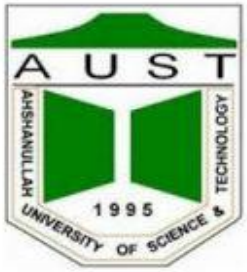
Ahsanullah University of Science & Technology

Department of Computer Science & Engineering



ANY TO-LET

CSE 3224

Information System Design

&

Software Engineering Lab

Submitted By:

Solayman Hossain Emon	16.01.04.091
Nahid Hasan	16.01.04.106
Md Toasin Habib	16.01.04.107

Date of Submission : **2 April, 2019**

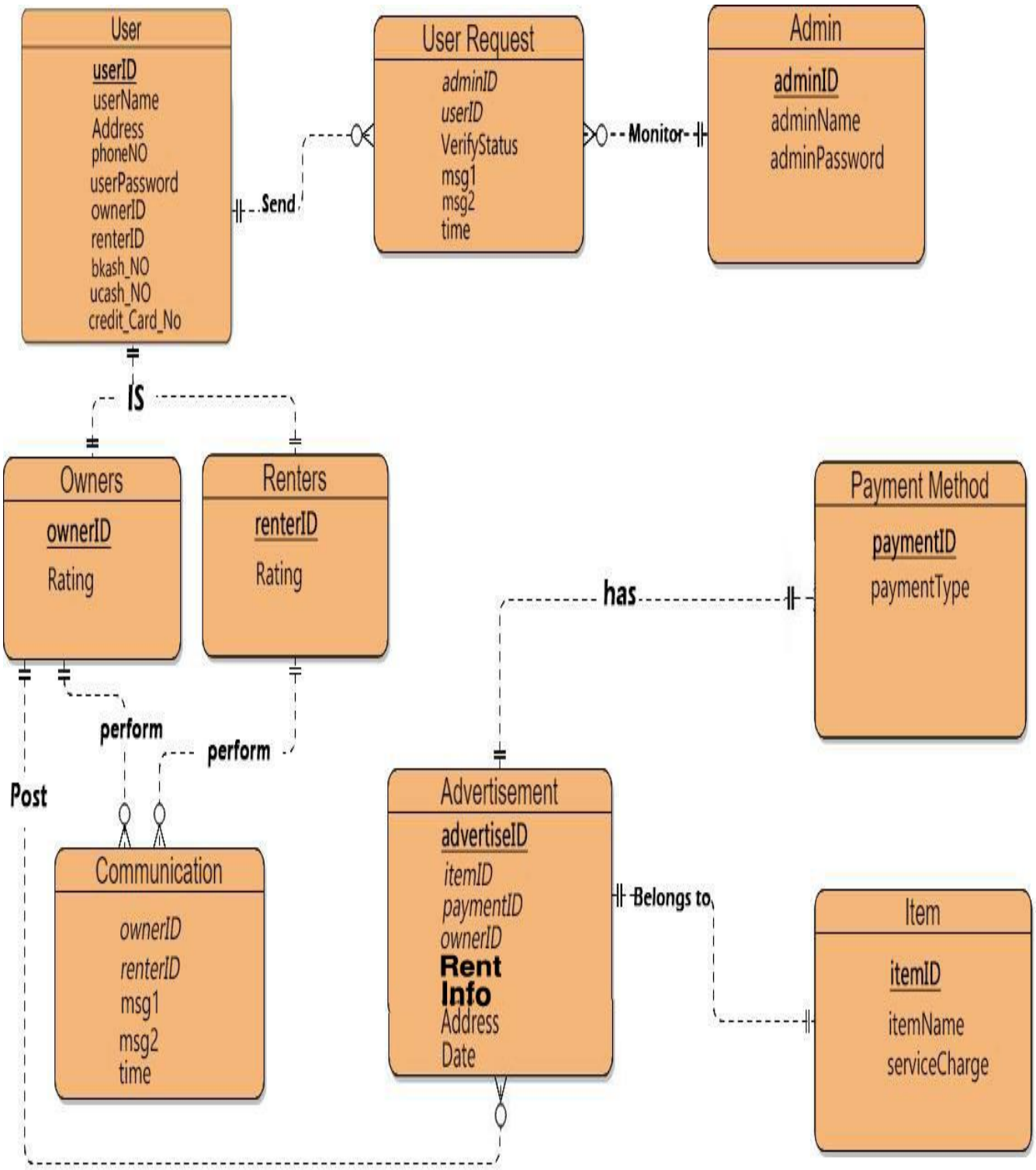
INTRODUCTION

To-let (available for renting) is a very common term to the general people. Dhaka City has a population of around 18 million. So there require a huge number of to-let advertisement on every day. By focusing that we are trying to develop an application. Through this application , the users can give any kind of to-let advertisement and the renter can see that advertisement in different categories. They can also search their desired rent product by filtering according to the type, area & rent price.

NAME OF ENTITIES WITH KEYS

Entity	Primary key	Foreign key	Composite key
User	userID	ownerID, renterID	
User Request		userID, adminID	
Admin	adminID		
Owners	ownerID		
Renters	renterID		
Communication		ownerID, renterID	
Advertisement	advertiseID	paymentID, itemID, ownerID	
Payment Method	paymentID		
Item	ItemID		

ENTITY RELATIONSHIP (ER) DIAGRAM



SQL COMMANDS

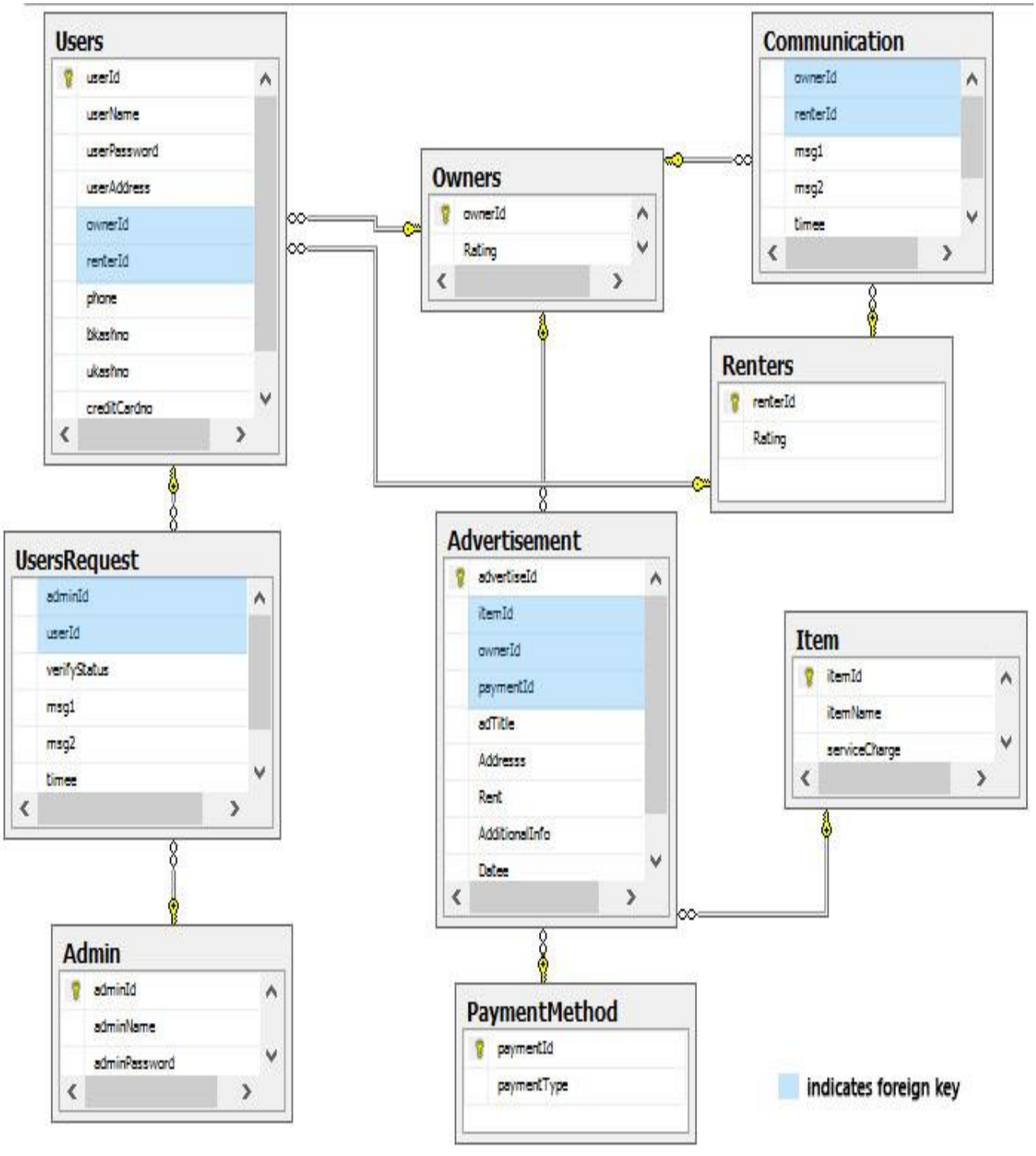
```
CREATE TABLE Owners
(
ownerId VARCHAR(500) NOT NULL ,
Rating VARCHAR(100) NOT NULL,
PRIMARY KEY (ownerId)
ON DELETE CASCADE
);
CREATE TABLE Renters
(
renterId VARCHAR(500) NOT NULL ,
Rating VARCHAR(100) NOT NULL,
PRIMARY KEY (renterId)
ON DELETE CASCADE
);
CREATE TABLE Item
(
itemId VARCHAR(500) NOT NULL ,
itemName VARCHAR(100) NOT NULL,
serviceCharge INT NOT NULL,
PRIMARY KEY (itemId)
);
```

```
CREATE TABLE PaymentMethod
(
paymentId VARCHAR(500) NOT NULL ,
paymentType VARCHAR(100) NOT NULL,
PRIMARY KEY (paymentId)
);
CREATE TABLE Admin
(
adminId VARCHAR(500) NOT NULL ,
adminName VARCHAR(100) NOT NULL,
adminPassword VARCHAR(100) NOT NULL,
PRIMARY KEY (adminId)
);
CREATE TABLE Users
(
userId VARCHAR(500) NOT NULL ,
userName VARCHAR(100) NOT NULL,
userPassword VARCHAR(100) NOT NULL,
userAddress VARCHAR(100) NOT NULL,
ownerId VARCHAR(500) NOT NULL ,
renterId VARCHAR(500) NOT NULL ,
phone VARCHAR(100) NOT NULL,
bkashno VARCHAR(100) ,
ukashno VARCHAR(100) ,
creditCardno VARCHAR(100) ,
PRIMARY KEY (userId),
FOREIGN KEY (ownerId) REFERENCES Owners (ownerId) ,
FOREIGN KEY (renterId) REFERENCES Renters (renterId)
);
```

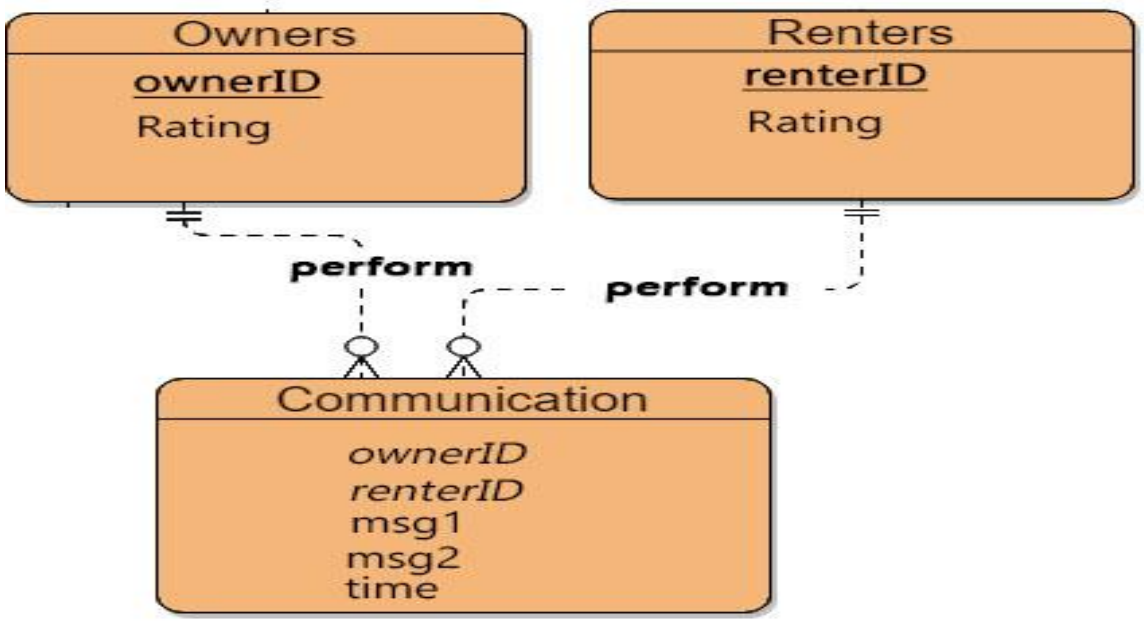
```
CREATE TABLE UsersRequest
(
adminId VARCHAR(500) NOT NULL ,
userId VARCHAR(500) NOT NULL ,
verifyStatus VARCHAR(100) NOT NULL,
msg1 VARCHAR(max) ,
msg2 VARCHAR(max),
timee DATETIME  NOT NULL,
FOREIGN KEY (adminId) REFERENCES Admin (adminId) ,
FOREIGN KEY (userId) REFERENCES Users (userId) ,

);

CREATE TABLE Communication
(
ownerId VARCHAR(500) NOT NULL ,
renterId VARCHAR(500) NOT NULL ,

msg1 VARCHAR(max) ,
msg2 VARCHAR(max),
timee DATETIME  NOT NULL,
FOREIGN KEY (ownerId) REFERENCES Owners (ownerId) ,
FOREIGN KEY (renterId) REFERENCES Renters (renterId) ,

);
```

```
CREATE TABLE Advertisement
(
advertiseId VARCHAR(500) NOT NULL ,
itemId VARCHAR(500) NOT NULL ,
ownerId VARCHAR(500) NOT NULL ,
paymentId VARCHAR(500) NOT NULL ,
adTitle VARCHAR(300) NOT NULL ,
Addresss VARCHAR(max) NOT NULL,
Rent INT NOT NULL,
AdditionalInfo VARCHAR(max) NOT NULL,
Datee DATETIME  NOT NULL,
FOREIGN KEY (itemId) REFERENCES Item (itemId) ,
FOREIGN KEY (ownerId) REFERENCES Owners (ownerId) ,
FOREIGN KEY (paymentId) REFERENCES PaymentMethod (paymentId) ,
PRIMARY KEY (advertiseId)
);
```

HIGHLIGHT PRIMARY & FOREIGN KEYS IN RELATIONAL MODEL



JUSTIFY THE MULTIPLICITIES



Here is the relation between product owners and Renters in ERD.

// Insert data into owners, renter & communication table

INSERT INTO Owners(ownerId, Rating)

VALUES ('nahid24owner', '5'), ('EmonQtowner', '5'),('nibirSpyowner', '5')

INSERT INTO Renters(renterId, Rating)

VALUES ('nahid24renter', '5'), ('EmonQtrenter', '5'),('nibirSpyrenter', '5')

INSERT INTO Communication(ownerId,renterId, msg1,msg2,timee)

VALUES ('nahid24owner','EmonQtrenter','hi i am the owner','', 30/3/2019),

('nahid24owner','EmonQtrenter','hi i am the owner','', 30/3/2019),

('nibirSpyowner','EmonQtrenter','hi i am the owner','', 30/3/2019),

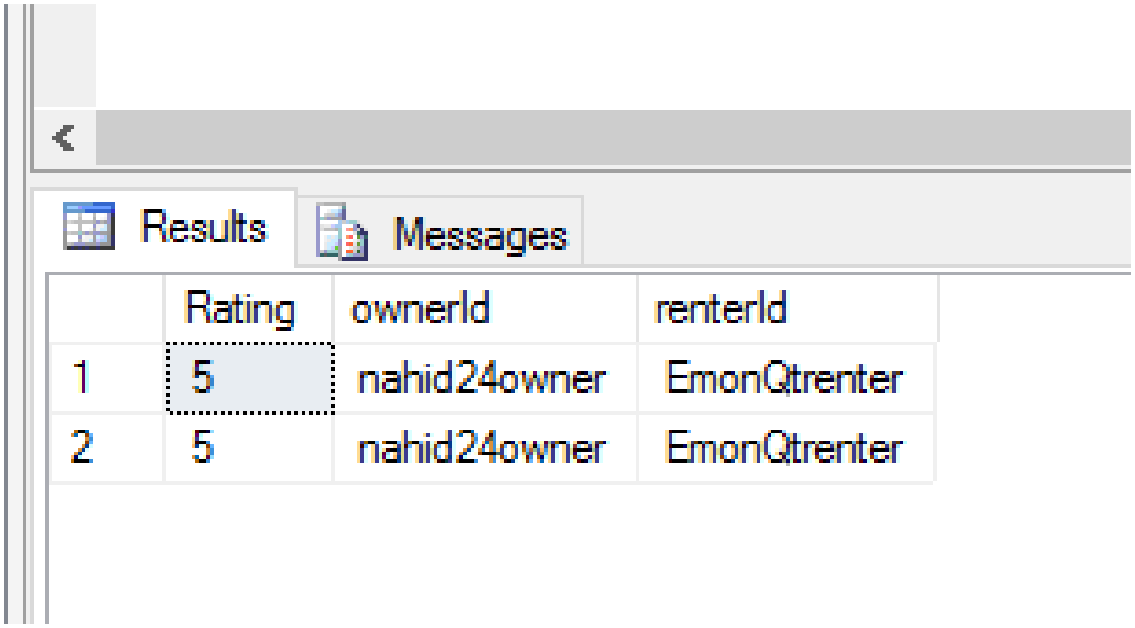
('nibirSpyowner','EmonQtrenter','hi i am the owner', '',30/3/2019)

// Perform a join operation between them

SELECT Owners.Rating,Owners.ownerId,Communication.renterId

FROM Owners

JOIN Communication ON Owners.ownerId=Communication.ownerId where Communication.ownerId='nahid24owner'



The displayed data proved that the multiplicities are correct.

CONCLUSION

In this report, At first we design a probable Entity Relationship Diagram of our project. Then observing the multiplicity and relation among them we create the sql command of their. After that we generate a relational model in Sql Server by running that queries . The generated model is so much closer to our ERD. The operation between the tables also perform well.