ASSUMPTION UNIVERSITY
Faculty of Science & Technology
Department of Information Technology

IT 3210 Database Management Systems

Apartment Management System

Submitted to	**:**	**A. Darun Kesrarat**	
Submitted by	**:**	Silvia Imaculada Soares	5538425
		Virgilia Maria Da Costa	5538427
		Su Myat Thu	5548106
		Sai Than Htike	5618394

Semester 2/2016

Description

Apartment management system is a system that provides to manage utility and day by day operations of apartment. The owner will control such as manage apartment information, monthly fee transaction, view tenant Information and check room availability. In current system, the owner had to use papers and files to manage the apartment. At the end of every month, they have to take out those papers and files in order to calculate the bill. It takes time to manage those things. Moreover, human errors will also cause in both minor or major mistakes. With this system, he can calculate and manage those things in a short period of time.

Scope

(1) Tenant

- Log In

  Tenant who is currently staying at the apartment has to log in with his/her username and password that was provided by the owner to see the information including monthly bill and contract detail.

- View Contract Information

  Tenant can see the detail of the contract including contract expired date and deposit amount.

- View Monthly Bill

  Tenant can check the total amount of the monthly bill including room fee, electricity fee, water fee and internet fee (if used).

(2) Owner

- Log In

  Owner must log in with his own username and password to access the privileges of the system.

- View Tenant Information

  Owner can see the information of tenant and the room number that he/she is staying.

- Manage Tenant Information

  Owner can add the information of the new tenant and his/her contract including the deposit amount. He can also update and delete that information when the tenant moved out.

- View Contract Information

  Owner can see the detail of the active and expired room contract along with who stay in that room.

- Manage Contract Information

  Owner can add or extend room contract information of the tenants.

- Control Monthly Bill

  Owner can fill up the amount for the monthly bill including room fee, electricity fee, water fill and internet fee.

- Receive Payment

  Owner can mark on the system as he received money for each rooms.

- Check Room Availability

  Owner can check how many rooms are available and occupied in his apartment.

Entity Relation Diagram

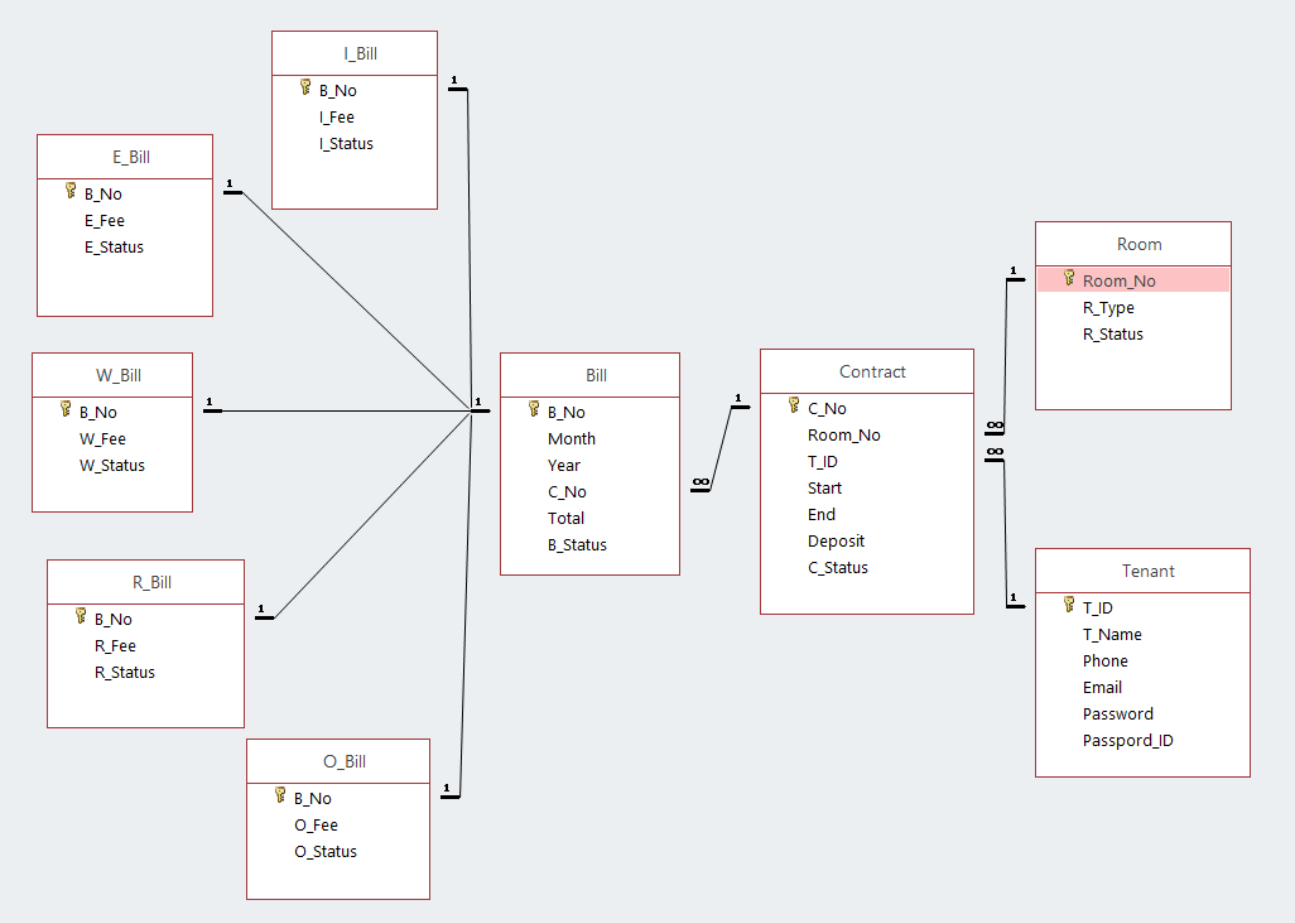


Figure 1: Entity Relationship Diagram

Data Dictionary

Tenant: Use for keeping the information of Tenants.				
Attribute	**Attribute Type**	**Field Size**	**Attribute Description**	**Sample**
T_ID	int	11	Auto Increment tenant's identification in the system.	3
T_Name	varchar	30	Tenant's name.	James Bond
Phone	varchar	15	Tenant's phone number.	0987564125
Email	varchar	30	Tenant's email address to login to the system.	bond007@gmail.com
Passport/ID	varchar	20	Tenant's passport number or national ID number.	US007007
Password	varchar	10	Tenant's password to access to the system.	bondjame

Figure 2: Data Dictionary – Tenant

Contract: Use for recording the detail of room contract.				
Attribute	**Attribute Type**	**Field Size**	**Attribute Description**	**Sample**
C_No	int	11	Auto Increment contract's number in the system.	3
Room_No	varchar	3	Room number and identification of the room.	203
T_ID	int	11	Tenant identification number.	3
Start	Date/Time		The date that start the contract.	01-Dec-16
End	Date/Time		The date that the contract will be expired.	01-Dec-17
Deposit	int	5	Deposit amount that tenant paid when the contract is made.	16500
C_Status	varchar	1	Active Contract = 1, Expired Contract = 0	1

Figure 3: Data Dictionary – Contract

Room: Use for tracking room status and room type.				
Attribute	**Attribute Type**	**Field Size**	**Attribute Description**	**Sample**
Room_No	varchar	3	Room number and identification of the room.	203
R_Type	varchar	1	Specifies type of the room. A = Superior B = Delux C = Execitive	A
R_Status	varchar	1	Available = 0 Occupied = 1	1

Figure 4: Data Dictionary – Room

Bill: Use for recording the amount of bill.				
Attribute	**Attribute Type**	**Field Size**	**Attribute Description**	**Sample**
B_No	int	11	Auto Increment bill's number in the system.	1
Month	Date/Time		Month of the bill.	April
Year	Date/Time		Year of the bill.	2017
C_No	int	11	Contract number of the bill.	3
Total	int	5	Total amount of the bill.	7570
B_Status	varchar	1	0 = Unpaid 1 = Paid 2 = Partial Paid	2

Figure 5: Data Dictionary – Bill

R_Bill: Use for recording the amount of room fee.				
Attribute	**Attribute Type**	**Field Size**	**Attribute Description**	**Sample**
B_No	int	11	Auto Increment bill's number in the system.	1
R_Fee	int	5	The amount of room fee.	5500
R_Status	varchar	1	0 = Unpaid 1 = Paid	1

Figure 6: Data Dictionary – R_Bill

E_Bill: Use for recording the amount of electricity fee.				
Attribute	**Attribute Type**	**Field Size**	**Attribute Description**	**Sample**
B_No	int	11	Auto Increment bill's number in the system.	1
E_Fee	int	5	The amount of electricity fee.	1350
E_Status	varchar	1	0 = Unpaid 1 = Paid	1

Figure 7: Data Dictionary – E_Bill

W_Bill: Use for recording the amount of water fee.				
Attribute	**Attribute Type**	**Field Size**	**Attribute Description**	**Sample**
B_No	int	11	Auto Increment bill's number in the system.	1
W_Fee	int	5	The amount of water fee.	130
W_Status	varchar	1	0 = Unpaid 1 = Paid	1

Figure 8: Data Dictionary – W_Bill

I_Bill: Use for recording the amount of internet fee.				
Attribute	**Attribute Type**	**Field Size**	**Attribute Description**	**Sample**
B_No	int	11	Auto Increment bill's number in the system.	1
I_Fee	int	5	The amount of internet fee.	590
I_Status	varchar	1	0 = Unpaid 1 = Paid	0

Figure 9: Data Dictionary – I_Bill

O_Bill: Use for recording the amount of other fee.				
Attribute	**Attribute Type**	**Field Size**	**Attribute Description**	**Sample**
B_No	int	11	Auto Increment bill's number in the system.	1
O_Fee	int	5	The amount of other fee. (Example – Late Payment fee)	0
O_Status	varchar	1	0 = Unpaid 1 = Paid	1

Figure 10: Data Dictionary – O_Bill

Interface Design and SQL Command

Tenant

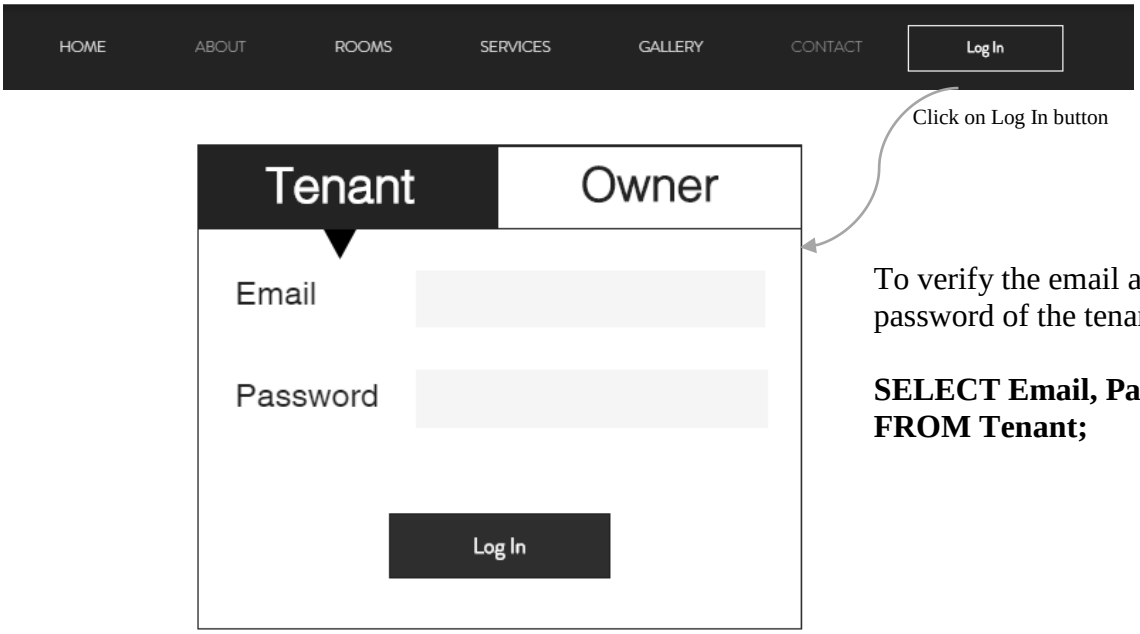


Figure 11: Tenant Login – Tenant can login to the system by filling his email address and password that owner provided.

Figure 12: Tenant's Profile – Tenant can view his personal information, room contract information and he can check monthly bill with status.

To get the information on the page, we use the following sql command to retrieve data from database.

SELECT * FROM ((Tenant INNER JOIN ((Room INNER JOIN (((Bill INNER JOIN E_Bill ON Bill.B_ID = E_Bill.B_ID) INNER JOIN R_Bill ON Bill.B_ID = R_Bill.B_ID) INNER JOIN W_Bill ON Bill.B_ID = W_Bill.B_ID) ON Room.Room_No = Bill.Room_No) INNER JOIN Contract ON Room.Room_No = Contract.Room_No) ON Tenant.T_ID = Contract.T_ID) INNER JOIN O_Bill ON Bill.B_ID = O_Bill.B_ID) INNER JOIN I_Bill ON Bill.B_ID = I_Bill.B_ID WHERE (((Tenant.Email)= 'bond007@yahoo.com'));

Owner

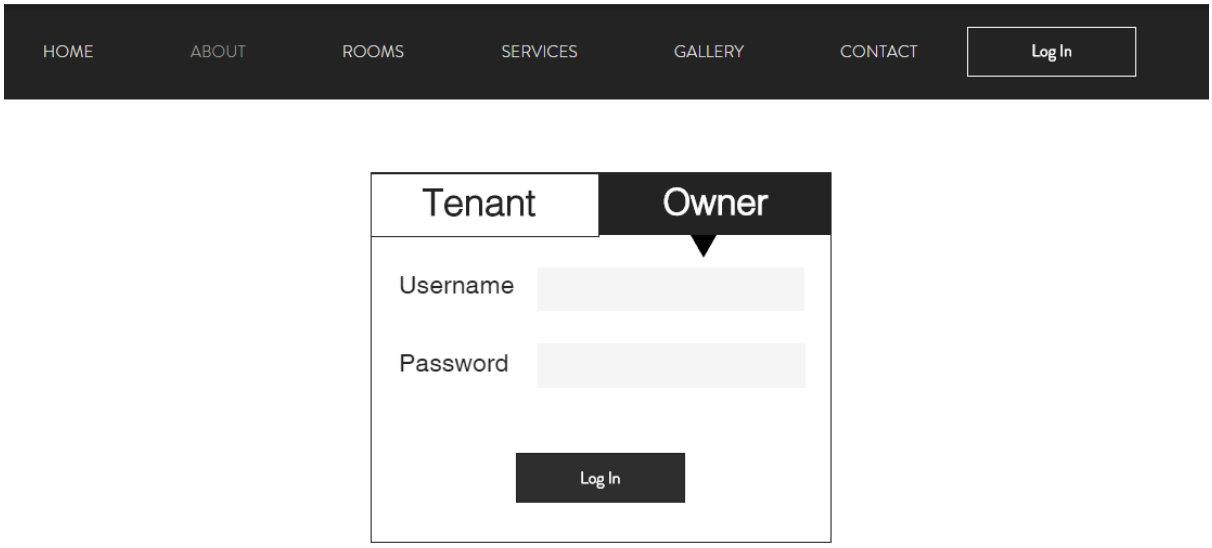
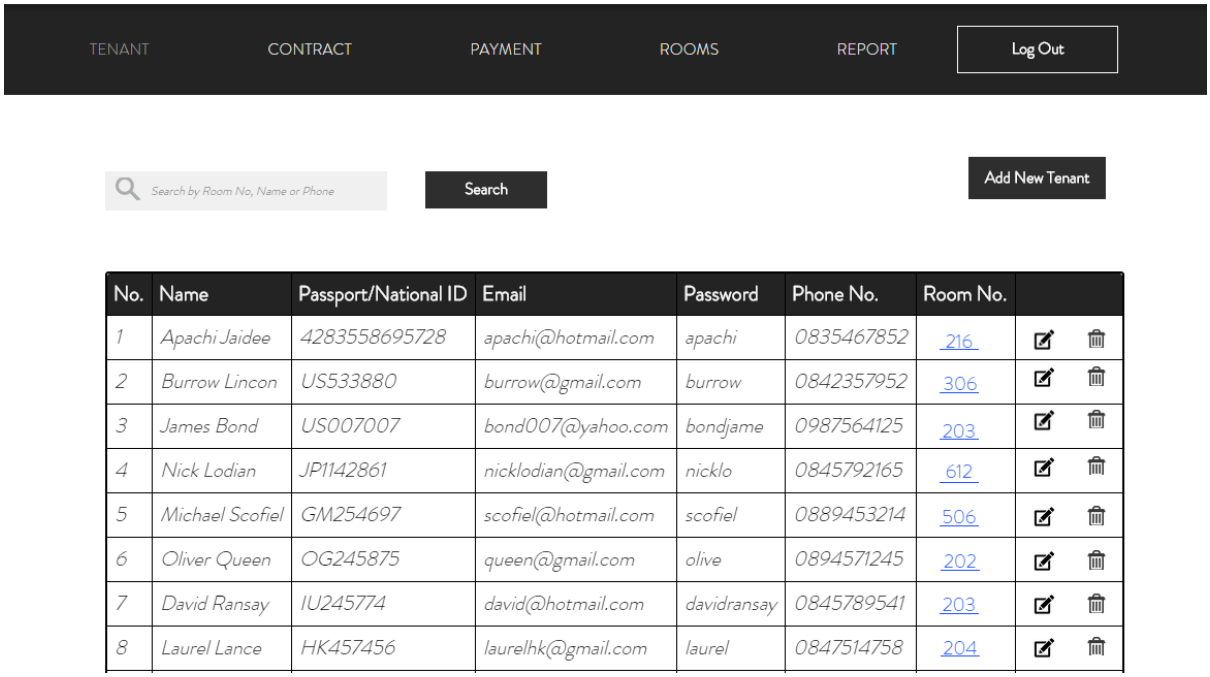


Figure 13: Login – Owner can login to the system by filling username and password.



No.	Name	Passport/National ID	Email	Password	Phone No.	Room No.	
1	Apachi Jaidee	4283558695728	apachi@hotmail.com	apachi	0835467852	216	
2	Burrow Lincon	US533880	burrow@gmail.com	burrow	0842357952	306	
3	James Bond	US007007	bond007@yahoo.com	bondjame	0987564125	203	
4	Nick Lodian	JP1142861	nicklodian@gmail.com	nicklo	0845792165	612	
5	Michael Scofiel	GM254697	scofiel@hotmail.com	scofiel	0889453214	506	
6	Oliver Queen	OG245875	queen@gmail.com	olive	0894571245	202	
7	David Ransay	IU245774	david@hotmail.com	davidransay	0845789541	203	
8	Laurel Lance	HK457456	laurelhk@gmail.com	laurel	0847514758	204	

Figure 14: Tenant Information – Owner can view the information of Tenant.

SELECT * FROM Tenant INNER JOIN Contract ON Tenant.T_ID = Contract.T_ID;

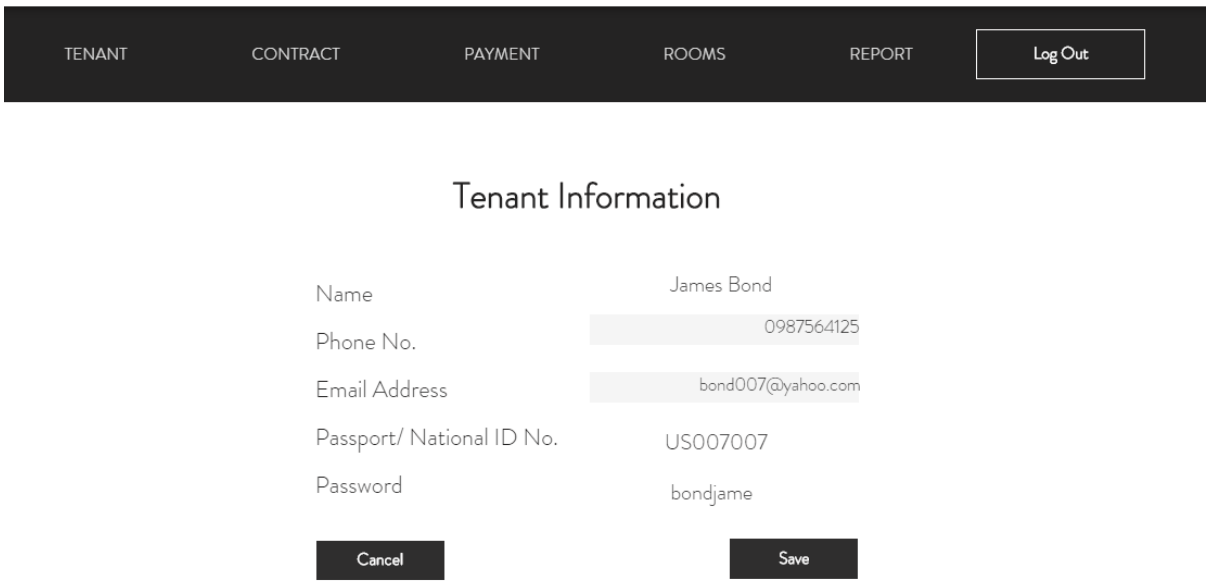


Figure 15: Edit Tenant Information – Owner can edit/update the information of Tenant.

UPDATE Tenant SET Email = 'bond007@yahoo.com', Phone = '0987564125' WHERE T_ID = 3;

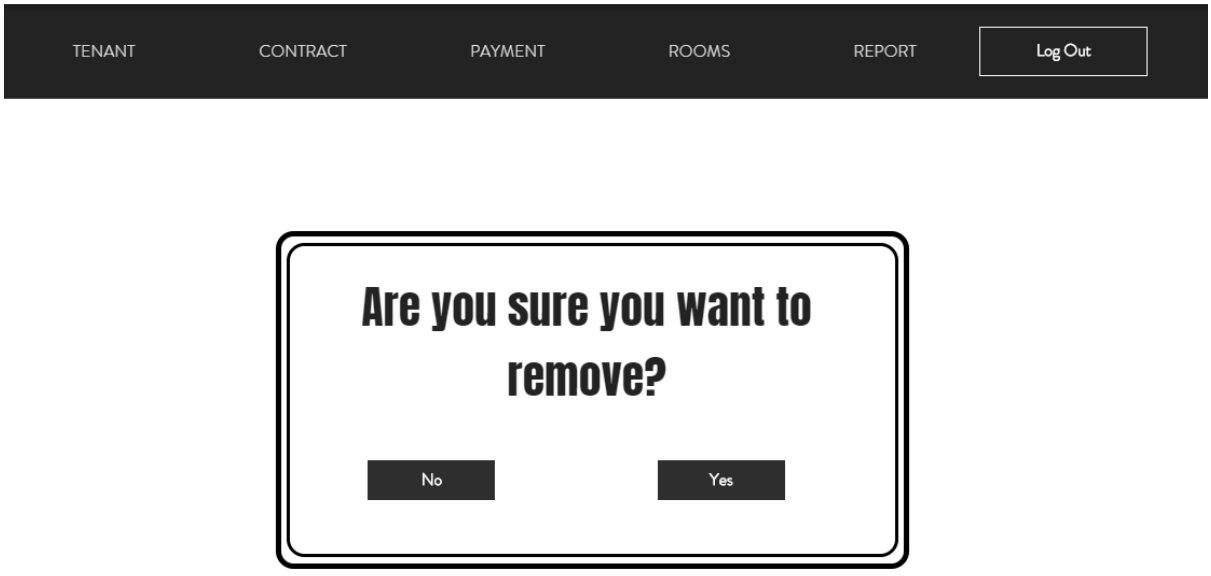


Figure 16: Remove Tenant Information – Owner can remove the information of Tenant.

DELETE FROM Tenant WHERE T_ID = 5;

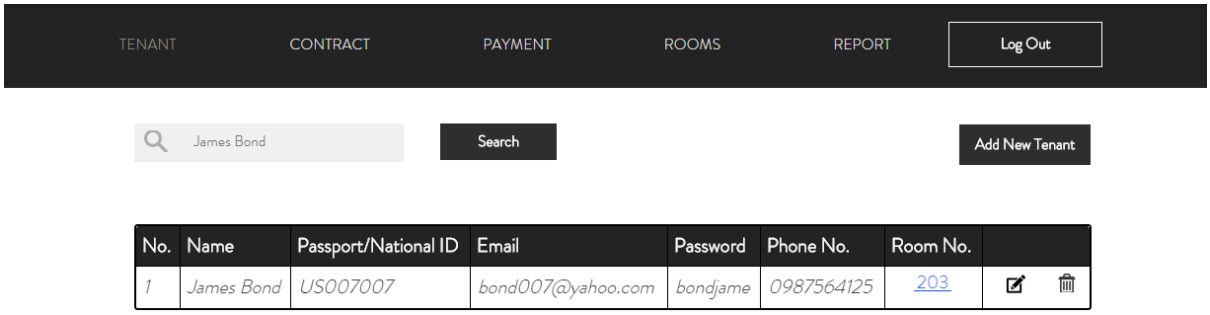


Figure 17: Search Tenant Information – Owner can search the information of Tenant by filling his/her name or room number.

SELECT * FROM Tenant INNER JOIN Contract ON Tenant.T_ID = Contract.T_ID WHERE Room_No = '203' OR T_Name LIKE '%James%';

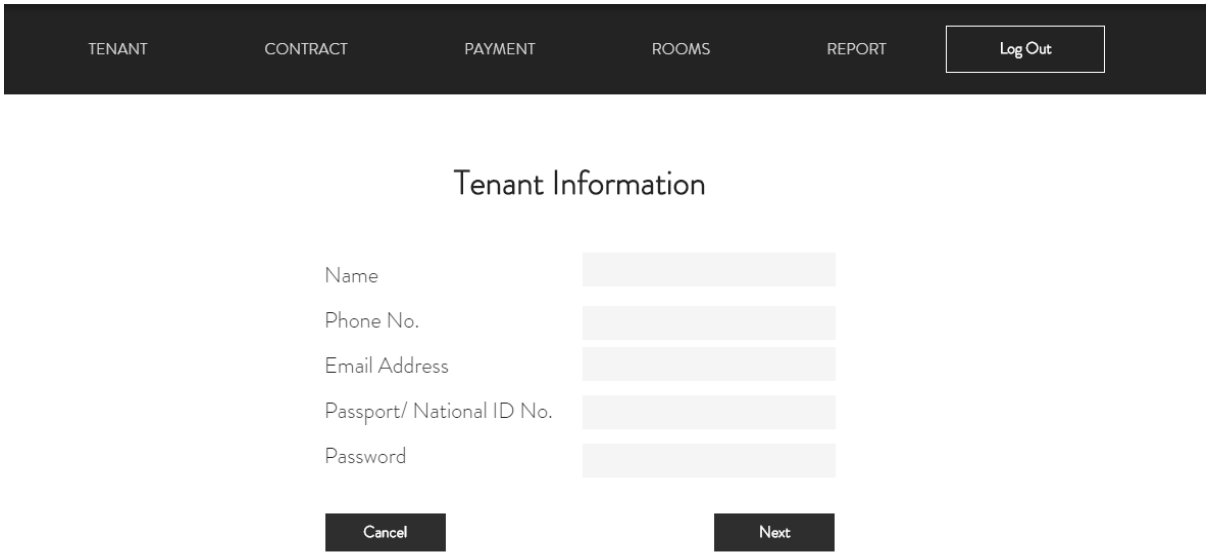


Figure 18: Add New Tenant – Owner can add new tenant by filling the information of Tenant and contract detail

INSERT INTO Tenant (T_Name, Phone, Email, Password, Passpord_ID) VALUES ('Burgle Danny', '0845768512', 'burgle@hotmail.com', 'burgle', 'AU1245486');

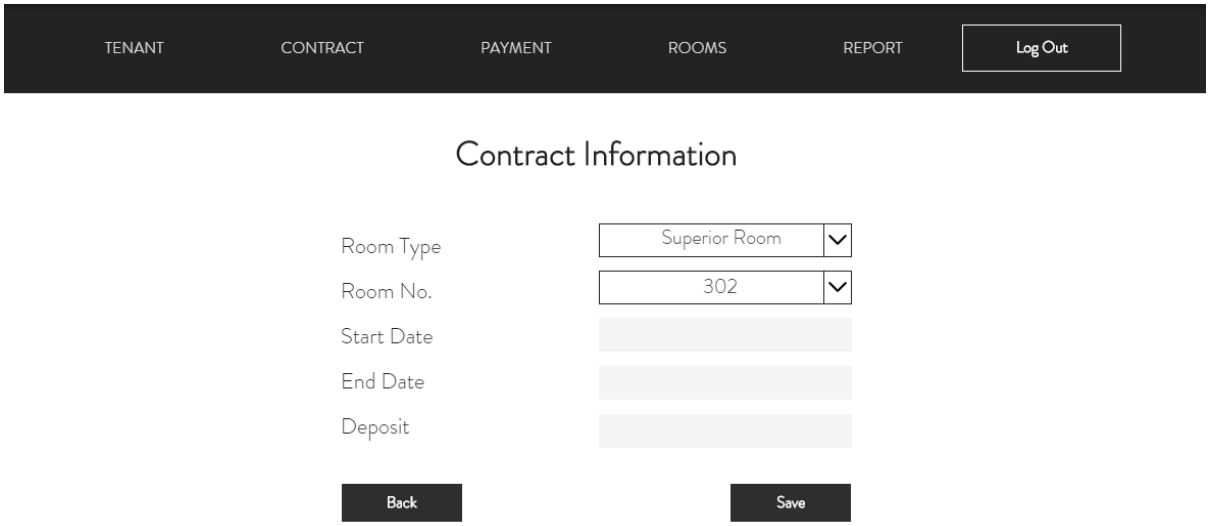


Figure 19: Add New Tenant – Owner can add new tenant by filling the information of Tenant and contract detail.

INSERT INTO Contract (Room_No, T_ID, Start, End, Deposit) VALUES ('302 ', '7', '01/May/2017', '01/May/2018', '16500');

UPDATE Room SET R_Status = '1' WHERE Room_No = '302';

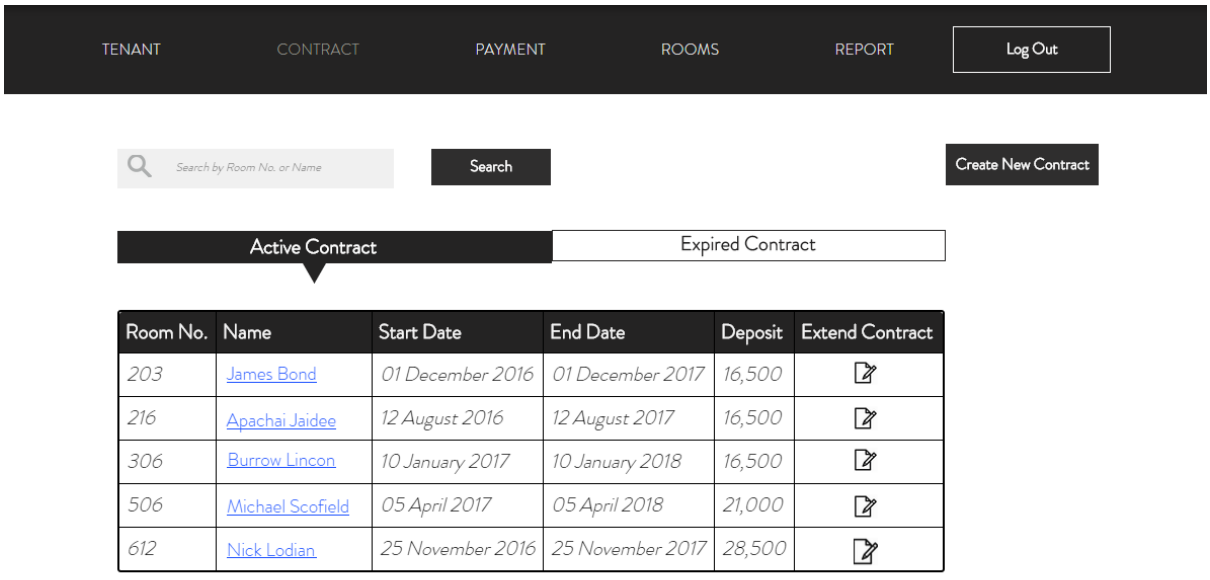


Figure 20: Active Contract – Owner can view the detail of active contract.

SELECT * FROM Contract INNER JOIN Tenant ON Contract.T_ID = Tenant.T_ID WHERE C_Status = '1';

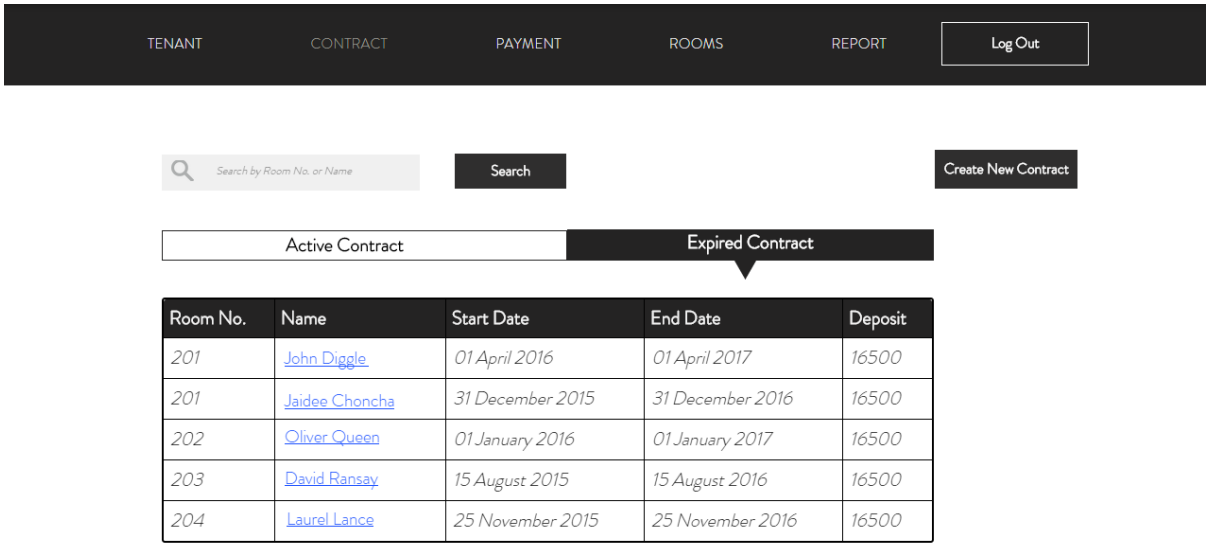


Figure 21: Expired Contract – Owner can view the detail of expired contract.

SELECT * FROM Contract INNER JOIN Tenant ON Contract.T_ID = Tenant.T_ID WHERE C_Status = '0';

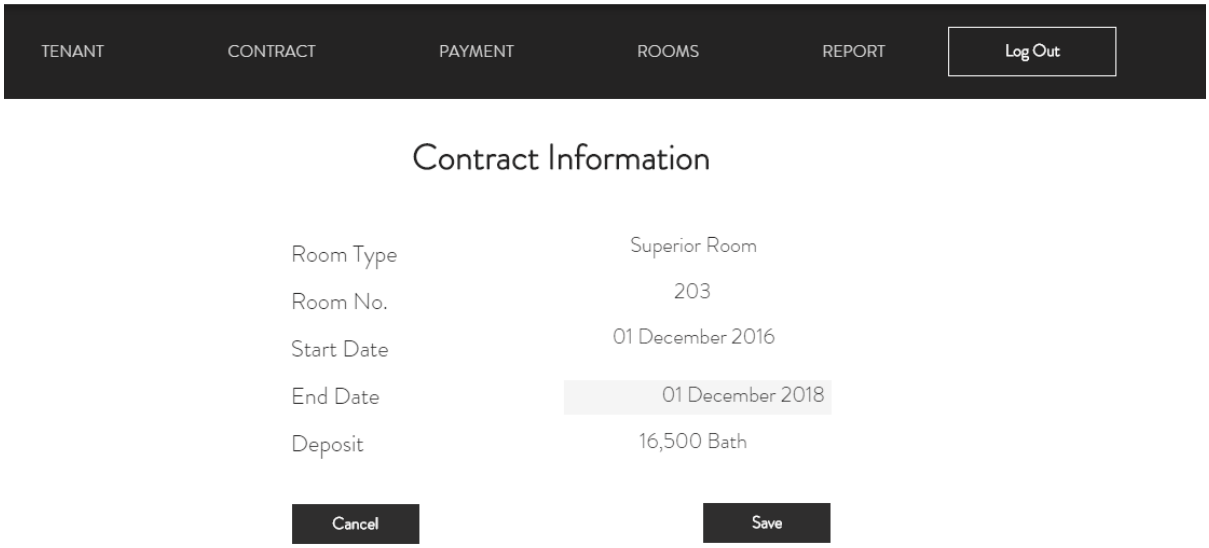


Figure 22: Extend Contract – Owner can extend the expired date of the contract.

UPDATE Contract SET End = '01 December 2018' WHERE T_ID = 3;

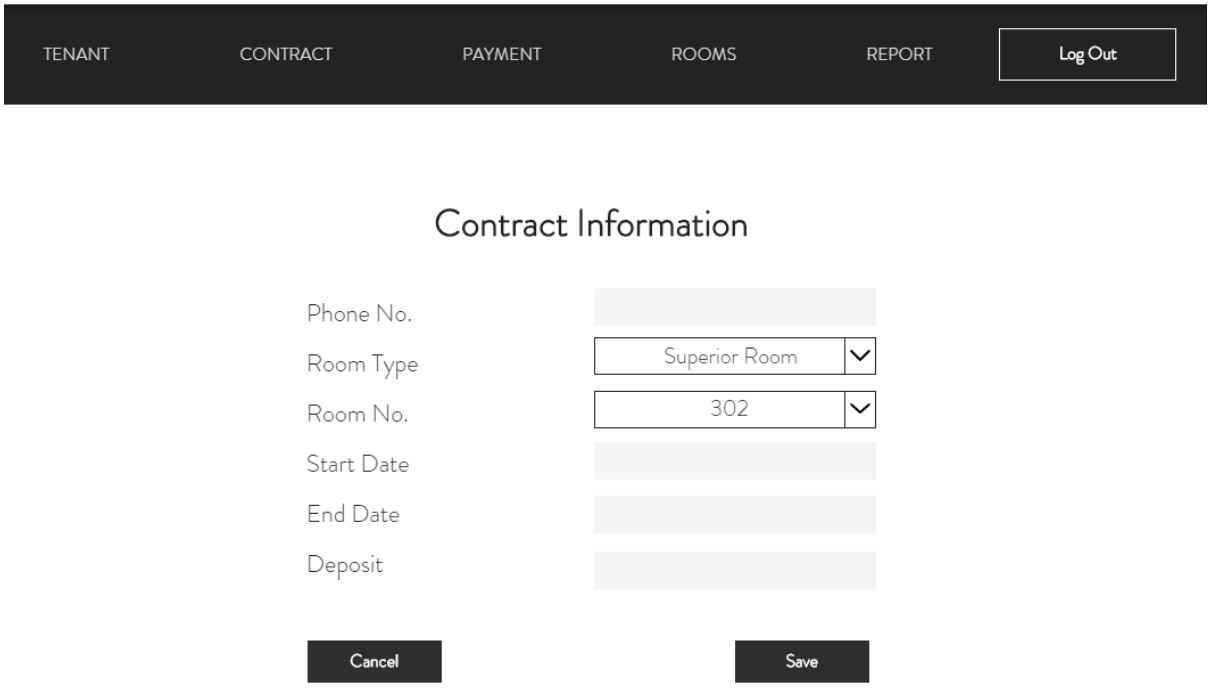


Figure 23: Create new contract – Owner can create new contract with the existing tenant.

INSERT INTO Contract (Room_No, T_ID, Start, End, Deposit) VALUES ('201', (SELECT T_ID FROM Tenant WHERE Phone = '0894571245'), '01/May/2017', '01/May/2018', '16500');

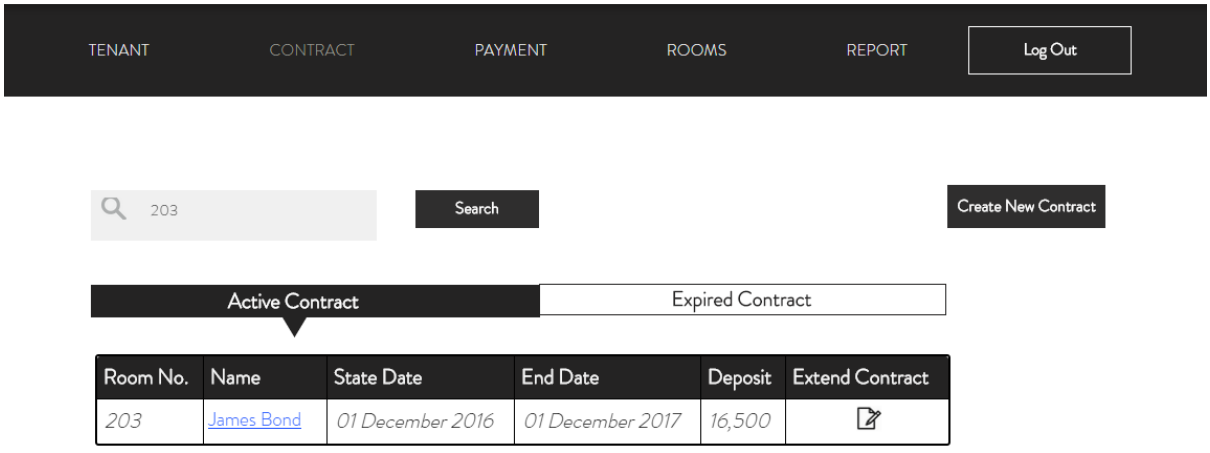


Figure 24: Search Contract – Owner can search the specific contract by filling room number.

SELECT * FROM Contract INNER JOIN Tenant ON Tenant.T_ID = Contract.T_ID WHERE Room_No = '203';

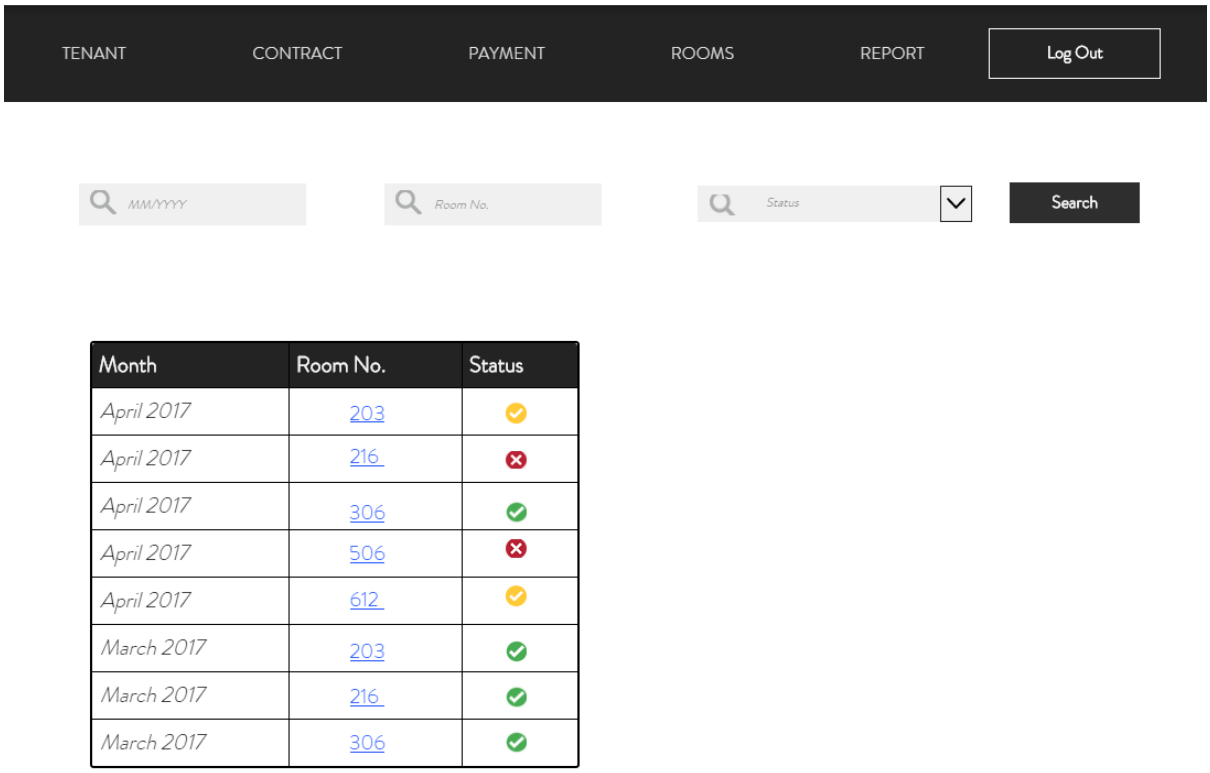


Figure 25: Bill – Owner can view the bill of all the occupied rooms.

SELECT * FROM Bill;

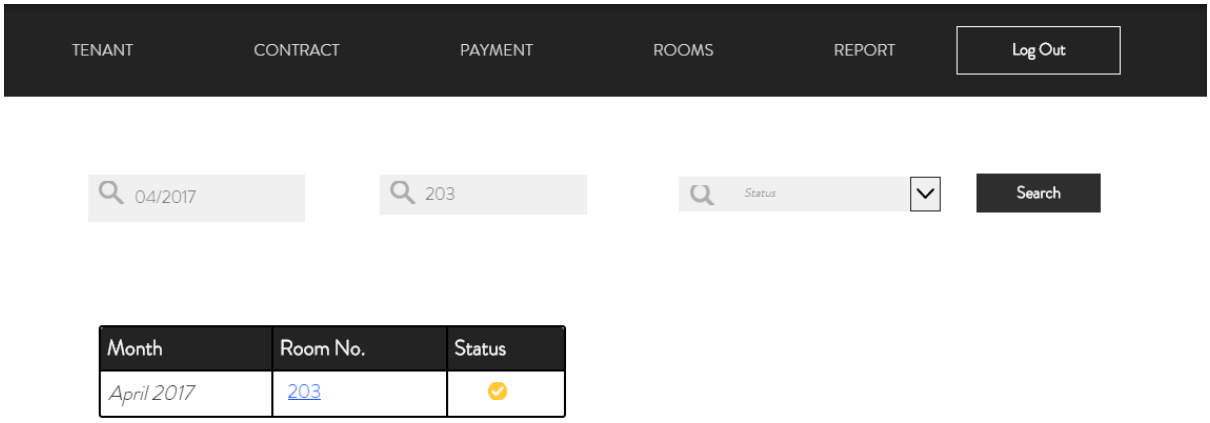


Figure 26: Search Bill – Owner can search the bill of specific room and month.

SELECT * FROM Bill INNER JOIN Contract ON Bill.C_No = Contract.C_No WHERE MONTH(Bill.Month) = 4 AND YEAR(Bill.Year) = 2017 AND Contract.Room_No = '203';

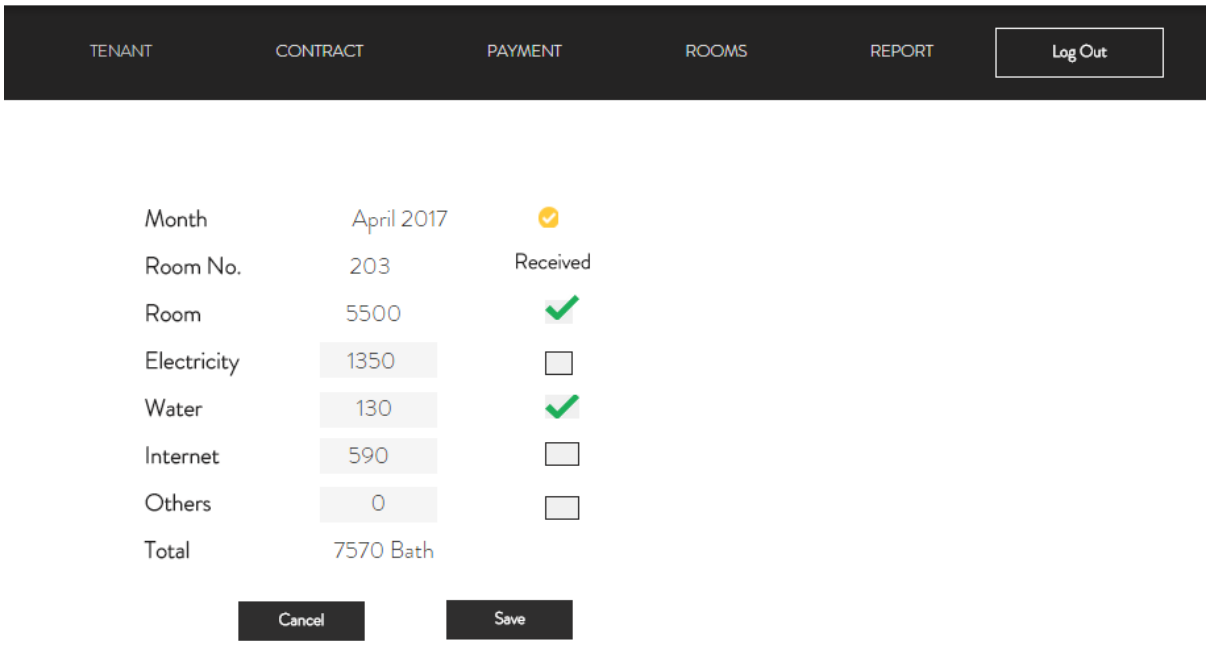


Figure 27: Bill Detail – Owner can view and add the detail of each room.

SELECT * FROM ((((Bill INNER JOIN E_Bill ON Bill.B_No = E_Bill.B_No) INNER JOIN O_Bill ON Bill.B_No = O_Bill.B_No) INNER JOIN R_Bill ON Bill.B_No = R_Bill.B_No) INNER JOIN W_Bill ON Bill.B_No = W_Bill.B_No) INNER JOIN I_Bill ON Bill.B_No = I_Bill.B_No WHERE C_No = '3';

INSERT INTO E_Bill VALUES (‘1’, ‘1350’, ‘0’);

INSERT INTO W_Bill VALUES (‘1’, ‘130, ‘0’);

INSERT INTO I_Bill VALUES (‘1’, ‘590, ‘0’);

INSERT INTO O_Bill VALUES (‘1’, ‘0’, ‘0’);

UPDATE R_Bill SET R_Status = ‘1’ WHERE B_No = ‘1’;

UPDATE W_Bill SET W_Status = ‘1’ WHERE B_No = ‘1’;

UPDATE Bill SET B_Status = ‘2’ WHERE B_No = ‘1’;

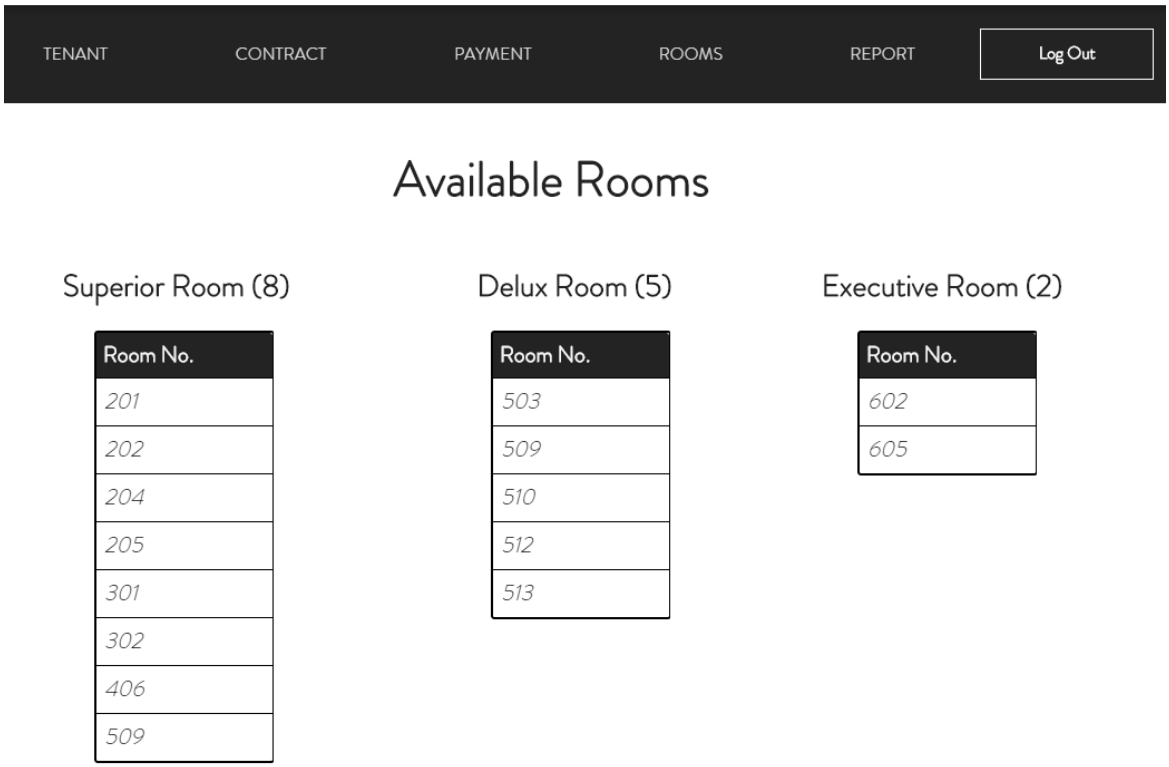


Figure 28: Available Rooms – Owner can check how many rooms are available.

SELECT * FROM Room WHERE R_Status = '0';

SQL Database Script

Database: ‘Seven_Twenty’

Table structure for table ‘Tenant’

```
CREATE TABLE `Tenant` ( `T_ID` INT(11) NOT NULL , `T_Name`VARCHAR(30) NOT NULL , `Phone` VARCHAR(10) NOT NULL , `Email` INT(30) NOT NULL, `Passport_ID` INT(20) NOT NULL , `Password` INT(10) NOT NULL , PRIMARY KEY(`T_ID`)) ENGINE = InnoDB;
```

Table structure for table Contract

```
CREATE TABLE `Contract` ( `C_No` INT(11) NOT NULL , `Room_NO`VARCHAR(3) NOT NULL , `T_ID` INT(11) NOT NULL , `Start` DATE NOT NULL , `End`DATETIME NOT NULL , `Deposit` DATETIME(5) NOT NULL , `C_Status` VARCHAR(1) NOTNULL , PRIMARY KEY (`C_No`), INDEX `FOREIGN
KEY` (`Room_NO`, `T_ID`)) ENGINE =InnoDB;
```

Table structure for table Room

```
CREATE TABLE ‘room` ( `Room_NO` VARCHAR(3) NOT NULL , `R_Type`INT(1) NOT NULL , `R_Status` VARCHAR(1) NOT NULL , PRIMARY KEY (`Room_NO`)) ENGINE = InnoDB;
```

Table structure for table Bill

```
CREATE TABLE ‘bill` ( `B_No` INT(11) NOT NULL , `Month` INT NOTNULL , `Year` INT NOT NULL , `Room_No` INT(5) NOT NULL , `Total` INT(5) NOTNULL , `B_Status` VARCHAR (1) NOT NULL , PRIMARY KEY (`B_No`), INDEX `FOREIGN
KEY`(`Room_No`)) ENGINE = InnoDB;
ALTER TABLE `bill` CHANGE `Month` `Month` DATETIME NOT NULL;
ALTER TABLE `bill` CHANGE `Year` `Year` DATETIME NOT NULL;
```

Table structure for table R_Bill

```
CREATE TABLE ‘R_Bill` ( `B_No` INT(11) NOT NULL , `R_Fee` INT(5)NOT NULL , `R_Status` VARCHAR (1) NOT NULL , PRIMARY KEY (`B_No`)) ENGINE = InnoDB;
```

Table structure for table E_Bill

```
CREATE TABLE ‘E_Bill` ( `B_No` INT(11) NOT NULL , `E_Fee` INT(5)NOT NULL , `E_Status` VARCHAR (1) NOT NULL , PRIMARY KEY (`B_No`)) ENGINE = InnoDB;
```

Table structure for table W_Bill

```
CREATE TABLE ‘W_Bill` ( `B_No` INT(11) NOT NULL , `W_Fee` INT(5)NOT NULL , `W_Status` VARCHAR (1) NOT NULL , PRIMARY KEY (`B_No`)) ENGINE = InnoDB;
```

Table structure for table I_Bill

```
CREATE TABLE ‘I_Bill` ( `B_No` INT(11) NOT NULL , `I_Fee` INT(5)NOT NULL , `I_S
tatus` VARCHAR (1) NOT NULL , PRIMARY KEY (`B_No`)) ENGINE = InnoDB;
```

Table structure for table O_Bill

```
CREATE TABLE ‘O_Bill` ( `B_No` INT(11) NOT NULL , `O_Fee` INT(5)NOT NULL , `
O_Status` VARCHAR (1) NOT NULL , PRIMARY KEY (`B_No`)) ENGINE = InnoDB;
```