Project Fancy Hotels
Group 57
CS 4400
Fall 2015
Phase II

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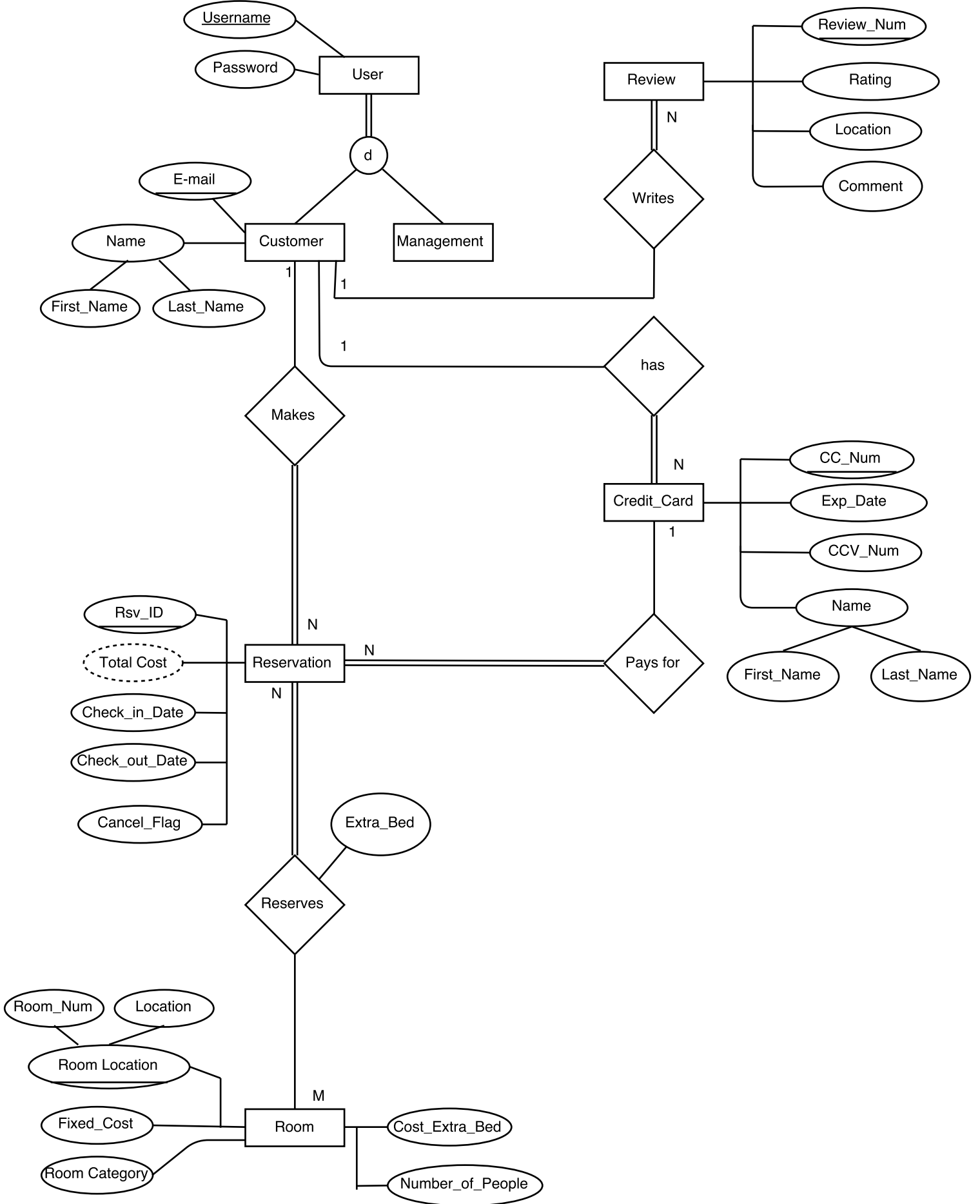mmaisuria3@gatech.edu
mmaisuria3
Monica Sweat, Section A

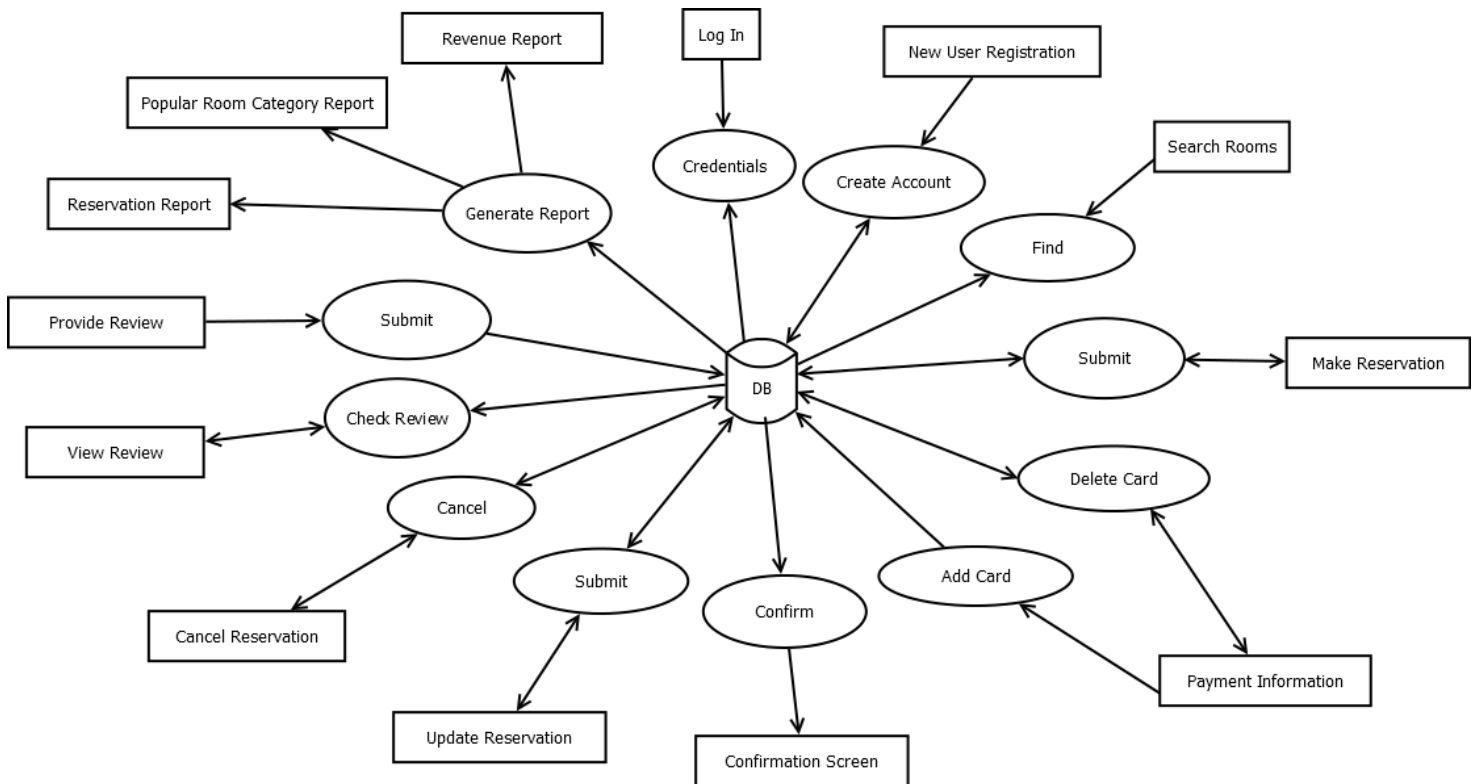Justin Hicks
jhicks39@gatech.edu
jhicks39
Monica Sweat, Section A

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

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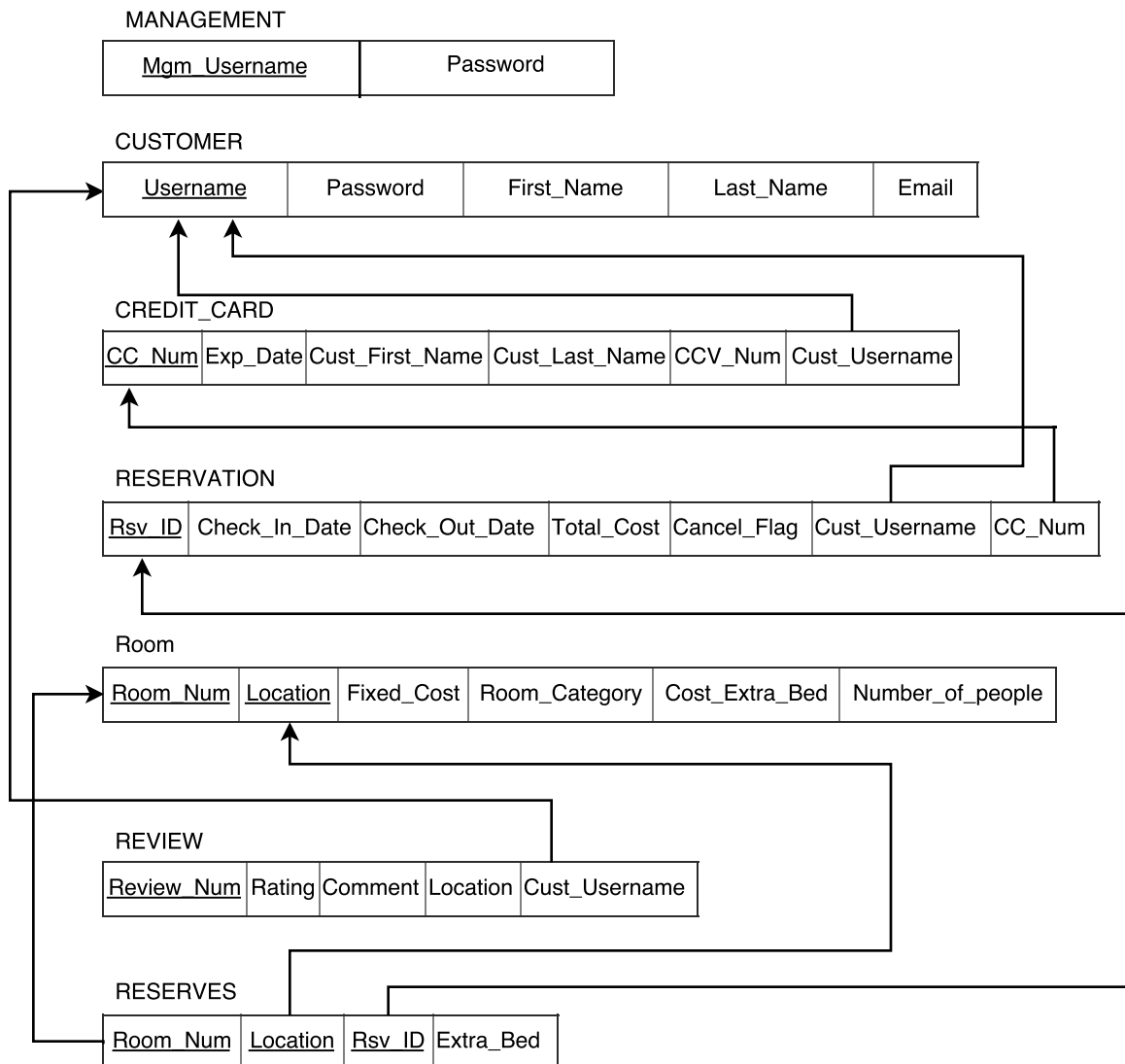Monica Sweat, Section A

PHASE II

Project Fancy Hotels_Group 57
EER Diagram

Username

Password

User

d

Review

Review_Num

Rating

Location

Comment

N

Writes

E-mail

Name

Customer

Management

First_Name

Last_Name

1

1

has

1

Makes

N

CC_Num

Exp_Date

Credit_Card

CCV_Num

Name

N

1

Rsv_ID

Total Cost

Reservation

N

N

Pays for

First_Name

Last_Name

Check_in_Date

Check_out_Date

Cancel_Flag

Extra_Bed

Reserves

Room_Num

Location

Room Location

Fixed_Cost

M

Room

Cost_Extra_Bed

Room Category

Number_of_People

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

CREATE TABLE MANAGEMENT (

       Mgm_Username   VARCHAR (30)   **NOT NULL**,

       Password   VARCHAR(30)   **NOT NULL**,

PRIMARY KEY (Mgm_Username)) ENGINE = InnoDB;

CREATE TABLE CUSTOMER (

       Username   VARCHAR (30)   **NOT NULL,**

       Password   VARCHAR(30)   **NOT NULL,**

       First_Name   VARCHAR (30)   **NOT NULL**,

       Last_Name   VARCHAR (30)   **NOT NULL,**

       Email   VARCHAR (30)   **NOT NULL,**

       PRIMARY KEY (Username),

       UNIQUE (Email)) ENGINE = InnoDB;

CREATE TABLE CREDIT_CARD (

       CC_Num   CHAR(16)   **NOT NULL,**

       Exp_Date   DATE   **NOT NULL,**

       CCV_NUM   CHAR(3)   **NOT NULL,**

       Cust_First_Name   VARCHAR (30)   **NOT NULL,**

       Cust_Last_Name   VARCHAR (30)   **NOT NULL,**

       Cust_Username   VARCHAR (30)   **,**

       PRIMARY KEY (CC_Num),

       FOREIGN KEY (Cust_Username) **REFERENCES** CUSTOMER (Username)

       ON DELETE SET NULL **ON UPDATE** CASCADE) ENGINE = InnoDB;

CREATE TABLE RESERVATION (

Rsv_ID	CHAR(6)	**NOT NULL,**
Check_In_Date	DATE	**NOT NULL,**
Check_Out_Date	DATE	**NOT NULL,**
Total_Cost	DECIMAL(19, 2)	**NOT NULL,**
Cancel_Flag	BOOLEAN	**DEFAULT 0,**
Cust_Username	VARCHAR (30)	,
CC_Num	CHAR(16)	,

PRIMARY KEY (Rsv_ID),

FOREIGN KEY (Cust_Username) **REFERENCES** CUSTOMER (Username)

ON DELETE SET NULL **ON UPDATE** CASCADE,

FOREIGN KEY (CC_Num) **REFERENCES** CREDIT_CARD (CC_Num)

ON DELETE SET NULL **ON UPDATE** CASCADE) ENGINE = InnoDB;

CREATE TABLE ROOM (

Room_Num	INT	**NOT NULL,**
Location	VARCHAR (30)	**NOT NULL,**
Fixed_Cost	DECIMAL(19, 2)	**NOT NULL,**
Room_Category	VARCHAR (30)	**NOT NULL,**
Cost_Extra_Bed	DECIMAL (19, 2)	,
Number_Of_People	INT	**NOT NULL,**

PRIMARY KEY (Room_Num, Location)) ENGINE = InnoDB;

CREATE TABLE REVIEW (

Review_Num	CHAR(6)	**NOT NULL,**
Rating	VARCHAR(15)	**NOT NULL,**
Comment	VARCHAR(100)	**NOT NULL,**
Location	VARCHAR (30)	**NOT NULL,**
Cust_Username	VARCHAR (30)	**,**

PRIMARY KEY (Review_Num),

FOREIGN KEY (Cust_Username) **REFERENCES** CUSTOMER (Username)

ON DELETE SET NULL **ON UPDATE** CASCADE) ENGINE = InnoDB;

CREATE TABLE RESERVES (

Room_Num	INT	**NOT NULL,**
Location	VARCHAR (30)	**NOT NULL,**
Rsv_ID	CHAR (6)	**NOT NULL,**
Extra_Bed	BOOLEAN	**DEFAULT 0,**

PRIMARY KEY (Room_Num, Location, Rsv_ID),

FOREIGN KEY (Room_Num, Location) **REFERENCES** ROOM (Room_Num, Location)

ON DELETE CASCADE **ON UPDATE** CASCADE,

INDEX (Room_Num, Location),

FOREIGN KEY (Rsv_ID) **REFERENCES** RESERVATION (Rsv_ID)

ON DELETE CASCADE **ON UPDATE** CASCADE) ENGINE = InnoDB;

Assumptions:
1.   Customers have enough money to make a reservation.
2.   Managers already have their login, so they do not need to sign up.
3.   User can have multiple user name with different emails.
4.   We want to keep as much information about past transactions as possible to keep track of revenues and histories in the Database. (That is why we used have ON DELETE SET NULL where it is possible).
     4.1.   All null types in the database will correspond to the foreign key when its parent has been deleted
     4.2.   Any null types in the database occur only when a table's foreign key refers to a deleted table.
5.   A database administrator can delete any tuple from any table in the database. (That is why we used only ON DELETE SET NULL or ON DELETE CASCADE).
6.   No tuple will have a null value within itself when created for the first time.
7.   A customer can only delete a card after the end date of their existing reservation(s).

Project Constraints:
1.   Customer must have a credit card to be able to make reservation because it is the only acceptable method of payment.
2.   When a new customer signs up for the first time, he or she must choose a password that satisfies the following security requirements:
     2.1.   At least eight characters long.
     2.2.   Must contain at least one uppercase (A-Z), one lower case(a-z) and one digit (0-9).
     2.3.   Must not contain user's name or username.
3.   If the customer's credit card expires, no payment can be made. The customer will have to enter new credit card information to be able to make a new reservation.
4.   Customers can only edit and/or cancel a reservation only if they have made reservation.
5.   The number of extra-beds a customer can request for, must be less or equal to the number of rooms for a specific reservation because any given room is allowed to have at most one extra bed.
6.   A Standard room can accommodate 2 people, a Family can accommodate 4, and a Suite can accommodate 4.
7.   Guests (customer included) that are 5 years of age or older count towards the room's capacity.
8.   The following policies would be applied when a reservation is canceled:
     8.1.   No refund for any cancellation made one day before the start date
     8.2.   80 % will be refunded for a cancellation made 3 days before the start date
     8.3.   100 % would be refunded for a cancellation made earlier than 3 days before the start date