# 数据库设计文档

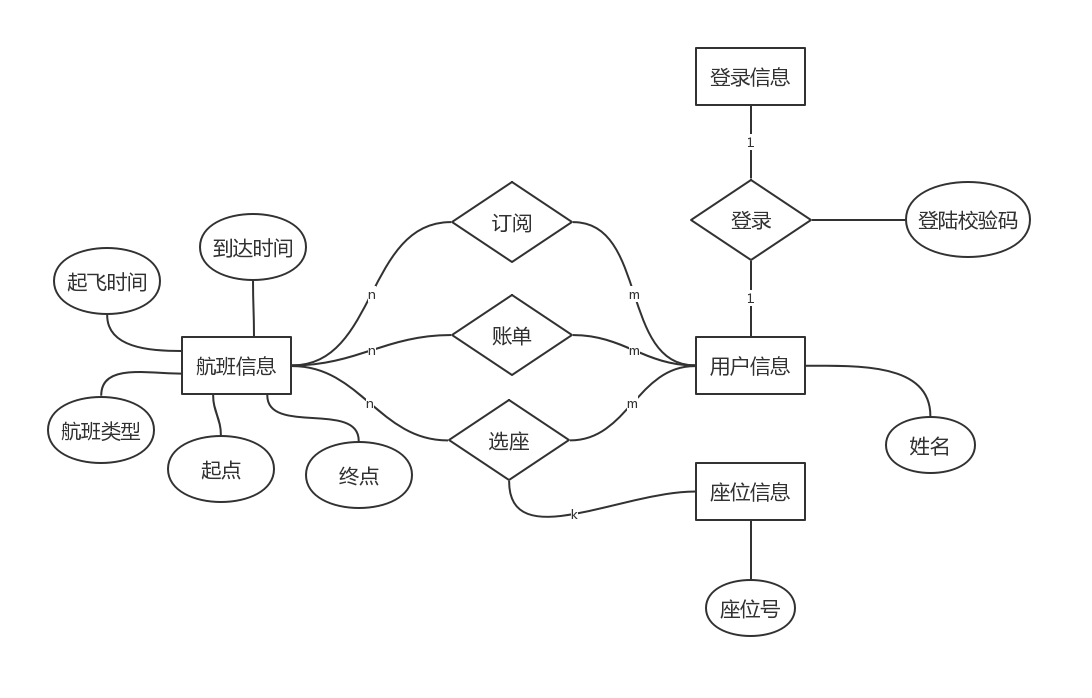
## 概要设计

航班订票系统首先需要分为两部分，第一部分是航空公司管理部分，在航空公司视角，能够录入每日的航班信息，能够修改已经加入的航班信息（如当前票价），能够统计今日的收入，能够统计某一阶段的收入，能够生成订单流水，能够查询某一航班的上座率等功能。第二部分是用户订阅部分，在用户视角，用户能够搜索和查询相应的航班信息，只需要输入时间和地点就可以获得所有航班列表，用户可以点击相应的航班列表进入选座状态，当用户选择了一个座位后并输入用户的uid后，进入交款界面（这说明订票人和乘坐人可以不一致），机票价格是实时变动的，所以每个用户购买的票价可能会出现不一样的情况。当用户缴纳费用后，可以打印账单作为凭证。最后用户可以选择打印取票通知，取票通知可以不用当前打印，也可以在下次登陆时系统给出打印提示。当用户没有打印取票通知时是可以进行退票操作的。

特别说明，账单和通知单应当生成PDF格式的文件，而航空公司收入统计信息应当生成XLS或者CSV格式的文件。

## 数据表设计

根据概要分析，我们做出如图的ER图：



首先是用户基本信息表，该表用来记录用户的uid、用户的姓名、用户总消费金额，理论上还需要用户的密码，但是本系统是一个演示系统就不需要输入密码了，直接输入用户uid即可进行登录。用户基本信息表的SQL语句如下：

create table user\_info(

uid int,

name varchar(16),

total\_cost decimal(10,2),

PRIMARY KEY(uid),

INDEX idx\_uid(uid)

) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

其中uid作为主键，并为uid建立索引。

其次是航班基本信息表，航班基本信息包括航班的fid，航班的类型，起点，终点，起飞时间，到达时间，是否包含食物，座位总数，当前机票的价格。

create table flight\_info(

fid int,

ftype varchar(63),

city\_from varchar(63),

city\_to varchar(63),

time\_begin datetime,

time\_end datetime,

food bool,

sitenum int NOT NULL,

income decimal(8,2)

PRIMARY KEY(fid),

INDEX idx\_fid (fid)

)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

其中fid为主键，并为主键添加索引。

接下来是所有的关系表，第一个关系表是用户登录表，理论上用户正确输入uid和密码并点击确定，系统校验正确后会返回一个正确的信号。根据常识，系统是不能够存储用户的明文密码的，因为这会带来很多安全问题（尽管演示系统根本就没有设置密码），所以当用户姓名密码校验成功后，系统会返回一个校验token，系统发送请求时带上token就会唯一确定一个用户。token在一定的时间内会自动删除，这样可以一定程度上避免信息泄露。虽然理论上是这么说，但是在本系统实验上，除非用户手动点击退出按钮，否则token不会自动删除。后续可以考虑增加触发器来自动删除过期的token。

create table user\_login(

ulid int AUTO\_INCREMENT,

uid int,

token varchar(63),

PRIMARY KEY(ulid),

INDEX idx\_uid(uid),

INDEX idx\_token(token)

)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

其中主键是ulid，理论上(uid,token)就可以作为主键，但是因为Mysql不能够将这两个作为主键，所以就再建立一个字段ulid作为主键。并且为uid何token分别建立索引，这样能够知道一个后可以快速得到另一个。

第二张关系表是选座信息表，选座信息表包括fsid，航班fid，用户uid，座位号。当用户输入uid和航班fid、座位号siteid后，就可以唯一确定某个航班的某个座位被某个人坐了。

create table flight\_site(

fsid int AUTO\_INCREMENT,

fid int,

uid int,

siteid int,

UNIQUE(fid,uid,siteid),

PRIMARY KEY(fsid),

FOREIGN KEY(fid) REFERENCES flight\_info(fid) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY(uid) REFERENCES user\_info(uid) ON DELETE SET NULL ON UPDATE CASCADE,

INDEX idx\_fuid(fid,uid,siteid)

)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

其中，fsid是因为不能够将三个外键作为候选码，所以索性就额外建立一个字段作为主键。但是座位号还是需要增加唯一性约束，所以必须建立一个额外的唯一性，所以有UNIQUE(fid,uid,siteid)。fid和uid作为外键约束并增加参照。最后建立fid,uid,siteid作为索引。

第三个关系是取票信息表，包括id，用户uid，航班fid，取票时间，起飞时间，备注信息以及是否退票。可能会有疑问，起飞时间在航班信息表中已经有了，这里为什么还要再增加一个起飞时间的字段呢？这是因为航班信息的取票时间是售票时预估的时间，而取票上的起飞时间有可能是延误了后真实的起飞时间。cancel字段是为了控制用户是否退票。退票时将所有有关的记录从数据库中删除明显不合理，这是因为以后的一些校验“证据”是需要保留的，所以这里增加一个cancel字段判断是否被退票而不是直接删除这条记录。

create table take\_ticket\_info(

id int AUTO\_INCREMENT,

bookid int,

uid int,

fid int,

time datetime,

offtime datetime,

extra varchar(255),

cancel tinyint(4),

PRIMARY KEY(id),

FOREIGN KEY(uid) REFERENCES user\_info(uid) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY(fid) REFERENCES flight\_info(fid) ON DELETE SET NULL ON UPDATE CASCADE,

INDEX idx\_id(id),

INDEX idx\_fuid(uid,fid),

UNIQUE(uid,fid)

)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

其中id作为主键，uid和fid是参照，并为id，(uid,fid)建立索引。因为一个用户只能领取唯一的一个通知单，所以还需要增加(uid,fid)的唯一性约束。

第四个关系是账单信息表，账单信息包括miid，用户uid，航班fid，交款金额，找零，实际交付，交款时间，取消字段。其中取消字段的原因和通知单中的取消操作的原因是一样的，为了留下存档。

create table money\_info(

miid int AUTO\_INCREMENT,

uid int,

fid int,

money\_in decimal(8,2),

money\_out decimal(8,2),

money decimal(8,2),

time datetime,

cancel tinyint(4),

UNIQUE(uid,fid),

PRIMARY KEY(miid),

FOREIGN KEY(uid) REFERENCES user\_info(uid) ON DELETE SET NULL ON UPDATE CASCADE,

FOREIGN KEY(fid) REFERENCES flight\_info(fid) ON DELETE SET NULL ON UPDATE CASCADE,

INDEX idx\_fuid(uid,fid)

)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

其中miid为主键，uid和fid是唯一性索引，一个用户只能为一个航班信息进行交款，uid和fid是参照，然后为(uid,fid)建立索引。

## API接口设计

因为本次实验采用浏览器-服务器模式，采用前后端交互的方式实现，而不是采用客户端-服务器的方式，那么前后端交互的途径就是API接口，通信方式采用JSON，使用的协议是HTTP协议。因为前后端交互的接口比较多，而且目前还处于不断修改的状态，所以这里只列举出一些典型的接口。

登录请求：

POST /login

request:{

uid:256 //发送用户的uid

}

response:{

code:0, //服务器正确处理校验代码

token: "123" //返回的用户凭证token

}

获得航班信息：

GET /flight\_info

request:{

token: "256",

time\_begin:20150302153221, //2015年3月2日15 32分

time\_end:20150320

from:"武汉",

to:"北京",

}

response:{

code: 0,

flightlist:[

{

fid:123,

ftype:"CT2015",

from:"武汉",

to:"北京",

begin:'20150302153221',

end:'20150303153221',

food: true,

sitenum: 25, //座位数

income: 340.56,

site\_not\_ok\_list:[1,2,3,4,5]

},

{},

{}

]

}

增加航班信息：

POST /post\_flight\_info

request:{

token: “256”,

ftype: "123",

from:" beijing",

to: "shanghai",

begin: "20160302",

end: " 20160316", //yyyymmddhhmmss

food: true, //是否有食物

sitenum : 25, //座位数

income: 802.5 //票价

}

response:{

code:0,

fid: 256

}

修改航班信息：

PUT /put\_flight\_info

request:{

token: 256,

fid: 123,

ftype: "CG1205", //optional

from:" beijing",

to: "shanghai",

begin: "20160302", //must

end: " 20166356", //must

food: true,

sitenum : 25,

income: 802.5

}

response:{

code:0

}

获取座位订阅信息：

GET /get\_book\_info

request:{

token: 569,

time: '201502030000', //预定时间,可选

fid\_list: [1,2,3]

}

response:{

code:0,

book\_info\_list:[

{

bookid: 123,

book\_uid:123,

uid:234,

fid:236,

time: '201603052130',

income: 340.56,

extra: "beizhu"

siteid: 25,

},

{}

]

}

订阅座位并缴费：

POST /book

request:{

token: 256,

book\_uid: 123,

uid: 256,

fid: 123,

time: "201503060203",

money\_in: 256, //交款金额

money\_out: 444, //找零

cost: 256.35, //实际交款

extra: "26356",

siteid: 25

}

response:{

code:0,

bookid:236

}

取消订单：

DELETE /book

request:{

token:569,

bookid: 256,

book\_uid: 236 //optional

uid: 263, //optional

}

response:{

code: 0,

money\_reply:256.32 (reback money, reback site, reback take\_ticket\_info)

}

获得取票信息打印：

GET /ticket\_info

request:{

token:985,

uid: 25, //optional

fid:658

}

response:{

code:0,

id: 256,

uid:256,

name:"xxx",

time: "201502320506",

fid:256,

ftype:"CF2563",

city\_from:"beijing",

city\_to:"shanghai",

offtime: "201302050603",

extra: "xxx",

siteid: 25

}

重新选择座位：

PUT /change\_site

request:{

token:598,

fid: 256,

uid: 489,

siteid: 5

}

response:{

code:0,

fid: 569,

uid: 256

}

获得账单信息：

GET /money\_info

request:{

token:256,

uid: 256, //optional

fid: 256

}

response:{

mmid:256,

book\_uid: 256,

uid: 256,

book\_name: "xxx",

name: "xx",

fid: 256,

time: "201503020405"

ftype: 653,

offtime: "201503020605",

money\_in: 256.32,

money\_out: 10.32,

money:246.00,

}

## 客户端界面设计

暂时未开始设计。