# 

**PROGRAMME**: IFY ENGINEERING AND COMPUTING

**MODULE NAME:** Information system development

**MODULE CODE:** LIPC1261

**TUTOR:** Moath Al-Doori

**NAME:** Cezary Stanislaw Szwalbe

**PNUMBER:** P2446634

**Assignment title: Car Rent System**

**Word count:**

**Date: 06/06/19**

1. **Abstract:**

My car rent system generally serves to provide customers with renting several types of cars by using easy online interface. External layout is based on Html and CSS language. To rent a car user has to open his own account and after logged in he will be allowed to transfer selected cars to his basket. The connection between interface and the database called middle layer allows administrators to adding, deleting and editing whole information about cars. To make this relation I have used C sharp language. Internal layer which is the database uses SQL language. Whole of these functionality are provided my with visual studio software. Due to lack of experience, I struggled with SQL syntax which was not automatically correcting and I have lost over two hours to find a typo.

1. **Introduction**

Information systems are undoubtedly necessary for today’s world, and as the most important part them preparation process we can place planning that prevents the loss of time in next processes and makes entire project easier to do. Next steps are creating, testing and deploying. To make my car rent system I have used exactly the same method, which conducted to the completion of the project on time with satisfactory amount of functions. The database used in the system consists of four tables where through retrieving and storing data provide us with saving information for instance about types of chosen cars or country of registered users. Four types of programing language were used to make this project, HTML and CSS for generally view, C# to create functions and SQL to operation inside the database.

1. **Analysis**

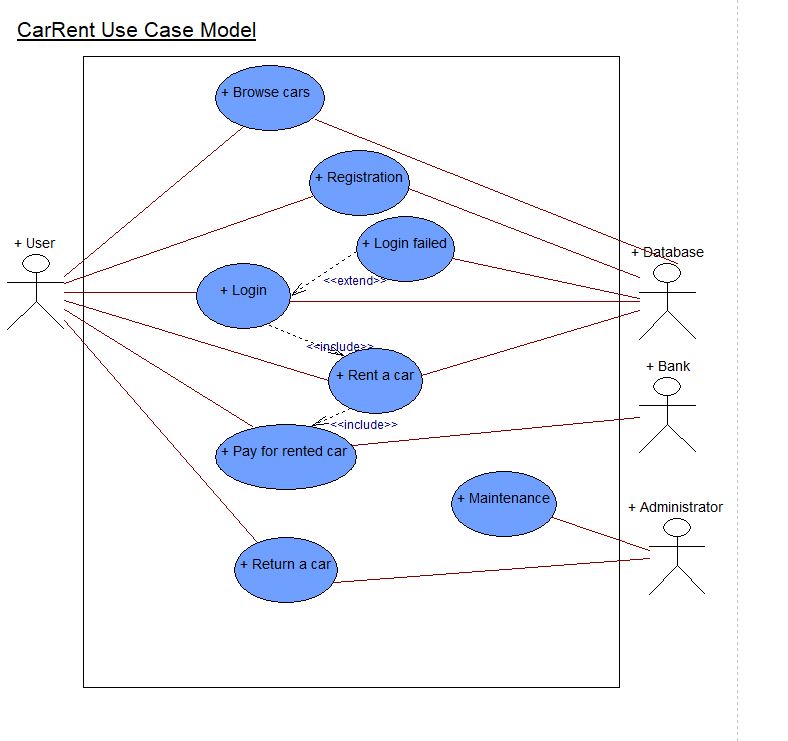
****

Figure: 1

Use Case model clearly shows relationships between use cases and actors.

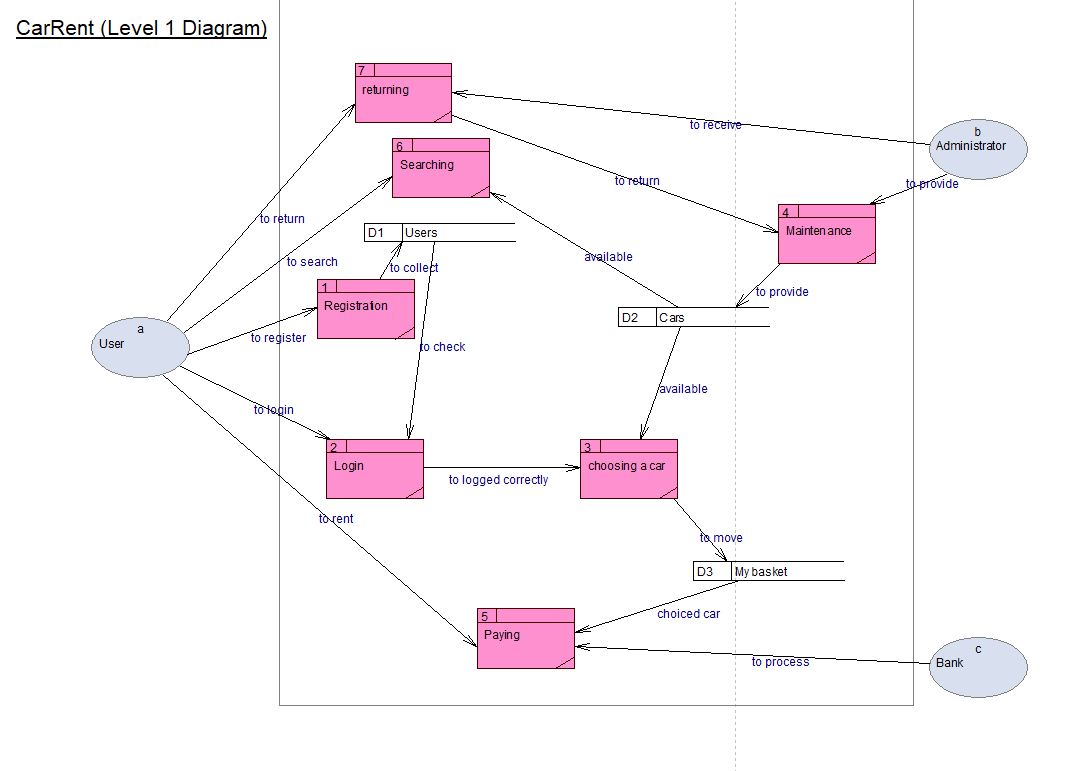


Figure: 2

Data Flow Diagram is a representation of data flowing inside the system, shows information about inputs and outputs of each entity. The difference between two of diagrams is very visible, it is manifested by the fact that DFD is an extended version of UC diagram because it shows boundaries and limitations. In contrast to UC diagram, DFD diagram shows the data source, destinations and how will be transform and used in the system.

**4. Design considerations:**

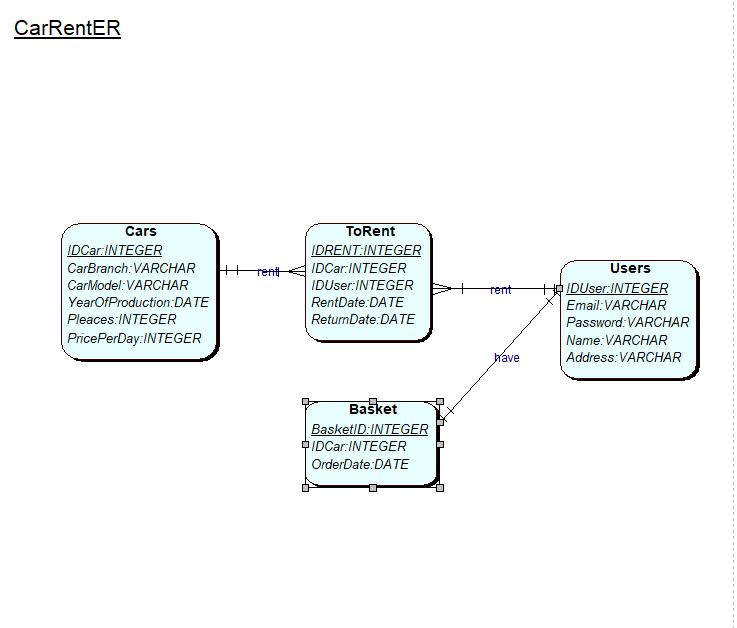


Figure: 3

Entity relationship diagram is about how the whole data is storing in the database and shows connections between tables.

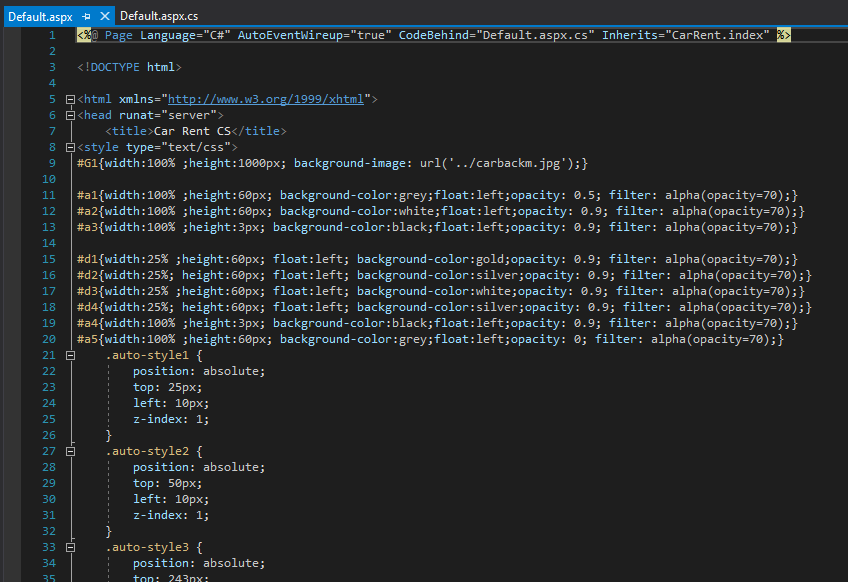


Figure: 4

By using a combination of HTML language with CSS allowed me to get a professional website appearance. Photo in the background adapted to the overall purpose of page and transparency of some divs in a simple way affect the good user experience. Quiet colours provide the impression of overall harmony.

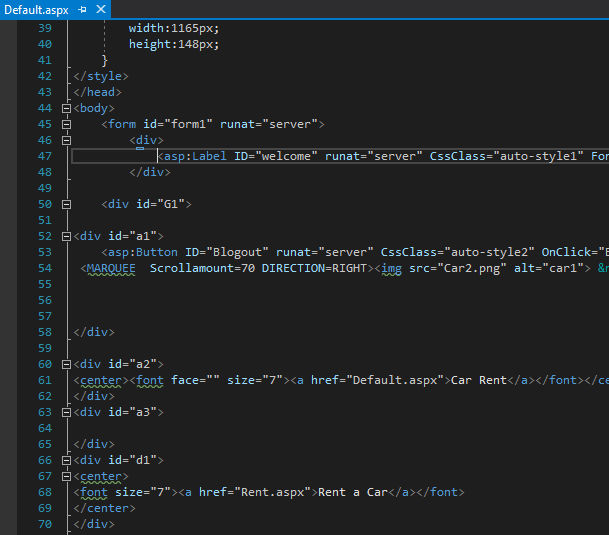


Figure: 5

General page appearance enriched with moving elements implemented by the appropriate use of the MARQUEE command distinguishes my project from others adding originality. The colour and size of the fonts used must not have the effect of merging with the rest.

1. **Implementation:**

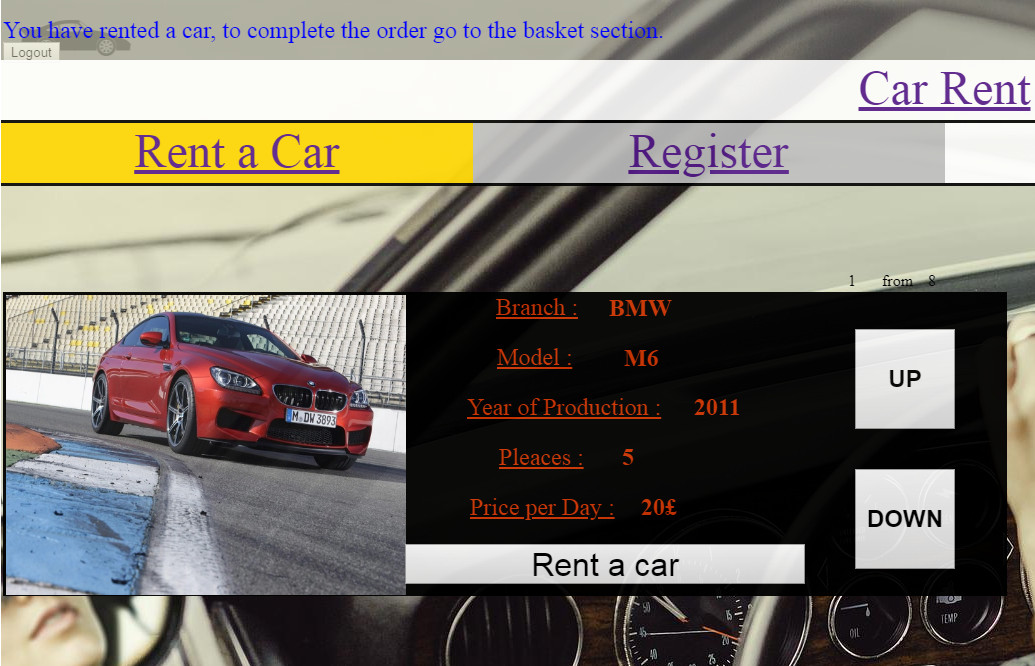


Figure: 6

Choosing your car is only possible after logging in to the system, so when you are logged in and click on the rental button, at the top of the page a command will be displayed informing you about the success of this operation. At this moment a record was added to the "basket" table in the database, which contains information about the id of the car, user id and the order date. It is possible by using the C sharp connection with SQL.

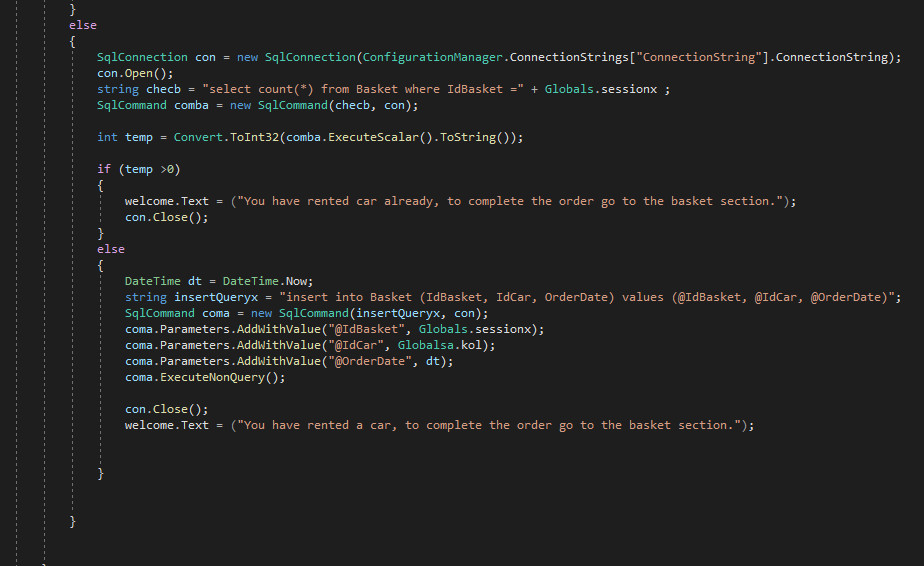


Figure: 7

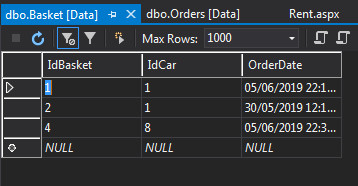
By using C sharp in conjunction with SQL it is possible to connect the user interface and database.

Figure: 8

Figure nr 8 shows the correct addition of the record of the selected car and indicates the one-to-one connection of the id basket attribute with Id user.

1. **Testing**

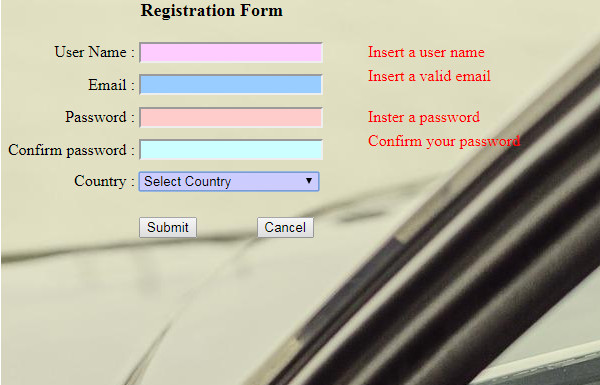


Figure: 9

The above photo shows that to complete the registrations, you must complete all fields successfully. In case of omitting one of them, the information informing the user is displayed where he made a mistake. Entering two different passwords also prevents the user from registering.

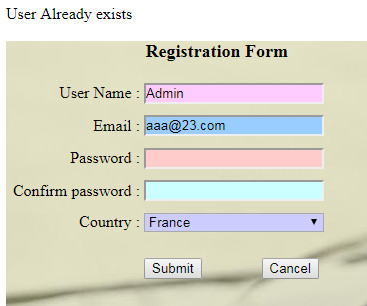


Figure: 10

While the user will use the user's existing name in the registration process, a message will be displayed to him, only to filling all fields correctly and providing a unique name allows to complete the registration successfully and redirects to the main page.

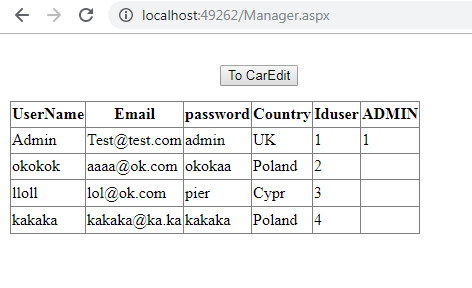


Figure: 11

All the information about users can be checked in the administrator tools

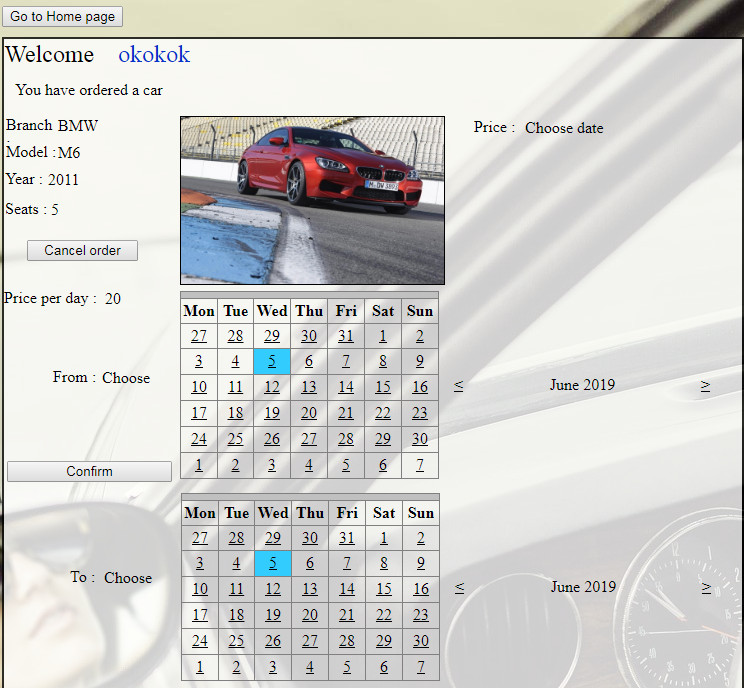


Figure: 12

After logging in and selecting the car, go to the basket section where we can choose the date of renting the car. Only the correct date selection from both calendars results in the transfer of information to the database. If the user does not select one of the dates or mark the day that has already passed or on the first calendar will indicate the date earlier than on the second one a message will be displayed to correct his choice of dates.

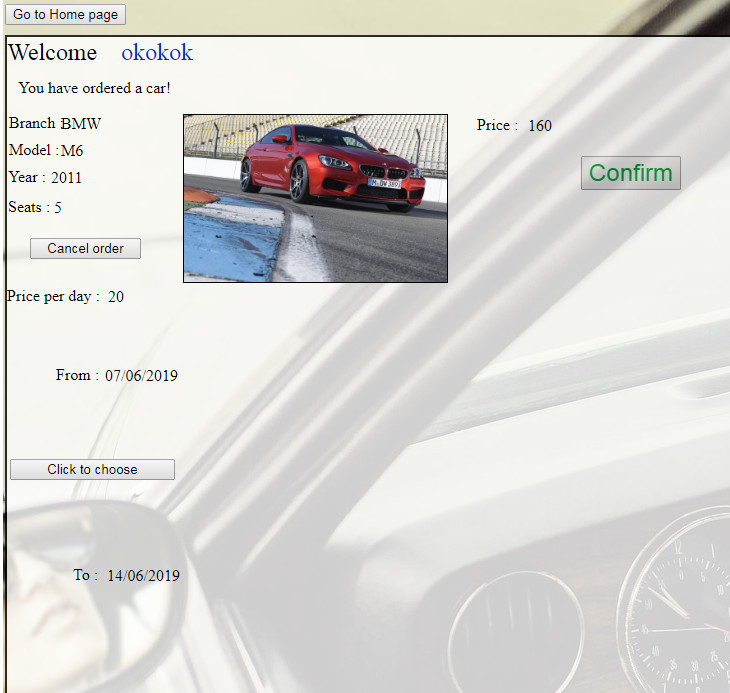


Figure: 13

When the correct date is selected and the user clicks the button to confirm, the price is calculated and shown to the user and the operation confirmation button appears on the screen.

1. **Conclusion:**

Generally, I'm happy with the work I've done. The project interested me and absorbed for many hours which gave me a lot of experience and influenced the decision about choosing future research connected with doing such projects. I could find many surfaces where I could still improve my work such as the possibility of choosing more than one car at a time or viewing all available cars at once not one after the other. Knowledge and experience I will try to use in future projects of this type. Applications of this type have unlimited possibilities, and connection to the database allows for the creation of large services such as social.

1. **Referencing**

SHARP, J. (2010) Microsoft Visual C# 2010 Step by Step. New York: Microsoft Press

KARWIN, B. (2014) SQL Antipatterns. Dallas, Texas The Pragmatic Bookshelf