# 

**PROGRAMME**: IFY ENGINEERING AND COMPUTING

**MODULE NAME:** Visual Web Development

**MODULE CODE:** LIPC1263

**TUTOR:** Salimah Mohamed/Burket Ali

**NAME:** Cezary Stanislaw Szwalbe

**PNUMBER:** P2446634

**Introduction**

The assumption of my project “An Online DVD shop” was to give the user the opportunity to buy DVD movies by using a simple to use interface based on HTML in conjunction with CSS connected to the database. In order for the user to buy any DVD, he is first required to create an account on my website. I tried to perform the most reliable system by adding validations and limiting the access of some functions without first selecting objects that are necessary to perform related operations. For example, the edit button did not appear without selecting any of the videos. In performing the most consistent appearance of the whole page, I used CSS styles. To ensure the administrator the ability to edit pictures on the main page, I used the input output function. Matched to the back of the page, the colours of the fonts are clearly visible so that the user can move freely between the sections. The database used in my system consists of five tables to which data is transferred and retrieving using SQL language which in combination with the C # creates all functionalities.

**Design Considerations**

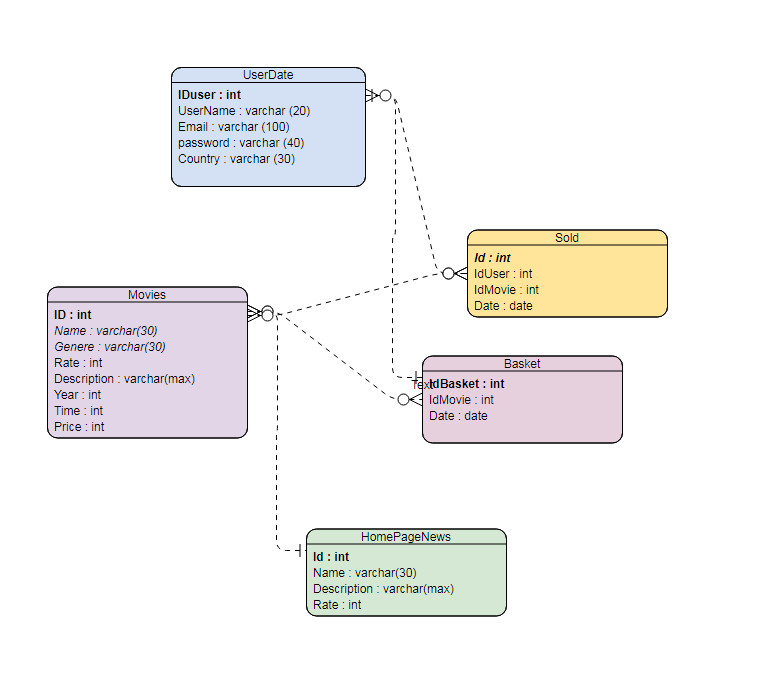
****

Figure: 1

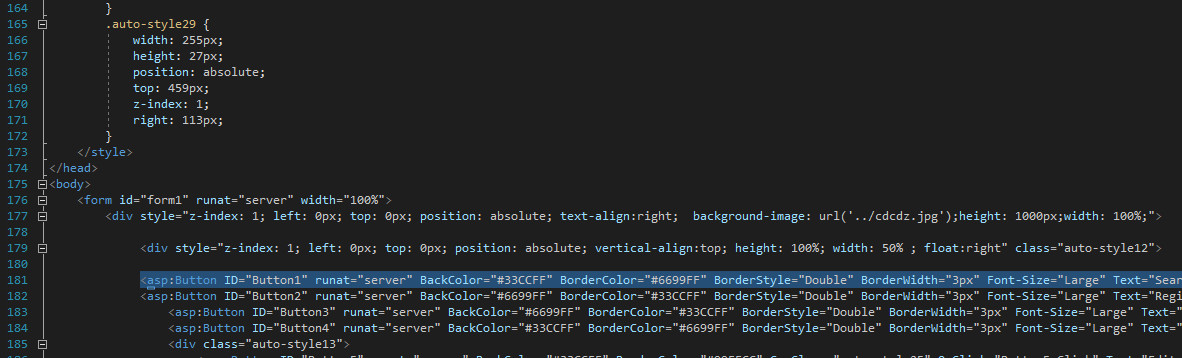
Data relationship diagram shows all my tables, including the type of data recording and connections between them. Storing data is very important in the case of large services because the time of access to data can be significantly extended in badly designed databases, repeating the same records should not take place and to avoid this, I use many to many relation in part of the my solution. Creating a consistent scheme can significantly reduce the time needed for the subsequent stages of creating an information system.

Figure 2

I used CSS Styles and HTML to get a nice look, the biggest difficulty was cantering the whole page, I achieved it by creating a div always occupying 50 percent of the screen which the rest of the elements will align with.

**Implementation**

Figure 3

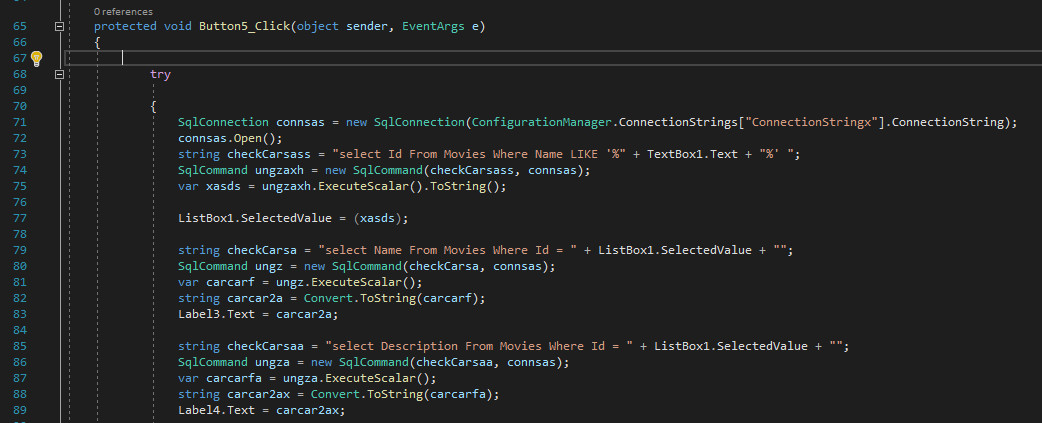
In order to be able to search for videos on the website without problems, I have equipped system with a tool for finding movies, it finds movies even without entering the whole name but only a part of it if the user does not remember the whole name.

Figure 4

To achieve a satisfying effect of this function, I have used the above-mentioned code, it is based on the use of the LIKE command in the SQL syntax. After selecting a video from the list, the user can get acquainted with the details of the film, but in order to transfer the film to the basket first, it is obliged to log in.

Figure 5

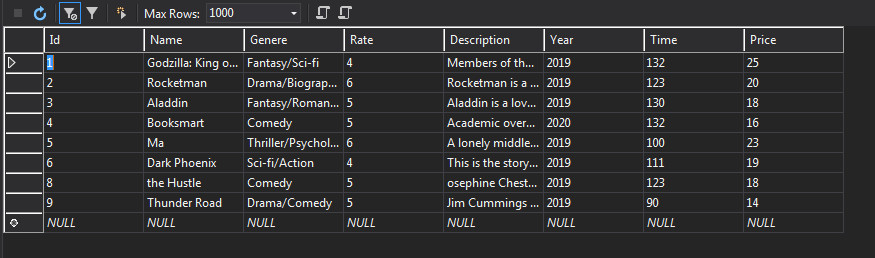
Administrator account has the ability to edit the removal and adding videos by a specially constructed tool, after completing one of these operations, a success message is displayed on the screen. 

Figure 6

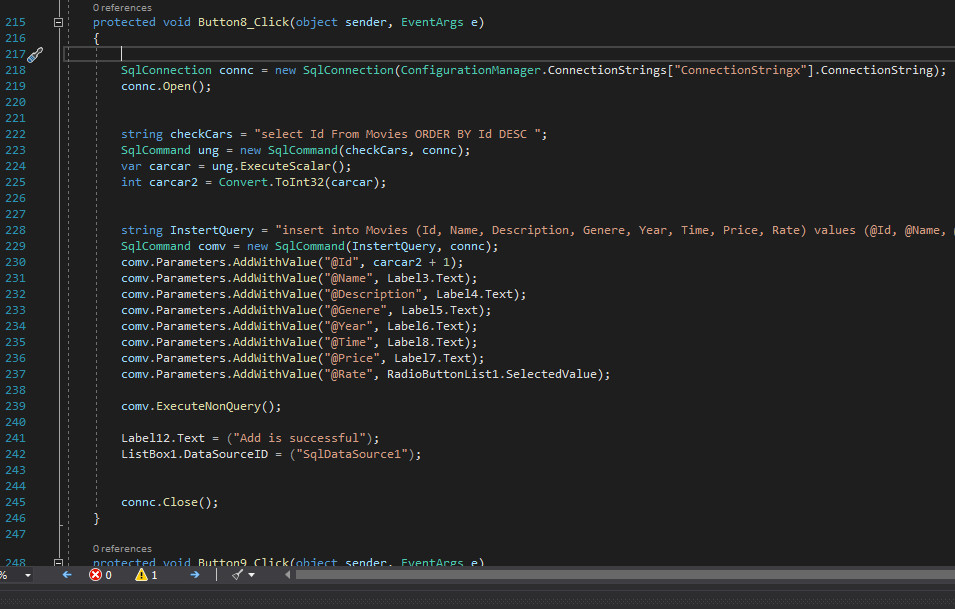
All changes made are saved directly in the Movies table.

Figure 7

Figure number 7 shows an example of using the C sharp connection with SQL to create functions for adding new movies to one of the database tables using the insert into command.

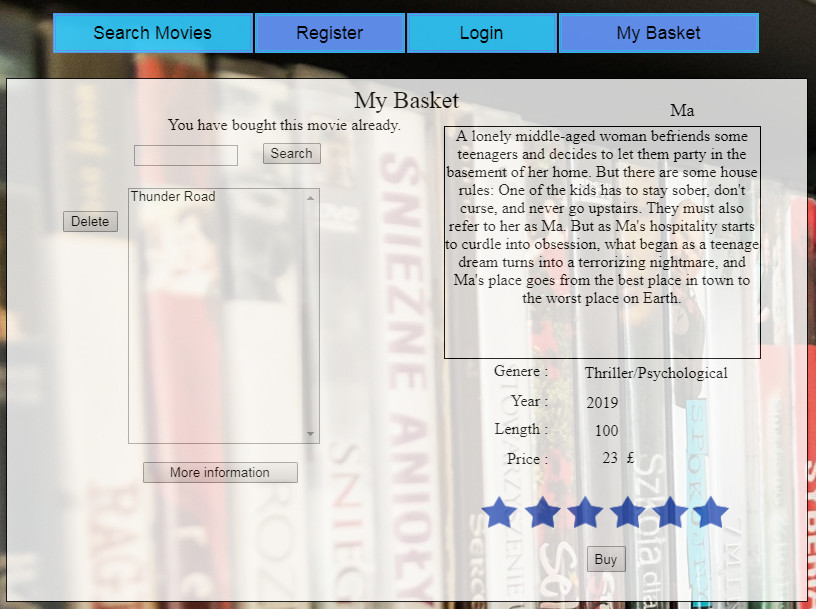


Figure 8

In the basket section the user can purchase selected films, if the user has previously bought the same movie, the message informing him will be displayed.

**Testing**

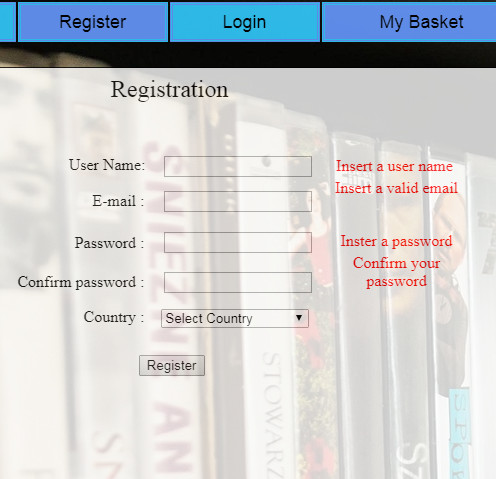
****

Figure 9

If the user who wants to register does not complete all fields, the validation system will forbid him from going forward and will advise him to fix the errors first

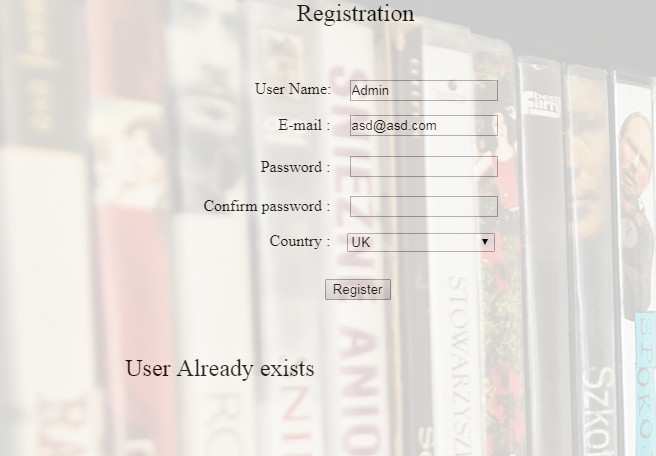


Figure 10

If user use an existing username to register, he will be prevented from creating an account and a command informing him about it will be shown to him.

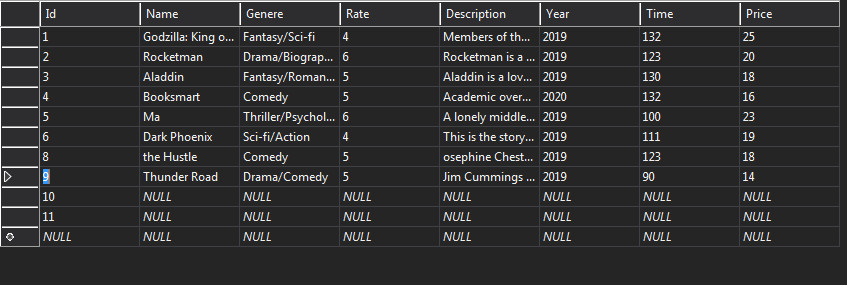


Figure 11

When a new movie is introduced to the database, there is a possibility to create an empty record because I have not implemented validations in this section however, this is not a big problem because it does not cause system errors and you can quickly edit such records.

**User Documentation**

****

Figure 12

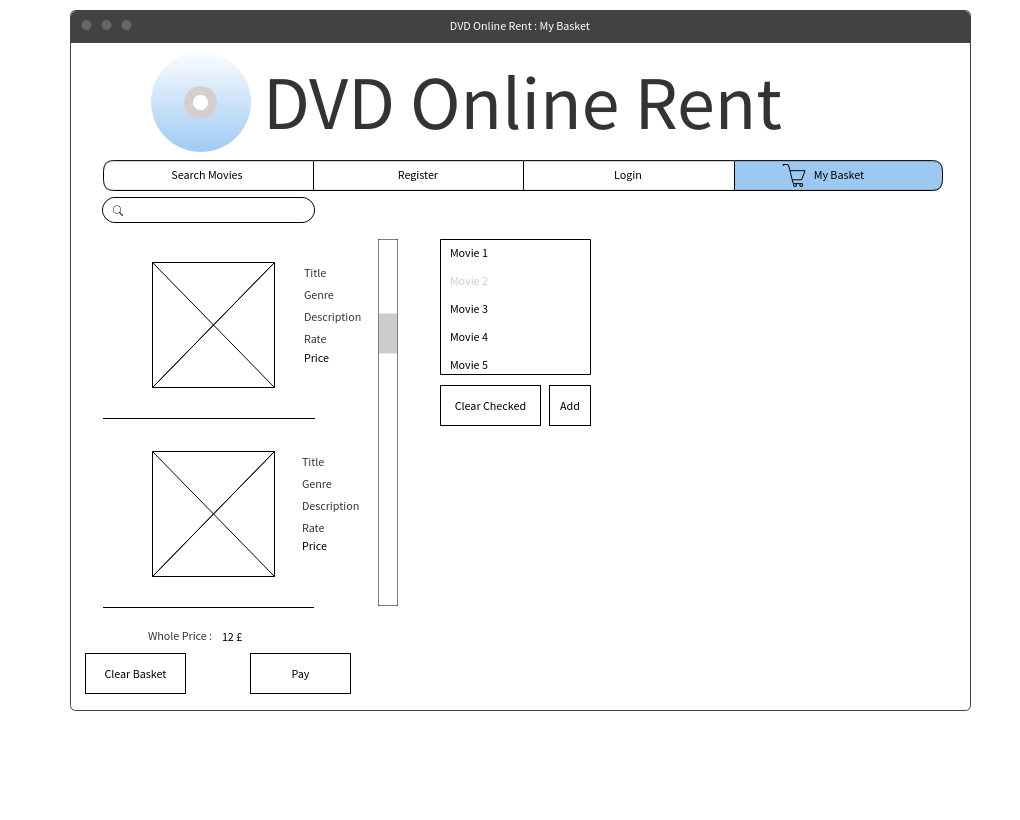
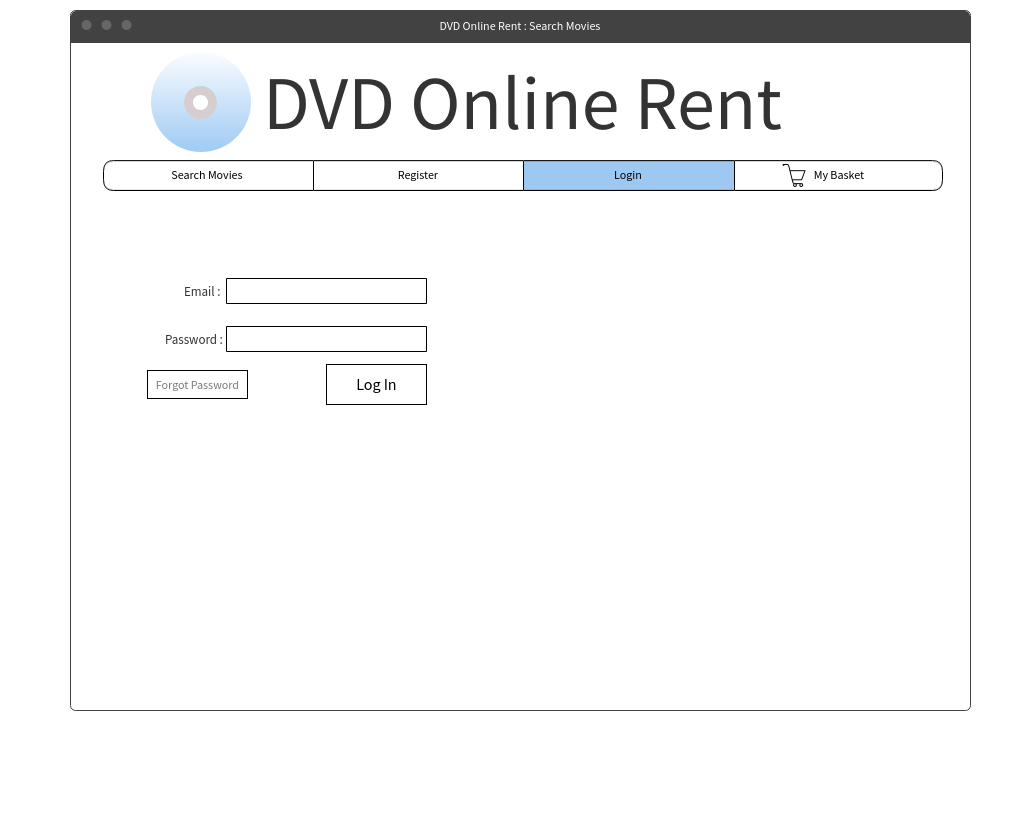
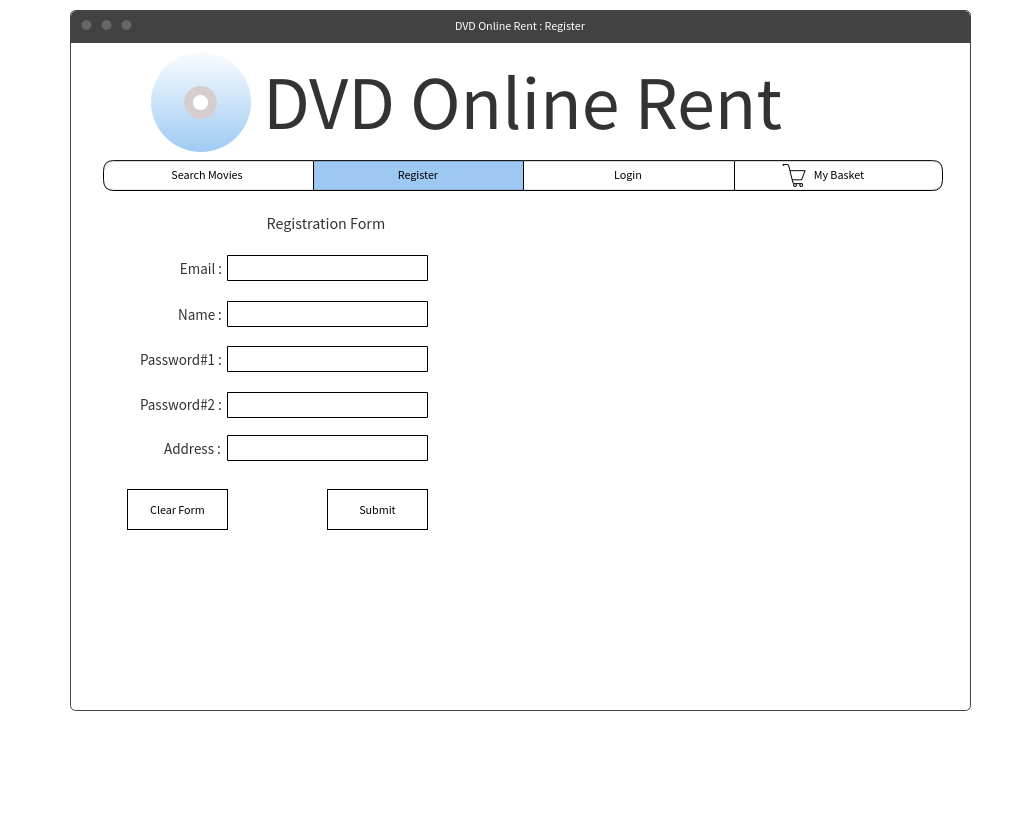
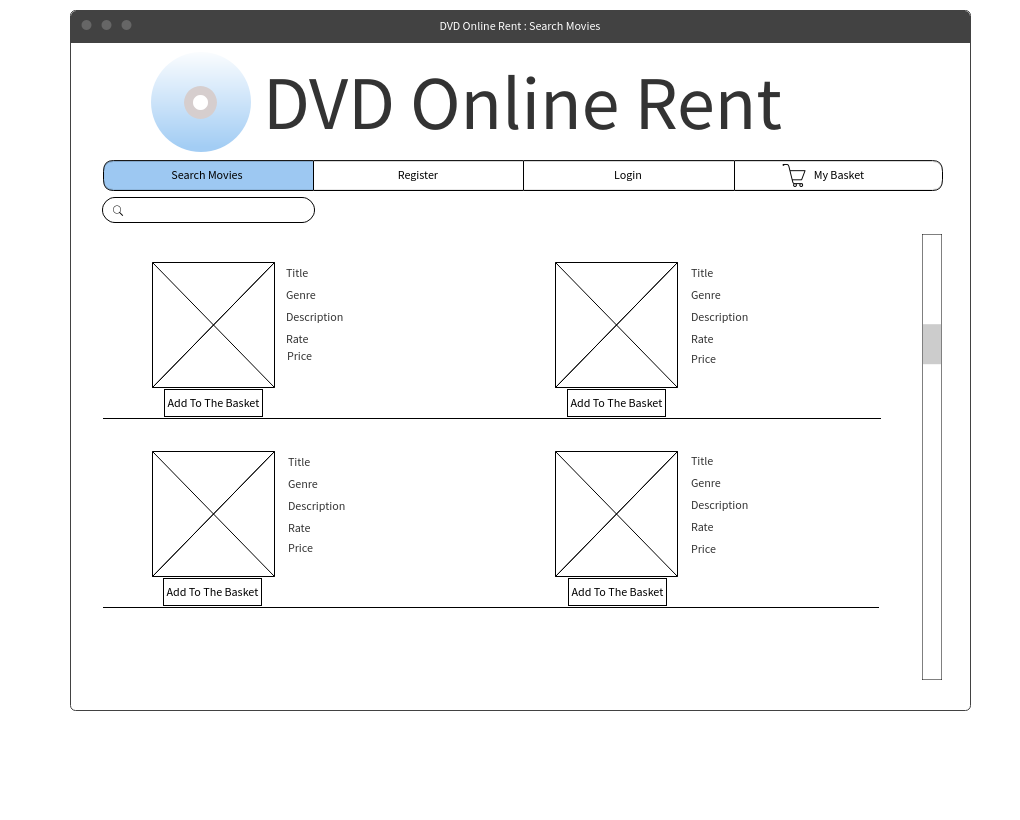
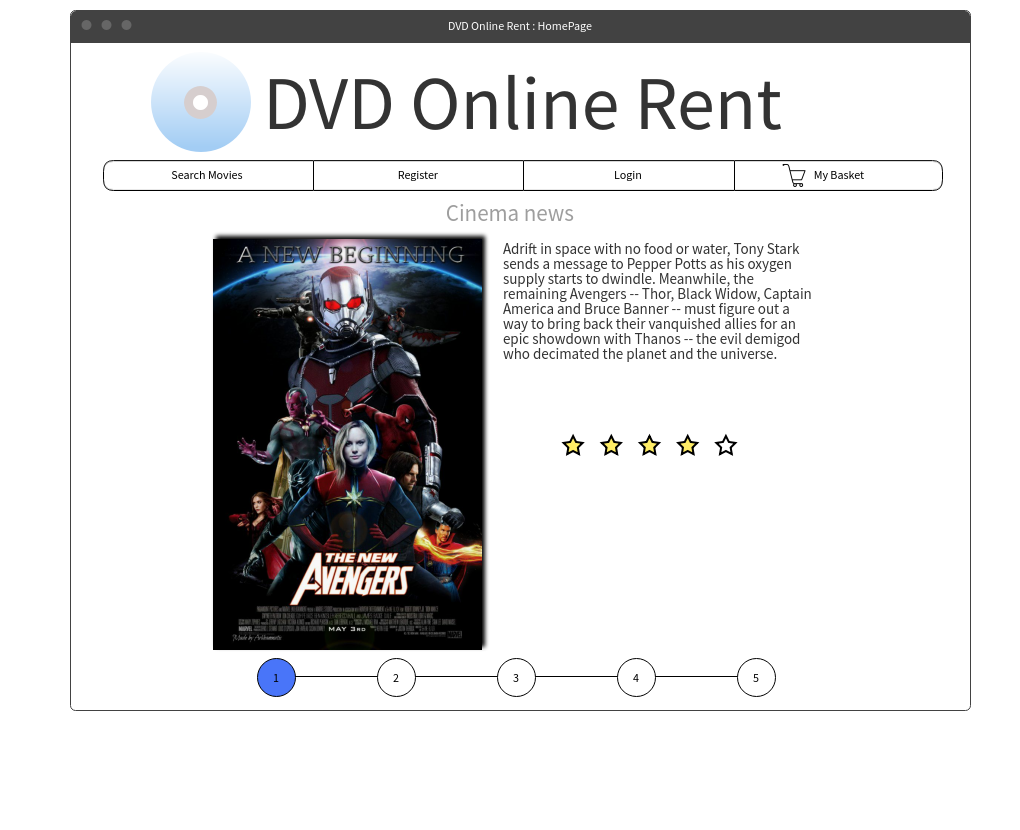
The whole system is coherent and ready for operation after switching on. The main page is intended to show film novelties and interest of the user through inserted photos. The administrator can edit photos and descriptions of the title page. Visible names of the tabs instruct the user how to make a movie purchase. As shown in figure 3 and 8, after logging in, the user is ready to choose his own films and after the transfer to the bookmark basket can buy them. The use of simple solutions prevents the user from getting lost on the site.

**Critical analysis and conclusion**

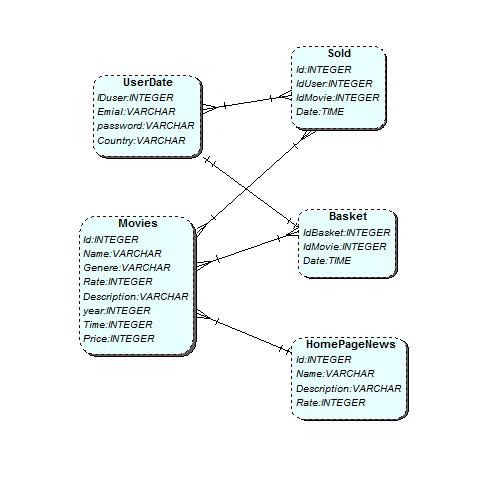
Summarizing the time spent on this project provided me with a lot of experience. I tried to demonstrate creativity and at the same time the simplicity of my solutions. Creating more functions made me happy, which means to me that I was not mistaken in choosing my field of study. I think that creating such projects is the direction of my future education. I tried to make the project as clean and user-friendly as possible, but there are many surfaces on which I could do more. Information systems are still a growing field of science, computers are replacing people more and more. In order to be able to implement my system for public use, it would be necessary to add payment functions for films and extend the database with new tables.

**Appendices**

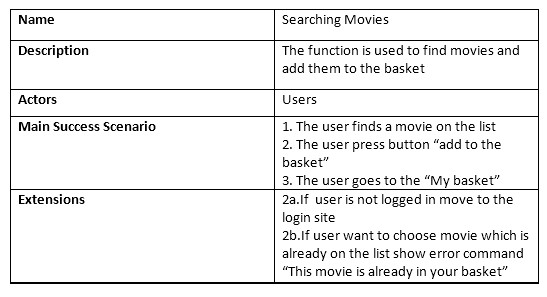
Appendix 1 – Wireframes

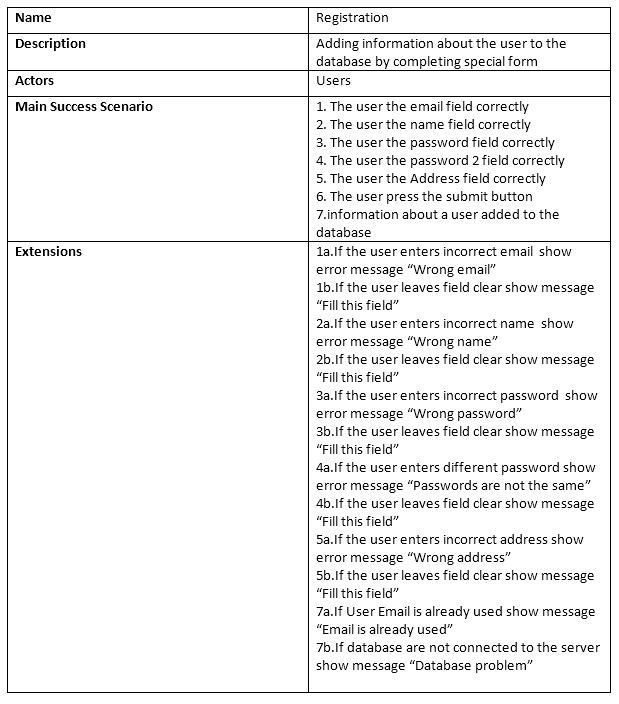


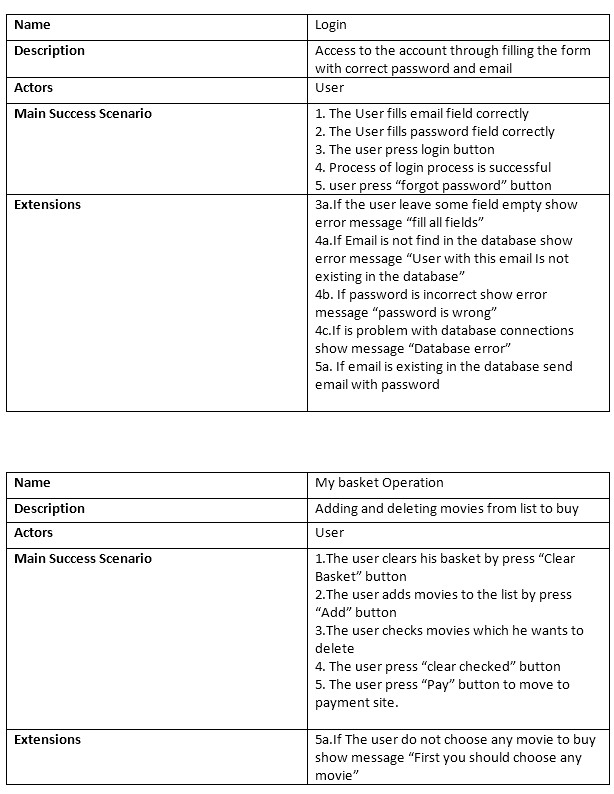
Appendix 2 – Entity Relationship Diagram



Appendix 3 - Use case diagram







**Bibliography**

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

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