CSCE 310 Final Project

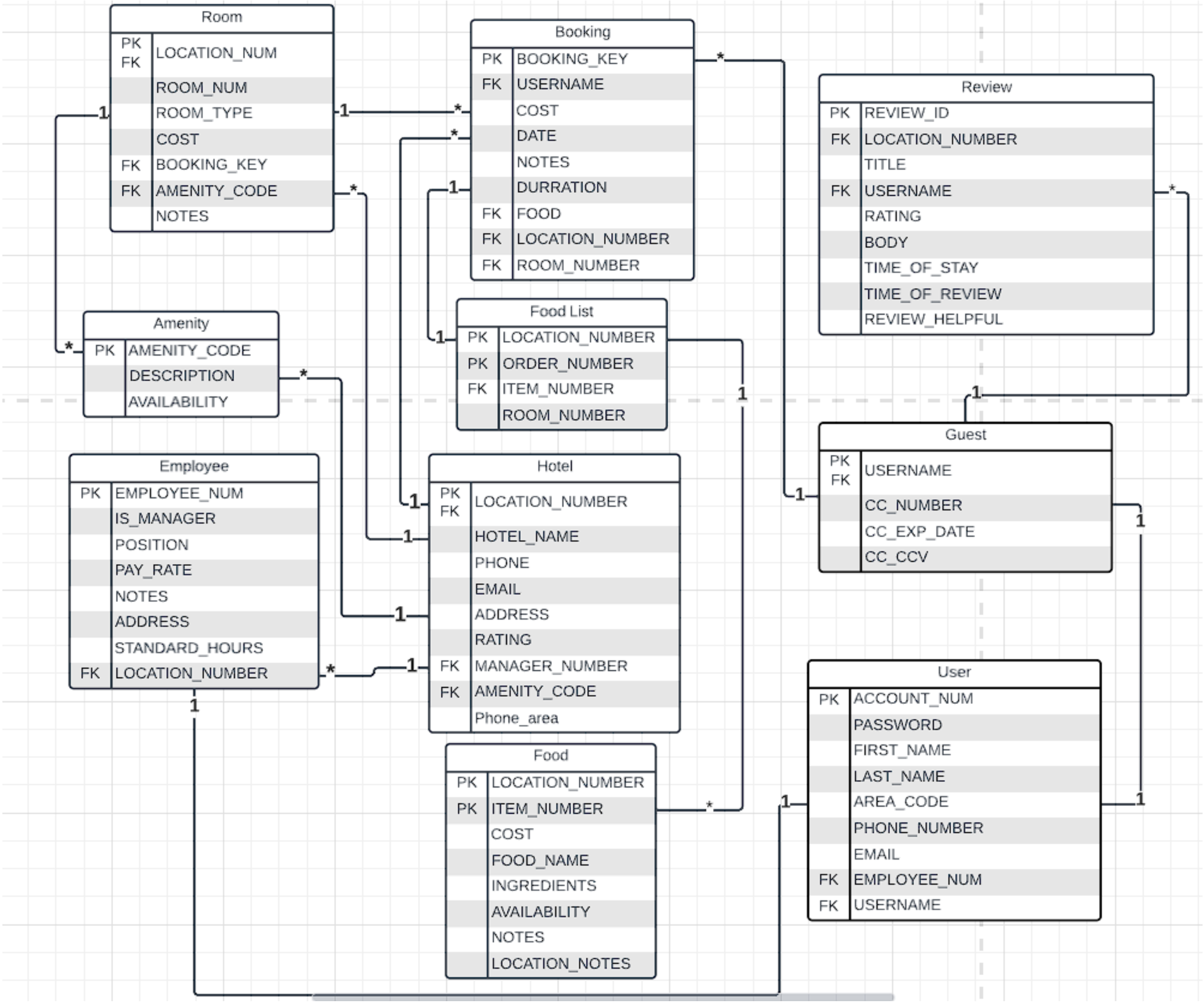
Group #24

Brandon Bartolomeo
Michael Mirhosseini
Alex Tung
Garrett Van Etten

5/5/23

Final Updated ERD:

Submitted ERD:



Repo

https://github.com/michaelmir915/CSCE310_Database

Individual Responsibilities:
Brandon

File Responsibility:
roomCharges.php (User page)
manageInventory.php (Employee Page)

Functionalities: Functionality Set 3

- Add items to orders
- Remove Items from orders
- Update prices on menu items
- Update availability of menu items

Insert:

```
INSERT INTO food_list (LOCATION_NUMBER, ORDER_NUMBER, ITEM_NUMBER,
ROOM_NUMBER) VALUES ('$location_number', '$orderNum', '$itemNumber', '1')
```

Join:

```
SELECT * FROM food
INNER JOIN food_list ON food.LOCATION_NUMBER = food_list.LOCATION_NUMBER
AND food.ITEM_NUMBER = food_list.ITEM_NUMBER
ORDER BY ORDER_NUMBER
```

Select:

```
SELECT * FROM food
```

Update:

```
UPDATE food SET FOOD_AVAILABILITY = FOOD_AVAILABILITY - 1 WHERE
LOCATION_NUMBER = '$location_number' AND ITEM_NUMBER = '$itemNumber'
```

Delete:

```
DELETE FROM food_list WHERE food_list.LOCATION_NUMBER = '$location_number'
AND food_list.ORDER_NUMBER = '$orderNum'
```

Michael

File Responsibility:
newBookings.php (user page)
SearchResults.php (user page)
currentBookings.php (user page)
manageBooking.php (manager page)

Functionalities: Functionality Set 2

Insert:

```
INSERT INTO booking (room_number, booking_start, booking_end, booking_cost,
username, location_number) VALUES ('$room_number', '$checkin', '$checkout',
'$booking_cost', '$username', '$location_number')"
```

Join:

```
SELECT room.*, room.room_cost
        FROM room
        LEFT JOIN booking
        ON room.room_number = booking.room_number
        AND booking.booking_start <= '$checkout'
        AND booking.booking_end >= '$checkin'
        WHERE booking.room_number IS NULL"
```

Delete:

```
DELETE FROM booking WHERE booking_key = '$booking_key'
```

Alex

File Responsibility:
review.php (user and manager page)
edit-review.php (helper function code)
delete-review.php (helper function code)
hotel.sql (database creation [review section])

Functionalities: Functionality Set 4

- Add comments, statuses, or reviews
- View comments, statuses, or reviews
- Update comments, statuses, or reviews
- Delete comments, statuses, or reviews

Insert:

```
INSERT INTO review (REVIEW_ID, LOCATION_NUMBER, USERNAME, REVIEW_TITLE,
REVIEW_RATING, REVIEW_BODY, REVIEW_TIME_OF_STAY, REVIEW_DAYS_STAYED,
REVIEW_TIME_CREATED, REVIEW_HELPFUL)
VALUES ('$review_id', '$location_num', '$username', '$title',
'$review_rating', '$review_body', '$review_stay_date', '$review_stay_len',
'$review_create_date', '$review_helpful');
```

Select:

```
SELECT * FROM review;
```

Update:

```
UPDATE review SET REVIEW_BODY='$new_review_body' WHERE
REVIEW_ID='$review_id';
```

Delete:

```
DELETE FROM review WHERE REVIEW_ID=$review_id;
```

Garrett

File Responsibility:

config.php (sets up database connection)
navbar.php (generates navbar on each page and manages the session)
login.php (page that logs a user in)
logout.php (script to logout a user)
register.php (this page adds new users to the database)
welcome.php (this is the default redirect page for profiles)
updateaccount.php (this is how normal users change their account details)
updateAccountEmployee.php (this is how employee update their account)
manageAccounts.php (this is the employee roster and editing capabilities)

Functionalities: Functionality Set 4

- Allow users to login
- Generating and mangaign UAC
- Change what a user can update
- How to find all relevant data for a user
- How adding a user changes the functionality

Insert

register user:

```
INSERT INTO user_hotel (USERNAME, PASSWORD, FIRST_NAME, LAST_NAME,
USER_EMAIL, USER_AREA_CODE, USER_NUMBER) VALUES ('$username',
'$hash_password', '$first_Name', '$last_Name', '$email', $area_code,
$phone_number)
```

register guest:

```
INSERT INTO guest (USERNAME, CC_NUMBER, CC_CCV, CC_EXP) VALUES
('$username', '$cc_number', $cc_cvc, '$cc_exp')
```

Hire employee:

```
INSERT INTO employee (EMPLOYEE_NUMBER, IS_MANAGER, POSITION, EMPLOYEE_PAY,
EMPLOYEE_NOTES, EMPLOYEE_STANDARD_HOURS, LOCATION_NUM, EMPLOYEE_EMAIL)
VALUES ($empid, '$manager', '$position', $empPay, '$empNotes',
'$empHours', $location, '$empemail')
```

Update

Update user information

```
UPDATE user_hotel SET password = '$hash_password', FIRST_NAME =
'$first_Name', LAST_NAME = '$last_Name', USER_EMAIL = '$email',
USER_AREA_CODE = $area_code, USER_NUMBER = $phone_number  WHERE
ACCOUNT_NUMBER = $accNum
```

Update guest information

```
UPDATE guest SET CC_NUMBER = '$cc_number', CC_CCV = $cc_cvc, CC_EXP =
'$cc_exp' WHERE USERNAME = $username
```

Updateing employee’s user account

```
UPDATE user_hotel SET password = '$hash_password', FIRST_NAME =
'$first_Name', LAST_NAME = '$last_Name', USER_EMAIL = '$email',
USER_AREA_CODE = $area_code, USER_NUMBER = $phone_number  WHERE
ACCOUNT_NUMBER = $accNum
```

Update employee account

```
UPDATE employee SET POSITION = '$postion', EMPLOYEE_ADRESS = '$adress',
EMPLOYEE_EMAIL = '$empEmail', EMPLOYEE_STANDARD_HOURS = '$hours' WHERE
EMPLOYEE_NUMBER = $empid
```

Update account for hiring

```
UPDATE user_hotel SET EMPLOYEE_NUMBER = $empid WHERE ACCOUNT_NUMBER =
$accountid
```

Update employee account as a manager

```
UPDATE employee SET POSITION = '$position', EMPLOYEE_PAY = $empPay,
EMPLOYEE_NOTES = '$empNotes', EMPLOYEE_STANDARD_HOURS = '$empHours',
LOCATION_NUM = $location, EMPLOYEE_EMAIL = '$empemail' WHERE
EMPLOYEE_NUMBER = $empid
```

Select

Checking login information

```
SELECT ACCOUNT_NUMBER, USERNAME, PASSWORD, EMPLOYEE_NUMBER FROM user_hotel
WHERE username = '$username'
```

Checking existing usernames for registering

```
SELECT USERNAME FROM user_hotel WHERE username = ?
```

Checking current employe data

```
SELECT * FROM employee WHERE EMPLOYEE_NUMBER = $empid
```

Getting current suer data

```
SELECT * FROM user_hotel WHERE USERNAME = '$username'
```

Manager view: checking employee data

```
SELECT * FROM employee WHERE EMPLOYEE_NUMBER = $empid
```

DELETE

Firing employee

```
DELETE FROM employee WHERE EMPLOYEE_NUMBER = $empid
```

Join

View full employee data as manager to fill roster

```
SELECT * FROM user_hotel INNER JOIN employee ON user_hotel.EMPLOYEE_NUMBER
= employee.EMPLOYEE_NUMBER
```