**reference document**

no external library is used (but some referencing codes)

introduction of my HMS

index.html

The whole system starts from this html file, whose name is a special “index” that helps the web server to find the directory. It also plays a role of the navigation page for the system, which can lead to the user and staff interfaces.

register.html

This page is structured with a form, handing input username, password, real name, passport ID, phone number and email address to the database system after submission.

In the following, all php files have the codes of connection to the database system.

onregister.php

This file executes the operation of checking the input validation, creating “roomers” table in the database and inserting posted information into this table. After all these operations, the page jumps to login.html.

login.html

This page is structured with a form, handing input username, password to the database system after submission.

onlogin.html

This file checks the validation of the input, then search in the table roomers to find valid username and password in record, meanwhile checking whether password is corresponding to the username. If all these validations are passed, a cookie named “username” is created for future usage. Finally, the page jumps to room\_book.html.

room\_book.html

This page is structured with a form, handing check-in time, check-out time and selected room type to the database system after submission.

onbook.php

Once the information are handed in, the check-in-out dates are measured not to be overdue or in wrong stream order. Next the system creates a “rooms” table containing information about rooms: roomname(exp. 101), floor(exp. 2), roomNo(exp. 13) and roomtype(exp. VIP room). Then all room information of Sunny Isle are inserted in. Then a table named “booked\_rooms” is created with properties of username, room information and check-in-out dates to record who booked which room for which days. It is empty initially. Afterwards the system filter all possible rooms for the user and shows them in the interface. The user then submit wanted room to the system.

onsubmitroom.php

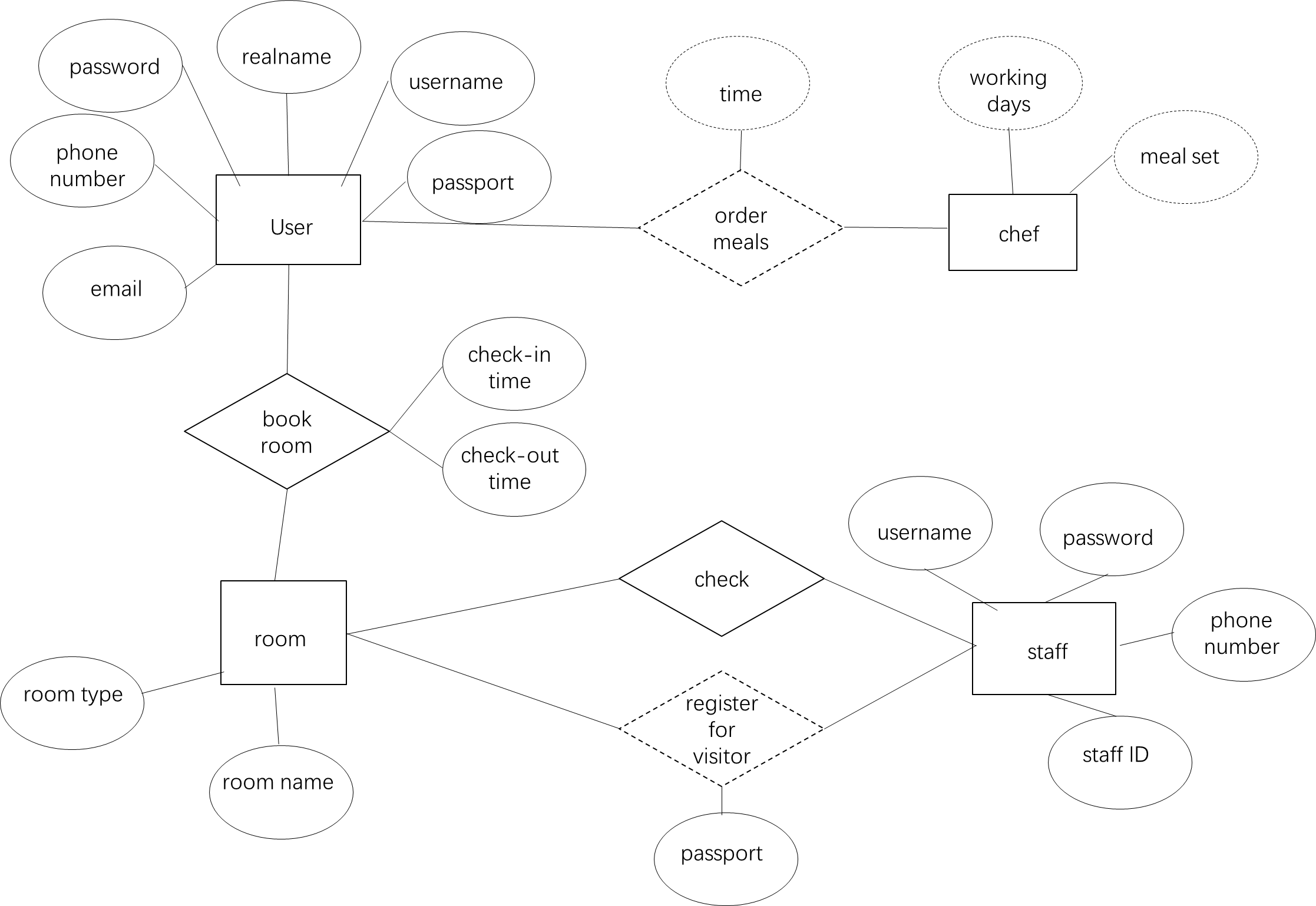
All information from cookies or posts are stored in the “booked\_room” table indicating that the room is occupied.

deleteapp.php and ondelete.php

The system also provides user a choice to delete a room he or she booked before.

stafflogin.html staffregister.html onstafflogin.php onstaffregister.php checkapp.html checkapp.php

These files include the register and login options for staff, they can search for a room name and check its availability condition.



M

N

1

N

M

M

N

M

（entity-relationship diagram）

“” part is from the php codes.

staff register for visitor:

a = SELECT roomname FROM booked\_rooms WHERE username =

(SELECT username FROM roomers WHERE passport = “the passport ID that visitor gives”)

if a is not empty

lead the user to one of the rooms.

staff can change booking status:

on the interface, after searching for a room, staff can choose to submit a form that change and delete room information.

INSERT INTO booked\_rooms (username, roomname, checkin, checkout)

VALUES（“staff submit”, “staff submit”, “staff submit”, “staff submit”）

DELETE FROM booked\_rooms WHERE roomname = “staff submit”

\*chef part:

create a table for chef and initialize it:

CREATE TABLE chef (

chefname VARCHAR(255),

workday1 DATE,

workday2 DATE,

workday3 DATE)

INSERT INTO chef (chefname, workday1, workday2, workday3)

VALUES (‘Charlie’, ‘Monday’, ‘Tuesday’, ‘Wednesday’),

…

(‘somebody’, ‘somemeal’, ‘Sunday’, ‘’, ‘’)

a meal table, one chefname may have several meal sets(several lines are inserted).

CREATE TABLE meals (

chefname VARCHAR(255),

mealset VARCHAR(255))

INSERT INTO meals (chefname, mealset)

VALUES (‘Charlie’, ‘chicken’),

(‘Charlie’, ‘salad’))

user order:

first in php interface a user choose order time(order time is then stored in a cookie).

SELECT chefname FROM chef WHERE

(workday1 = “user submit time” OR workday1 = “user submit time” OR workday1 = “user submit time”)

show user the available chef list on an interface, and he or she may click a “order” button for each chef jumping to a new interface to choose one meal set.

information selected from:

SELECT \* FROM meals WHERE chefname = “user click button”

CREATE TABLE orders (

username VARCHAR(255),

ordertime TIME,

mealset VARCHAR(255))

each time user clicks a chef button and submit a meal, one line is inserted into the table orders.

INSERT INTO orders (username, ordertime, mealset)

(“username cookie”, “ordertime cookie”, “user submit mealset”)

the “orders” table is available for chef and they can make meals.