Online Movie Booking System

Group MLB_05.2_02

Table of Contents

Hypothetical scenario

Today, people need to perform their activities at a high speed by their own. In an environment like that, booking movies when they arrive at the venue can be frustrating. And also, people might need to know the details of the movies before they book a movie ticket. Because of above mentioned reasons, online movie booking systems are used to provide reliable, efficient and user-friendly environment to both users and theater staff.

When users enter to the system, they can view details about both now showing and upcoming movies. Movie show times and ticket prices are also displayed with movie details. In order to book a movie, users should select a movie first. Then they have to enter the no of tickets they need and select the desired seating positions. After that users are proceeded to payments section, which is displayed only for registered users. If the users are not registered in the system, they have to register in the system in order to confirm their payment methods. The system allows both cash and card payments done by the customers. The cash payments should be done in the preferred movie theater.

Requirement analysis

Purpose

The purpose of this document is to show the requirements needed to develop the system. This document includes the functional and non-functional requirements necessary to build the system.

Scope and overview

The required system is an online movie booking system. The main objective of this system is to make the movie ticket booking process easier and more efficient. This system will allow users (customers) to search for present and upcoming movies, watch movie trailers, view movie details, and book tickets online. Furthermore, it will provide the capability of updating and deleting theater details, updating seat details, updating and deleting movie details for the administrators. Apart from these functions, the system will also provide the user the capability of registering and logging in to the system.

Functional and Non-functional requirements

Functional requirements

These requirements are the features that are expected from the system, the features that the customer or the client wants.

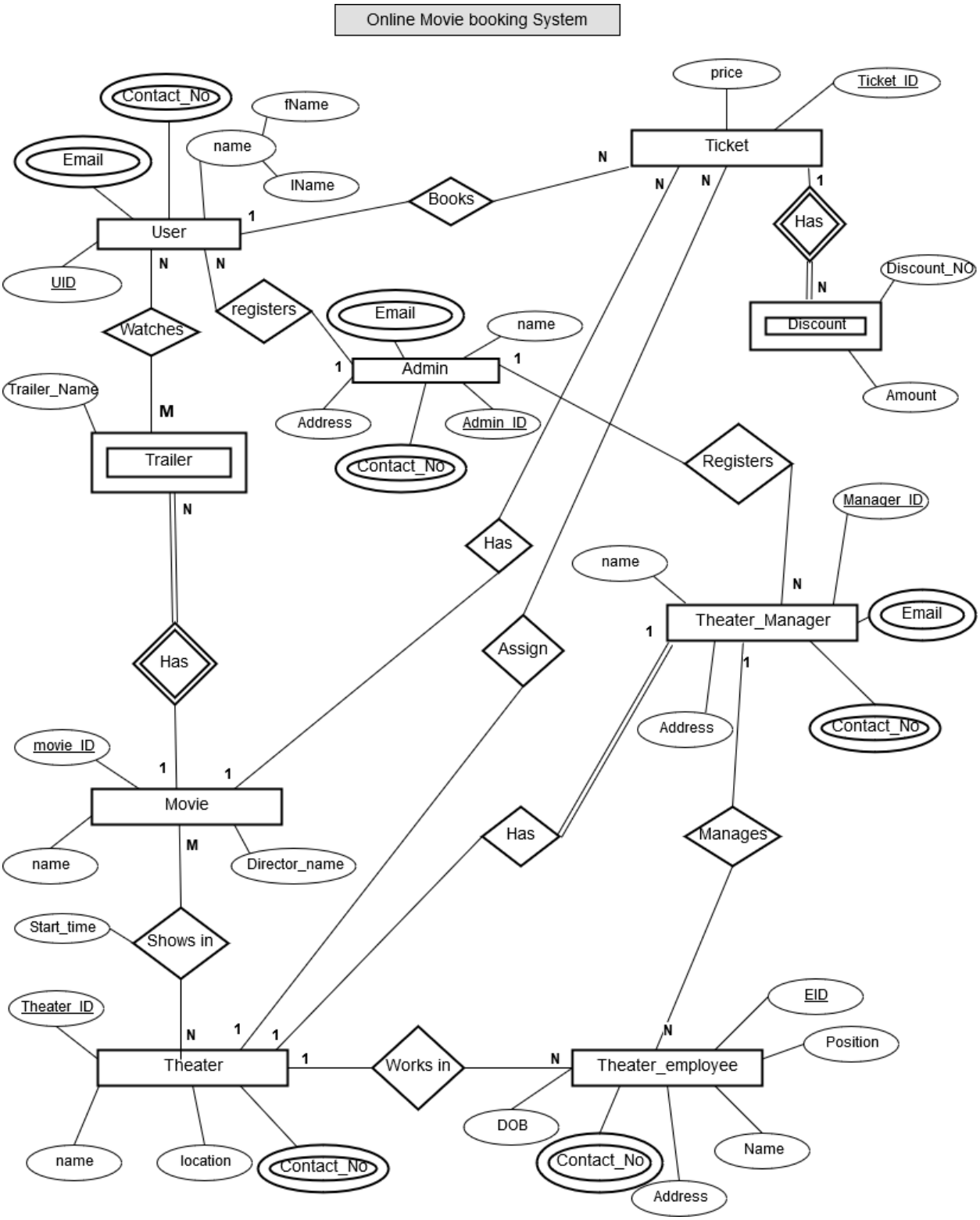
- Search for movies
    - When the customer searches for a movie, the system should display the relevant movie, if it is available in theaters or if it is an upcoming movie.
- View movie details
    - When the customer selects this option, the system should display the details of the movie, the show times of the nearest movie theaters and the ticket prices.
- Watch movie trailer
    - When the customer selects this option, the system should play the movie trailer.
- Book tickets online
    - By selecting this option, the customer can pay and book movie tickets for the required show time from any movie theater.
- Updating and deleting theater details and movie details
    - The administrators should be able to update, delete the theater details and the movie details.
- Updating seat details
    - The system should update the seat availability after every booking made and display the available seats.

Non-functional requirements

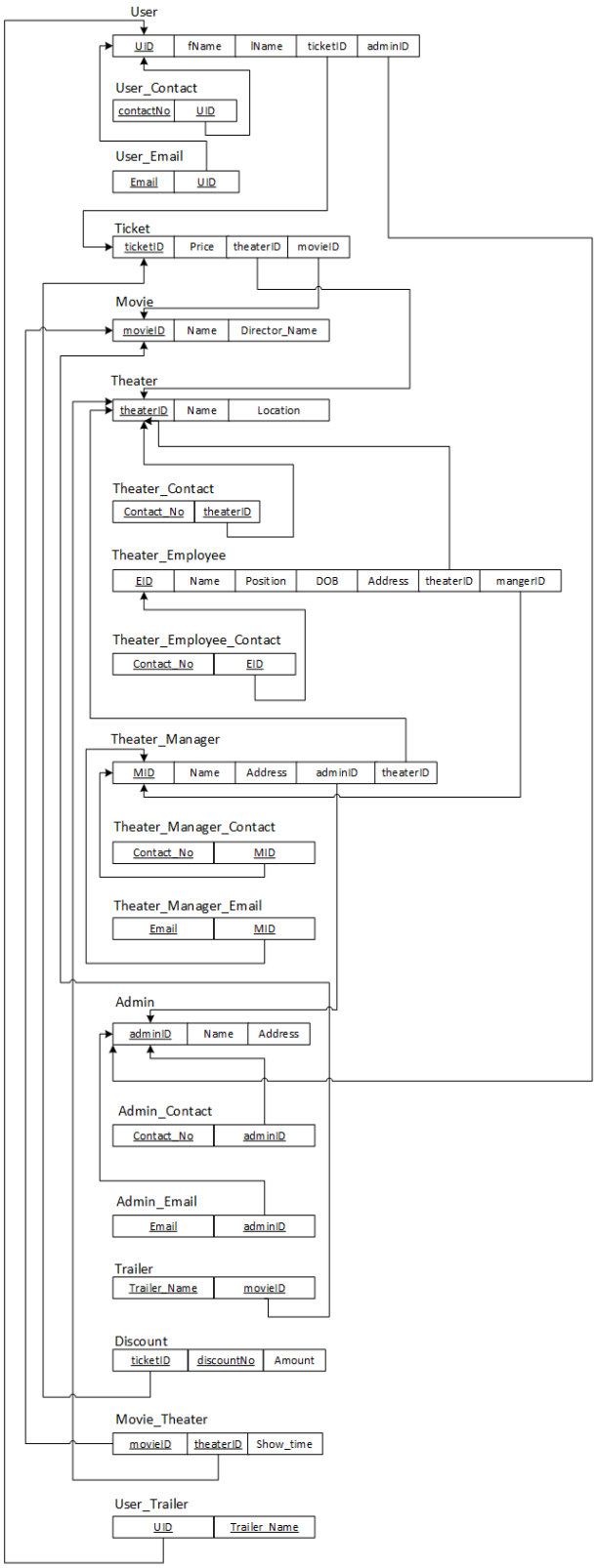
These requirements are the features that are used to judge the functionality of the system. Although, the clients do not directly request these features, they are necessary for a system.

- User-friendliness
    - The system should be easy to use. The user should be able to navigate through the system easily. The given details or instructions should be easy to understand.
- Speed
    - The system should be able to respond to the user requests very quickly. It should be free from lagging.
- Security
    - Only legal users should be able to access the system. The system is only accessible to those who have a username and password.
- Quality
    - The system should be very clear and good in quality.

ER Diagram


Online Movie booking System
Contact_No
fName
Email
name
lName
User
UID
Books
price
Ticket_ID
Ticket
Has
Discount_NO
Discount
Amount
Watches
registers
Email
name
Admin
Address
Admin_ID
Contact_No
Trailer_Name
Trailer
Registers
Manager_ID
Has
name
Theater_Manager
Email
Assign
Address
Contact_No
movie_ID
Movie
name
Director_name
Has
Manages
Start_time
Shows in
Theater_ID
Theater
Works in
Theater_employee
EID
Position
DOB
Contact_No
Name
Address
name
location
Contact_No

Relational Schema



SQL Commands

```
--Creating admin table (there are no foreign keys in the admin table)

Create table Admin(

Admin_ID char(5) NOT NULL,

Name varchar(30) NOT NULL,

Address varchar(30) NOT NULL,

constraint CHK_Admin check(Admin_ID like'[A-Z][A-Z][0-9][0-9][0-9]'),

constraint PK_Admin primary key (Admin_ID))

--primary key of the Admin table is Admin_ID

--Admin_ID consist with 2 Letters and 3 integers ex:"AD345"

--Admin_ID is refferenced by Admin_contact and Theater_MAnager tables

--Inserting Details to Admin Details

insert into Admin(Admin_ID,Name,Address) values ('AB001','Sachinthana
Bandara','New Kandy Rd Kaduwela')

insert into Admin(Admin_ID,Name,Address) values ('AC201','Kamal Bandara','New
Kandy Rd Delgoda')

insert into Admin(Admin_ID,Name,Address) values ('BD001','Dakshitha
Somasiri','New Kandy Rd Rathupaswala')

insert into Admin(Admin_ID,Name,Address) values ('EB001','Gayan
Laksiri','Colombo Street Kollupitiya')

insert into Admin(Admin_ID,Name,Address) values ('ED001','Sachin
thanushka','Colombo Street Maradana')

insert into Admin(Admin_ID,Name,Address) values ('EI301','Damith
Dissanayake','Peradeniya Kandy')

insert into Admin(Admin_ID,Name,Address) values ('DI761','Charith
Asalanka','Kadugannawa  Kandy')

insert into Admin(Admin_ID,Name,Address) values ('EF301','Hasitha
Ekanayake','New Kandy Rd Padiwaththa')

insert into Admin(Admin_ID,Name,Address) values ('AI401','Sachintha
Bandara','Kurunegala Rd Mawathagama')
```

```
insert into Admin(Admin_ID,Name,Address) values ('EF601','Danuri
Dissanayake','New Kandy Rd Baththaramulla')

--Drop table Admin

 --Delete Admin where Admin_ID =

--select *

--from Admin


--Creating Movie Table(there are no foreign keys in the table )

Create table Movie(

Movie_ID char(4),

Name  varchar(30),

Director_Name varchar(30),

constraint CHK_Movie check(Movie_ID like '[A-Z][0-9][0-9][0-9]'),

constraint PK_Movie primary key (Movie_ID))

--Primary key is the Movie_ID and it can start with 1 letter and 3 integer
numbers ex:"E301"


--Inserting details to the Movie table

insert into Movie values('A301','Chakde Srilanka','Ekanayake EK')      A301
B301 C301 D301 E301 F301 I301 J301 K301

insert into Movie values('B301','Chakde India','Rahman Ali')

insert into Movie values('C301','Raksha Bawana','Row Danny')

insert into Movie values('D301','Nim Thera','Ekanayake JK')

insert into Movie values('E301','How To Train Your Dragon','Rowly LS ')

insert into Movie values('F301','Love is Blind','Ramy Tom ')

insert into Movie values('G301','Jai Ho','ER Rahmath')

insert into Movie values('H301','Theri Meri Kahani','Vijay Prakash')

insert into Movie values('I301','Dictator','Jhonny Wilkinson')

insert into Movie values('J301','Sayuri','Ekanayake EK')
```

```
insert into Movie values('K301','Godzila','Joe Denly')


--drop table Movie

--delete from Movie where Movie_ID =

--    select *

--from Movie

--creating Admin_contact table

create table Admin_Contact(

ContactNo char(10),

Admin_ID char(5),

CONSTRAINT chk_phone CHECK (ContactNo not like '%[^0-9]%'),

constraint PK_Admin_Contact primary key(ContactNo,Admin_ID),

constraint FK_Admin_Contact foreign key (Admin_ID) references
Admin(Admin_ID));

--phone number is a 10 digit number

--foreign key is the Admin id which is referenced by Admin Table

--primary key is made out of Contact No and the Admin_ID

--Drop table Admin_contact

--inserting data into the table

insert into Admin_Contact values('0812422882','AC201')

insert into Admin_Contact values('0712422893','AC201')

insert into Admin_Contact values('0712423892','AB001')

insert into Admin_Contact values('0712428892','AB001')

insert into Admin_Contact values('0772422892','EB001')

insert into Admin_Contact values('0762322892','EB001')

insert into Admin_Contact values('0813442892','ED001')

insert into Admin_Contact values('0722425892','ED001')
```

```
insert into Admin_Contact values('0112422892','DI761')
insert into Admin_Contact values('0338422892','DI761')
--SELECT * FROM Admin_Contact
--delete from Admin_Contact
--WHERE Admin_ID =

--Creating Admin_Email table

create table Admin_Email (
Admin_ID char(5),
Email varchar(30),
constraint CHK_Admin_Email check(Email like '%_@_%'),
constraint FK_Admin_ID foreign key (Admin_ID) references Admin(Admin_ID),
constraint PK_Admin_Email primary key(Admin_ID,Email));

--AdminID is refferenced by the ADmin table
--Email should consist @
--Drop table


insert into Admin_Email values ('AC201','kamal@gmail')
insert into Admin_Email values ('AC201','amal@gmail')
insert into Admin_Email values ('AB001','samal@gmail')
insert into Admin_Email values ('AB001','janilal@gmail')
insert into Admin_Email values ('EB001','kjamal23@gmail')
insert into Admin_Email values ('EB001','k98amal@gmail')
insert into Admin_Email values ('EB001','qwkamal@gmail')
insert into Admin_Email values ('DI761','okwamal@gmail')
insert into Admin_Email values ('DI761','kd9amal@gmail')
insert into Admin_Email values ('ED001','kam87al@gmail')
```

```
--select *
--from Admin_Email
--delete Admin_Email where Admin_ID =

--Creating the Trailer Table

create table Trailer(
Movie_ID char(4),
trailer_Name varchar(20),
constraint FK_Trailer foreign key(Movie_ID) references Movie(Movie_ID),
constraint PK_Trailer primary key(Trailer_Name,Movie_ID))
-- Movie_ID is referenced by the Movie table
--drop table Trailer

insert into Trailer values('A301','Funny scenes')
insert into Trailer values('B301','Fight scene')
insert into Trailer values('B301','Love scenes')
insert into Trailer values('C301','Commedy scenes')
insert into Trailer values('D301','Love scenes')
insert into Trailer values('E301','Commedy scenes')
insert into Trailer values('F301','Movie trailer')
insert into Trailer values('G301','Commedy scenes')
insert into Trailer values('H301','Songs')
insert into Trailer values('I301','Fight scene')
insert into Trailer values('J301','Love scenes')
insert into Trailer values('K301','Songs')

--select * from Trailer
```

```
--delete Trailer where Movie_ID ='K301'



--Creating Theatre table

create table Theatre (
Theatre_ID char(4),
Name varchar(20),
location varchar(20),
constraint CHK_Theatre check (Theatre_ID like'[A-Z][A-Z][A-Z][0-9]'),
constraint  PK_Theatre Primary key (Theatre_ID))
--Theatre_ID is consist with 3 letters and 1 number
-- location shows the gps location
--drop table Theatre

--Inserting data to Theatre table

Insert into Theatre values ('ABC1','Regal','#sjajkssa..sjasjh#')
Insert into Theatre values ('ABC2','Legal','#kjajkssa..sjasjh#')
Insert into Theatre values ('ABC3','Kegalle Cinema ','#Lksjajkssa..sjasjh#')
Insert into Theatre values ('ABC4','Lanka Cinema','#lskajsiasujaishaj')
Insert into Theatre values ('ABD2','Scope Cinema','#khdjajkssa..sjasjh#')
Insert into Theatre values ('ABH1','Newlanka Cinema','#sjsjkssa..sjh#')
Insert into Theatre values ('EBC1','Co-op Lanka Cinema','#osu=sjh#')
Insert into Theatre values ('EBC2','Colombo Cinema','#usyjajkssa..sjasjh#')
Insert into Theatre values ('EBC5','Serandib Cinema','#klsjajkssa..sjasjh#')
Insert into Theatre values ('ECC1','Hellow Cinema','#oa..sjasjh#')

--select * from Theatre
--delete Theatre where  Theatre_ID =
```

```
--creating the table movie theatre

create table Movie_Theatre(

Movie_ID char(4),

Theatre_ID char(4),

Show_Time time,

constraint FK_Movie_Theatre1 foreign key (Movie_ID) references
Movie(Movie_ID),

constraint FK_Movie_Theatre2 foreign key (Theatre_ID) references
Theatre(Theatre_ID),

constraint PK_MovieTheatre primary key (Movie_ID,Theatre_ID))

--time data type is used in show time

--Movie_ID is reffered by Movie table and Theatre_ID is reffered from Theatre
Table


--drop table Movie_Theatre

insert into Movie_Theatre values('A301', 'ABC1','08:23:00')

insert into Movie_Theatre values('A301', 'ABC2','10:23:00')

insert into Movie_Theatre values('B301', 'ABC1','09:25:00')

insert into Movie_Theatre values('C301', 'ABC3','16:50:00')

insert into Movie_Theatre values('E301', 'ECC1','12:15:00')

insert into Movie_Theatre values('F301', 'EBC5','15:55:00')

insert into Movie_Theatre values('D301', 'EBC5','10:35:00')

insert into Movie_Theatre values('A301', 'EBC2','11:25:00')

insert into Movie_Theatre values('A301', 'ABC4','14:45:00')

insert into Movie_Theatre values('A301', 'ABD2','09:25:00')


--SELECT * FROM Movie_Theatre
```

```
--DELETE Movie_Theatre WHERE Theatre_ID =

--last 00000 refers the date


--Creating the theatre Contacts table

create table Theatre_Contacts(

ContactNo char(10),

Theatre_ID char(4),

constraint CHK_Theatre_Contacts check (ContactNo not like '%[^ 0-9]% '),

constraint FK_Theatre_Contacts foreign key (Theatre_ID) references
Theatre(Theatre_ID),

constraint PK_Theatre_Contacts primary key (Theatre_ID,ContactNo ))

--Theatre ID is referenced from the table Theatre

--contact no is 10 digit number

--Primary key is consist with noth the ContactNo and the Theatre_ID where
Theatre contacts are multivalued attributes


--drop table Theatre_Contacts

 insert into Theatre_Contacts values ('0813422883','ABC1')

 insert into Theatre_Contacts values ('0112422883','ABC2')

 insert into Theatre_Contacts values ('0332422883','ABC3')

 insert into Theatre_Contacts values ('0712422883','ABC4')

 insert into Theatre_Contacts values ('0812522883','ABD2')

 insert into Theatre_Contacts values ('0112922883','ABH1')

 insert into Theatre_Contacts values ('0662422883','EBC1')

 insert into Theatre_Contacts values ('0112492783','EBC2')

 insert into Theatre_Contacts values ('0702422883','EBC5')

 insert into Theatre_Contacts values ('0772422883','EBC5')

 insert into Theatre_Contacts values ('0752422883','ECC1')
```

```
 --SELECT * FROM Theatre_Contacts

 --DELETE Theatre_Contacts WHERE Theatre_ID =

--Creating the Theatre_Manager table

create table Theatre_Manager(

MID char(3),

Name varchar(30),

Address varchar(40),

Theatre_ID char(4),

Admin_ID char(5),

constraint CHK_Theatre_MAnager check (MID like '[A-Z][0-9][0-9]'),

constraint FK_Theatre_Manager1 foreign key (Theatre_ID) references
Theatre(Theatre_ID),

constraint FK_Theatre_MAnager2 foreign key (Admin_ID) references
Admin(Admin_ID),

constraint PK_Theatre_Manager primary key(MID))

--primary key is MID consist withg 1 letter and 2 numbers

--Theatre_ID is refferenced by Theatre table and Admin_ID is referenced from
Admin table

--Primary key of the table is MID


--drop table Theatre_Manager;

--select * from Theatre_Manager


insert into Theatre_Manager values('A11','EKL Senadeera','NO :15 Malabe Rd
Kaduwela','ABC1','AB001')

insert into Theatre_Manager values('A12','MKL Gajaadeera','NO :16 Malabe Rd
Malabe','ECC1','AC201')

insert into Theatre_Manager values('B11','GHL Bandara','NO :12 Kandy Rd
Kadugannawa','ABC1','BD001')

insert into Theatre_Manager values('B21','TDKL Perera','NO :15 Kurunagala Rd
Mawathagama','EBC5','EB001')
```

```
insert into Theatre_Manager values('C11','TEKL Silva','NO :13 Kegalla
Polgahawela','EBC2','EI301')

insert into Theatre_Manager values('D11','GEKL Madumadawa','NO :87 Malabe Rd
Viaharamawatha','EBC1','AB001')

insert into Theatre_Manager values('E11','EML Kalpage','NO :44 Colombo
Kollupitiya','ABC4','EF301')

insert into Theatre_Manager values('F11','EL Almeda','NO :22 Mahaiyawa
Kandy','ABC3','AI401')

insert into Theatre_Manager values('G11','KJL Bandara','NO :17 Colombo
Kiribathgoda','ABC2','EF601')

insert into Theatre_Manager values('H11','SL Hasintha','NO :18 Malabe Rd
Kaduwela','ABC1','DI761')


 --SELECT * FROM Theatre_Manager

 --DELETE Theatre_Manager WHERE MID =


 --Creating Theatre Manager Email

 create table Theatre_Manager_Email(

 Email varchar(20),

 MID char(3),

constraint CHK_Theatre_Manager_Email check (Email like '%_@_%'),

constraint FK_Theatre_Manager_Email foreign key (MID) references
Theatre_Manager(MID),

constraint PK_Theatre_Manager_Email primary key (MID,Email));


--table is referenced by Theatre_Manager

--Primary key is a composite (Email + MID)


--Drop table Theatre_Manager_Email

insert into Theatre_Manager_Email values ('kamal@gmail.com','A11')

insert into Theatre_Manager_Email values ('SA3mal@gmail.com','A12')
```

```
insert into Theatre_Manager_Email values ('J89amal@gmail.com','A12')

insert into Theatre_Manager_Email values ('kal98@gmail.com','B11')

insert into Theatre_Manager_Email values ('SA90kl@gmail.com','B21')

insert into Theatre_Manager_Email values ('amal@gmail.com','C11')

insert into Theatre_Manager_Email values ('Sapu34amal@gmail.com','C11')

insert into Theatre_Manager_Email values ('amal@gmail.com','D11')

insert into Theatre_Manager_Email values ('jkamal@gmail.com','D11')

insert into Theatre_Manager_Email values ('malj98@gmail','H11')


--select* from Theatre_Manager_Email

--delete Theatre_Manager_Email where MID =


--creating Theatre_Manager_contacts table

Create Table Theatre_Manager_Contacts(

ContactNo char(10),

MID char(3),

constraint CHK_Theatre_Manager_Contacts check (ContactNo not like '%[^ 0-
9]%'),

constraint FK_Theatre_Manager_Contacts foreign key (MID) references
Theatre_Manager(MID),

constraint PK_Theatre_Manager_Contacts primary key (MID,ContactNo))

--number is a ten digit

--MID is referenced by the Theatre_Manager  table

--Primary key of the table is (MID + Theatre_MAnager)

--drop table Theatre_Manager_Contacts


Insert into Theatre_Manager_Contacts values ('0772345678','A11')

Insert into Theatre_Manager_Contacts values ('0812240678','A12')

Insert into Theatre_Manager_Contacts values ('0112445678','B11')
```

```
Insert into Theatre_Manager_Contacts values ('0352845678','B21')

Insert into Theatre_Manager_Contacts values ('0758375678','C11')

Insert into Theatre_Manager_Contacts values ('0662335778','D11')

Insert into Theatre_Manager_Contacts values ('0816345678','E11')

Insert into Theatre_Manager_Contacts values ('0729345678','F11')

Insert into Theatre_Manager_Contacts values ('0704345678','G11')

Insert into Theatre_Manager_Contacts values ('0775345678','H11')


--SELECT * FROM Theatre_Manager_Contacts

--delete Theatre_Manager_Contacts where MID =


--Creating theatre employee table

create table Theatre_Employee(

EID char(4),

Name varchar(20),

Position varchar(30),

Address varchar(50),

DOB date,

Theatre_ID char(4),

MID char(3),

constraint CHK_Theatre_Employee check (EID like '[A-Z][A-Z][0-9][0-9]'),

constraint FK_Theatre_Employee1 foreign key (Theatre_ID) references
Theatre(Theatre_ID),

constraint FK_Theatre_Employee2 foreign key (MID) references
Theatre_Manager(MID),

constraint PK_Theatre_Employee primary key (EID))


--EID consist with 2 letters and 3

--Theatre_ID and MID is referenced by Theatre and Theatre_Manager tables
respectively
```

```
--Primary key of the table is EID

--Drop table Theatre_Employee


 insert into Theatre_Employee values ('AB12','AK Lakshan','Officer','No 12
Colombo Street Kandy','1984/03/21','ABC1','A12')

 insert into Theatre_Employee values ('AC12','AK Sachin','Ticket Officer','No
13 Colombo Street Kandy','1985/02/11','ABC1','A11')

 insert into Theatre_Employee values ('AD12','TK Rajapaksha','Theatre
Officer','No 14 Colombo Street Kandy','1987/06/21','ABC1','A11')

 insert into Theatre_Employee values ('AE12','LM Malinga','Ticket
Officer','No 15 Colombo Street Kandy','1997/02/11','ABC1','A11')

 insert into Theatre_Employee values ('AB21','TJ Herath','Projector
Manager','No 16 Colombo Street Kandy','1977/02/11','ABC1','A11')

 insert into Theatre_Employee values ('AB22','TK Sakshan','Accountant','No 17
Colombo Street Kandy','1978/02/22','ABC1','A11')

 insert into Theatre_Employee values ('AB23','K Nakshan','Officer','No 18
Colombo Street Kandy','1977/02/21','ABC1','A11')


-- select * from Theatre_Employee

--delete Theatre_Employee where EID =


 --Creating the table


 create table Theatre_Employee_Contacts(

 ContactNo char(10),

 EID char(4),

 constraint CHK_ContactNo check (ContactNO not like '%[^0-9]'),

 constraint FK_Theatre_Employee_Contact foreign key (EID) references
Theatre_Employee(EID),

 constraint PK_Theatre_Employee_Contact primary key (EID,ContactNo))


 --drop table Theatre_Employee_Contacts

 --Primary key is the EID and the contactNo
```

```
--EID is referenced from the Employee table


Insert into Theatre_Employee_Contacts values('0716432655','AB12')
Insert into Theatre_Employee_Contacts values('0716232655','AB12')
Insert into Theatre_Employee_Contacts values('0112432655','AC12')
Insert into Theatre_Employee_Contacts values('0718432655','AD12')
Insert into Theatre_Employee_Contacts values('0716433655','AE12')
Insert into Theatre_Employee_Contacts values('0756432655','AB22')
Insert into Theatre_Employee_Contacts values('0706432655','AB22')
Insert into Theatre_Employee_Contacts values('0661643265','AB12')
Insert into Theatre_Employee_Contacts values('0816432655','AB23')

--Select * from Theatre_Employee_Contacts
--delete Theatre_Employee_Contacts where EID =

--creating tickets table

create table Ticket(
Ticket_ID char(4),
price float,
Theatre_ID char(4),
constraint CHK_Ticket check(Ticket_ID like '[a-z][0-9][0-9][0-9]'),
constraint FK_Ticket foreign key(Theatre_ID) references Theatre(Theatre_ID),
constraint PK_Ticket primary key(Ticket_ID))

--drop table Ticket
--user ID is consist with 1 letter and 3 numbers

 insert into Ticket values ('A301','250.00','ABC1')
 insert into Ticket values ('B301','350.00','ABC2')
 insert into Ticket values ('C301','550.00','ABC3')
 insert into Ticket values ('A201','750.00','ABC4')
```

```
  insert into Ticket values ('A401','650.00','ABD2')
  insert into Ticket values ('A991','350.00','ABC1')
  insert into Ticket values ('D301','250.00','ABC1')

  --DELETE Ticket whrere
  --select * from Ticket

  --creating Discount table

  create table Discount(
  Discount_No char(2),
  Ticket_ID char(4),
  Amount float,
  constraint CHK_Discount check (Discount_No like '[A-Z][0-9]'),
  Constraint FK_Discount foreign key (Ticket_ID) references
Ticket(Ticket_ID),
  constraint PK_Discount primary key (Discount_No,Ticket_ID))
  --Drop table Discount

  insert into Discount values ('A1','B301','50.00')
  insert into Discount values ('B1','A301','15.00')
  insert into Discount values ('C1','C301','70.00')
  insert into Discount values ('D1','B301','50.00')
  insert into Discount values ('E1','A201','60.00')
  insert into Discount values ('F1','B301','50.00')
  insert into Discount values ('G1','A401','50.00')
  insert into Discount values ('H1','B301','80.00')

  --SELECT * FROM Discount
  --DELETE Discount where Discount_No =
```

```
--creating user table

      create table User_(
      UID char(6),
      Lname varchar(20) not null,
      Fname varchar(30) not null,
      Ticket_ID char(4),
      Admin_ID char(5),
      constraint CHK_User_ check (UID like '[A-Z][A-Z][A-Z][0-9][0-9][0-9]'),
      constraint FK_User_1 foreign key (Ticket_ID) references
Ticket(Ticket_ID),
      constraint FK_User_2 foreign key (Admin_ID) references Admin(Admin_ID),
      constraint PK_User_ primary key (UID))


   --drop table User_
   --Ticket_ID and the Admin_ID refferenced by Ticket and Admin table
respectively
   --primary key of the table is UID


   Insert into User_ values ('JKL567','Sajitha','Premasiri','B301','AB001')
   Insert into User_ values ('JJL667','ajitha','Premadasa','A401','AB001')
   Insert into User_ values ('JLL567','Lasitha','Premakumara','B301','BD001')
   Insert into User_ values ('JML567','Kamal','Hemasiri','C301','AB001')
   Insert into User_ values ('JMK978','Amal','Siripala','D301','AB001')
   Insert into User_ values ('ABL867','Lakshan','Kumaradasa','B301','ED001')
   Insert into User_ values ('JLJ567','Sachintha','Lakshan','B301','AB001')
   Insert into User_ values ('JOL577','Sahasara','Kodagoda','A401','AB001')
   Insert into User_ values ('JPL867','Salitha','Bandara','B301','AB001')
```

```
  -- delete User_ where UID =


  --creating user contacts table

   create table User_Contact(
      ContactNo char(10),
      UID char(6),
      constraint CHK_User_Contact check (ContactNo like '[0-9][0-9][0-9][0-
9][0-9][0-9][0-9][0-9][0-9][0-9]'),
      constraint FK_User_Contact foreign key (UID) references User_(UID),
      constraint PK_User_Contact primary key (ContactNo,UID))


      --UID is reffered from User_ table
      --composite primary key is (contactNo ,UID)
      --Drop table User_Contact


      Insert into User_Contact values ('0716543234','JKL567')
      Insert into User_Contact values ('0666543234','JLL567')
      Insert into User_Contact values ('0779543234','JKL567')
      Insert into User_Contact values ('0706543234','JKL567')
      Insert into User_Contact values ('0718543234','JKL567')
      Insert into User_Contact values ('0786543234','JKL567')


      -- DELETE from User_Contact where UID =
      --creating user email table


      create table User_Email(
      Email varchar(20),
      UID char(6),
      constraint CHK_User_Email check (Email like '%_@_%'),
```

```
	constraint FK_User_Email foreign key (UID) references User_(UID),
	constraint PK_User_Email primary key (UID,Email))

	--Drop trailer
	--UID is referenced by the User_
	--composite primary key is
	Insert into User_Email values ('kamal@gmail.com','JKL567')
	Insert into User_Email values ('amal@gmail.com','JLL567')
	Insert into User_Email values ('samal@gmail.com','JLL567')
	Insert into User_Email values ('bimal@gmail.com','JKL567')
	Insert into User_Email values ('kds3amal@gmail.com','JKL567')
	Insert into User_Email values ('ka23mal@gmail.com','JKL567')
	Insert into User_Email values ('Lkamal@gmail.com','JKL567')
	Insert into User_Email values ('slkamal@gmail.com','JKL567')

	--delete from User_Email where =

	--creating User_Trailer table

	create table User_Trailer(
	UID char(6),
	Trailer_Name varchar(20),
	Movie_ID char(4),
	constraint FK_User_Trailer1 foreign key (UID) references User_(UID),
	constraint FK_User_Trailer2 foreign key (Movie_ID) references
Movie(Movie_ID),
	constraint PK_User_Trailer primary key (UID,Movie_ID,Trailer_Name))

	--drop table User_Trailer
	--UID and Movie_ID is referenced by theUser and  Movie tables
respectively
```

```
insert into User_Trailer values ('JKL567','Fight scene1','A301')
insert into User_Trailer values ('JKL567','Fight scene2','B301')
insert into User_Trailer values ('JJL667','Fight scene3','A301')
insert into User_Trailer values ('JJL667','Fight scene5','A301')
insert into User_Trailer values ('JJL667','Fight scene6','A301')
insert into User_Trailer values ('JMK978','Comedy scene1','B301')
insert into User_Trailer values ('JMK978','Love scene1','B301')
insert into User_Trailer values ('JMK978','Fight scene7','B301')

    delete from User_Trailer ;
```

Special Security Requirements

Since Online movie booking system is a transaction based system, few security requirements are used when creating the system.

- **Using a QR code to confirm the payment at the movie theater:** Customers who are choosing the payment method as cash will be given a QR code relevant to the payment made by them. This code can be scanned at the movie theater and can be used to confirm their booking details.

- **Using special login credentials for different stakeholders:** Special login credentials will be provided to system administrators and theater staff to avoid unauthorized users login to the system as system administrators.

- **Granting different access privileges to different user levels:** Different access privileges will be granted to different use levels. I.e. An unregistered user can't proceed to the payment section without having a proper user account.

- **Confidentiality:** Users' sensitive data such as credit card details will not be disclosed through the system for any reason.