HMS Reference Document

1. External libraries: jQuery. Source:

https://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js

2. A brief walk through of HMS:

a. On login page, user can change guest login or staff login and guest login is in default. If password is wrong or missing, proper prompt will show up. If guest doesn't have an account, he/she is free to register by inputting unique username, password, valid first name, valid last name, passport number, telephone number and valid e-mail address. Staff accounts are already in database and staffs don't have to register.

b. On homepage, every guest is free to visit homepage of HMS, but he/she cannot book any rooms or check his/her reservation if he/she is not logged in. Any visitor can view the page "About Us". Once the guest is logged in, he/she can book a room by clicking the button "Book My Room!".

c. On any page of HMS, guests and staffs are allowed to logout, except the login page. Additionally, guests are free to return to homepage by clicking "Sunny Isle" on the top of the page except homepage.

d. After entering the book page, a guest is required to choose the type of room, check-in date and check-out date as well. Functions written in JavaScript will check if the guest has selected type of room, valid check-in and check-out date. After this, the function will check if there is any valid room for this type during their intended stay period.

e. Next, a guest could choose whether to book a randomly assigned room or choose a specific room on the next page. If choose a randomly assigned room, the system will randomly assign a room for the guest and return to the homepage.

f. If a guest chooses to select a specific room, he will be showed the plan view of the floors of the hotel. Moreover, he/she shall select a specific floor and then select a room. Or, he/she can skip this step and book a randomly assigned room by clicking the button "Skip".

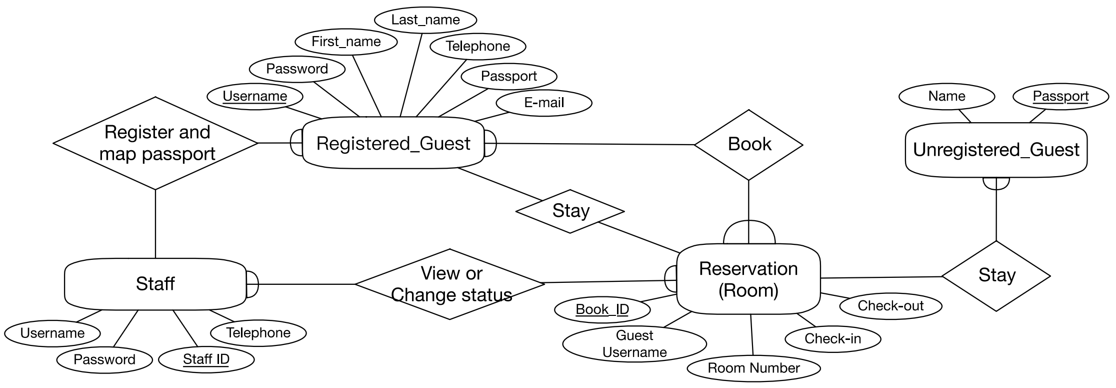
g. A guest could view or cancel his/her reservation by clicking the button "View/Cancel Reservation" on navigation bar on every page if the guest has logged in. After entering reservation page, he/she see a list of all his/her reservation with room type, room floor, room number, check-in date and check-out date. Also, he/she is allowed to cancel any reservation in the list.

h. As for staffs' operation, a staff is able to see all booked dates of a specific room or all reservation during a specific period. To see all dates of a room, he/she shall select specific floor and room. Then the list of reservation of this room shows on the right side of the page. To see reservation of a period, he/she shall select start date and end date. Then the list of reservation of this period shows below.

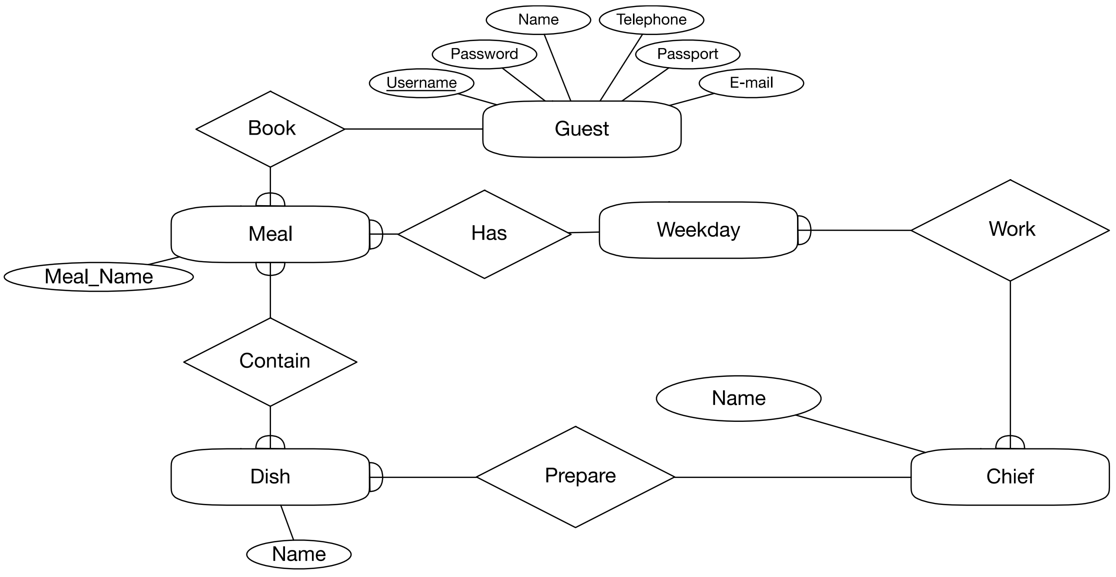
3. Mandatory:

a) Entity-Relationship Diagram:

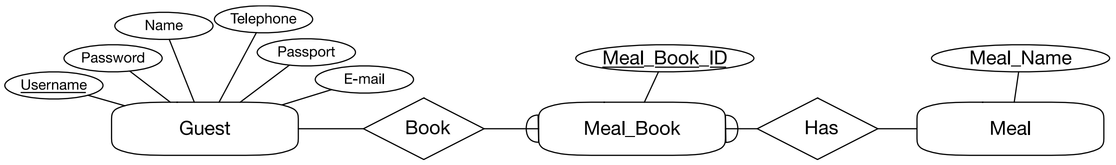
i. Guest & Room



ii. Meal, Dishes, Chief & Guest



iii. Guest & Meal_Book_Record



b) SQL “create table” statements:

```
Create TABLE `Registered_Guest` (
  `Username` varchar(255) NOT NULL,
  `Password` varchar(255) NOT NULL,
  `First_name` varchar(255) NOT NULL,
  `Last_name` varchar(255) NOT NULL,
  `Telephone` varchar(255) NOT NULL,
  `Passport` varchar(255) NOT NULL,
  `E-mail` varchar(255) NOT NULL,
  PRIMARY KEY (Username),
  UNIQUE (Username),
  UNIQUE (Passport)
);

CREATE TABLE `Unregistered_Guest` (
  `Name` varchar(255) NOT NULL,
  `Passport` varchar(255) NOT NULL,
  PRIMARY KEY (Passport),
  UNIQUE (Passport)
);

CREATE TABLE `Reservation` (
  `Book_ID` int(11) NOT NULL AUTO_INCREMENT,
  `RoomNumber` int(11) NOT NULL,
  `Username` varchar(255) NOT NULL,
  `Check-In` Date NOT NULL,
  `Check-Out` Date NOT NULL,
  PRIMARY KEY (Book_ID),
  CONSTRAINT f_r_key FOREIGN KEY (Username) REFERENCES
Registered_Guest(Username),
  UNIQUE (RoomNumber)
);

CREATE TABLE `Staff` (
   `Username` varchar(255) NOT NULL,
   `Password` varchar(255) NOT NULL,
   `Telephone` varchar(255) NOT NULL,
   `Staff_ID` varchar(255) NOT NULL,
   PRIMARY KEY (Staff_ID),
   UNIQUE (Username),
   UNIQUE (Staff_ID)
);
```

2.

```
CREATE TABLE `Meal_Book` (
  `Meal_Book_ID` int NOT NULL AUTO_INCREMENT,
  `Guest_Username` varchar(255) NOT NULL,
  `Meal_Name` varchar(255) NOT NULL,
  PRIMARY KEY (Meal_Book_ID),
  UNIQUE (Meal_Book_ID),
  CONSTRAINT f_m_key FOREIGN KEY (Meall_Name) REFERENCES Menu(Meal_Name),
  CONSTRAINT f2_m_key FOREIGN KEY (Guest_Username) REFERENCES
Registered_Guest(Username)
);

CREATE TABLE `Menu` (
    `Meal_Name` varchar(255) NOT NULL,
    `Dish_1` varchar(255) NOT NULL,
    `Dish_2` varchar(255) NOT NULL,
    `Dish_3` varchar(255) NOT NULL,
    `Dish_4` varchar(255) NOT NULL,
    `Dish_5` varchar(255) NOT NULL,
    `Dish_6` varchar(255) NOT NULL,
    `DIsh_7` varchar(255) NOT NULL,
    `Dish_8` varchar(255) NOT NULL,
    `Dish_9` varchar(255) NOT NULL,
    `DIsh_10` varchar(255) NOT NULL,
    PRIMARY KEY (Meal_Name),
    UNIQUE (Meal_Name)
);

CREATE TABLE `Weekday_Menu` (
  `Weekday` int NOT NULL,
  `Meal_1` varchar(255) NOT NULL,
  `Meal_2` varchar(255) NOT NULL,
  `Meal_3` varchar(255) NOT NULL,
  `Meal_4` varchar(255) NOT NULL,
  `Meal_5` varchar(255) NOT NULL,
  `Meal_6` varchar(255) NOT NULL,
  PRIMARY KEY (Weekday),
  CONSTRAINT f_w_m_key FOREIGN KEY (Weekday) REFERENCES Chief_Work(Weekday),
  UNIQUE (Weekday)
);

CREATE TABLE `Chief_Work` (
    `Weekday` int NOT NULL,
    `Chief_1` varchar(255) NOT NULL,
    `Chief_2` varchar(255) NOT NULL,
    `Chief_3` varchar(255) NOT NULL,
    `Chief_4` varchar(255) NOT NULL,
    `Chief_5` varchar(255) NOT NULL,
    `Chief_6` varchar(255) NOT NULL,
    `Chief_7` varchar(255) NOT NULL,
    `Chief_8` varchar(255) NOT NULL,
    PRIMARY KEY (Weekday),
    CONSTRAINT f_c_w_key FOREIGN KEY (Weekday) REFERENCES Weekday_Menu(Weekday),
    UNIQUE (Weekday)
);

CREATE TABLE `Chief_Dish` (
    `Chief_Name` varchar(255) NOT NULL,
    `Dish_1` varchar(255) NOT NULL,
    `Dish_2` varchar(255) NOT NULL,
    `Dish_3` varchar(255) NOT NULL,
    `Dish_4` varchar(255) NOT NULL,
    `Dish_5` varchar(255) NOT NULL,
    `Dish_6` varchar(255) NOT NULL,
    `Dish_7` varchar(255) NOT NULL,
    `Dish_8` varchar(255) NOT NULL,
    `Dish_9` varchar(255) NOT NULL,
    `Dish_10` varchar(255) NOT NULL,
    PRIMARY KEY (Chief_Name),
    UNIQUE (Chief_Name)
);
```

c) A brief discussion:

i. A registered guest is allowed to book many rooms and a room can be booked by several registered guest in different period. For the table registered guest, the username is the primary key and the passport is the unique key. In 'Unregistered_Guest' table, the guest has a name and a passport. The passport is the primary key of that table.

ii. In the table of room reservation, 'Book_ID' is the primary key and 'Username' is the foreign key which references to the table 'Registered_Guest''s 'Username'. Each time, the staff wants to check status of a room, a 'Book_ID' is unique to select a record and 'Username' is to find the information of guest.

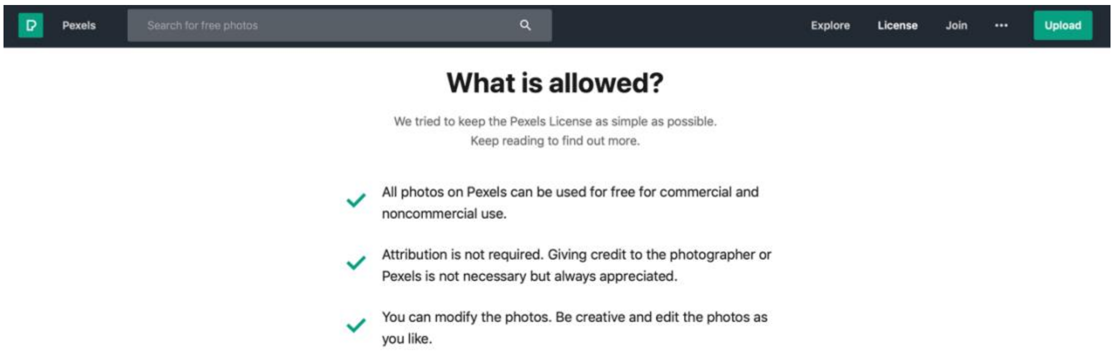
iii. The staff has a username, password, telephone number and staff ID. While 'Staff_ID' is the primary key and 'Username' is unique of this table.

iv. As for booking meal. The basic idea is that, a registered guest (with username) can check what meals are served on each day ('Week_Menu'). Additionally, he can see what are contained in one meal ('Menu'). If the guest has a preference chief, he shall check if the chief works on that day ('Chief_Work'). Once the guest finds the preference chief, he is allowed to check what dishes he can make or prepare ('Chief_Dish'). Management is also allowed to schedule what days chief works based on that table. In 'Weekday_Menu', 'Weekday' is the primary key and foreigh key which references to 'Chief_work(Weekday)'. In the table 'Mune', 'Meal_Name' is the primary key and unique key.

v. Like room reservation, there are reservations for meals as well. 'Meal_Book_ID' is like 'Book_ID' in room reservation. And it automatically increases. 'Guest_Username'is the foreign key which references to the guest' username who booked the meal. 'Meal_Name' is another foreign key relates to the name of a specific meal.

Recourse Licenses:

1. jQuery. Source: https://apps.bdimg.com/libs/jquery/2.1.4/jquery.min.js
2. Images are all from a website called "**Pexels**" which provides totally free images. Homepage: https://www.pexels.com. License: https://www.pexels.com/photo-license/



What is allowed?

We tried to keep the Pexels License as simple as possible.
Keep reading to find out more.

✓ All photos on Pexels can be used for free for commercial and noncommercial use.

✓ Attribution is not required. Giving credit to the photographer or Pexels is not necessary but always appreciated.

✓ You can modify the photos. Be creative and edit the photos as you like.

a) login1.jpg: https://www.pexels.com/photo/palm-trees-at-night-258154/
b) vip.jpg: https://www.pexels.com/photo/photo-of-living-room-1457842/
c) LDB.jpg: https://www.pexels.com/photo/two-white-bed-mattresses-near-wall-1097439/
d) LSB.jpg: https://www.pexels.com/photo/bed-bedroom-ceiling-chandelier-262048/
e) SSM.jpg: https://www.pexels.com/photo/bed-with-white-bed-sheet-and-pillowcase-set-1267438/

The code and related files are placed in my cslinux sever. The index.php is in public_html/src/index.php
http://cslinux.nottingham.edu.cn/~scyzz2/src/index.php