Project Fancy Hotels
Group 57
CS 4400
Fall 2015
Phase III

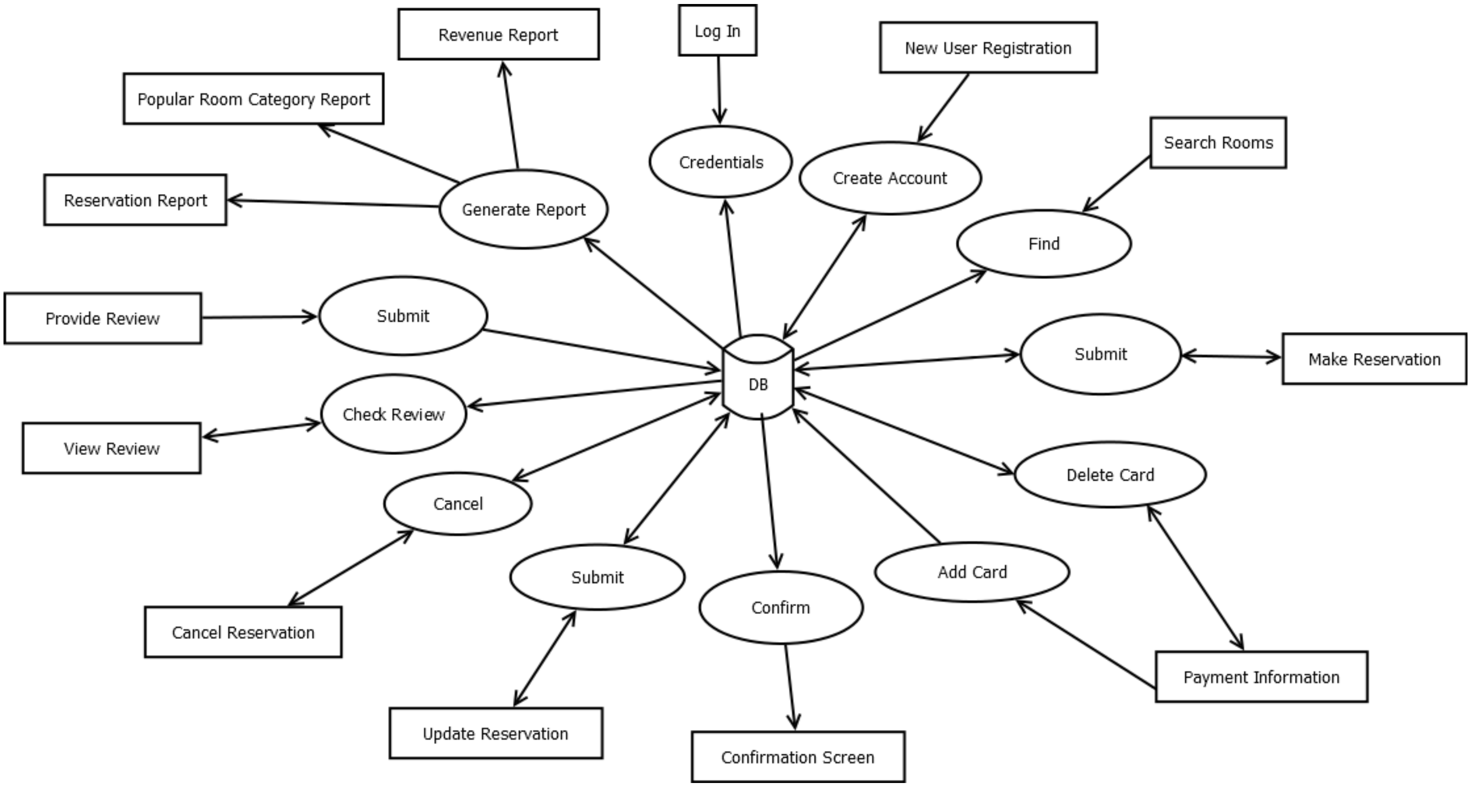
Mayurbhai Maisuria
mmaisuria3@gatech.edu
mmaisuria3
Monica Sweat, Section A

Justin Hicks
jhicks39@gatech.edu
jhicks39
Monica Sweat, Section A

Salim Moctar G. Sabo
msabo6@gatech.edu
msabo6
Shamkant B. Navathe, Section B

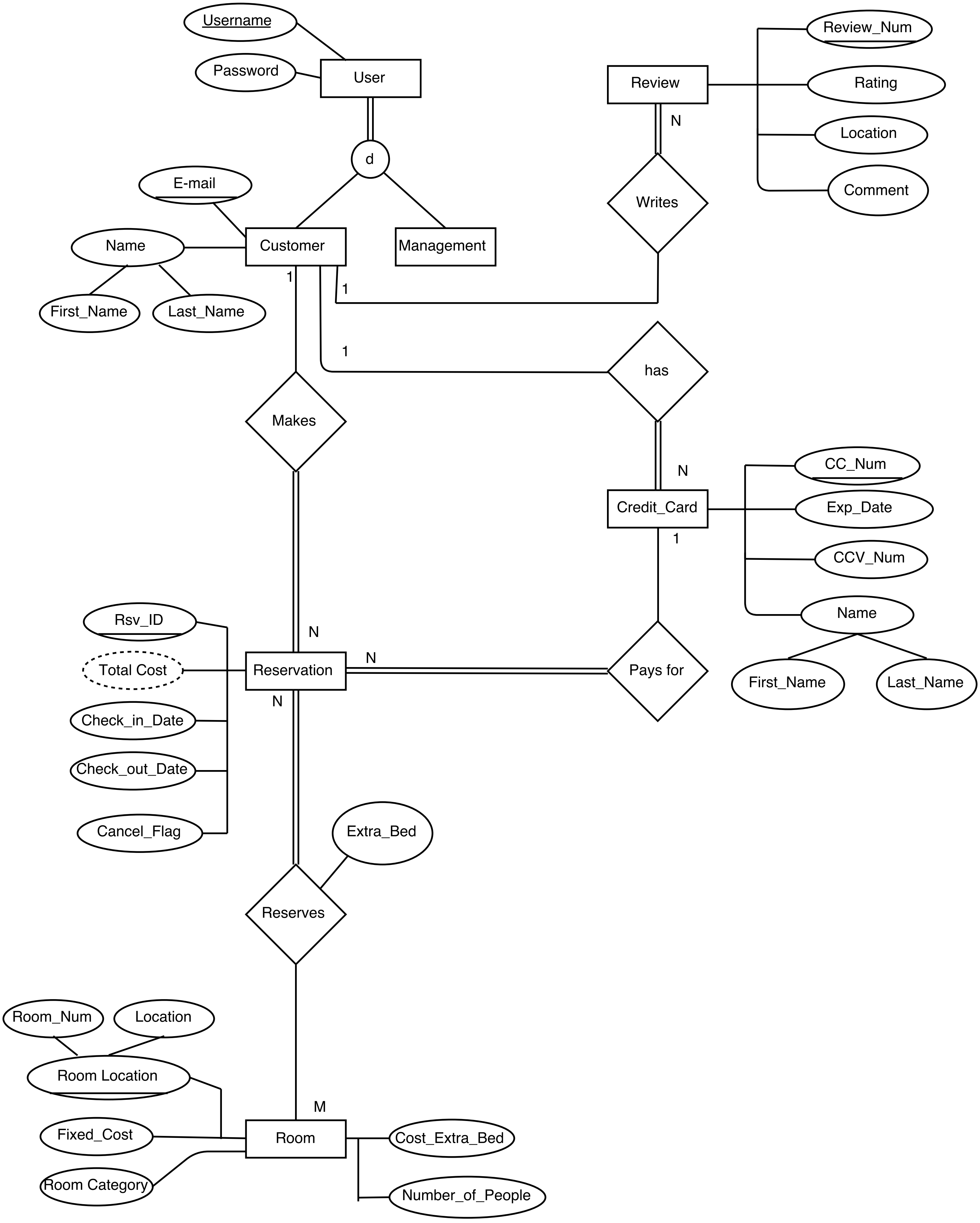
Zo Hming Thang
zthang3@gatech.edu
zthang3
Monica Sweat, Section A

For SQL tasks and data, see separate .txt files.


Revenue Report
Log In
New User Registration
Popular Room Category Report
Credentials
Create Account
Search Rooms
Reservation Report
Generate Report
Find
Provide Review
Submit
DB
Submit
Make Reservation
Check Review
View Review
Cancel
Delete Card
Submit
Add Card
Confirm
Cancel Reservation
Payment Information
Update Reservation
Confirmation Screen

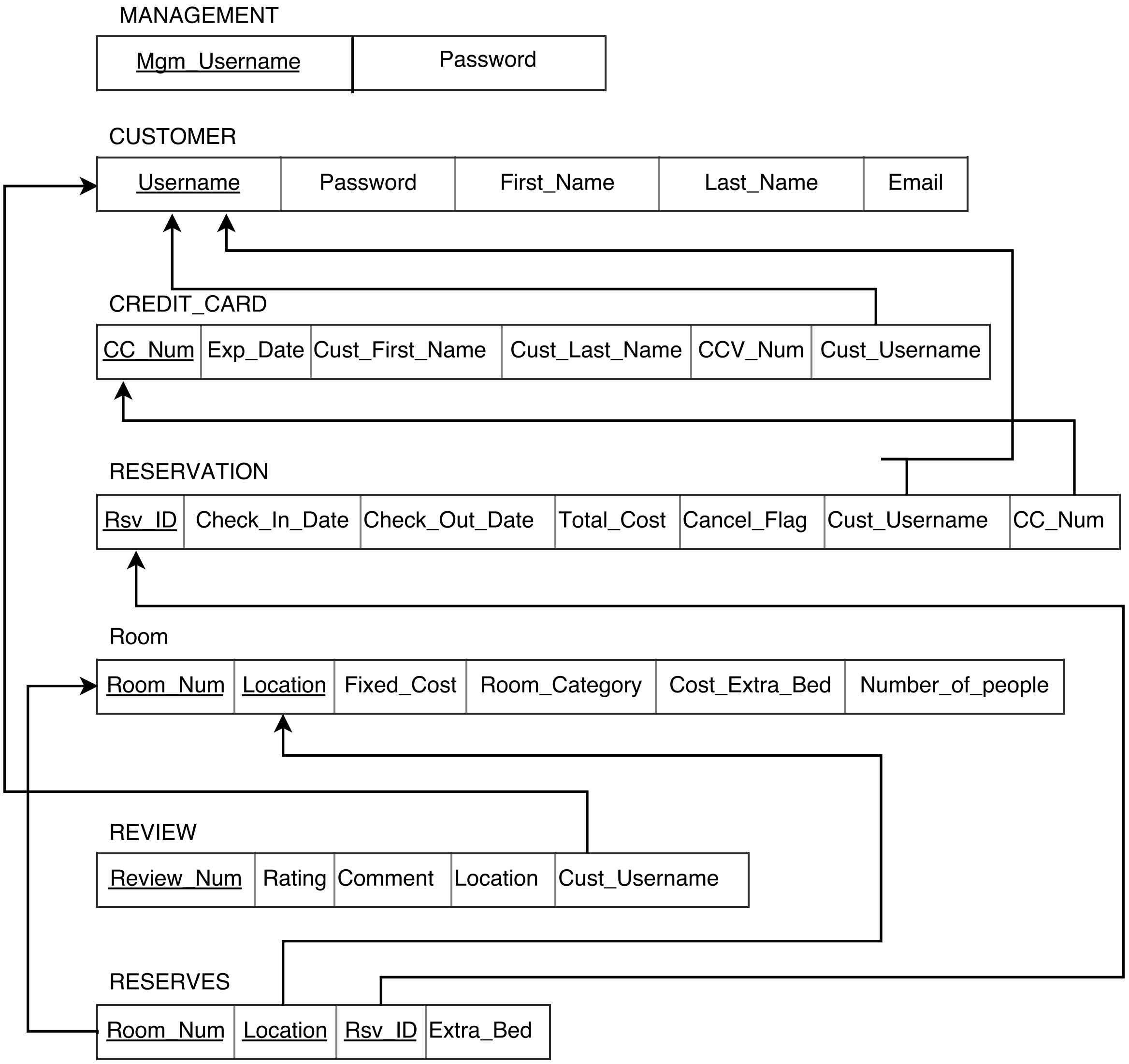
PHASE III

Project Fancy Hotels_Group 57
EER Diagram



PHASE III

Project Fancy Hotels_Group 57
Relational Schema Diagram



MANAGEMENT

Mgm_Username	Password

CUSTOMER

Username	Password	First_Name	Last_Name	Email

CREDIT_CARD

CC_Num	Exp_Date	Cust_First_Name	Cust_Last_Name	CCV_Num	Cust_Username

RESERVATION

Rsv_ID	Check_In_Date	Check_Out_Date	Total_Cost	Cancel_Flag	Cust_Username	CC_Num

Room

Room_Num	Location	Fixed_Cost	Room_Category	Cost_Extra_Bed	Number_of_people

REVIEW

Review_Num	Rating	Comment	Location	Cust_Username

RESERVES

Room_Num	Location	Rsv_ID	Extra_Bed

CS 4400
Fall 2015
Group Number 57
Phase III
SQL Statements for each task

```
#Task1: Customer Login
#Assume $dbuser and $dbpass are the values used to login
SELECT Username, Password
FROM CUSTOMER
WHERE Username = 'User1' AND Password = 'Password1';
```

```
#Task2: Manager login
SELECT Mgm_Username, Password
FROM MANAGEMENT
WHERE Mgm_Username = 'dbuser' AND Password = 'dbpass';
```

```
#Task3: Register/Create Account
#Assume $Newdbuser $Newdbpass $dbfirstN $dblastN $dbemail.
INSERT INTO CUSTOMER
(Username, Password, First_Name, Last_Name, Email)
VALUES
('Newdbuser', 'Newdbpass', 'dbfirstN', 'dblastN', 'Newdbemail')
```

```
#Task4: Search Rooms
#Assume $dblocation $dbstartDate $dbendDate
SELECT *
FROM ROOM
WHERE Location = 'Atlanta' AND (Room_Num, Location)
NOT IN (SELECT Room_Num, Location
FROM RESERVES NATURAL JOIN RESERVATION
WHERE Cancel_Flag = 0 AND
((Check_In_DATE <’dbendDate’) AND  (Check_Out_DATE >’dbstartDate’)));
```

```
#Task5: Add Credit Card
# Assume $dbFname, $dbLname, $dbuser $dbcc_Num,$expDate, $dbCCV
INSERT INTO CREDIT_CARD
(Cust_First_Name,Cust_Last_Name,Cust_Username, CC_Num, Exp_Date,CCV_NUM)
VALUES
 (‘dbFname’, ‘dbLname’, ‘dbuser’ ,‘dbcc_Num’, ‘expDate’, ‘dbCCV’);
```

```
#Task6: Delete Credit Card
#Assume $dbuser, $0000000000000000 (CC num)
#step 1: check condition if the card can be deleted
SELECT *
FROM RESERVATION
WHERE Cust_Username=’dbuser’ AND DATEDIFF(Check_Out_Date,CURDATE())>0 AND
CC_num = ‘1000000000000000’;
# step 2: Delete card
DELETE
FROM CREDIT_CARD
WHERE Cust_Username=’User1’ AND CC_num = ‘1000000000000000’ ;
```

```
#Task7: Reserving a room.
#Assume $dblocation $dbstartDate $dbendDate, credit card $dbcc_Num, and
#dbtotalCost, which will be calculated (nested query)
#Assume ‘dbExtraBed’ as 1 or 0; whether the customer wants an extra bed or not.
INSERT INTO RESERVATION
(Rsv_ID, Check_In_Date, Check_Out_Date, Total_Cost, Cust_Username, CC_Num)
VALUES
('dbRsv_ID', 'dbstartDate', 'dbendDate',
(SELECT DATEDIFF('dbendDate, 'dbstartDate') * (SELECT Fixed_Cost FROM ROOM
WHERE Room_Num = 'dbRoom_Num') + (SELECT Cost_Extra_Bed FROM ROOM
WHERE Room_Num = 'dbRoom_Num') * '1'),
'dbuser', 'dbcc_Num');
INSERT INTO RESERVES (Room_Num, Location, Rsv_ID, Extra_Bed)
VALUES ('dbRoom_num', 'dblocation','dbrsv_id', 'dbExtraBed');
```

```
# Task8: Confirmation Screen
SELECT Rsv_ID FROM RESERVATION WHERE Cust_Username = "dbuser";

# Assumption: User can’t make a reservation on the start date.
# Task9: Update Reservation
SELECT *
FROM ROOM NATURAL JOIN RESERVES
WHERE Location = 'dblocation' AND RSV_ID = 'dbrsv_id'
 AND (Room_Num, Location) NOT IN
        (SELECT (Room_Num, Location)
FROM (RESERVES NATURAL JOIN RESERVATION)
        WHERE(Cancel_Flag = 0 AND RSV_ID = 'dbrsv_id' AND
((Check_In_DATE <’dbendDate’) AND  (Check_Out_DATE >’dbstartDate’));
#this will update the reservation into the database
UPDATE RESERVATION SET  Check_In_Date =  'dbstartDate' , Check_Out_Date =
'dbendDate' , Total_Cost=  'dbtotalCost'  WHERE RSV_ID = 'dbrsv_id'



#Task9: Cancel Reservation
UPDATE RESERVATION SET Cancel_Flag = 1, Total_Cost=  'dbnewtotalCost' WHERE
Rsv_ID = "dbrsv_ID"

#Task10: View Review
SELECT Rating, Comment FROM REVIEW WHERE Location = 'dblocation' ORDER BY
Rating;


#Task11: Write Review
#Assume $dbreviewnum, $dbrating, $dbcomment
SELECT DISTINCT Location
FROM RESERVATION
NATURAL JOIN ROOM
NATURAL JOIN RESERVES
WHERE Cust_Username = 'dbuser'

INSERT INTO REVIEW
```

```
(Location, Rating, Comment, Cust_Username)
VALUES
(dblocation, dbrating, dbcomment, dbuser);
```

```
#Manager Tasks
#Task13: View Reservation Report
SELECT Month(Check_In_Date) AS "Month", Location ,COUNT(*) AS "Total Number Of
Reservations"
FROM RESERVATION NATURAL JOIN RESERVES GROUP BY Month(Check_In_Date),
Location ORDER BY Month(Check_In_Date), Location;


#Task14: View popular room-category report
#get only August and September
SELECT month( Check_In_Date ) AS  "Month", Room_Category, Rsv_ID, Location,count(*) as "Total"
FROM RESERVATION
NATURAL JOIN ROOM
NATURAL JOIN RESERVES where Month(`Check_In_Date`)=8
GROUP BY Room_Category, location
ORDER BY Total DESC
LIMIT 5



#Task15: View Revenue Report

SELECT Month(Check_In_Date), Location, Sum(Total_Cost) AS "Total Revenue of Reservations"
FROM RESERVATION
NATURAL JOIN RESERVES
NATURAL JOIN ROOM
GROUP BY Month(Check_In_Date), Location;
```

```
#Created a view for all relevant Reservation/Reserves/Room information
CREATE VIEW All_Reservation_Data AS
SELECT Rsv_ID, Room_Num, Location, Check_In_Date , Check_Out_Date, Fixed_Cost, Total_Cost ,
Room_Category, Cost_Extra_Bed, Extra_Bed
FROM
RESERVATION
NATURAL JOIN ROOM
NATURAL JOIN RESERVES
```