**旅游帮大学生旅游网**

高层设计说明书

|  |  |
| --- | --- |
| **DOC ID:** | NEIVMS\_HLD |
| **VERSION:** | V1.1 |
| **DATE:** | 2016年1月15日 |

**摘 要**: 从开发人员的角度，对旅游帮大学生旅游网

进行详细设计

关键字: 旅游帮大学生旅游网 详细设计

类图、时序图、界面和数据库

**旅游帮网站项目组**

**版本历史**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **版本号** | **日期** | **作者** | **修订原因** | **评审日期** | **评审人** |
| 1.0 | 2015年12月9日 | 齐敬佩  吴树智 | 初稿 |  |  |
| 1.1 | 2016年1月15日 | 齐敬佩  吴树智 | 加入DAO层的接口 |  |  |
| 2.0.1 |  |  |  |  |  |
| 2.0.3 |  |  |  |  |  |

**文档批准**

|  |  |  |  |
| --- | --- | --- | --- |
| **角色** | **人员** | **签字** | **日期** |
| 项目负责人 | 李海涛 |  |  |
| 课题负责人 | 李海涛 |  |  |
| 项目组长 | 齐敬佩 |  | 2016年1月15日 |
| 项目组成员 | 吴树智 |  | 2016年1月15日 |
|  |  |  |  |

目录

[1.导言 4](#_Toc440619692)

[1.1概述 4](#_Toc440619693)

[1.2目的 4](#_Toc440619694)

[1.3范围 4](#_Toc440619695)

[1.4 参考资料 4](#_Toc440619696)

[2.系统结构 4](#_Toc440619697)

[2.1功能描述 4](#_Toc440619698)

[2.2系统逻辑结构图 6](#_Toc440619699)

[3.模块设计 9](#_Toc440619700)

[3.1用户登录和注册 10](#_Toc440619701)

[3.2管理员登录 12](#_Toc440619702)

[3.3攻略处理 14](#_Toc440619703)

[3.4评论攻略 16](#_Toc440619704)

[3.5寻找同行驴友 17](#_Toc440619705)

[3.6申请成为同行驴友 19](#_Toc440619706)

[3.7封禁用户 20](#_Toc440619707)

[4.网页界面 22](#_Toc440619708)

[4.数据库操作层的接口设计 26](#_Toc440619709)

[4.1说明 26](#_Toc440619710)

[4.2接口定义 26](#_Toc440619711)

[4.3城市CityDaoImpl： 27](#_Toc440619712)

[4.4景点AttractionDaoImpl: 27](#_Toc440619713)

[4.5美食FoodDaoImpl: 27](#_Toc440619714)

[4.6 攻略Strategy DaoImpl:攻略有3种类型：城市、景点、美食 27](#_Toc440619715)

[4.7同行驴友PartnerDaoImpl 28](#_Toc440619716)

[4.8评论CommentDaoImpl 28](#_Toc440619717)

[5.数据库设计 29](#_Toc440619718)

[5.1 ER图 29](#_Toc440619719)

[5.2数据库表： 29](#_Toc440619720)

## 1.导言

### 1.1概述

本章对该文档的目的、功能范围、术语、相关文档及参考资料进行说明

### 1.2目的

本文档旨在使设计人员遵循统一的高层设计书写规范，降低项目实现的风险，以利于项目的实现、测试、维护和版本升级。

### 1.3范围

本文档用于软件设计阶段的概要设计，包括软件总体结构设计、构件设计、构件之间的接口设计、界面设计以及数据结构设计。

### 1.4 参考资料

## 2.系统结构

### 2.1功能描述

|  |  |
| --- | --- |
| 功能名称 | 简介 |
| **注册** | **浏览者必须通过邮箱注册，然后通过邮件进行激活** |
| **登录** | **用户登录后，可以获得一定的经验值并且获得更多的权利。** |
| **退出** | **未登录用户登录后可以退出** |
| **查询城市的旅游攻略** | **通过搜索进行城市旅游攻略的查看** |
| **查询城市下景点的攻略** | **景点拥有独立于城市的攻略，用户可以查看** |
| **查询城市下美食的攻略** | **美食也是城市的一大特色，本网站提供城市下美食的介绍** |
| **投诉和建议** | **在网站主界面留有管理员的电话及邮箱，用户可以联系管理员。** |
| **发表旅游攻略（登录后）** | **用户可以将自己的旅游攻略在网站发表，会获得经验值。旅游攻略可以包括图片、文字、美食、交通、花费介绍等** |
| **修改自己的旅游攻略（登录后）** | **用户如果觉得自己的攻略不合适可以进行修改，修改后攻略会显示最后修改时间** |
| **删除自己的旅游攻略（登录后）** | **用户如果觉得自己的攻略不合适可以进行删除** |
| **评论攻略（登录后）** | **用户对攻略进行评论** |
| **查看他人部分信息（登录后）** | **用户可以查看其他用户的昵称、性别、联系方式、生日、等级、信用度** |
| **查看自己的用户中心（登录后）** | **用户中心包含用户的昵称、等级、性别、年龄、联系方式、生日、信用度、系统消息、用户消息** |
| **修改自己的个人信息（登录后）** | **可以对自己的性别、年龄、联系方式、生日，登录密码等信息进行修改** |
| **发布招募“驴友”的信息（登录后）** | **用户可以招募旅行目的地和旅行时间相近的用户作为“驴友”，发布者在有用户申请成为“驴友”时，可以点击“同意”按钮来成为“同行驴友”，发布者在出发前也可以取消成为“同行驴友”** |
| **申请成为“驴友”（登录后）** | **用户可以通过本网站申请成为发布招募“驴友”的用户的“同行驴友”** |
| **查看申请成为“驴友的人的信息”（登录后）** | **发布者可以查看申请者的昵称、等级、信用度、联系方式（可以是重新补充的）等信息** |
| **评价“驴友”（成为“驴友”并且登录后）** | **用户可以对自己的“驴友”进行文字评价并且为驴友的信用度打分** |
| **增加城市信息（管理员）** | **如果获得了未在数据库的城市信息，管理员就可以将此城市的信息添加到数据库中** |
| **修改城市信息（管理员）** | **如果已在数据库的城市的信息不正确，管理员就可以将此城市的错误信息进行修改** |
| **删除城市信息（管理员）** | **如果已在数据库的城市不具有景点的价值，管理员就可以将此城市的全部信息进行删除** |
| **增加城市下景点的信息（管理员）** | **如果获得了未在数据库的景点信息，管理员就可以将此景点添加到数据库中** |
| **修改城市下景点的信息（管理员）** | **如果已在数据库的景点的信息不正确，管理员就可以将此景点的错误信息进行修改** |
| **删除城市下景点的信息（管理员）** | **如果已在数据库的景点不具有景点的价值，管理员就可以将此景点的全部信息进行删除** |
| **修改攻略（管理员）** | **管理员可以修改网站私有的攻略，并且可以在用户同意的前提下修改用户的攻略** |
| **删除攻略（管理员）** | **如果管理员认为用户的攻略触犯法律或者对其他用户造成不利的影响，可以通过留言警告该用户并且删除该攻略** |
| **设置成精华攻略（管理员）** | **如果管理员认为用户的攻略和网站的要求相符，就可以将该攻略设为“精华攻略”，并给予用户一定的经验值** |
| **删除评论（管理员）** | **如果管理员认为某些评论触犯了法律或者对其他用户造成不利的影响，管理员可以删除该评论并且警告用户，管理员没有修改用户评论的权利** |
| **给用户留言（管理员）** | **管理员与用户联系的渠道，可以警告用户或者提醒用户** |
| **增加美食攻略（管理员）** | **如果获得了未在数据库美食的攻略，管理员就可以将此美食添加到数据库中** |
| **修改美食攻略（管理员）** | **如果已在数据库的美食的信息不正确，管理员就可以将此美食的错误信息进行修改** |
| **删除美食攻略（管理员）** | **如果已在数据库的美食攻略已不具有参考价值，管理员就可以将此美食的全部信息进行删除** |
| **删除“同行驴友”招募信息（管理员）** | **如果管理员认为招募信息触犯了法律，管理员可以删除此招募信息并且警告该用户** |
| **发布系统公告（管理员）** | **发布重要的信息，包括系统维护、网站升级、错误说明** |
| **删除系统公告（管理员）** | **如果管理员认为系统公告不正确，就可以删除此系统公告** |
| **解封用户（管理员）** | **如果用户被盗号，被他人恶意利用后，被管理员封禁此号，若经过管理员同意，可由管理员解封账号。** |
| **封禁用户（管理员）** | **如果管理员认为用户在网站内的行为触犯了法律或者对其他用户造成不利的影响，则管理员可以封禁该用户** |
| **备份数据库（数据库管理员）** | **系统管理员可以每过一段时间将数据库备份，以防数据丢失** |
| **恢复数据库（数据库管理员）** | **如果数据库中的数据出现丢失、出错的情况，系统管理员可以恢复数据库** |

### 2.2系统逻辑结构图

#### 2.2.1城市管理相关类



#### 2.2.2景点管理相关类



#### 2.2.3美食管理相关类



#### 2.2.4城市查询时序图



#### 2.2.5删除城市时序图



#### 2.2.6修改美食时序图



## 3.模块设计

如果类图中的Dao层和DaoImpl(数据库操作类接口类)有不同的地方，优先按照接口类中的方法声明来创建方法。

### 3.1用户登录和注册





#### 3.3.1任务描述

登录和注册功能提供登录和注册的必需界面，并且提供给用户登录、注册、找回密码、修改密码等基本功能。

#### 3.3.2输入

用户输入登录信息（或通过用户名选项的下拉菜单选取已登录且已保存的用户名），可选择是否保存自己的用户名

#### 3.3.3输出

成功进入主界面或提示用户名密码不匹配

### 3.2管理员登录





#### 3.2.1任务描述

登录和注册功能提供登录和注册的必需界面，并且提供给管理员登录、注册、找回密码、修改密码等基本功能。

#### 3.2.2输入

管理员输入登录信息（或通过用户名选项的下拉菜单选取已登录且已保存的用户名），可选择是否保存自己的用户名

#### 3.2.3输出

成功进入主界面或提示用户名密码不匹配

### 3.3攻略处理



##### 发表一篇城市的攻略的时序图



#### 3.3.1任务描述

攻略处理功能提供用户搜索的界面，以此进入攻略界面，并且提供给用户发布攻略、修改自己的攻略、删除自己的攻略等功能。用户在发布攻略的时候可以自由选择发布城市攻略、景点攻略、美食攻略任意一项。在编写攻略期间可以进行多次保存。

#### 3.3.2输入

用户可以输入攻略的内容，输入完毕后点击保存按钮

#### 3.3.3输出

攻略的发布者，发布时间以及攻略的内容

### 3.4评论攻略



#### 3.4.1任务描述

用户可以进入攻略界面，自由选择城市攻略、景点攻略和美食攻略任一个进行评论，也可以对自己的评论进行删除，但是不允许修改自己的评论，允许用户多次评论同一攻略

#### 3.4.2输入

评论的内容

#### 3.4.3输出

当用户点击保存按钮时，提示用户评论成功或评论失败。

### 3.5寻找同行驴友



##### 发布招募驴友



#### 3.5.1任务描述

寻找同行驴友功能提供给用户发布申请、取消申请、编辑自己的信息、寻找同行信息等功能

#### 3.5.2输入

发布者的出发时间、出发地点、结束时间、目的地、招募的人数、联系方式、自己的性别、对申请者的要求等

#### 3.5.3输出

当用户点击确定按钮时，网页提示用户申请成功或申请失败，如果申请失败则会保存用户填写的信息和错误原因进行回显。

### 3.6申请成为同行驴友



#### 3.6.1任务描述

申请成为同行驴友功能提供给用户申请成为他人驴友、取消申请成为他人驴友以及编辑自己的信息的功能

#### 3.6.2输入

申请者的性别、参加的人数、联系方式、自己的要求等

#### 3.6.3输出

当用户点击确定按钮时，网页提示用户申请成功或申请失败

### 3.7封禁用户





#### 3.7.1任务描述

封禁用户功能提供给管理员可以封禁某些行为不当的用户，管理员也可以解禁某封禁用户

#### 3.7.2输入

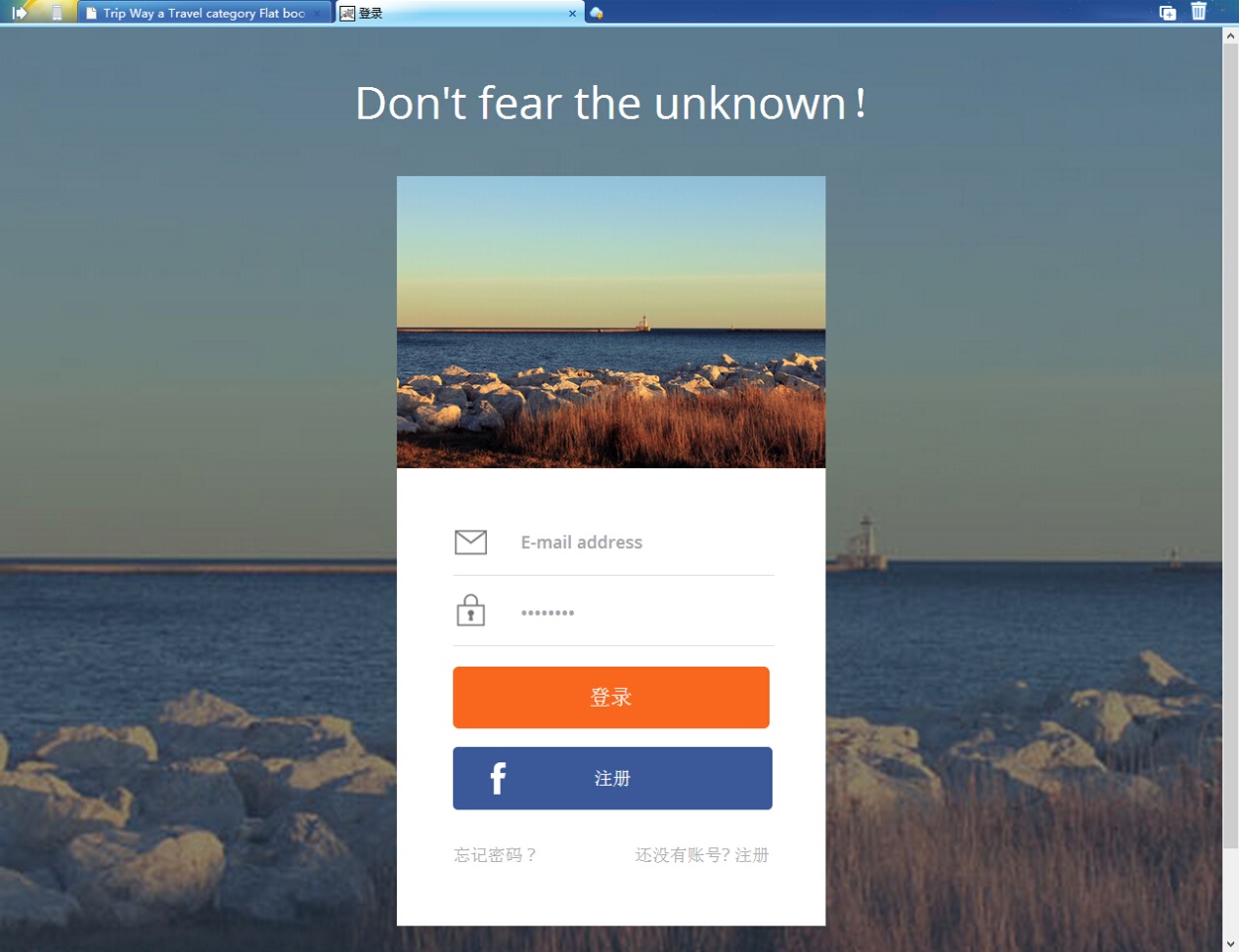
管理员输入要封禁（解禁）用户的id

#### 3.7.3输出

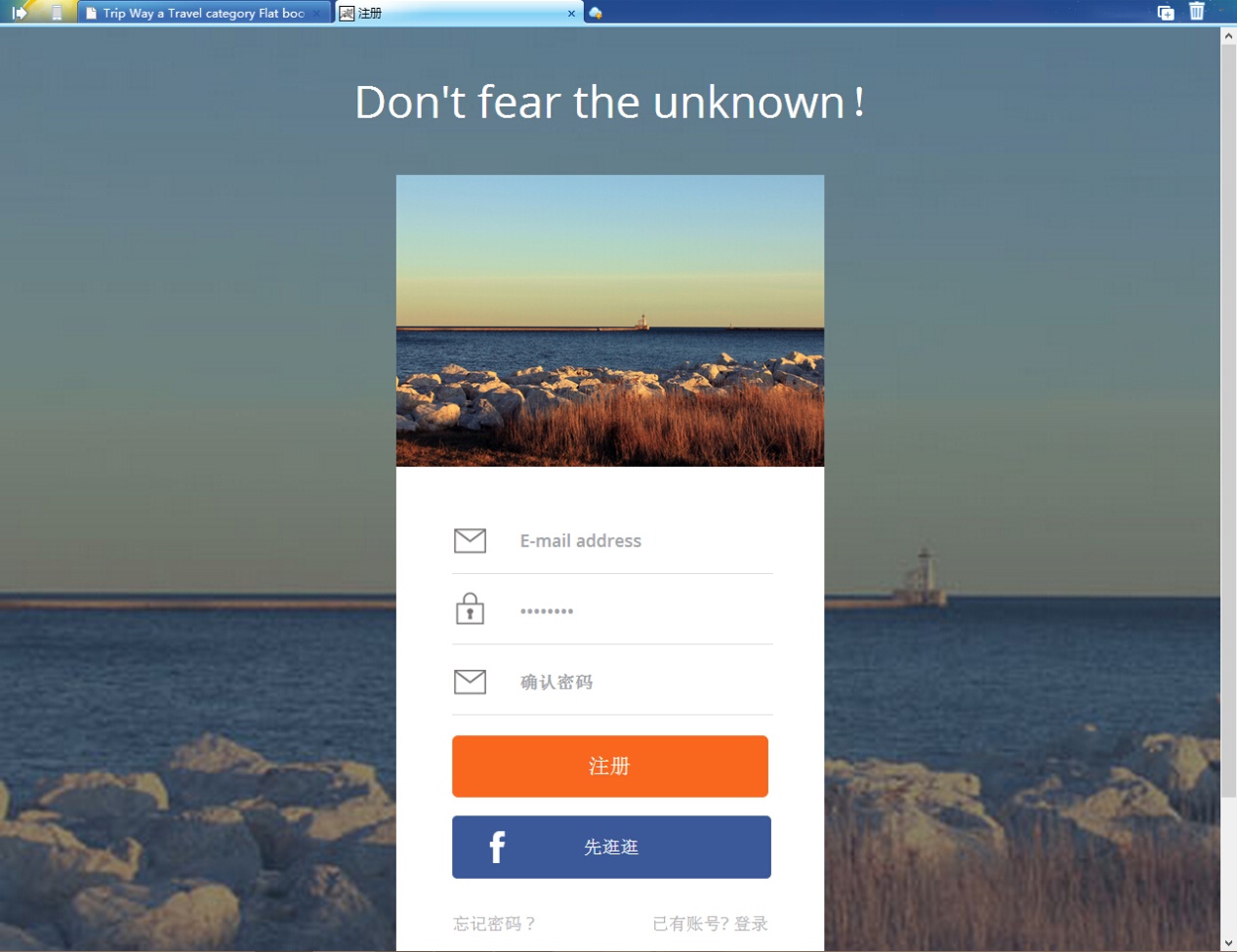
当管理员点击确定按钮时，网页会提示封禁（解禁）成功或封禁（解禁）失败

### 4.网页界面

#### 4.1登录界面



#### 4.2注册界面



#### 4.3首页









#### 4.4更多页面请看“系统原型图”

## 4.数据库操作层的接口设计

### 4.1说明

创建多个接口，让数据库操作类（Dao层的各个类）实现对应的接口。这样可以实现更好的分工合作，并且在更换数据库时，只需要更换数据库操作类就可以了。

如果类图中的Dao层和DaoImpl(数据库操作类接口类)有不同的地方，优先按照接口类中的方法声明来创建方法。

### 4.2接口定义

1.用户UserDaoImpl:

findByUsername(String name):User//按照用户名查找用户

findByUserid(String id):User//按照用户id查找用户

changePassword():void //修改密码

edit():void//修改用户信息

add():void//添加新用户

ban():void//封禁用户

unban():void//解禁用户

### 4.3城市CityDaoImpl：

findByName(String name):String

findAll():String//查询所有城市，返回一个City对象的List数组

add():String

delete():String

edit ():String

### 4.4景点AttractionDaoImpl:

findByName():String

FindAll():String

add():String

delete():String

edit ():String

findByAttid():Strategy//查询某个城市下的景点

### 4.5美食FoodDaoImpl:

findAll():List<Food>//查询所有美食

add():void

delete():void

edit():void

findById():Food

findByCItyid():List<Food>//查询某个城市下的美食

### 4.6 攻略Strategy DaoImpl:攻略有3种类型：城市、景点、美食

findAll():List<Strategy>

findByStrid():Strategy

findByName():Strategy

findByid():Strategy

findByType()://按照类型查询攻略

add():void

edit():void

levelup():void//设置成精华攻略

### 4.7同行驴友PartnerDaoImpl

findAll()

add()

findByCityid

applyFor()//申请同行

delete

edit

### 4.8评论CommentDaoImpl

findAll()

findByComid()//加载单个评论

findByUserid()//按照用户id查询评论

add()

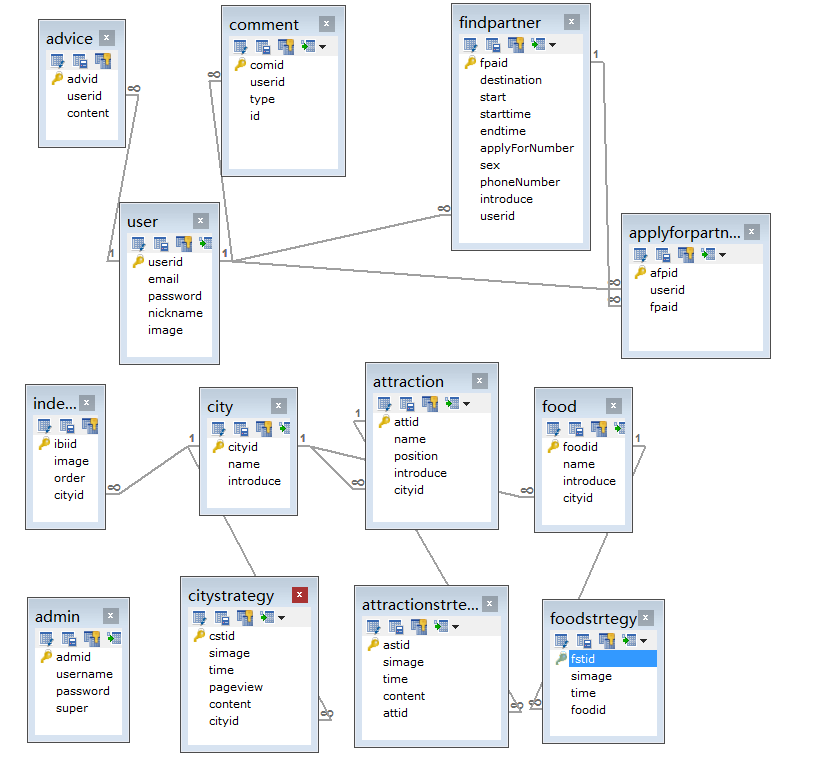
delete()

edit()

findByStrid()//查出是对哪一篇攻略的评论

## 5.数据库设计

### 5.1 ER图



### 5.2数据库表：

总原则：按照项目现实意义来命名，见名知意

属性命名：用英文单词，尽量用全称。

主键命名：在每个表的主键编号id上，如果字母数量超过6个，就进行缩写，缩写成5个字母，如strategyid缩写成：srtid。

外键命名：和被引用的主键名字相同 。

大小写原则：首个单词小写，后面的单词首字母大写。

CREATE DATABASE Travel;

-- drop database Travel;

-- show tables;

USE Travel;

CREATE TABLE `user` -- 用户表（加上俩点是为了跟关键字区别开来）

(

userid CHAR(32) PRIMARY KEY,-- 用户编号

email CHAR(50), -- 邮箱,用邮箱登录

`password` CHAR(32),-- 密码

nickname CHAR(32), -- 昵称

image VARCHAR(200)-- 头像

);

CREATE TABLE city-- 城市

(

cityid CHAR(32) PRIMARY KEY,-- 城市id

`name` VARCHAR(100), -- 城市名

introduce VARCHAR(500) -- 介绍

);

CREATE TABLE cityStrategy -- 城市攻略 注意：把城市攻略保存成一个html文件，用户浏览时直接把这个文件内嵌在网页中

(

cstid CHAR(32) PRIMARY KEY, -- id

simage VARCHAR(200), -- 小封面图片

`time` DATETIME ,-- 发表时间

pageview INT,-- 浏览量

content VARCHAR(200), -- 保存的html的位置

cityid CHAR(32),-- 所属城市的id

FOREIGN KEY (cityid) REFERENCES city(cityid) -- 外键

);

CREATE TABLE indexBigImage-- 首页大图

(

ibiid CHAR(32) PRIMARY KEY, -- 编号

image VARCHAR(200) ,-- 保存位置

`order` INT UNIQUE AUTO\_INCREMENT, -- 序号，如每次只读取序号最大的4张图片（自增的属性必须是主键或者unique）

cityid CHAR(32),-- 所指向的城市的id号

FOREIGN KEY (cityid) REFERENCES city(cityid) -- 外键

);

CREATE TABLE findPartner -- 寻找同行驴友

(

fpaid CHAR(32) PRIMARY KEY,-- id

destination VARCHAR(100), -- 目的地

`start` VARCHAR(100), -- 出发地

starttime DATETIME, -- 出发时间

endtime DATETIME, -- 结束时间

applyForNumber INT, -- 已申请人数（这在做起来有点小难）

sex BOOLEAN, -- 性别是否为男性(0False为女，1True为男)

phoneNumber CHAR(32),-- 联系方式（手机号、微信、qq都可以）

introduce VARCHAR(2000), -- 同行介绍

userid CHAR(32), -- 外键引用的userid

FOREIGN KEY (userid) REFERENCES `user`(userid)

);

-- 以上数据库为要用到的，下面的数据库为暂时不用的--

CREATE TABLE applyForPartner -- 申请同行

(

afpid CHAR(32) PRIMARY KEY ,-- 申请同行id

phoneNumber char(32) ,-- 联系方式

number INT,--同行人数

sex BOOLEAN ,-- 性别

remark VAECHAR(500), -- 备注

userid CHAR(32) ,-- 申请用户的id

fpaid CHAR(32), -- 同行驴友招募的id

FOREIGN KEY (userid) REFERENCES `user`(userid),

FOREIGN KEY (fpaid) REFERENCES `findPartner`(fpaid)

);

CREATE TABLE attraction-- 景点

(

attid CHAR(32) PRIMARY KEY,-- 景点id

`name` VARCHAR(100) , -- 景点名

`position` VARCHAR(200),-- 景点位置

introduce VARCHAR(500), -- 介绍

cityid CHAR(32) ,-- 所属城市

FOREIGN KEY (cityid) REFERENCES city(cityid)-- 外键声明

);

-- select \* from attraction

CREATE TABLE attractionStrategy -- 景点攻略

(

astid CHAR(32) PRIMARY KEY, -- id

simage VARCHAR(200), -- 小封面图片

`time` DATETIME ,-- 发表时间

content VARCHAR(200), -- 保存的html的位置

attid CHAR(32),-- 所属景点的id

FOREIGN KEY (attid) REFERENCES attraction(attid) -- 外键

);

CREATE TABLE food-- 美食

(

foodid CHAR(32) PRIMARY KEY,-- 美食id

`name` VARCHAR(100) , -- 美食名字

introduce VARCHAR(500), -- 介绍

cityid CHAR(32) ,-- 所属城市

FOREIGN KEY (cityid) REFERENCES city(cityid)-- 外键声明

);

CREATE TABLE foodStrategy -- 美食攻略

(

fstid CHAR(32) PRIMARY KEY, -- id

simage VARCHAR(200), -- 小封面图片

`time` DATETIME ,-- 发表时间

foodid CHAR(32),-- 所属的美食

FOREIGN KEY (foodid) REFERENCES food(foodid) -- 外键

);

CREATE TABLE `comment`-- 评论（只有评论，没有回复）

(

comid CHAR(32) PRIMARY KEY,-- 评论id

userid CHAR(32) ,-- 引用的用户id，是这个用户发的评论

`type` CHAR(32) ,-- 评论类型，对城市的评论为c，对景点的评论为a，对美食的评论为f

id CHAR(32),-- 指向的攻略的id，因为攻略有3种，所以这里不添加外键声明了

FOREIGN KEY (userid) REFERENCES `user`(userid)

);

CREATE TABLE advice-- 投诉及建议

(

advid CHAR(32) PRIMARY KEY,-- id

userid CHAR(32), -- 用户id

content VARCHAR(1000), -- 投诉或者建议内容

FOREIGN KEY (userid) REFERENCES `user`(userid)

);

CREATE TABLE admin -- 管理员表，用于管理员登录

(

admid CHAR(32) PRIMARY KEY,-- id

username CHAR(32) NOT NULL,-- 登录名称

`password` CHAR(32) NOT NULL, -- 密码

`super` BOOLEAN -- 是否为超级管理员，超级管理员可以对普通管理员进行管理，

);