Technical University of Cluj-Napoca

Order Product Application

Homework 2

Sucitu Stefania

30421

March 2016

1. Abstract

The application which is going to be presented in what follows serves a very practical and useful purpose. Its main aim is to facilitate the ordering of products, using a very customer friendly interface, in an easy and comfortable way. The backbone of this project is a SQL database which stores all the data this application needs and handles. The user has the possibility to enter new info about the desired field, modify already existing fields and also permanently remove them. The principal functionality of the application presented here is to operate on certain fields of a data base, without using any data base management software, therefore being aimed at users with not so much knowledge in the field of databases, SQL query language or any related subjects.

2. Introduction

The application developed here comes in handy when dealing with multiple users that need to be connected to multiple products. It can be of great use for a manager of an online store for instance. He could manage his customers as well as the products he has in his storage unit, as well as handle the incoming orders of products. This application facilitate the use of databases as not anyone finds the task of working with the SQL query language a simple, easy and basically effortless job. This way anyone can work with a database without knowing the least about these structures, as the application in discussion relies on a well-organized structure of commands. The ease of working with databases in the currently developed application is assured by the graphical user interface provided. Coming packed with explicit choices and functionalities linked with them, from the graphical user interface, all the work and functions can be performed in the easiest ways, of course, this being sustained by the rigorous methods the program relies on.

3. Aim of the Application

As mentioned previously, the application comes in the help of uninitiated in the art of SQL language users. The main objective of this mini project is to develop an application which offers a simple to understand user interface which holds different operations on database like data fields, relationships and complex tables. The user is asked to choose an option according to the task he wants the program to perform, one out of the three choices provided: client operations, product operations and placing an order related operations. Going further, the user is prompted with a window based on his previous choice. Each one of the window comes with multiple functionalities, depending on the aim of the task he wants to be performed, such as adding a new field, updating an already existing one, removal of an existing field and visualizing the contents of the table linked to the domain the user works on: client, product and order.

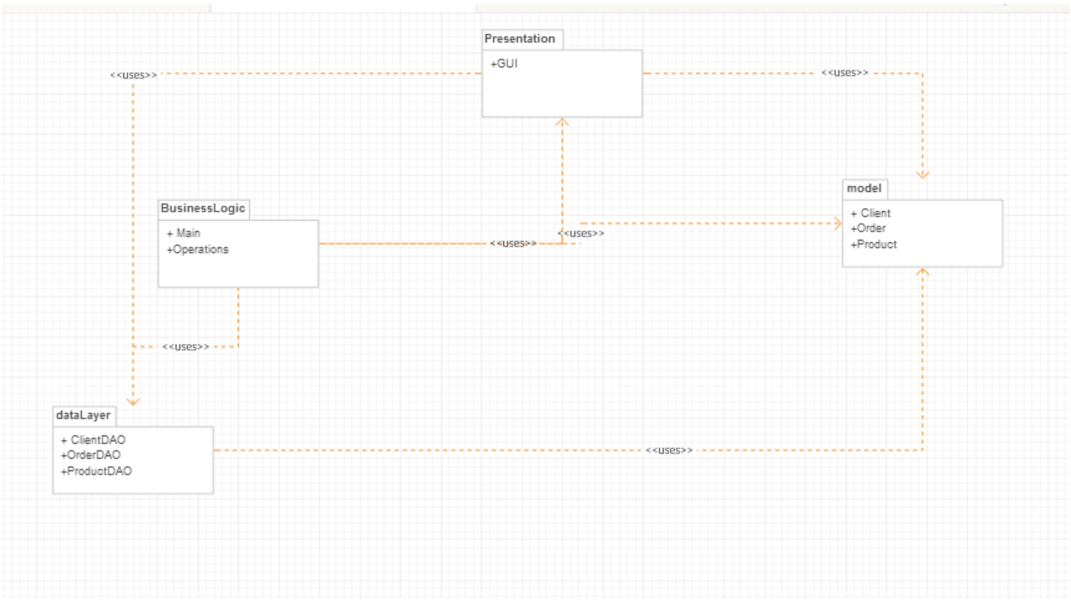
One of the main feature of this application is user interface windows, where all the user application interactions take place. What would be considered more important is the fact that the user should be able to use the interface given, without any help, in other words, the general look of the window should be self-explanatory.

4. Problem Analysis

4. 1 Modeling

The first step into solving the given problem is to analyze and decide on what should be the main purpose of the application and how can it be done, in a structured and well developed program. The packages and class design should be of great interest as one good initial design could save a lot of time in the later steps of the development process.

The main aspect to be considered into the development of the problem we are dealing with would be the representation of the database tables as entities in our project, therefore a package dedicated to this will be created with the required entities. The package model with the classes Client, Product and Order will be the mirror of the database used as storage unit for the application. Going further there will be required a connection to the database and the access to the data base should be performed quick, in a secure and controlled way and also in an efficient manner. A separate package will hold this functionality so the data Layer package is created, containing the ClientDAO, ProductDAO and OrderDAO classes. The mentioned classes will serve the purpose of connecting with the database and will execute any operation requiring access to the database fields. To be noted out of these operations are add new row, update row and remove row. A certain additional functionality to the application is required so another package must be added. BusinessLogic will hold the position of manager of these operations. As mentioned above, the user interface is of great importance in this project, therefore the additional commands for creating and maintaining the functionality of the user interface require the creation of the presentation package. The presentation package will represent the graphical user interface place of birth.



After deciding what packages should be included in our developing process we go on planning the functional aspects of the application. We will need a way to input data so JTextFiles will be chosen for this task. We will need to have the possibility of viewing data so we chose to display it in a pleasant, appealing way by putting the data in a beautifully structured JTable. Certain functionalities will be given the execution triggers from the JButtons. All of these elements will be organized using JFrames and JPanels so they could offer a clean, organized and neat representation.

In the following moments the main methods this application is based on will be presented. The start point will be the java program and data base link. This is done through a try catch structure. First a connection will be established, next a statement will be created and a result set follows. If any of these attempts do not succeed there is an exception thrown which will stop the execution. The same mechanism is used for all three of the entities of the application.

The functionalities which were mentioned above will be implemented similarly. The addition, update and removal of data entities will be done through statement execution updates using SQL syntax.

INSERT INTO *table_name* VALUES (*value1*,*value2*,*value3*,...);

UPDATE *table_name*SET *column1*=*value1*,*column2*=*value2*,...
WHERE *some_column*=*some_value*;

DELETE FROM *table_name* WHERE *some_column*=*some_value*;

SELECT * FROM *table_name*;

4. 2 Class Design

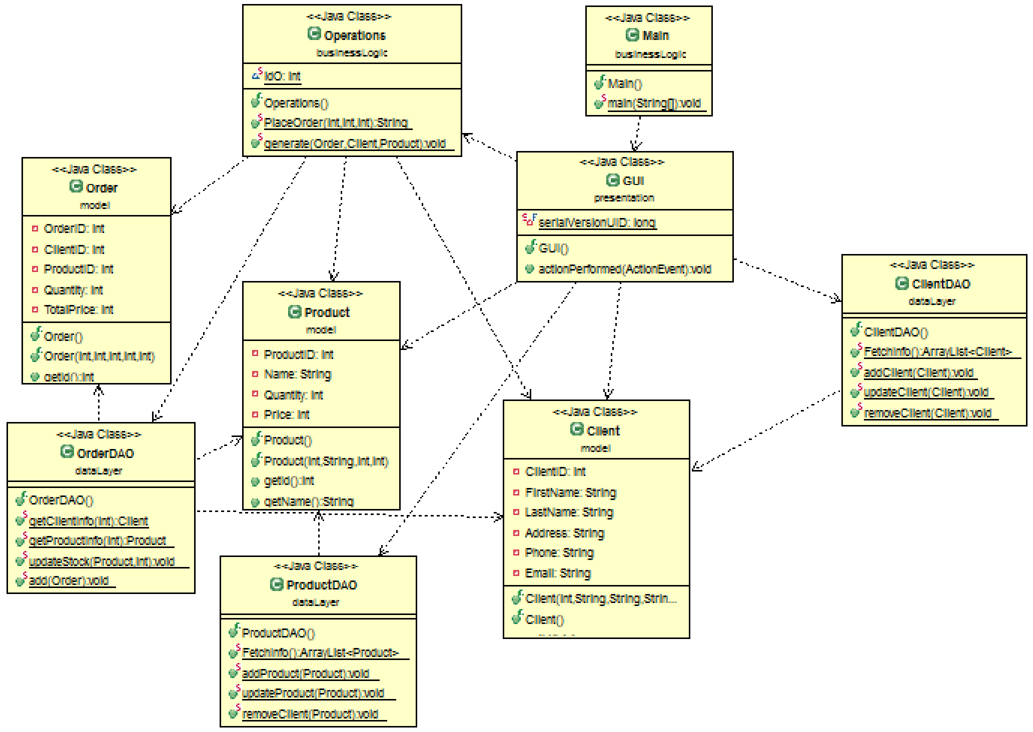
When designing the classes in one project, one should keep in mind as well as focus on what the class is aimed to do and it should that action and only that. As a consequence, the program in discussion is made up of multiple classes, each serving a different but specific purpose. Each class in the presented program serves a well known and simply and easily understandable way.

Starting from the model package we have the main entities classes. The Product class serves the structure of the product type object. It constructs the object as well as containing getter and setter methods. The Order class and Client class, both part of the same package are constructed similarly, instantiating objects and the offering to modify or fetch the attributes of the discussed objects. Each one of these classes are the common feature between the java application and the database as they represent the tables of the database, including their entities such as name, address, quantity and other representing features.

Going further, the data Layer package contains as well three classes, one for each of our main entities. Client DAO, Product DAO and Order DAO serve the functionalities of communicating with the data base, through specific SQL statements.

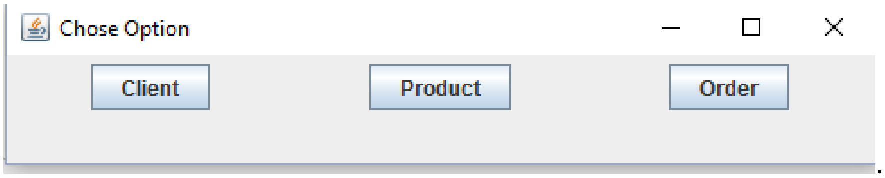
The application requires the possibility of placing orders therefore the Operations class is created. This class serves the order placement purpose and also contains methods which will provide the necessary data for the good functioning and proper execution of this option presented by the application.

The most complex class of this entire project is the GUI class, which by its name strongly suggests what it really and in fact does. It creates an initial window which offers the possibility of choosing one of the three options: Client, Product and Order, which will later change the course if execution. Each one of this options prompts a new window with the functionalities: add, edit info or remove and view options for product and client, choosing to input the desired fields to be changed. If the order option is to be selected, the user will be taken to a window where he can chose and already existing product id and client id and select a quantity and if the required quantity is in stock then the order can be placed successfully. Each of the options presented above are executed when the assigned button is pressed, an action listener triggering the required executions.

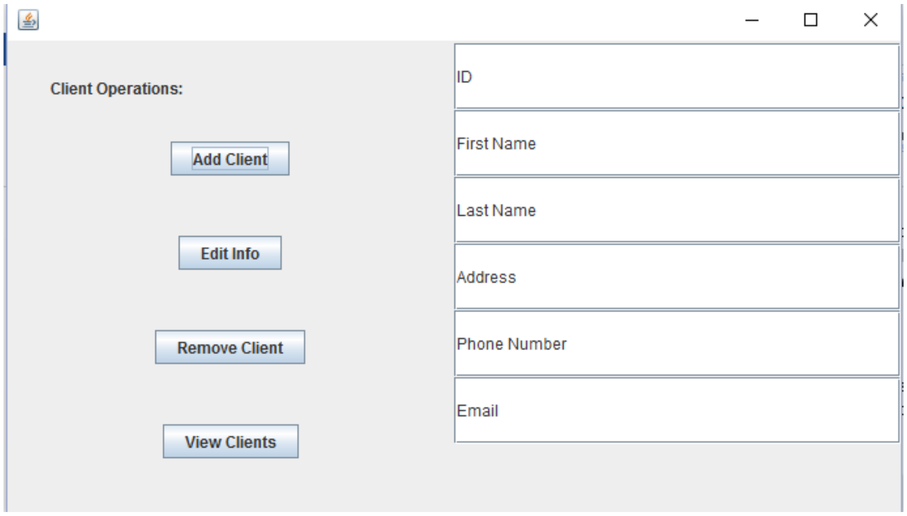


4. 3 User Interface

As mentioned above, the user interface is one of the most important and main features of this application. Firstly and foremost, the user is prompted with the following window:

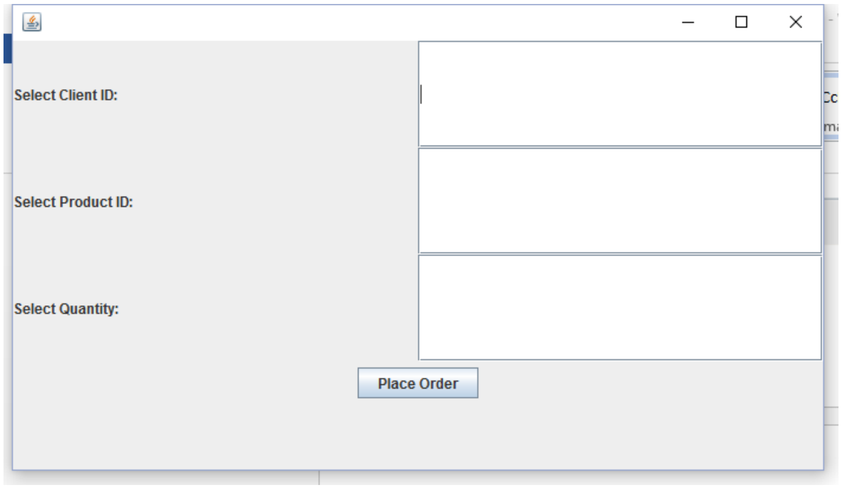


Next the user choses an option, for example Client and the next window pops up:



The choice of operations in visible. The user can either add or remove a client by entering data in the right side placed text boxes and can as well edit info the same way. By pressing the view button, a table will be displayed, showing what the linked table in the data base holds. A similar window will be displayed for the product option as well, the only difference will be the smaller number of features presented.

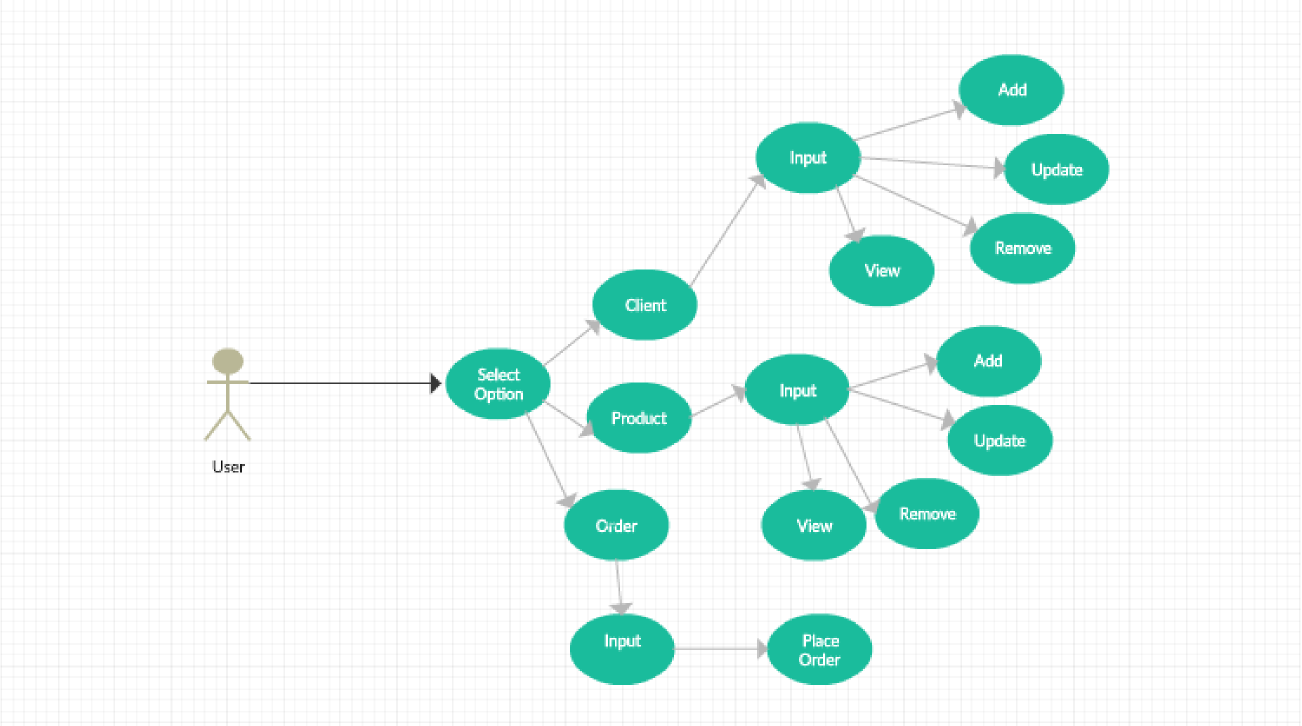
If the order option is selected in the first window the next window is being displayed:



The selection of client and product is done via the id feature and a quantity can be inputted. When the place order button is pressed, underlying tests are being held to ensure the credibility of the order placement process and only if the requirements are fulfilled an order will be placed, meaning it will be added as an entry to the order table and an electronic bill will be generated.

5. Use Case

To make sure the application is working properly we need a protagonist. This protagonist will be the user. Prompted with the above described windows he will have to make some choices, one triggering others decisions to be made. When working with the client and product areas, the user will have to input data first, then chose what to do with it. When one of this choices is selected, the input data is tested and in case it does not fulfill or match the requirements exceptions will be thrown and the execution stopped. If all goes as planned and everything is executed successfully then the changes could be noticed directly in the database management software or through the view options offered by this application.



6. Implementing and testing

The most used technique of testing was the insertion of the *System.out.println*(…) method in different stages of the development process. Using this method came in handy when the result of a certain operation was needed to be tested for proper working .The main advantage of this method is that it offers quick feedback and it may help locate the unusual behavior source very fast. On the other hand, the fast performance comes with a major disadvantage, its size and many lines of code if it is needed repeatedly and also the impracticality aspect in some cases.

Trial and error was the next step to obtain a perfectly working operation. After developing all the required method, any programmer would think that it should work. Unfortunately, this in not always the case. For the first test of proper performing, some input data will be entered, and a proper result should be obtained, of course considering the entered data is of proper type. If the obtained result does not matches the expected one, the first step is to check if the algorithm is correct and make the needed changes and alterations if necessary, which most of the time are.

A good technique and approach is to verify the correctness of the small modules included into the large project. If the correctness and proper performance of the basic parts is ensured, the errors in the final step of the testing are less likely to appear.

If by any chance, the method implemented does not give the desired results, even after many and may modifications, then a logical rethinking must be performed. Taking the problem step by step may give the result through a new approach, or it may help you figure out what is not correct in your first approach. This method may be very effective but it will surely cost more time. Despite that, in some cases, figuring out a new approach takes less time that figuring out the mistake in a poorly implemented and rapidly, in the urge of the moment developed solution.

7. Results

The most important objectives this project aimed to satisfy have been accomplished. A user friendly interface has been delivered, with its functionalities implemented. All the methods could be further developed and there is also space for improvements in the future. The application offers the basic operations on database entities and records which are fully functional and good to go, ensuring the overall success of the project.

8. Conclusion

The project helped dusting off the database entities and notions knowledge acquired during the years I have studied in the informatics domain. This project also help me get a better understanding of how the initially simple looking but in fact complex operations on databases work. It was also a good exercise of design, the way each functionality should be put in the right place, so it could offer the best results. One of the main aspects which I understood from developing this application is the fact that a good initial design and organized initial approach helps a lot when working with the object oriented programming entities. If the base is done correctly and efficiently it saves a lot of time when developing the solution and also excludes some of rethinking processes usually mandatory. Another aspect to be mentioned is that if the initial design of the classes is done properly, the needed calls for the methods and objects can be done in far less words, which is helpful in large, complex projects.

9. Further Development

The application presented in this project serves an enormous potential for further implementations of more and more complex aspects .To be noted as future enhancements would be the inclusion of more complex operation in terms of fields and their sizes, meaning larger values, as well as formats. As the user interface interaction window resembles a fill in form, the application could be developed to offer different functionalities, such as automatic fill in and fall down tables. The operations offered and implemented could be also further developed, in a manner in which more operation of different complexities could be added. One step forward would be to change the input methods by offering numbers and certain characters responsible for inserting different data, for example "@".

As far as the graphical user interface is concerned, if all the above mentioned enhancements are to be implemented they would need a graphical implementation as well. All the extra added operations will require a button for selecting it and if we were to implement a larger number of features participating to the operation, input text boxes need to be added to the final design. If we were to talk strictly about the present application the graphical user interface can be modified so it has a more attractive look. The functionality and ease of use of the present developed application should be enough for the moment, none of them requiring more explicit visualization. One feature that could be eventually mentioned could be the way the input should be typed in, a hint may be offered to the user at the top of the window.