



ÇANKAYA UNIVERSITY

Faculty of Art and Science
Department of Mathematics and Computer Science
MCS 492 Graduation Project II

CINEMASOFT : A CINEMA BOOKING SYSTEM

Buket Çakmak 200822011

Instructor: Assist. Prof. Dr. Tolga PUSATLI

May 2012 Ankara

CONTENTS	2
1. INTRODUCTION	3
1.1 OBJECTIVES	3
1.2 LIMITATIONS	3
2. DATABASE DESIGN	4
2.1 COLLECTION OF REQUIREMENTS	5
2.2 CONCEPTUAL DESIGN	7
2.3 LOGICAL DESIGN	8
3. USERS	9
3.1 CUSTOMER	9
3.2 EMPLOYEE	9
3.3 CINEMA MANAGER	9
3.4 ADMINISTRATOR	9
4. USER INTERFACES	10
4.1 FIRST SCREEN	11
4.2 ADMINISTRATOR FORM	11
4.3 CINEMA MANAGER FORM	12
4.4 EMPLOYEE FORM	13
4.5 ADD/REMOVE THEATRE FORM	14
4.6 ADD/REMOVE EMPLOYEE FORM	14
4.7 ADD/REMOVE MOVIE FORM	15
4.8 RESERVATION AND SELLING FORM	15
5. SAMPLE CODES FROM PROGRAM	16
5.1 ADDING A NEW EMPLOYEE PROCESS	16
5.2 REMOVING AN EMPLOYEE PROCESS	16
6. TESTING	16
7. FUTURE WORKS	17
7.1 THREE-TIER MODELING	17
7.2 PERFORMANCE AUGMENTATION PROCESS	18
7.3 PRINTING TICKET	18
8. CONCLUSION	18
9. REFERENCES	19
10. BIBLIOGRAPHY	19

1.INTRODUCTION

1.1 OBJECTIVES

This project is based on a scenario. In this scenario the customer wants to know all movies with their show times in the city when he decides to go to cinema. He can do this through the Internet or by a phone call.

If he uses the Internet, he will learn all movies with their show times and make his choice then will select the movie and show time and submit identification information (his name, phone number for communication etc.) to complete the booking the process.

If he prefers calling, he will meet the employee who work at the cinema and take information from the employee about movies, its show time, cinema's address etc. then he selects a movie and its show time, and he gives some personal information about himself (e.g. his name, phone number and number of people for seats) and booking process will be completed.

In summary, the aim of this project is to satisfy potential customers' requests to see all movies in show along with their show time and place in Ankara, and reserve seats.

1.2 LIMITATIONS

Our project has the following limitations. Firstly we are studying the cinemas that are only located in Ankara. Secondly our program has no information about cinema's time of incorporation, other shops or etc. in it, it has only about cinemas' theatres. Thirdly it should be said that the customer would be able to booking process only; not buying cinema ticket. Employee's side will do the booking and selling process. Finally we must determine that; in this system the customer will not select seat number. The authorized people in this process will be the employee who works at the cinema.

Also, a printing system will not be in this project. In future, a ticket printing system can be entegrated. (Section 7.3)

2. DATABASE DESIGN

The database-planning phase begins when a customer requests to develop a database project. It is set of tasks or activities, which decide the resources required in the database development and time limits of different activities. The phases of database design is as shown below:

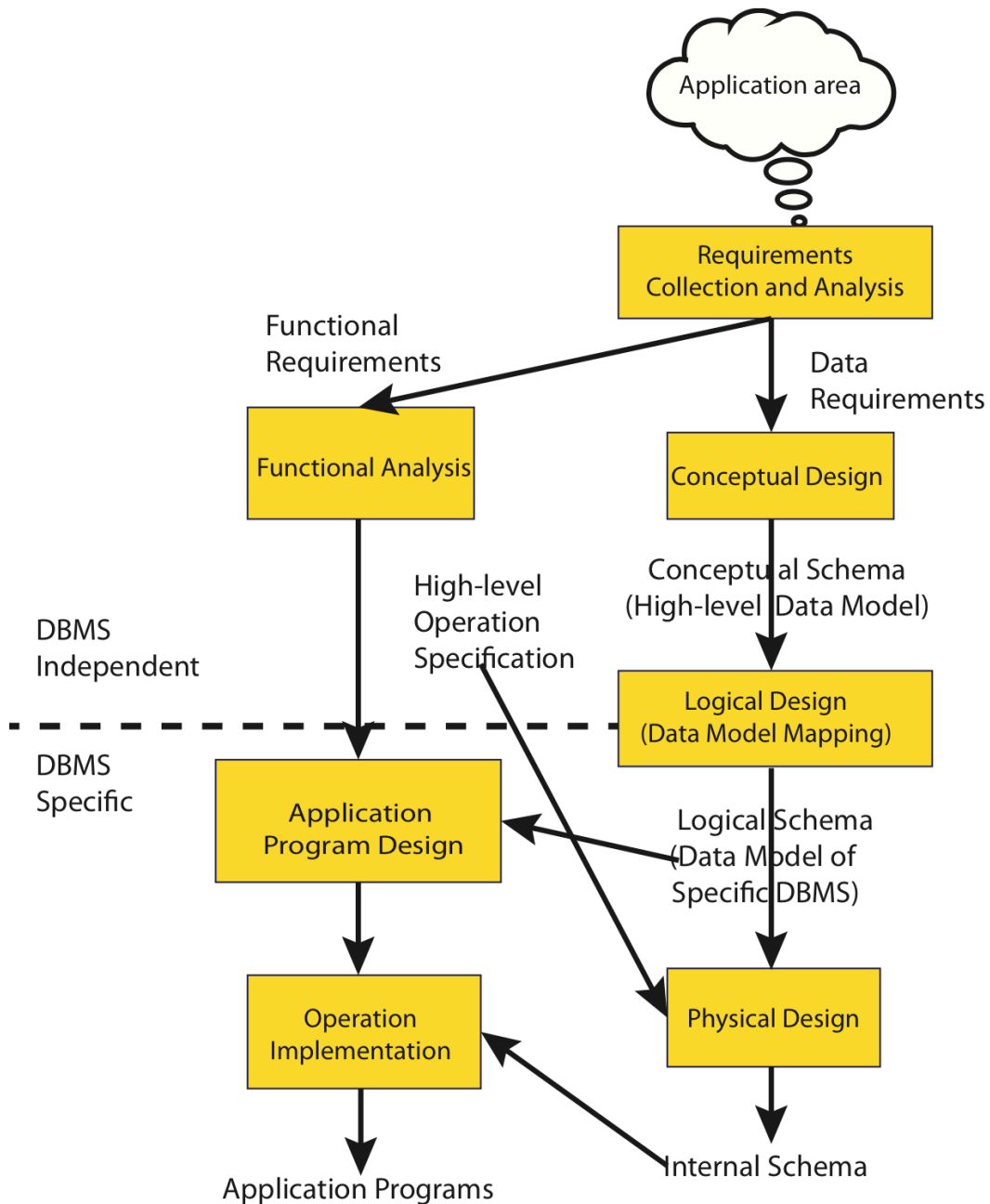


Figure 2.1 : The main phases of database design [1]

2.1 COLLECTION OF REQUIREMENTS

Requirements analysis is done in order to understand the problem, which is to be solved. That is very important activity for the development of database system. The requirements and the collection analysis phase produce both data requirements and functional requirements. The data requirements are used as a source of database design. The data requirements should be specified in as detailed and complete form as possible.

In parallel with specifying the data requirements, it is useful to specify the known functional requirements of the application. These consist of user-defined operations that will be applied to the database (retrievals and updates). The functional requirements are used as a source of application software design. Steps in preparing this graphical user interface model presented on Figure 2.2 :

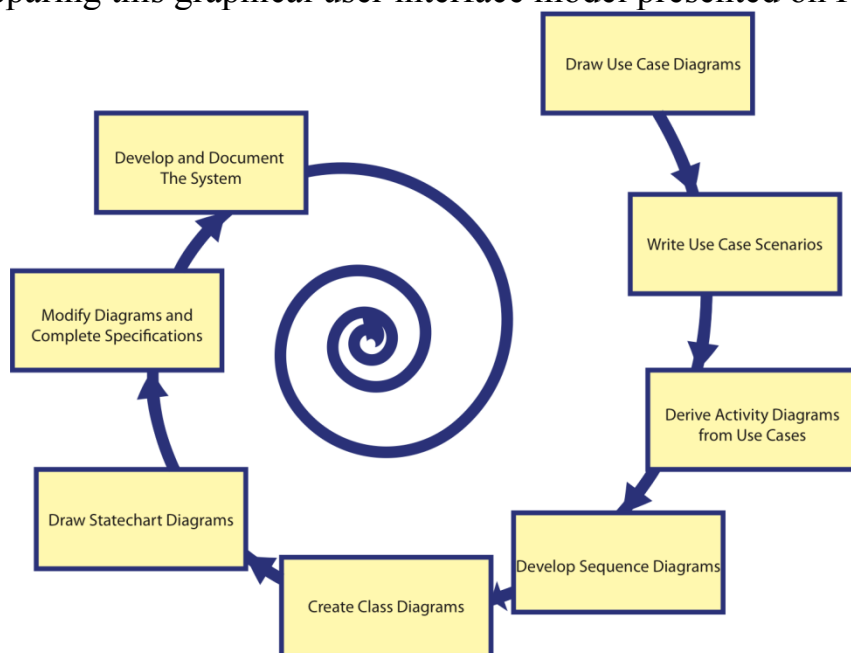


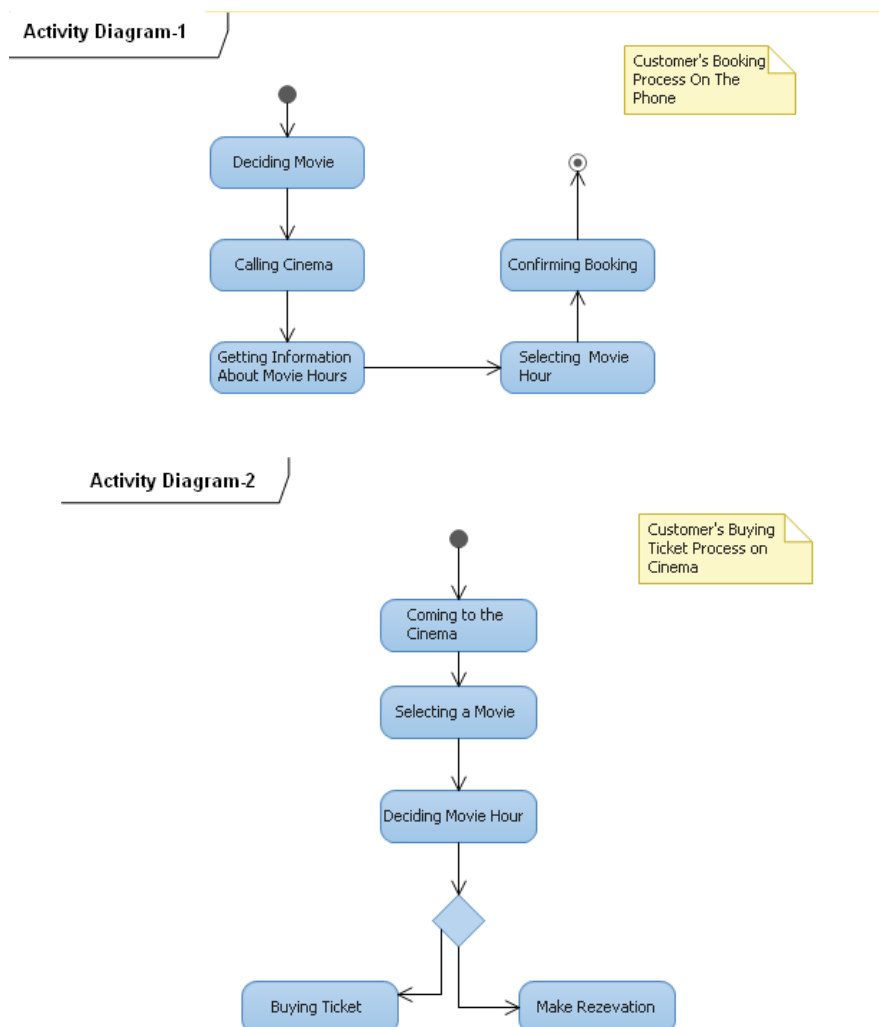
Figure 2.2 : Functional Requirements Cycle [2]

Although full functional design is excluded in this project, we prepare use case diagrams (Figure 2.3) as examples that this part is necessary. If we continue on this way, we must draw activity diagram, sequence diagram etc. Here are sample diagrams in the following:

Figure 2.4 : Use case scenario example ;

Use-case name :	Successfully make a system-assisted booking	Use-case reference no :	0001
Initiating event :	Customer contacts cinema, requesting a booking		
Key outputs :	None		
Key state changes :	New booking added to system		
Script : A booking request is received (e.g. by phone or over the Net) The details of the requested booking will be input to the system. The system will locate a suitable seat or seats. It means a seat is free at the time in question. The system allocates the booking.			

Figure 2.5 : Activity Diagrams;



2.2 CONCEPTUAL DESIGN

Once all the requirements have been collected and analyzed, the next step is to create a conceptual schema for the database, using a high-level conceptual data model. This phase is the conceptual design.

The result of this phase is an Entity-Relationship (ER) diagram or UML class diagram. It is a high-level data model of the specific application area. It describes how different entities (objects, items) are related to each other. It also describes what attributes (properties) each entity has. It includes the definitions of all the concepts (entities, attributes) of the application area.

In this part, ER diagram is drawn with the information, which is taken from the cinema company, and restrictions are specified. We stated to prepare an interface from functional requirements (GUI). For interface, we used the programming language C#.

C# is the first component oriented language in the C and C++ family of languages. It is a simple, modern, object oriented and type- safe programming language derived from C and C++. As it inherits from C++, C# is a powerful programming language; additionally, it is easy to code by thanks to its user friendly IDE interface provided by Microsoft.

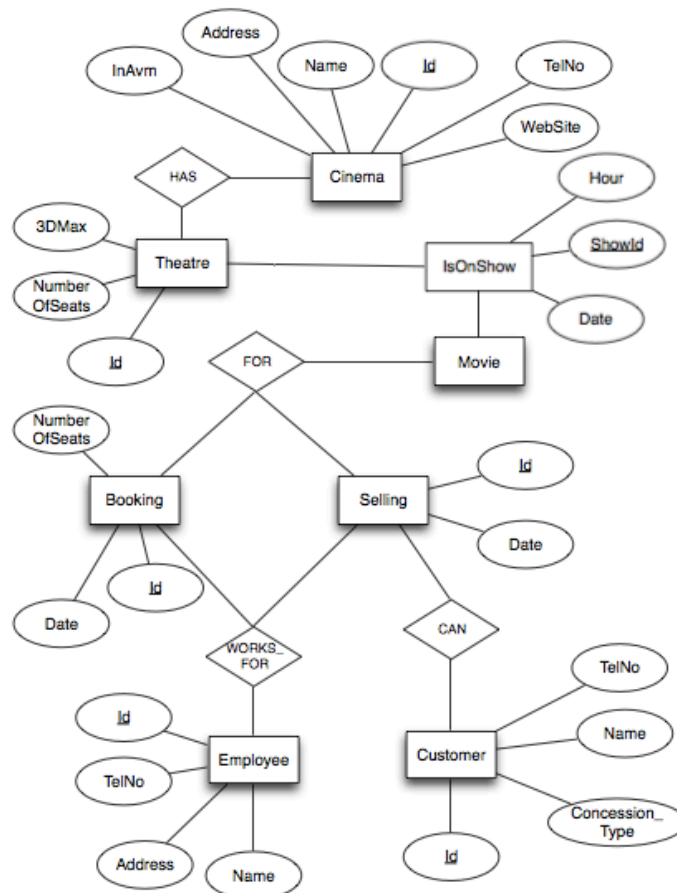


Figure 2.6 : ER Diagram of Cinemasoft

2.3 LOGICAL DESIGN

There are rules how the ER model or class diagram is transferred to relation schemas.

The relation schemas are the basis for table definitions. In this phase (if not done in previous phase) the primary keys and foreign keys are defined.

While preparing the logical design part, we applied relational data model. To do this, we decided to use SQL Server 2008. SQL-Compliant means it use the ANSI (American National Standard Institute) version of Structured Query Language or 'SQL'. Structured Query Language is a command that allow us to modify or retrieve information from the database. SQL Server is also a Relational Database Management System (RDBMS) and it supports relational data model and it can connect with C# easily.

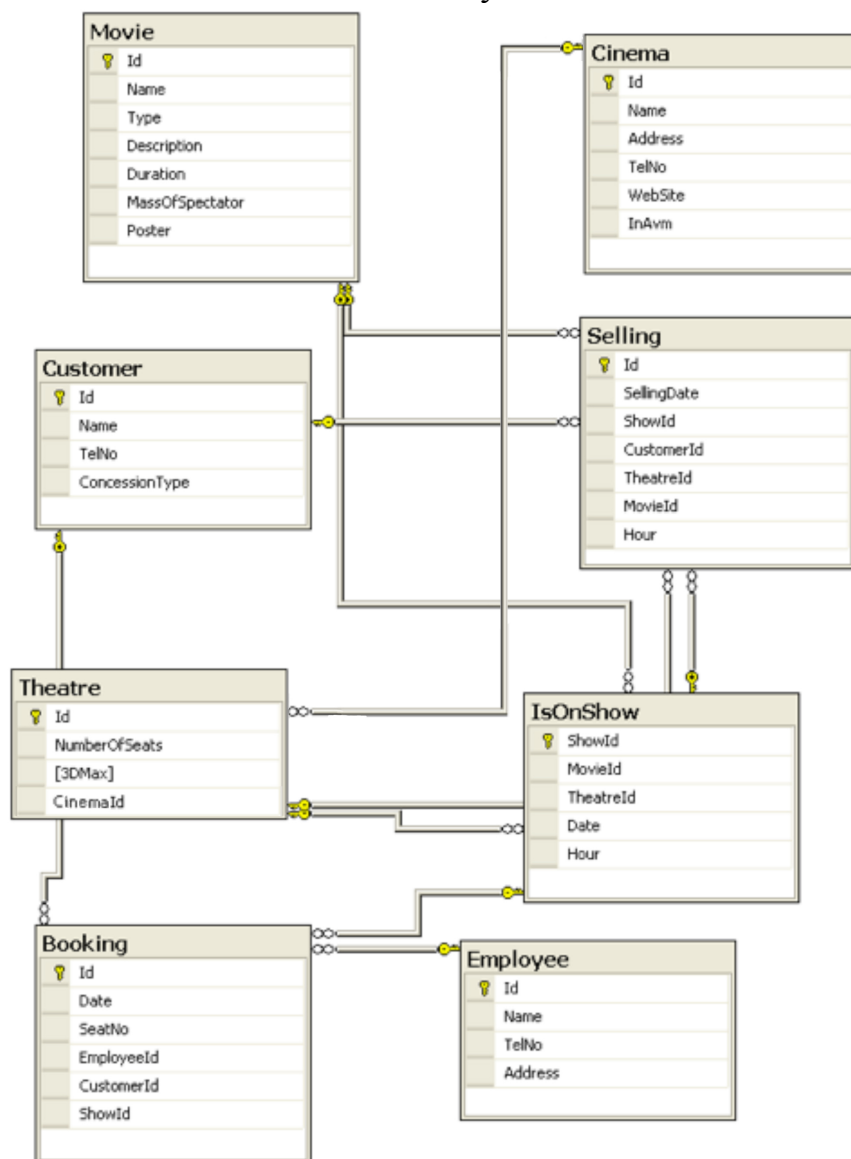


Figure 2.7 : Relational Data Model of Cinemasoft

3. USERS

In this program there are three user types : Customer, Employee, Cinema Manager and Administrator.

3.1 CUSTOMER

The customers are to use the web interface. (Section 7)

- Selects cinema,
- Selects movie,
- Selects show time,
- Selects number of seats,
- Books seat.

3.2 EMPLOYEE

- Books seat,
- Takes the reservations,
- Receives the customer's booking request,
- Sells ticket,
- Selects seat and seat number,
- Adds/Removes customers.

3.3 CINEMA MANAGER

- Adds/Removes seat,
- Adds/Removes theatre,
- Adds/Removes movie,
- Adds/Removes show time,
- Adds/Removes employees,

3.4 ADMINISTRATOR

- Adds/Removes seat,
- Adds/Removes theatre,
- Adds/Removes movie,
- Adds/Removes show time,
- Adds/Removes employees,
- Books seat,
- Sells ticket,
- Selects seats and seat number.

UseCaseDiagram

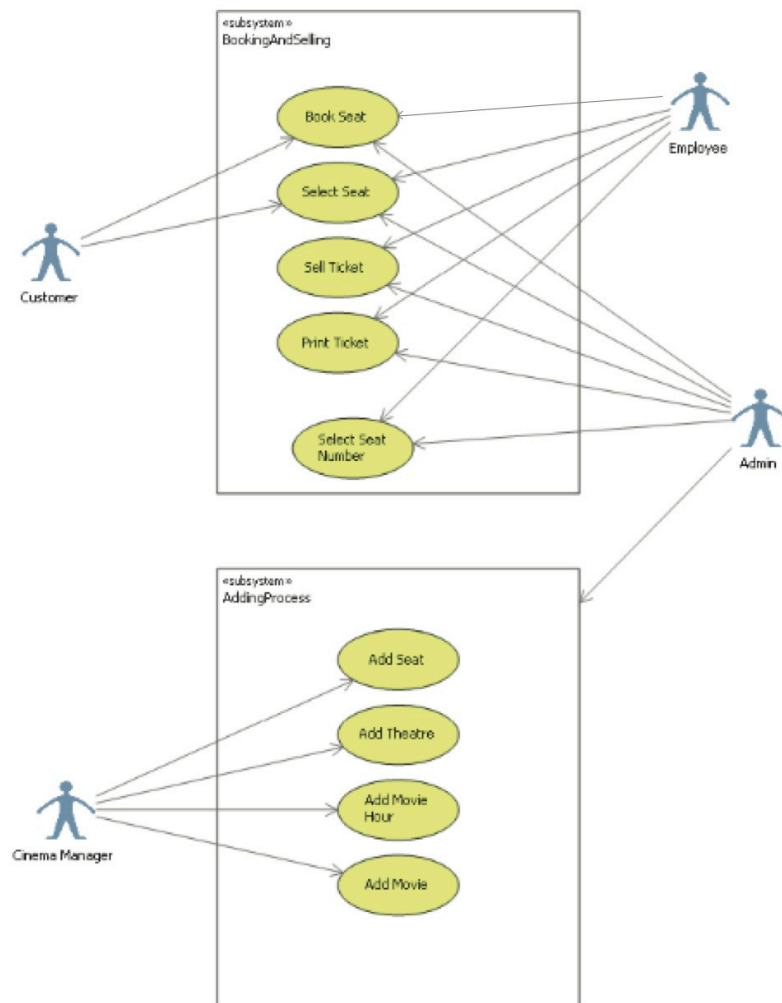


Figure 3.1 : Use Case Diagram

4. USER INTERFACES

4.1 FIRST SCREEN

On the first screen, user must select his statue to login :



Figure 4.1 : First Page of Cinemasoft Program

In this part, user can choose his statue, quit program or take information about Cinemasoft software.

4.2 ADMINISTRATOR FORM

If the user choose his statue as an administrator, admin login page will appear on the screen :

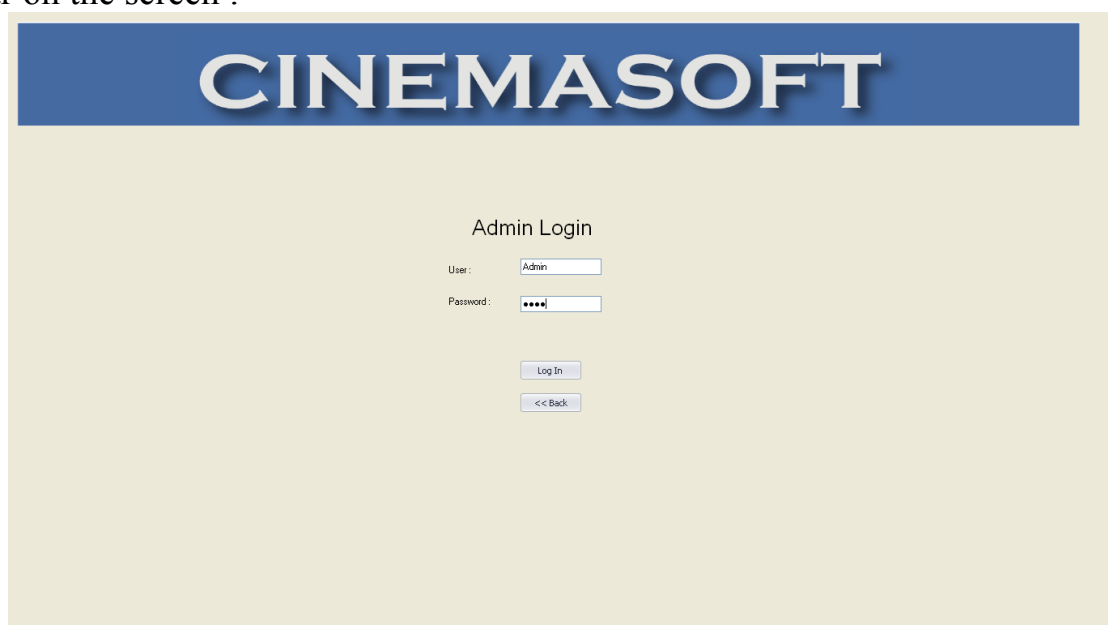


Figure 4.2 : Administrator Login Screen of Cinemasoft Program

After the login process, admin can see his process list :

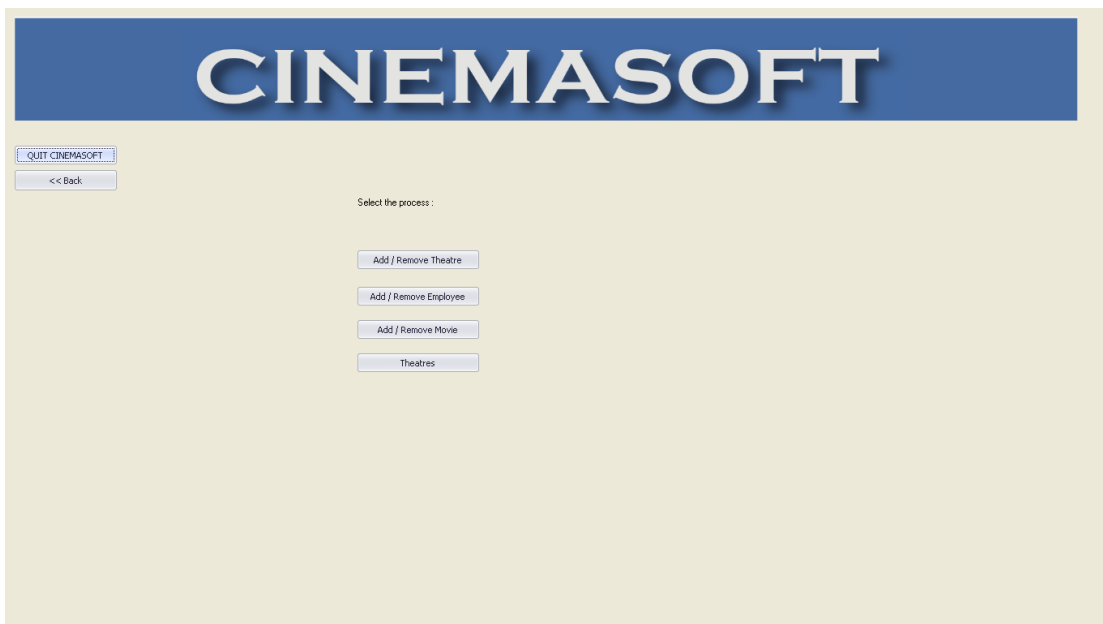


Figure 4.3 : Administrator Process Form of Cinemasoft Program

4.3 CINEMA MANAGER FORM :

When the cinema manager log in the system, he will see his process list like the administrator :

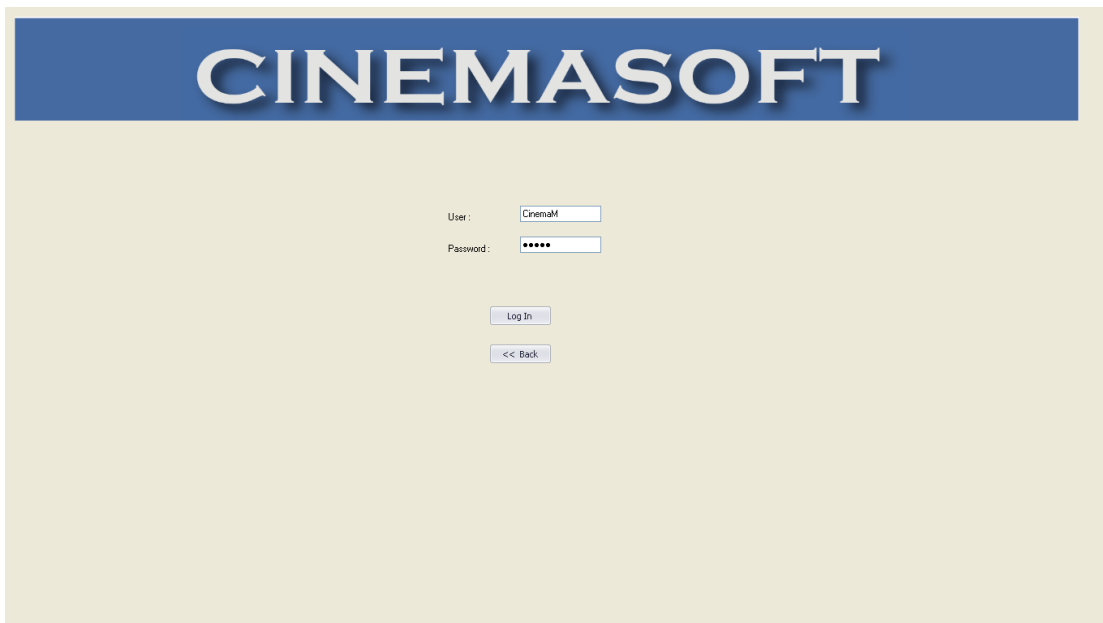
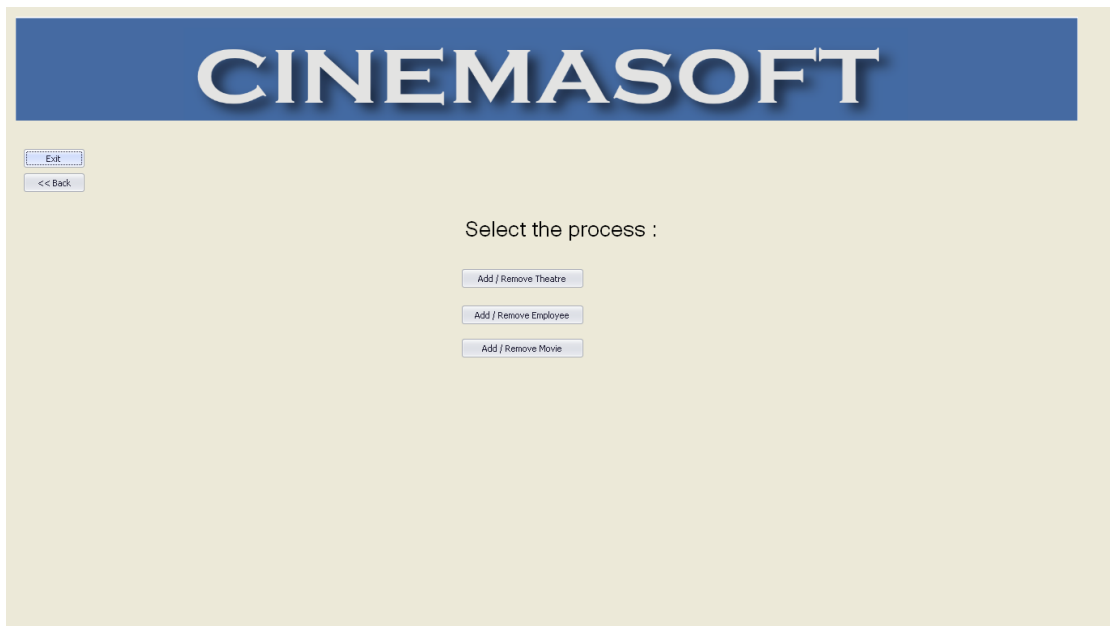


Figure 4.4 : Cinema Manager Login Screen of Cinemasoft Program



The screenshot shows the 'CINEMASOFT' application window. At the top is a blue header with the title in white. Below the header, on the left, are two buttons: 'Exit' and '<< Back'. In the center, the text 'Select the process :' is displayed. Below this text are three buttons stacked vertically: 'Add / Remove Theatre', 'Add / Remove Employee', and 'Add / Remove Movie'.

Figure 4.5 : Cinema Manager Process Form of Cinemasoft Program

4.4 EMPLOYEE FORM :



The screenshot shows the 'CINEMASOFT' application window. At the top is a blue header with the title in white. Below the header, in the center, is the text 'Employee Login'. Underneath this text are two input fields: 'User:' with the text 'Employee' and 'Password:' with masked characters '*****'. Below the input fields are two buttons: 'Log In' and '<< Back'.

Figure 4.6 : Employee Login Screen of Cinemasoft Program

When an employee logged in with his or her name, ticket selling and reservation form will be appear on the screen only, an employee can not see other add/remove process forms.

4.5 ADD/REMOVE THEATRE FORM :

In this screen, administrator or cinema manager can see all theatre list and their properties, he can add new one, or remove one of them from the list.

Id	NumberOfSeats	ThreeDMax
1	100	<input checked="" type="checkbox"/>
2	50	<input checked="" type="checkbox"/>
3	100	<input checked="" type="checkbox"/>
4	150	<input checked="" type="checkbox"/>
5	200	<input checked="" type="checkbox"/>
6	220	<input checked="" type="checkbox"/>
7	450	<input type="checkbox"/>

Figure 4.7 : Add/Remove Theatre Screen of Cinemasoft Program

4.6 ADD/REMOVE EMPLOYEE FORM :

In this screen, administrator or cinema manager can see all employee names on a list, he can add new one, or remove one of them from the list.

Id	Name	TelNo	Address
1	Neiman San	03124523456	Yenimahalle Ank...
2	Fatih Mavi	05327885940	Eryaman Ankara
3	Selin Yeşil	05436978056	Bağkent Ankara
*			

Figure 4.8 : Add/Remove Employee Screen of Cinemasoft Program

4.7 ADD/REMOVE MOVIE FORM :

In this screen, administrator or cinema manager can see all movie names on the list and their properties, he can add new one, or remove one of them from the list.

Id	Name	Type	Description	Duration
2	One Day	Romantic	Description ...	130
3	Soldier	Adventure	A tale of a soldier...	98

Figure 4.9 : Add/Remove Movie Screen of Cinemasoft Program

4.8 RESERVATION AND SELLING FORM :

In this screen, administrator or employee can see all theatres on a list, he or she can choose one of them and complete booking or reservation process.

Theatre 4

Clear

Exit

Movie: One Day

Movie Hours:

- 11:00
- 13:40
- 15:40
- 17:40
- 20:00
- 22:00

SCENE

Figure 4.10 : Reservation/Selling (Theatre) Screen of Cinemasoft Program

5. SAMPLE CODES FROM PROGRAM

5.1 ADDING A NEW EMPLOYEE PROCESS

While adding an employee on employee list we need a connection to connect our database. For doing this, we created a database connection on SQL Server which its name is 'sqlConn', and write our query then.

```
private void simpleButton1_Click(object sender, EventArgs e)
{
    SqlConnection sqlConn = new SqlConnection(Properties.Settings.Default.ConnectionString);
    sqlConn.Open();
    String thisQuery = "INSERT INTO Employee (" + " Name, TelNo, Address" + ") VALUES ('" +
        @textBox1.Text.ToString() + "', '" + @textBox2.Text.ToString() + "', '" + @textBox3.Text.ToString() + "')";
    SqlCommand thisCommand = new SqlCommand(thisQuery, sqlConn);
    thisCommand.ExecuteNonQuery();
    sqlConn.Close();

    SqlConnection con = new SqlConnection(Properties.Settings.Default.ConnectionString);
    string showTable = "Select Id, Name, TelNo, Address from Employee";
    SqlDataAdapter adap = new SqlDataAdapter(showTable, con);
    DataSet EmployeeTable = new DataSet();
    con.Open();
    adap.Fill(EmployeeTable);
    con.Close();
    dataGridView1.DataSource = EmployeeTable.Tables[0];
}
```

5.2 REMOVING AN EMPLOYEE PROCESS

```
private void simpleButton2_Click(object sender, EventArgs e)
{
    SqlConnection sqlConn = new SqlConnection(Properties.Settings.Default.ConnectionString);
    sqlConn.Open();
    String thisQuery = "DELETE FROM Employee WHERE Name='" + @textBox1.Text.ToString() + "' AND TelNo='" + @textBox2.Text.ToString() +
        "' AND Address='" + @textBox3.Text.ToString() + "'";
    SqlCommand thisCommand = new SqlCommand(thisQuery, sqlConn);
    thisCommand.ExecuteNonQuery();
    sqlConn.Close();

    SqlConnection con = new SqlConnection(Properties.Settings.Default.ConnectionString);
    string showTable = "Select Id, Name, TelNo, Address from Employee";
    SqlDataAdapter adap = new SqlDataAdapter(showTable, con);
    DataSet EmployeeTable = new DataSet();
    con.Open();
    adap.Fill(EmployeeTable);
    con.Close();
    dataGridView1.DataSource = EmployeeTable.Tables[0];
}
```

6. TESTING

In this part, we tested our program by getting help from different people and prepared test reports.

Reference No : 001	Date : 05.05.2012
Name : Demet Çakmak	
Scenario :	Cinemasoft program was started. On the first page, Cinema Manager was choosen as a user. On the login screen, there was no textbox cursor automatically on the login name textbox area. Mrs.Çakmak tried to write something while using keyboard but can not do this.
Outcome :	On the cinema manager login page, cursor must be located on the first textbox (user name textbox) automatically.

Figure 6.1 : Test Report – 1

Reference No : 002	Date : 04.05.2012
Name : Tolga Pusatlı	
Scenario :	Cinemasoft program was started. Administrator user type was selected. When Mr.Pusatlı entered "admin" as a user name on the first textbox, program did not accept this syntax, but it did not give an error. Then he tried "ADMIN" and it gived the same result. But after then, he tried "Admin", program accepted this syntax and continue the process.
Outcome :	On the admin form, program codes at background must be looked over again. Admin user name textbox was case-sensivite.

Figure 6.2 : Test Report - 2

7. FUTURE WORKS

7.1 THREE-TIER MODELING

In future, user and business requirements and expectations may change. There will be different studies on this project. For example, this project can be used on the web. We can transform this project, while keeping our database. Also we can transport this project to mobile media. While doing this transport process, our database will not be changed. This is known as *three-tier modeling*. Three-tier architecture is a linear organizational structure that is the mode for all communication.

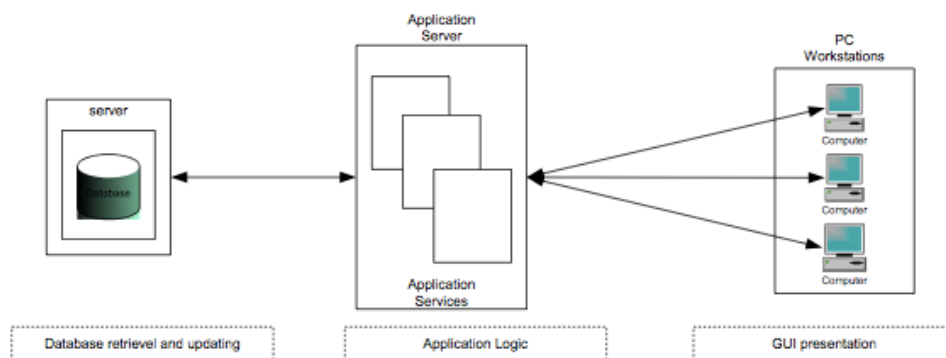


Figure 7.1 : Three-Tier Architecture Model [3]

Furthermore user requests can be changed in future; for example customers may want online buying process over the internet.

7.2 PERFORMANCE AUGMENTATION PROCESS

Another and most important future work can be 'performance augmentation process'. If our project grows, increasing performance process requirement may be emerged (concurrency control, caching etc.)

7.3 PRINTING TICKET

And finally, a printing system can be integrated to print cinema tickets as I said under limitations header. (Section 1.2)

8. CONCLUSION

Nowadays, traditional reservation ways of cinema ticketing is dying. It's new age where technology dominates human life. With the software and technological devices, exceptions are reduced and even terminated. Also, people prefer easy, quick and safe way for every part of his life.

This project is designed to meet the requirements of a cinema booking and selling system. It has been developed in C# and the database has been built in SQL Server 2008 keeping in mind the specifications of the system.

In our project: with this cinema ticketing system; cinema companies can satisfy comfortable facilities to their customers. The relationship between cinema manager, employee, and customer satisfy a good communication to complete ticketing process.

With this platform we developed, we are hoping to reduce time wasting, avoid misunderstandings, provide easy data flow, customer pleasure, and less hard work. We believe that we have accomplished our goals and satisfied with the code we developed.

9. REFERENCES:

- [1] *Database Systems – Models, Languages Design and Application Programming*, Ramez Elmasri, Shamkant B.Navathe, Sixth Edition, 2011, Pearson
- [2] *System Analysis and Design*, Kenneth E.Kendall, Julie E.Kendall, Eighth Edition, 2010, Pearson
- [3] *3-Tier Architecture*, Channu Kambalyal, 2005

10. BIBLIOGRAPHY

- [1] *Database Systems - Concepts Languages and Architectures*, Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Riccardo Torlone, 1999, Mc Graw Hill
- [2] *Object Oriented Analysis and Design*, John Deacon, 2005, Pearson