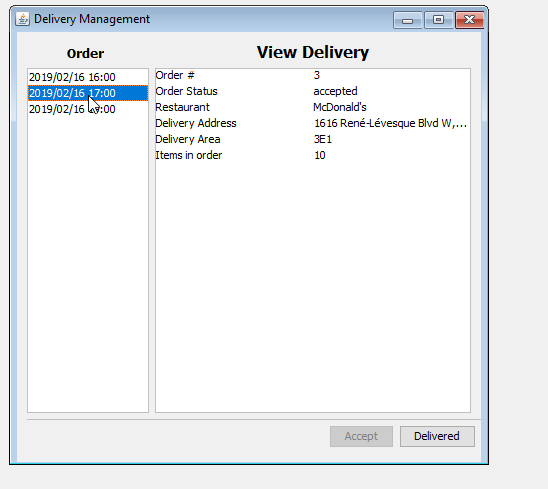
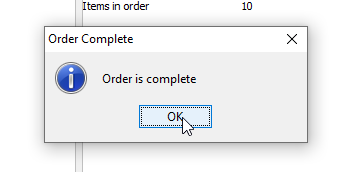
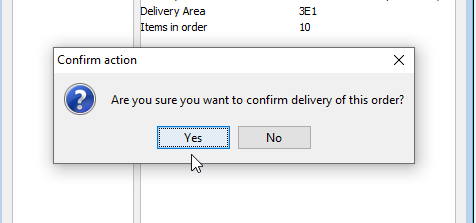
# Implementation of the project

## Realization of the project

This project was realized by using custom data classes for the Restaurants, Users, and Menu with Items. The project uses the MVC design pattern while also reusing a view and controller for multiple purposes (i.e. Add, Edit, Delete a restaurant with one Controller and View) in order to simplify maintenance in the future.

## Screenshots





# Project Discussion

As we did not learn database implementation, and implementing one would have complicated the project for the rest of the team, I (Michael) decided to create multiple classes for the different data type (User, Restaurant, Order, Item) which would be held in various ArrayLists. Those contained data manipulation methods in different classes (Users, Restaurants, Orders, Menu), which are then initiated as objects in the Main class when the application is executed, along with some example data for development.

We were able to complete almost the entire project, the part missing was the ability for clients to edit and delete their account.