Hotel Management System

Document

Breakertt | DBI-cw2 | 4.22.2019

1. Project Overview

1.1. Main purpose

This project is a hotel management system for Sunny Isle to accept more customers and manage books more convenient.

1.2. Developer

GAO Tianyi – scytg1@nottingham.edu.cn

1.3. Web Framework

1.3.1. Back-end

- php7
- mySQL

1.3.2. Front-end

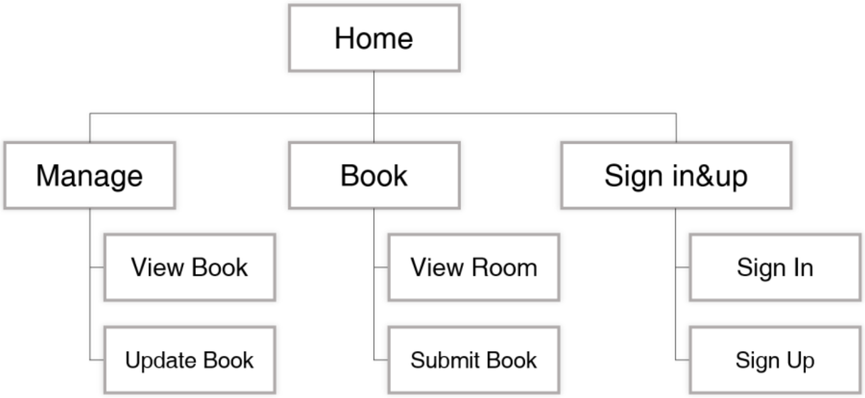
- jQuery (https://jquery.com/)
  A JS query library, for easier element select and data editing.
- Bootstrap (https://getbootstrap.com/)
  A design library based on jQuery.
- Font Awesome (https://fontawesome.com/)
  A font library including icons.
- DataTables (https://datatables.net/)
  A data table plugin of jQuery.
- GIJGO (https://gijgo.com/)
  jQuery plugin, offering date range picker.

1.4. Database Initial Script

‘`initDatabase.sql`’ in ‘`/src/`’

2. Site info

2.1. Site Map



2.2. Content Types

- **Room**
  - **Room Number**
  - **Room Type**
- **Book**
  - **Book Date**
  - **Book Status**
  - **Room Number**
  - **Customer Name**
- **User**
  - **User Name**
  - **User Type**

2.3. Page templates

Home Manage Book Sign-in&up

3. Database

Database initialize Script: ‘`initDatabase.sql`’

3.1. customer

Coloum	Type	Null	Comment
cID	BIGINT	false	
username	varchar(255)	false	username
password	varchar(255)	false	
truename	varchar(255)	false	Real Name
passportid	varchar(25)	false	Passport ID
telephone	varchar(20)	false	Telephone Number
email	varchar(255)	true	Email Address

3.2. staff

Coloum	Type	Null	Comment
sID	BIGINT	false	
username	varchar(255)	false	username
password	varchar(255)	false	
truename	varchar(255)	false	Real Name
staffid	varchar(25)	false	Passport ID
telephone	varchar(20)	false	Telephone Number
email	varchar(255)	true	Email Address

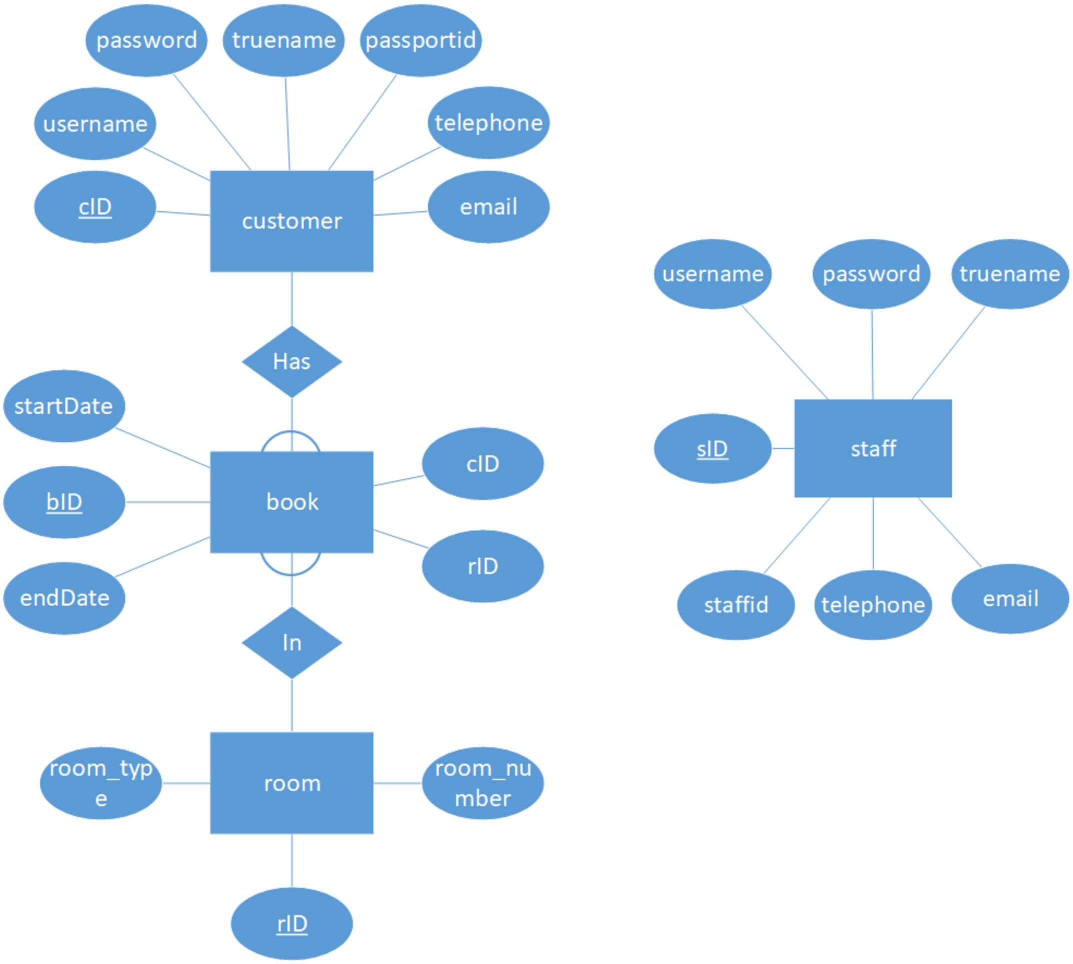
3.3. room

Coloum	Type	Null	Comment
rID	BIGINT	false	
room_number	varchar(5)	false	emxample:101
room_type	varchar(255)	false	vip/double/single_s/single_l

3.4. book

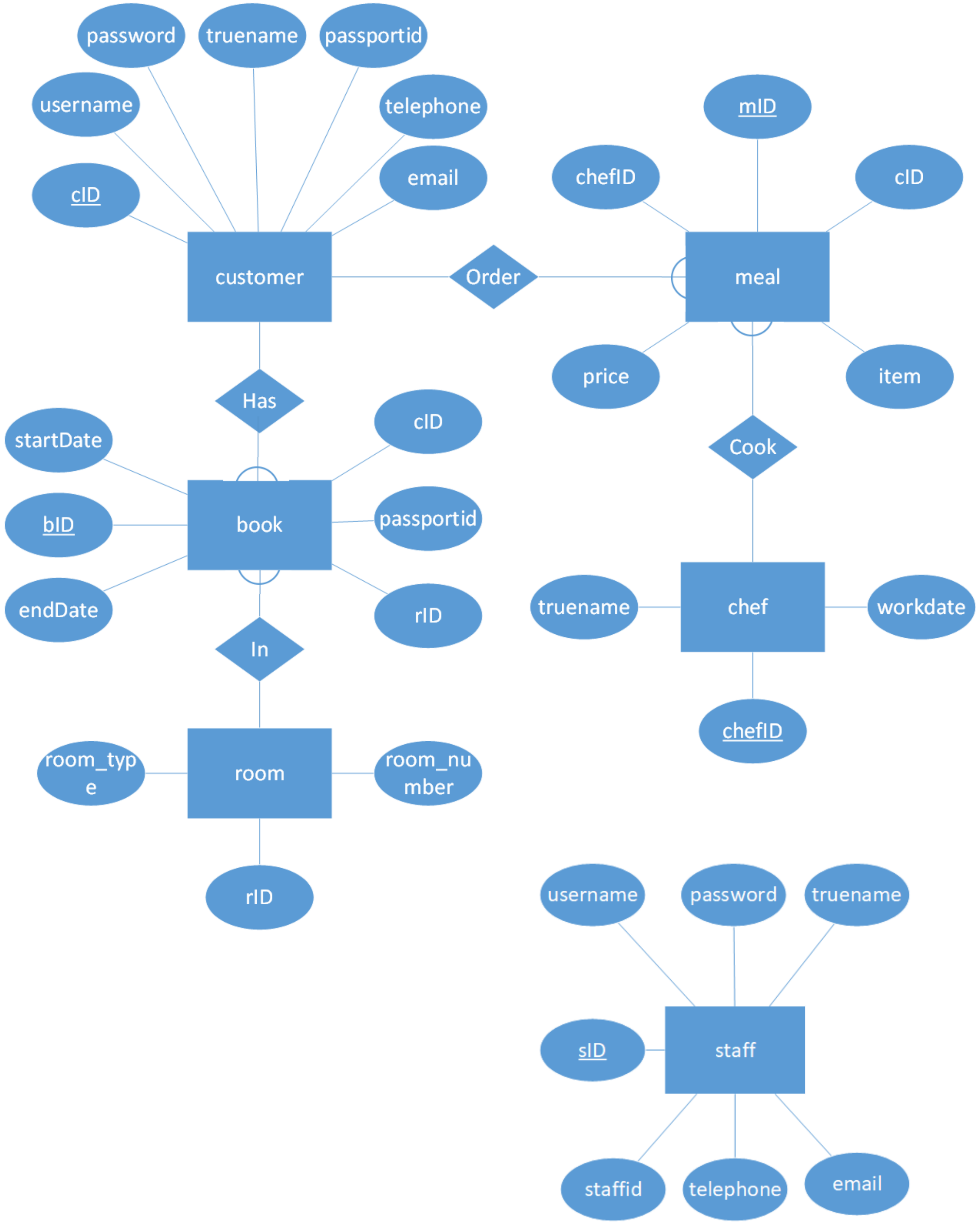
Coloum	Type	Null	Comment
bID	BIGINT	false	
startDate	DATE	false	
endDate	DATE	false	
status	INT	false	-1:invalid 0:booked 1:checked in
cID	BIGINT	false	customer ID / foreign key
rID	BIGINT	false	room ID / foreign key

3.5. ER Diagram for Current Database



3.6. Further Implementation

a. ER Diagram for Further Database



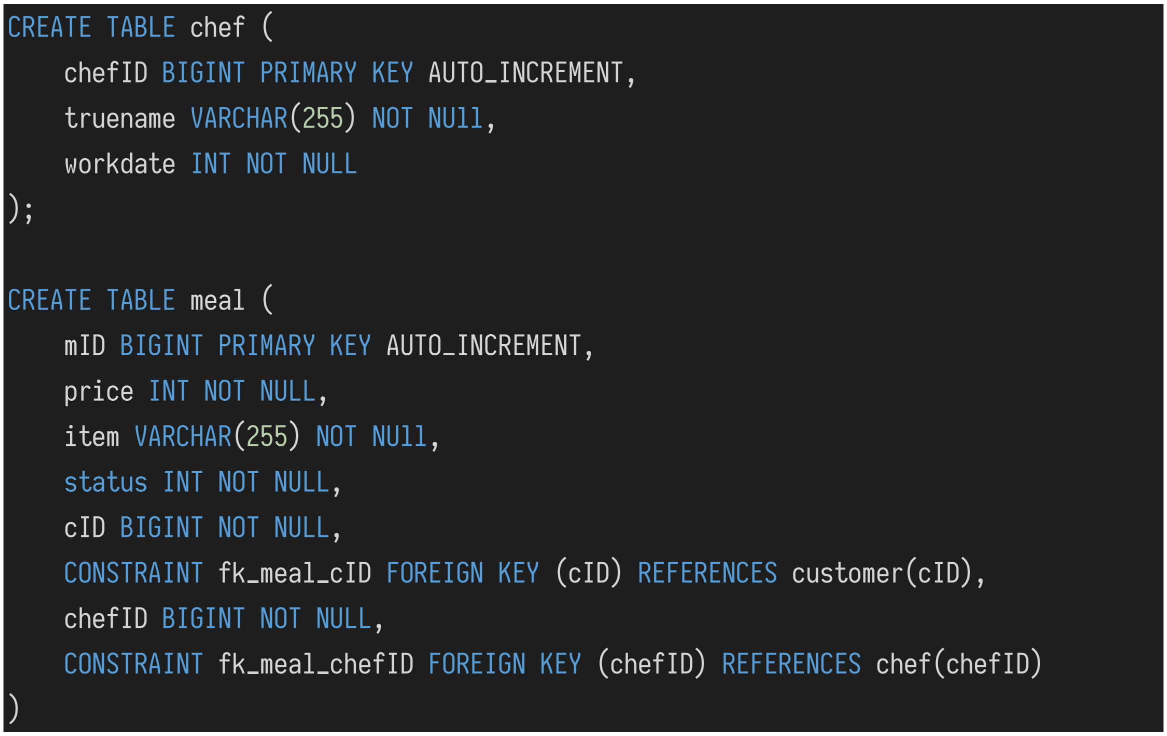
b. SQL statement

Core Function ~~strikethrough effect:~~

As only table ‘book’ is changed, so only statement of ‘book’ is offered in document.

```
CREATE TABLE book (
    bID BIGINT PRIMARY KEY AUTO_INCREMENT,
    startDate DATE NOT NULL,
    endDate DATE NOT NULL,
    status INT NOT NULL,
    passportid VARCHAR(25) NOT NULL,
    cID BIGINT,
    CONSTRAINT fk_cID FOREIGN KEY (cID) REFERENCES customer(cID),
    rID BIGINT NOT NULL,
    CONSTRAINT fk_rID FOREIGN KEY (rID) REFERENCES room(rID)
)
```

Additional Function ~~strikethrough effect:~~



```
CREATE TABLE chef (
    chefID BIGINT PRIMARY KEY AUTO_INCREMENT,
    truename VARCHAR(255) NOT NUll,
    workdate INT NOT NULL
);

CREATE TABLE meal (
    mID BIGINT PRIMARY KEY AUTO_INCREMENT,
    price INT NOT NULL,
    item VARCHAR(255) NOT NUll,
    status INT NOT NULL,
    cID BIGINT NOT NULL,
    CONSTRAINT fk_meal_cID FOREIGN KEY (cID) REFERENCES customer(cID),
    chefID BIGINT NOT NULL,
    CONSTRAINT fk_meal_chefID FOREIGN KEY (chefID) REFERENCES chef(chefID)
)
```

c. Explanation

Core Function ~~strikethrough effect:~~

I just add one key ‘passportid’ in ‘book’ table for non-login book and constrain on one man one book.

To let non-login user to submit a book, ‘cID' can be left blank and ‘passportid’ should have value.

To avoid one user, stay in multiple rooms, system will not allow user book more than one room unless user add extra passportid.

Staff table is already implemented in my coursework.

Additional Function ~~strikethrough effect:~~

For additional part, I added one table for chefs `**chef**` and one table for meal order `**meal**`.

For every meal order, the specific item selected by user, price and item will be saved, and a chef will be assigned, and the customer ID of the customer who orders the meal will also be recorded.

For every chef, they have an ID, truename, and workdate, workdate should be a number from 1-7 standing for Monday to Sunday every week.

4. Back-end Functions (API)

4.1. Sign up

- File
  /func/signup_post.php
- Request Method
  Post
- Description
  Handle sign up request, register a new user and set a login session

4.2. Sign in

- File
  /func/signin_post.php
- Request Method
  Post
- Description
  Handle sign in request and set a login session, will return false if duplicate username.

4.3. Sign out

- File
  /func/signout_post.php
- Description
  Handle sign in request and set a login session

4.4. Update book

- File
  /func/managebook_post.php
- Request Method
  Post
- Description
  Update status of current book

4.5. Check available rooms

- File
  /func/checkroom_get.php
- Request Method
  GET
- Description
  Return all available rooms which meet the requirement and not have been booked yet.

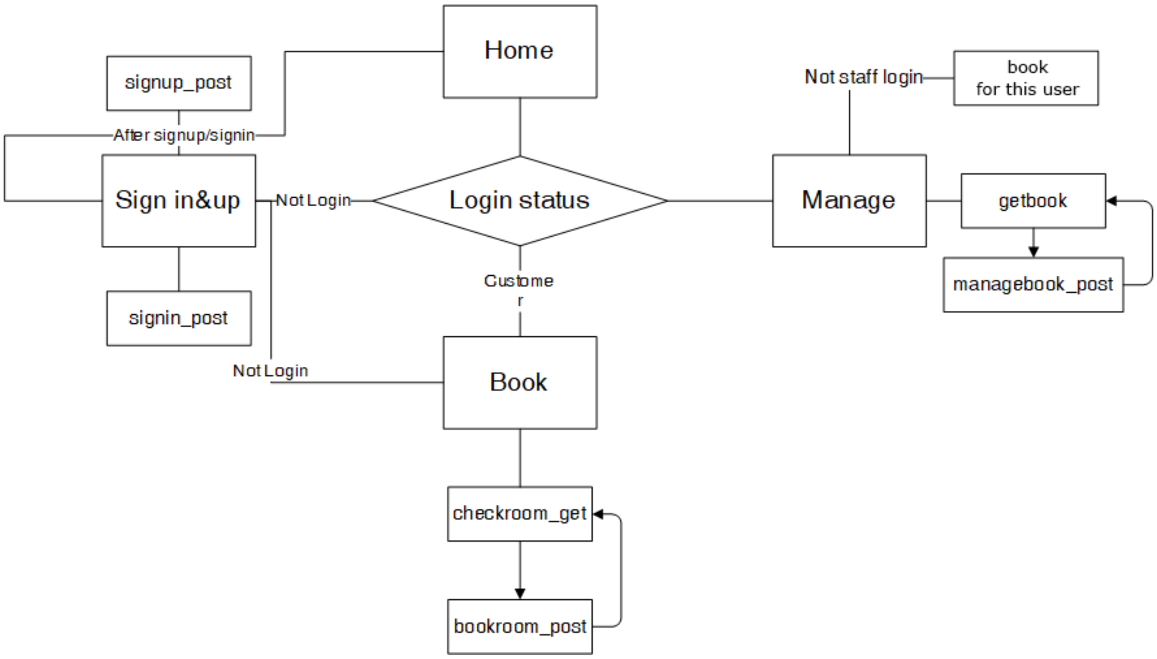
4.6. Book rooms

- File
  /func/bookroom_post.php
- Request Method
  POST
- Description
  Submit a book.

4.7. Get books

- File
  /func/getbook.php
- Description
  Get all valid book.

5. System Walkthrough



.