AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY
DHAKA-1208, BANGLADESH.

Department of Computer Science and Engineering
Spring 2019

Program: Bachelor of Science in Computer Science and Engineering
Course No: CSE 4126
Course Title: Distributed Database System Lab

Final Project Submission

Date of Submission: October 14, 2019

Submitted to:
Mr. Faisal Muhammad Shah
Assistant Professor
Department of CSE, AUST

Tasfia Zahin
Lecturer,
Department of CSE, AUST.

Submitted By:
Syed Sanzam 16.01.04.042
Nusrat Hossain Anika 16.01.04.045
MD. Faisal Ahmed 16.01.04.046

Lab Group: A2

Movie Theater Management System

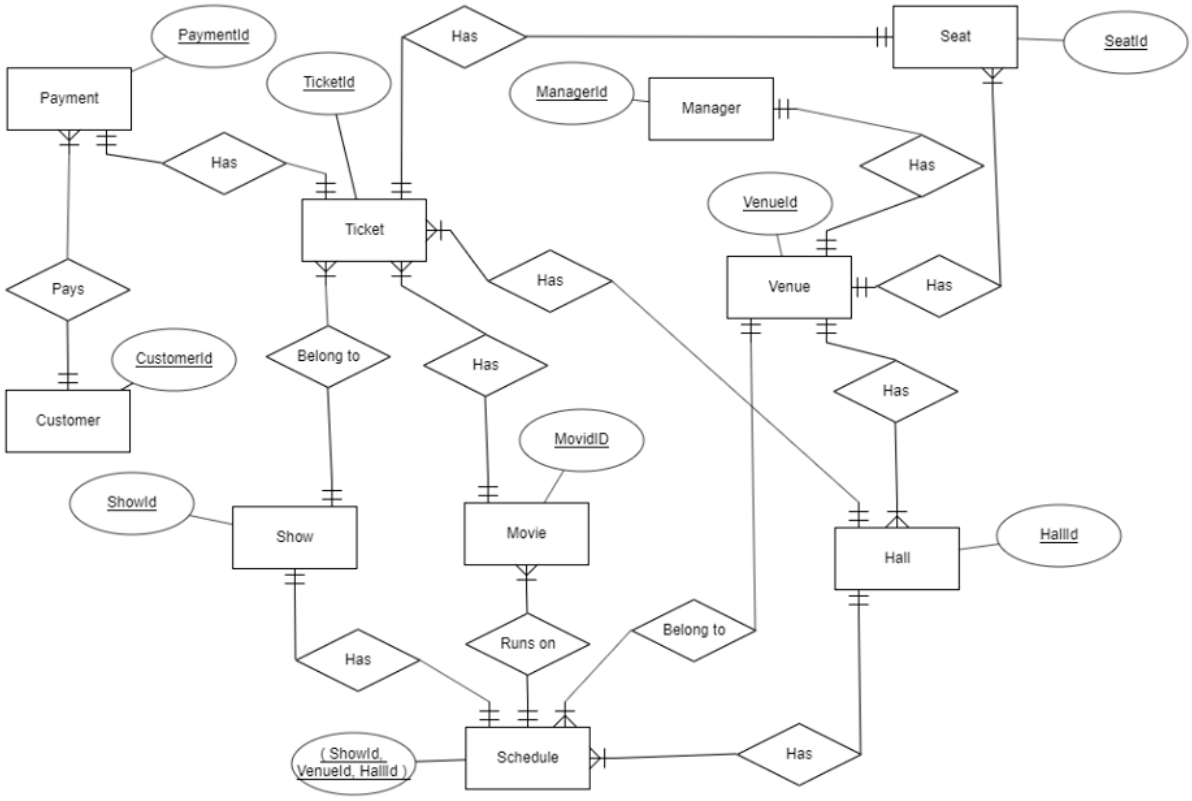
Introduction:

It is an automated theatre management system which is capable of handling most of the ticket-related and scheduling operations in a distributed manner. This system was designed to keep track of movies along with showtimes, available seats, entitled employees and last but not least, ticket sales. It provides a friendly interface for users to interact with. In this system, all the necessary information about a film is presented in a convenient and orderly fashion to make the experience as effortless as possible.

Motivation:

Films and Movies play a very important role in defining the cultural growth of an individual. Going to the cinemas has been the culture of almost all the people of today's generation. It is a time when family and friends can spend some time altogether by stealing some hours from their busy schedule. This project provides an application that allows the movie theatre manager and the viewers to handle all the cinema activities in a manner that is both efficient and convenient for all the people involved.

Entity-Relation Diagram:



Global Schema:

- MOVIES(**MOVIEID**, TITLE, CATEGORY,RELEASEDATE, LANGUAGE)
- CUSTOMER(**CUSTOMERID**, CUSTOMERNAME, CUSTOMERCONTACT)
- VENUE(**VENUEID**, VENUNAME, LOCATION)
- MANAGER(**MANAGERID**, V_ID, MANAGERNAME, MANAGERCONTACT)
- HALL(**HALLID**,V_ID, HALLNAME, CATEGORY, TOTALSEATS)
- SHOW(**SHOWID**, SHOWTIME)
- SCHEDULE(**V_ID**, **H_ID**, **S_ID**, M_ID, **SHOWDATE**)
- SEAT(**SEATID**, V_ID, POSITION, PRICE)
- TICKETS(**TICKETID**, V_ID, H_ID, S_ID, M_ID, ST_ID, M_DATE)
- PAYMENT(**PAYMENTID**, C_ID, T_ID, V_ID, AMOUNT, PAYMENTDATE)

Fragmentation Schema:

The fragmentation will mostly depend on the 'Location' attribute. For example,

Venue1 = SL Area = 'Panthapath'

Venue2 = SL Area = 'Bashundhara'

Venue3= SL Area = 'Banani'

Allocation Schema:

The data are divided into three locations, they are *Panthapath*, *Bashundhara* and *Banani*. For the sake of various technical reasons, we will assume Panthapath as Site 1, Bashundhara as Site 2 and Banani as Site 3. Therefore, all the instances of the relations are present in the three aforementioned sites. These three sites are connected over a local network and are able to perform both local and global applications.

Functions & Procedures Definition:

- Display_Hall: Shows the information of halls on the basis of input(location) taken in the main procedure.
- Display_Show: Shows the information of a show.
- Display_Schedule: Shows the information of schedule of a particular venue.
- Display_Seat: Shows the information of available seats.
- Display_Movies: Shows the list of the movie currently running on that venue.
- Display_Manager: Shows the information of the manager of that venue.
- Buy_Ticket: Book a ticket for specific show in a specific venue
- Movie_Specific_Information: It displays the venues, dates, halls, show times, schedules and available tickets for a specific movie.

Contribution:

16.01.04.042:

- Established the database connection between three sites.
- Developed functions and procedures for specific operations.
- Divided the code into various segments for design and readability purposes.

16.01.04.045:

- Created tables.
- Inserted data based on the primary key and foreign key references.
- Designed triggers for each operation.

16.01.04.046:

- Designed the Entity-Relationship Diagram of the system.
- Created a menu-like interface for user convenience.
- Added some queries that can best reflect the entire system