《数据库系统原理》课程设计

系统实现报告

题目名称： 简单的飞机订票系统

学号及姓名：\_\_\_\_\_\_\_\_\_14061003刘重伟\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_14061015张文镔\_\_\_\_\_\_\_\_\_\_\_

2016年 11 月 23 日

**一．系统功能需求分析**

基于Use-Case方法分析系统功能需求。

## 1.1系统用例

图表 1‑1系统用例图

此用例图将需要实现的基本功能展示出来，并将参与者与这些功能的联系以箭头的形式表示出来。展示了用例之间以及同用例参与者之间是怎样相互联系的。

## 1.2系统用户

客户：在本网站订票的乘客

## 1.3注册登录用例

前置条件：用户登录网站

基本路径：

1.系统显示主界面

2.用户点击注册或登录，用例开始

3.系统显示注册或登录界面

4.用户填写信息

5.系统进行验证信息

6.注册或登陆成功

7.若为注册则直接登录，若为登录，直接跳转至查询机票页面

扩展路径：若信息不符合要求，弹出错误提示

后置条件：如果用户没有中途取消注册，则将用户信息保留在数据库中。

## 1.4查票用例

前置条件：用户已经登录系统

基本路径：

1.用户点击机票查询

2.系统转到机票查询界面

3.用户填写出发地、目的地、出发时间和航空公司

扩展路径：

1.若用户输入信息不完善，则给出提示。

2.旅行信息查询：

(1)用户点击点击查询结果列表中的目的地。

(2)查询页面左侧会显示目的地城市的相关景点，酒店等。

3.航空公司查询：

(1)用户点击查询结果列表中的航空公司。

(2)页面跳转到航空公司界面，在这个界面您能够对该公司进行评价或者查看其他用户对该公司的评价。

4.查询结果排序：

(1)查询结果默认按起飞时间排序。

(2)用户点击降落时间，查询结果按照降落时间升序排列。

(3)用户点击准点率，查询结果按照准点率降序排列。

(4)用户点击票价，查询结果按照票价升序排列。

(2)用户点击起飞时间，查询结果按照起飞时间升序排列。

后置条件：若信息完善，系统显示查询结果。

## 1.5订票用例

前置条件：用户查询机票后系统给出候选结果

基本路径：

1.用户选择一个结果点击预订

2.界面跳转到机票订单页面

3.用户选择订票数量，订票数量需要与订票人数量一致

4.用户选择订票人，用户可根据需要选择为自己订票或者为自己的好友订票

5.界面跳转到选座界面，用户可以根据需要选择和订票数量数目一致的飞机座位。

后置条件：系统记录订票结果。

## 1.6订单管理用例

前置条件：用户转到订单界面

基本路径：

1.用户填写订单号、订票日期、乘机人

2.用户选择订单状态进行查询，订单状态包括已退票，已出票、已取消、已申请、待提交，当用户完成支付后订单状态由待提交变为已申请。

3.界面显示历史订单的查询结果。

4.用户选择需要进行操作的订单。

5.退票：

(1)用户点击退票选项

(2)系统提示“确认退票？”

(3)用户选择“确认”或“取消”

6.订单信息导出：

(1)用户单击导出订单选项

(2)界面弹出保存路径选项，用户填写保存路径。

(3)用户单击确定。

(4)订单信息会以excel表格形式保存到相应路径中。

扩展路径：

1. 若退票时间在起飞前一天，则不予退票。
2. 用户不能对已经生成的订单信息进行修改。

后置条件：

1. 若退票成功则删去此订票记录。
2. 若导出成功则会在相应路径中出现以时间戳命名的excel文件。

## 1.7机票推荐用例

前置条件：用户登录

基本路径：

1. 用户登录机票查询页面。
2. 低价机票：
3. 在机票查询界面低价机票模块，数据库会搜索最近一周内票价最低的机票推荐给用户。
4. 用户单击相应机票
5. 界面跳转到订票界面，订票界面默认选择之前的机票。
6. 推荐机票：
7. 在机票查询界面推荐机票模块，数据库还会查询用户之前的用户浏览和搜索历史来生成用户推荐的机票。

(2)用户单击相应机票

(3)界面跳转到订票界面，订票界面默认选择之前的机票。

扩展路径：跳转失败提示跳转失败，保留当前页面

后置条件：无。

## 1.8用户信息管理用例

前置条件：用户登录

基本路径：

1. 用户进入个人信息界面。
2. 修改注册信息：
3. 用户单击注册信息选项
4. 页面跳转到个人信息修改页面
5. 用户单击电子邮箱能够修改本人的电子邮箱地址
6. 用户单击联系方式能够修改自己的电话号码
7. 用户单击密码能够修改自己的登录密码
8. 订单查看：
9. 用户单击查看历史订单选项。
10. 界面订单查询模块会显示用户按时间排序最近的4次订单。
11. 用户单击详细查询。
12. 界面跳转到订单查询界面，用户能够查询更多订单信息。
13. 查询历史查看：
14. 用户点击查询历史查看选项
15. 界面浏览历史查看模块会显示用户之前的查询历史。
16. 用户点击相应的查询历史。系统给会根据这些记录自动为用户推荐用户想去城市的机票。
17. 界面跳转到查票界面，查票的搜索的目的地、航空公司默认为查询历史记录的内容。

扩展路径：

1.更改登录信息失败提示保存失败，保留当前页面

2.跳转到查票界面，查票的搜索的目的地、航空公司默认为查询历史记录的内容。

后置条件：数据库进行相应修改。

## 1.9航空公司用例

前置条件：界面跳转到航空公司界面

基本路径：

1. 用户单击航空公司介绍选项。
2. 界面航空公司简介模块会出现航空公司的详细介绍。
3. 界面航空公司评价模块会出现航空公司的系统评分。系统评分是根据用户的总体评分和该航空公司的准点率为该航空公司评分的。
4. 用户单击评价选项。
5. 界面用户评价模块会显示最近的用户评价。用户能够为其他用户对该公司服务的评价点赞或者反对。
6. 用户能够在页面底部输入自己的评论并且为该航空公司的服务水平进行打分。

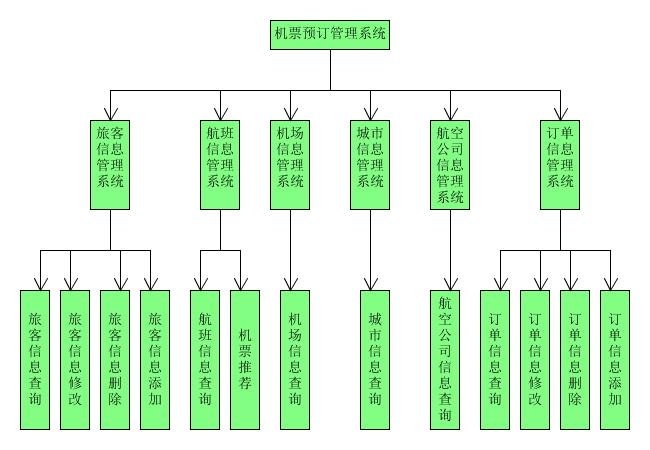
扩展路径：设置失败提示保存失败，保留当前页面

后置条件：将系统管理员的设置更改并保存。

## 二．系统功能结构设计

## 2.1系统功能结构图

系统功能整体结构如下图:



图表2-1 系统功能结构图

## 2.2系统功能详细介绍

**2.2.1旅客信息管理系统**

旅客信息管理系统主要对系统用户的基本信息进行保存与管理，在用户登录验证信息时进行使用。同时，数据库还会保存用户的一些使用习惯，比如经常选择的航班、经常到达的城市，用以给用户提供有价值的航班推荐信息。同时，用户可以在个人信息管理页面对个人信息进行管理，比如查看信息、修改姓名密码等。

**2.2.2航班信息管理系统**

航班信息管理系统用于对航班的基本信息进行存储、分析与使用。首先，系统记录各个航班的时间、起飞城市、降落城市、价格等信息。当用户进行航班搜索时，系统会根据用户要求对数据库进行检索，选择出符合要求的航班供用户选择。同时，系统会对航班的价格、起降时间进行分析，提供给用户更加优质的出行选择。

**2.2.3机场信息管理系统**

机场信息管理系统用于对各个机场的信息进行存储与使用。用户可以查询各个机场的基本信息。

**2.2.4城市信息管理系统**

城市信息管理系统用于对各个业务所覆盖的城市的信息进行存储与使用。用户可以查询各个城市的基本信息。

**2.2.5航空公司信息管理系统**

航空公司信息管理系统用于对各个航空公司的信息进行存储与使用。用户可以查询各个航空公司的基本信息。

**2.2.6订单信息管理系统**

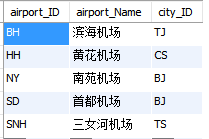
订单信息管理系统用于对用户各个订单的信息进行存储与操作。用户可以在订单管理页面查询自己的所有订单，包括已经预定而未支付、已经支付、已经退票的订单。页面会显示每一个订单的基本信息，包括起降时间、起降城市、价格等信息。同时用户可以对已经订票而未支付的进行支付，对已经订购或已经支付的订单进行退票。

## 2.3系统功能实现方法

本系统采用Browser/Server架构。前端jsp页面与用户交互，提供功能选项，并将用户的选择提交给后端。依据功能要求，实现的主要页面有用户登录页面、机票查询页面，座位选择页面、支付页面、用户信息管理页面、订单管理页面、机票推荐页面。通过这些页面用户可以进行完整的订票操作。用户输入的选择等信息被提交到后端，服务器对这些信息进行检查并对数据库进行检索、修改等操作，最后将结果反馈给前端页面，功能实现。

**三、数据库基本表的定义**

**3.1 airport表**



图表 3-1 airport表

属性说明：

airport\_Name：机场的名称

airport\_ID：机场编号

city\_ID：所在城市的编号

建表语句：

**CREATE** **TABLE** `runoob`**.**`Airport`

**(**

`airport\_ID` **CHAR(**3**)** **NOT** **NULL** **,**

`airport\_Name` **VARCHAR(**10**)** **NOT** **NULL** **,**

`city\_ID` **VARCHAR(**4**)** **NOT** **NULL** **,**

**PRIMARY** **KEY(**`airport\_ID`**),**

**FOREIGN** **KEY** **(**`city\_ID`**)** **REFERENCES** City**(**`city\_ID`**)**

**);**

**3.2 city表**



图表 3-2 city表

属性说明：

city\_ID：城市编号

city\_Name：城市名

建表语句：

**CREATE** **TABLE** `runoob`**.**`City`

**(**

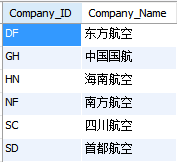
`city\_ID` **VARCHAR(**4**)** **NOT** **NULL** **,**

`city\_Name` **VARCHAR(**4**)** **NOT** **NULL** **,**

**PRIMARY** **KEY(**`city\_ID`**)**

**);**

**3.3 company表**



图表 3-3 company表

属性说明：

Company\_ID：航空公司编号

Company\_Name：航空公司名称

建表语句：

**CREATE** **TABLE** `runoob`**.**`Company`

**(**

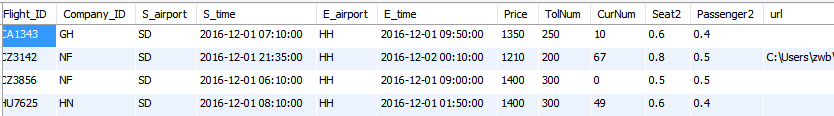
`Company\_ID` **VARCHAR(**10**)** **NOT** **NULL** **,**

`Company\_Name` **VARCHAR(**10**)** **NOT** **NULL** **,**

**PRIMARY** **KEY(**`Company\_ID`**)**

**);**

**3.4 flight表**



图表 3-4 flight表

属性说明：

Flight\_ID：航班编号

Company\_ID：航班所属航空公司

S\_time：起飞时间

E\_time：降落时间

S\_airport：起飞机场

E\_airport：降落机场

Price：票价

ToNum：总座位数

CueNum：当前剩余座位数

Seat2：二等座价格比例

Passenger2：儿童座价格比例

url：座位图文件所在位置

建表语句：

**CREATE** **TABLE** `runoob`**.**`Flight`

**(**

`Flight\_ID` **CHAR(**6**)** **NOT** **NULL** **,**

`Company\_ID` **VARCHAR(**10**)** **NOT** **NULL** **,**

`S\_airport` **CHAR(**3**)** **NOT** **NULL** **,**

`S\_time` **DATETIME** **NOT** **NULL** **,**

`E\_airport` **CHAR(**3**)** **NOT** **NULL** **,**

`E\_time` **DATETIME** **NOT** **NULL** **,**

`Price` **INT(**4**)** **NOT** **NULL** **,**

`TolNum` **int(**3**)** **NOT** **NULL** **,**

`CurNum` **int(**3**)** **NOT** **NULL** **,**

`Seat2` **decimal(**2**,**1**),**

`Passenger2` **decimal(**2**,**1**),**

`url` **VARCHAR(**40**)** **NOT** **NULL** **,**

**PRIMARY** **KEY(**`Flight\_ID`**),**

**FOREIGN** **KEY** **(**`Company\_ID`**)** **REFERENCES** `Company`**(**`Company\_ID`**),**

**FOREIGN** **KEY** **(**`S\_airport`**)** **REFERENCES** `Airport`**(**`airport\_ID`**),**

**FOREIGN** **KEY** **(**`E\_airport`**)** **REFERENCES** `Airport`**(**`airport\_ID`**),**

**CHECK** **(**`Price` **<** 2000**),**

**CHECK** **(**`TolNum` **>** 0 **AND** `TolNum` **<** 400**),**

**CHECK** **(**`TolNum` **>=** 0 **AND** `CurNum` **<** TolNum**),**

**CHECK** **(**`Seat2` **>** 0 **AND** `Seat2` **<** 1**),**

**CHECK** **(**`Passenger2` **>** 0 **AND** `Passenger2` **<** 1**)**

**);**

**3.5 passenger表**



图表 3-5 passenger表

属性说明：

Passenger\_ID：客户编号

Name：客户姓名

E\_mail：客户邮箱

passenger：客户密码

Typ：客户类别

建表语句：

**CREATE** **TABLE** `runoob`**.**`Passenger`

**(**

`Passenger\_ID` **CHAR(**8**)** **NOT** **NULL** **,**

`Name` **VARCHAR(**4**)** **NOT** **NULL** **,**

`E\_mail` **VARCHAR(**20**)** **NOT** **NULL** **,**

`password` **VARCHAR(**20**)** **NOT** **NULL,**

`Typ` **INT(**1**)** **NOT** **NULL** **,**

**PRIMARY** **KEY(**`Passenger\_ID`**),**

**CHECK** **(**`Typ` **=** 1 **OR** `Typ` **=** 2**)**

**);**

**3.6 ticket表**



图表 3-6 ticket表

属性说明：

Ticket\_ID：订单号

Flight\_ID：航班号

Passenger\_ID：乘客编号

seat：座位位置

state：订单当前状态

建表语句：

**CREATE** **TABLE** runoob**.**Ticket

**(**

Ticket\_ID **VARCHAR(**10**)** **NOT** **NULL** **,**

Flight\_ID **VARCHAR(**10**)** **NOT** **NULL** **,**

Passenger\_ID **CHAR(**8**)** **NOT** **NULL** **,**

seat **VARCHAR(**6**)** **NOT** **NULL** **,**

**PRIMARY** **KEY(**Ticket\_ID**),**

**FOREIGN** **KEY** **(**Flight\_ID**)** **REFERENCES** Flight**(**Flight\_ID**),**

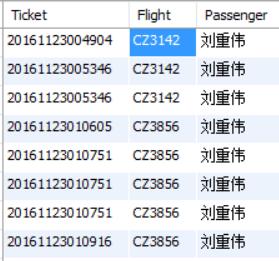
**FOREIGN** **KEY** **(**Passenger\_ID**)** **REFERENCES** Passenger**(**Passenger\_ID**)**

**);**

**3.7 视图**

设计了两个视图。

1.



图表 3-7-1视图search\_tic

视图创建语句：

**CREATE** **VIEW** search\_tic**(**Ticket**,**Flight**,**Passenger**)**

**AS**

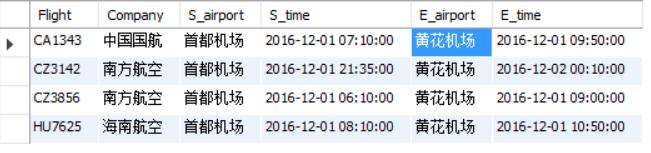
**SELECT** Ticket\_ID**,**Flight\_ID**,**Name

**FROM** ticket **,** passenger

**WHERE** passenger**.**Passenger\_ID **=** ticket**.**Passenger\_ID**;**

设计此视图可以方便地对订单进行查询，简化操作。

2.



图表 3-7-2视图search\_flt

视图创建语句：

**CREATE** **VIEW** search\_flt**(**Flight**,**Company**,**S\_airport**,**S\_time**,**E\_airport**,**E\_time**)**

**AS**

**SELECT** Flight\_ID**,**Company\_Name**,**a1**.**airport\_Name **AS** S\_airport**,**S\_time**,**a2**.**airport\_Name **AS** E\_airport**,**E\_time

**FROM** flight **,** company **,** airport **AS** a1 **,** airport **AS** a2

**WHERE** flight**.**Company\_ID **=** company**.**Company\_ID

**AND** flight**.**S\_airport **=** a1**.**airport\_ID

**AND** flight**.**E\_airport **=** a2**.**airport\_ID**;**

设计此视图可以方便地对航班进行查询，简化操作。

**3.8 索引**

对表flight按各个属性以升序和降序创建了索引以便于查询。

索引创建语句：

**ALTER** **TABLE** flight

**ADD** **INDEX** A\_Price**(**Price**);**

**ALTER** **TABLE** flight

**ADD** **INDEX** D\_Price**(**Price **DESC);**

**ALTER** **TABLE** flight

**ADD** **INDEX** A\_Stime**(**S\_time **ASC);**

**ALTER** **TABLE** flight

**ADD** **INDEX** D\_Stime**(**S\_time **DESC);**

**ALTER** **TABLE** flight

**ADD** **INDEX** A\_Etime**(**E\_time **ASC);**

**ALTER** **TABLE** flight

**ADD** **INDEX** D\_Etime**(**E\_time **DESC);**

**ALTER** **TABLE** flight

**ADD** **INDEX** A\_TolNum**(**TolNum **ASC);**

**ALTER** **TABLE** flight

**ADD** **INDEX** D\_TolNum**(**TolNum **DESC);**

**ALTER** **TABLE** flight

**ADD** **INDEX** A\_CurNum**(**CurNum **ASC);**

**ALTER** **TABLE** flight

**ADD** **INDEX** D\_CurNum**(**CurNum **DESC);**

**3.7 数据库完整性**

**3.7.1实体完整性**

每一个基本表都定义了主码以确保数据库实体完整性。各个表主码分别为:

|  |  |
| --- | --- |
| 表 | 主码 |
| Passenger | Passenger\_ID |
| City | city\_ID |
| Airport | airport\_ID |
| Company | Company\_ID |
| Flight | Flight\_ID |
| Ticket | Ticket\_ID |

图表 3-7-1各表主码

在对基本表进行更新操作时，数据库管理系统会检查主码值是否唯一、将要插入的值是否为空，如果主码不唯一或值为空则拒绝操作，这样便保证了数据库的实体完整性。

**3.7.2参照完整性**

为了保证参照完整性在以下几个表中设置了外码:

1.表Airport的外码为city\_ID，参照的是City表的city\_ID属性，并定义city\_ID不为空。当操作违反参照完整性时，采用系统默认处理策略，即拒绝执行。

2.表Flight的外码为Company\_ID、S\_airport、E\_airport，分别参照Company的Company\_ID属性、Airport的airport\_ID属性、Airport的airport\_ID属性。当操作违反参照完整性时，采用系统默认处理策略，即拒绝执行。

3.表Ticket的外码为Flight\_ID和Passenger\_ID，分别参照Fligh表的Flight\_ID属性、Passenger表的Passenger\_ID属性。当操作违反参照完整性时，采用系统默认处理策略，即拒绝执行。

**3.7.3自己定义的完整性**

1.在建表时，除了Flight表的Seat2属性和Passenger2属性，所有表上的属性都设置为NOT NULL。

2.使用CHECK语句。

在Passenger表中在元组上约束Typ=1或Typ=2。

在Flight表上，在元组上约束Price<2000，0<TolNum < 400，TolNum`>= 0 且CurNum`< TolNum，0<Seat2<1，0<Passenger2<1。

当操作不满足约束条件时，将被拒绝执行。

**四、触发器的定义与实现**

系统设置了两个触发器。

1.

DELIMITER **|**

**CREATE** **TRIGGER** ADD\_RECORD

**AFTER** **INSERT** **ON** runoob**.**ticket

**FOR** **EACH** **ROW**

**BEGIN**

**UPDATE** flight **SET** CurNum **=** CurNum **+** 1 **WHERE** Flight\_ID **=** **NEW.**Flight\_ID **;**

**END|**

DELIMITER **;**

作用为当ticket表中插入一个元组时，flight表中Flight\_ID为插入的ticket元组的Flight\_ID的元组的CurNum属性值加一，即一张票被预定后系统自动将相应航班的当前人数加一。

2.

DELIMITER **|**

**CREATE** **TRIGGER** DROP\_RECORD

**AFTER** **UPDATE** **ON** ticket

**FOR** **EACH** **ROW**

**BEGIN**

**IF** **NEW.state** **=** '3' **AND** **OLD.state** **=** '1' **THEN**

**UPDATE** flight **SET** CurNum **=** CurNum **-** 1 **WHERE** Flight\_ID **=** **NEW.**Flight\_ID **;**

**END** **IF;**

**END|**

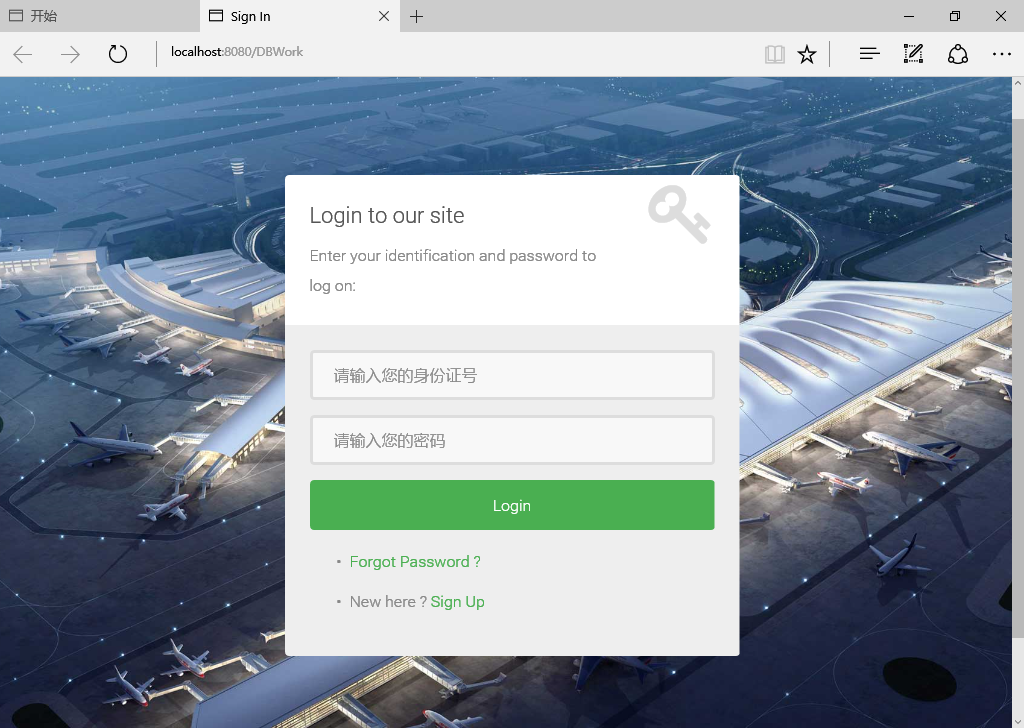
DELIMITER **;**

作用为当ticket表某一个元组的state属性由1更新为3时，flight表中Flight\_ID为插入的ticket元组的Flight\_ID的元组的CurNum属性值减一，即一张票被退票后系统自动将相应航班的当前人数减一。

**五、存储过程的定义与实现**

**六、系统实现结果**

## 6.1登录页面

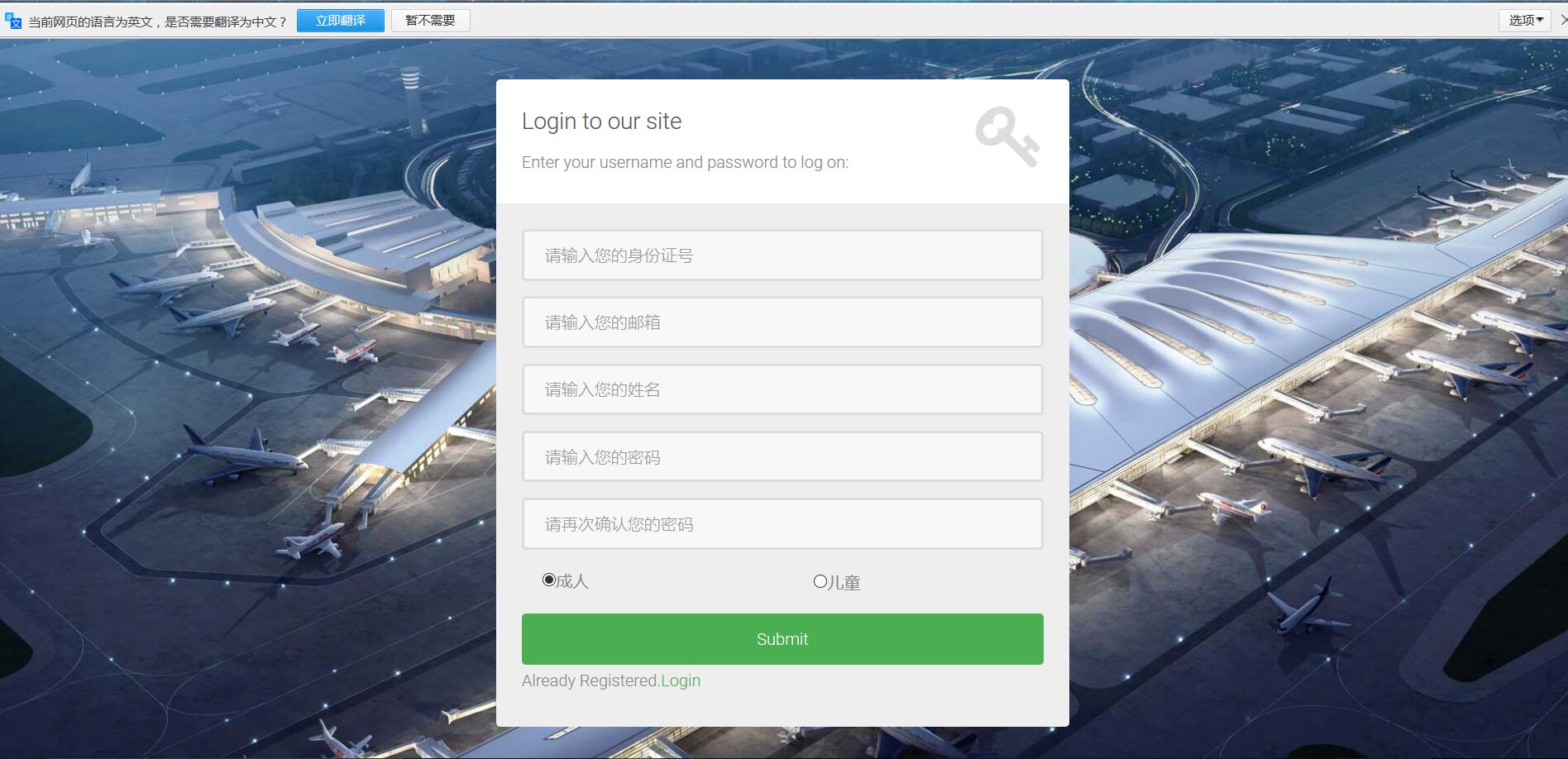


图表 6-1登录页面

页面介绍：

此页面为用户登录页面，如果用户已经注册过，直接输入账号和密码就可以直接登陆系统。如果是首次来访可以点击底部“Sign Up”按钮进行注册。

## 6.2注册页面

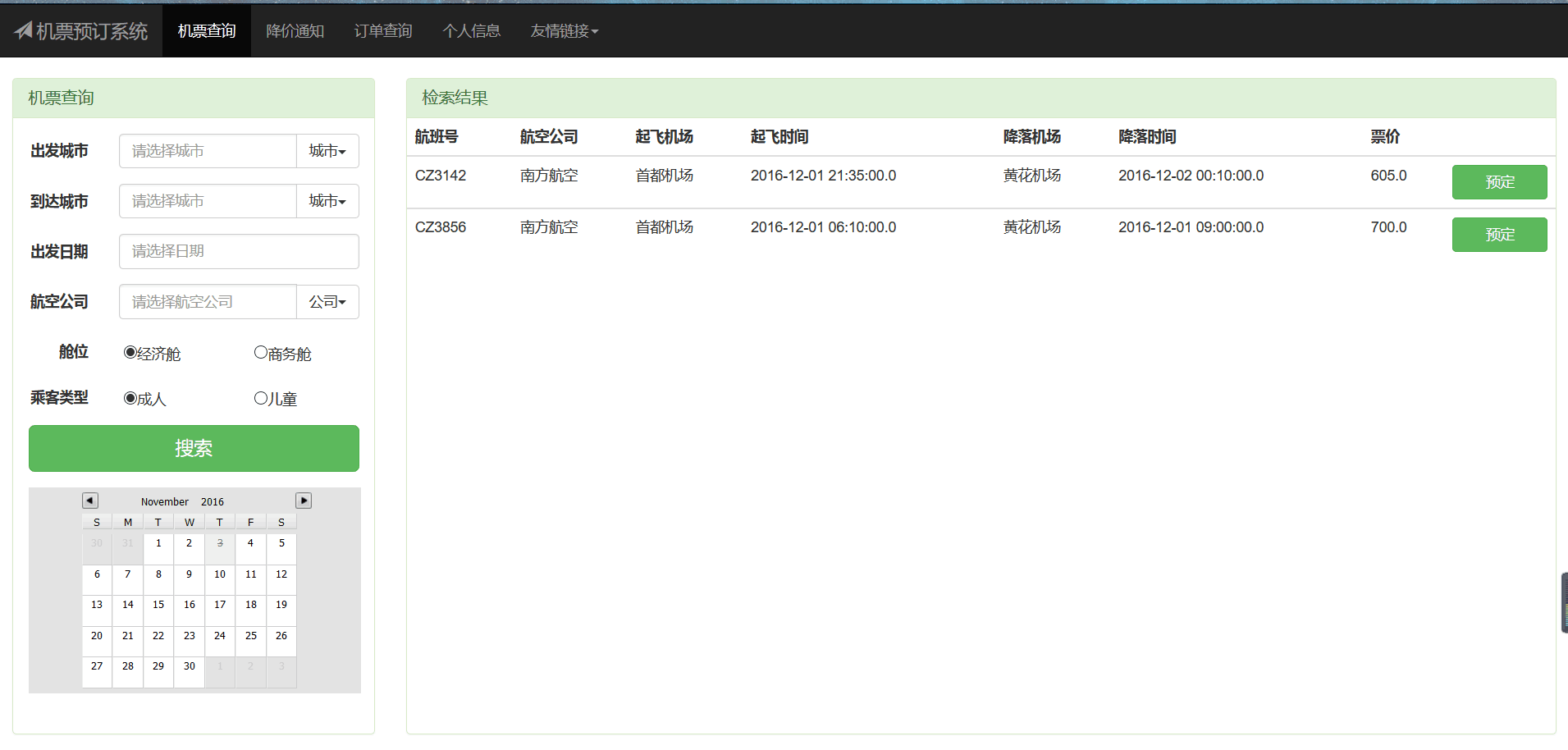


图表 6-2注册页面

页面介绍：

此页面为用户注册页面，注册信息有账号（身份证号）、邮箱、姓名、密码以及乘客类别（成人或儿童）。在填写注册信息时系统会检查格式，如果不符合格式会给出提示信息。信息无误后用户点击Submt完成注册，并转到登录页面。

## 6.3航班查询页面

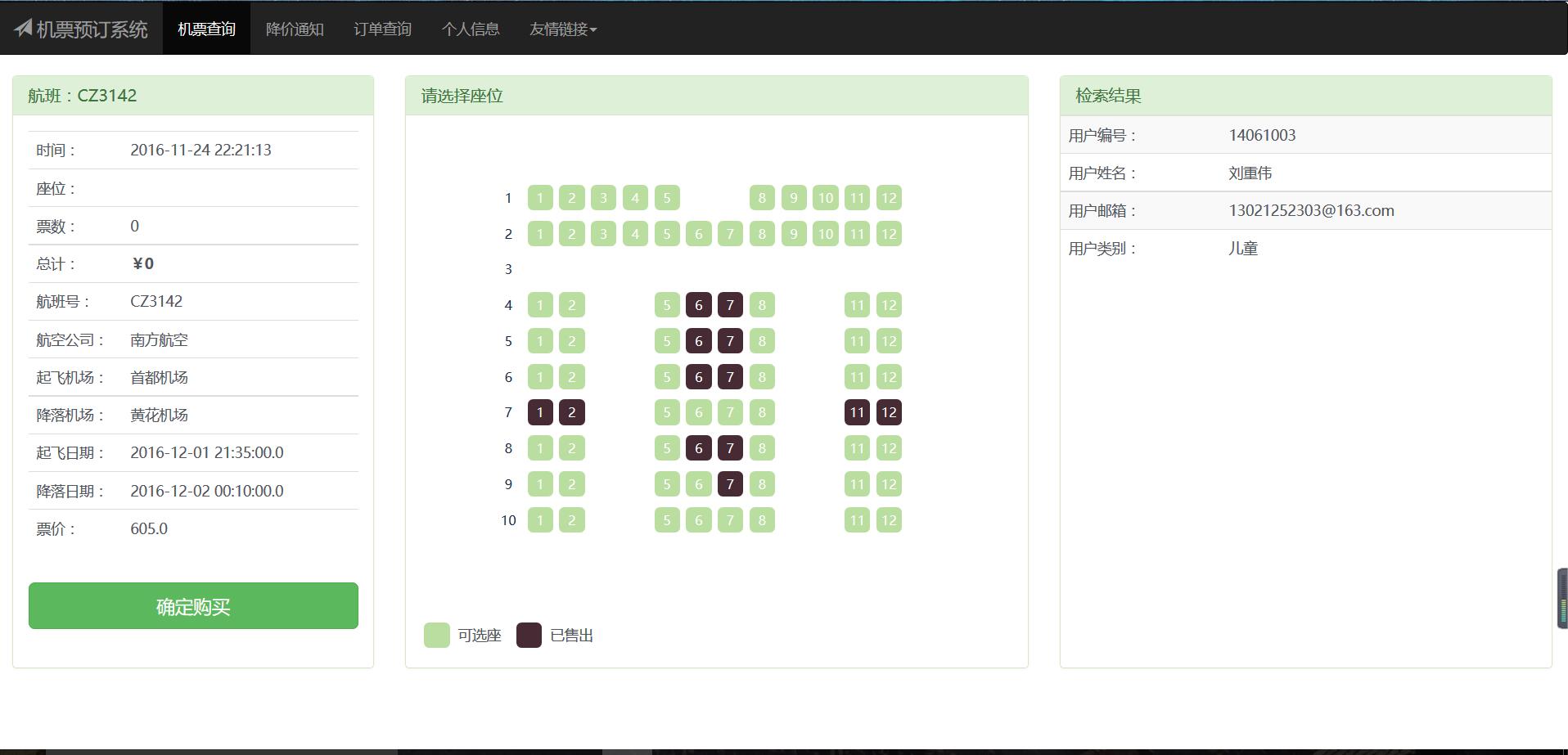


图表 6-3航班查询页面

页面介绍:

此页为航班查询页面，页面最上面一行是导航栏，最左侧为输入表单，用户输入出发城市、到达城市、出发日期、航空公司、舱位以及乘客类别信息，日期的输入可以在下方的日历中直接点击，输入完成之后点击“搜索”，系统开始在数据库中检索符合条件的航班。如果有符合条件的航班便会在页面右侧显示出来。用户选择满意的航班之后可以点击相应航班右侧的“预定”按钮进行预订。

## 6.4选座页面

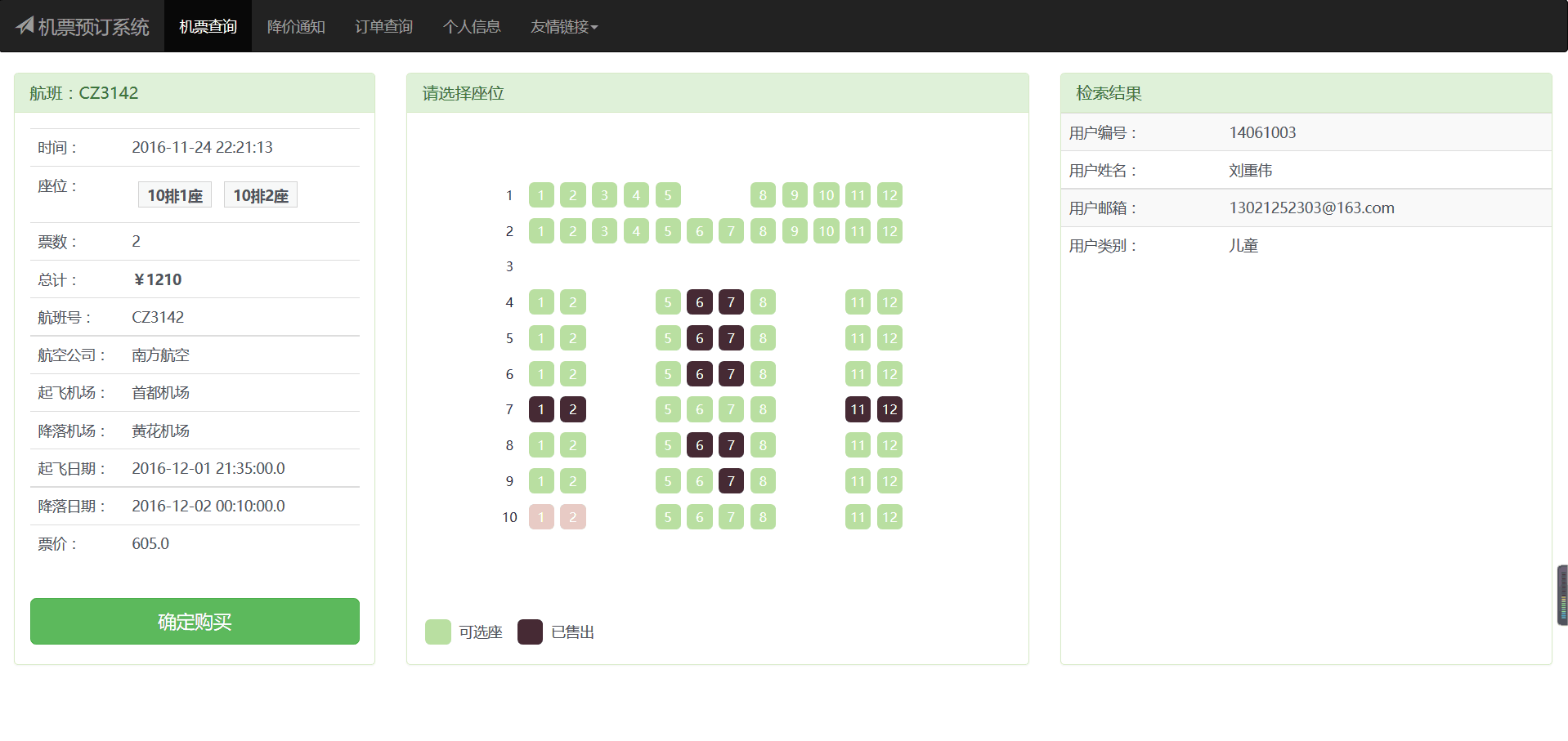


图表 6-4选座页面

页面介绍：

用户在搜索页面预定相应航班后会跳转到选座页面进行座位选择。页面左侧是航班信息。页面中间是座位分布图，其中黑色标识的表示已经被人预定，不可选择，而绿色标识的可以预订。页面右侧是用户个人信息。

## 6.5已经选座

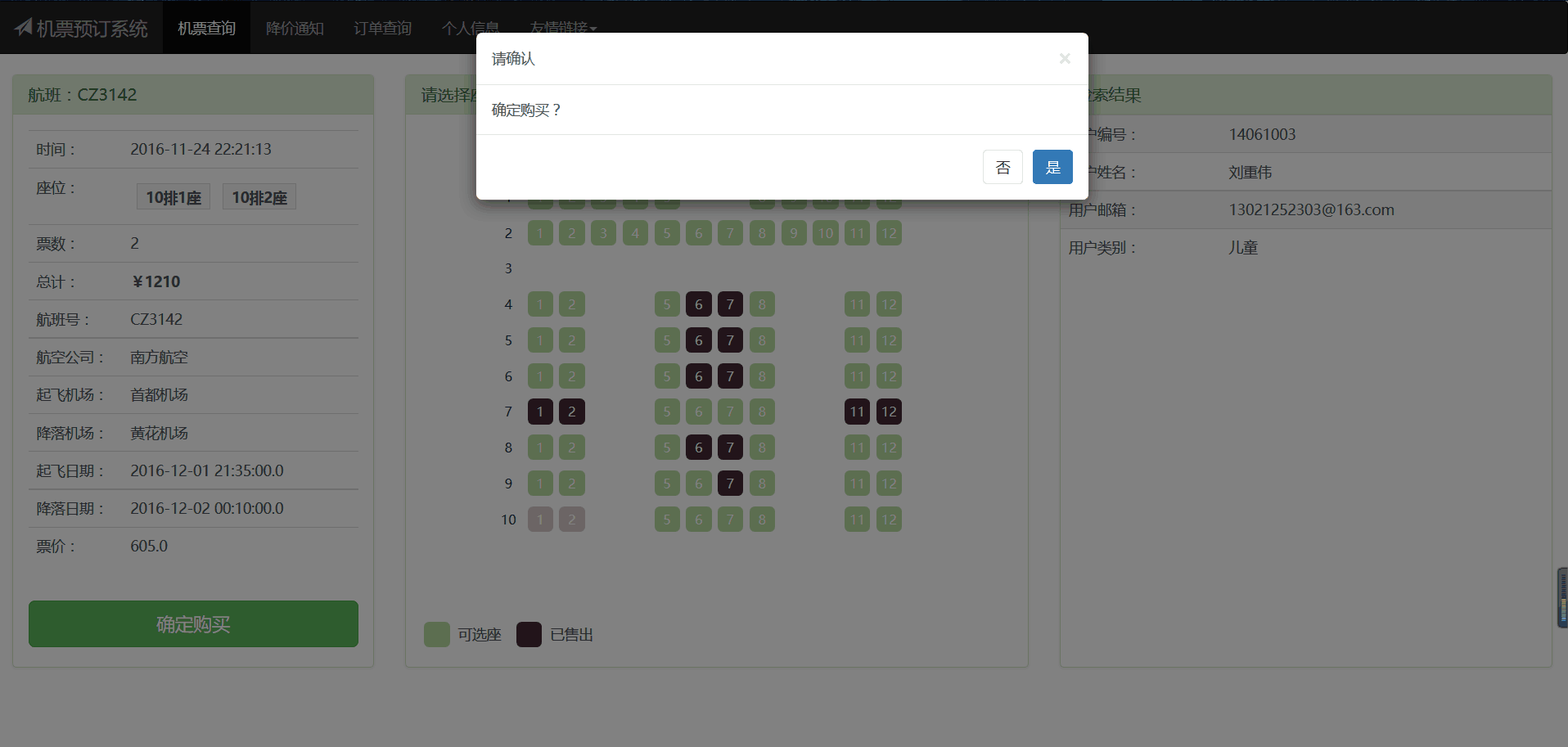


图表 6-5已经选座效果

页面介绍：

此图展示用户选择页面之后效果。可以看到页面中间出现了两个红色标记，表示的是用户选择的座位。同时在左侧部分“座位”项里显示了座位信息，票数和总价也有了相应改变。用户选择了座位之后，可以点击左侧最下面“确认购买”按钮进行机票购买。

## 6.6确认支付提示信息

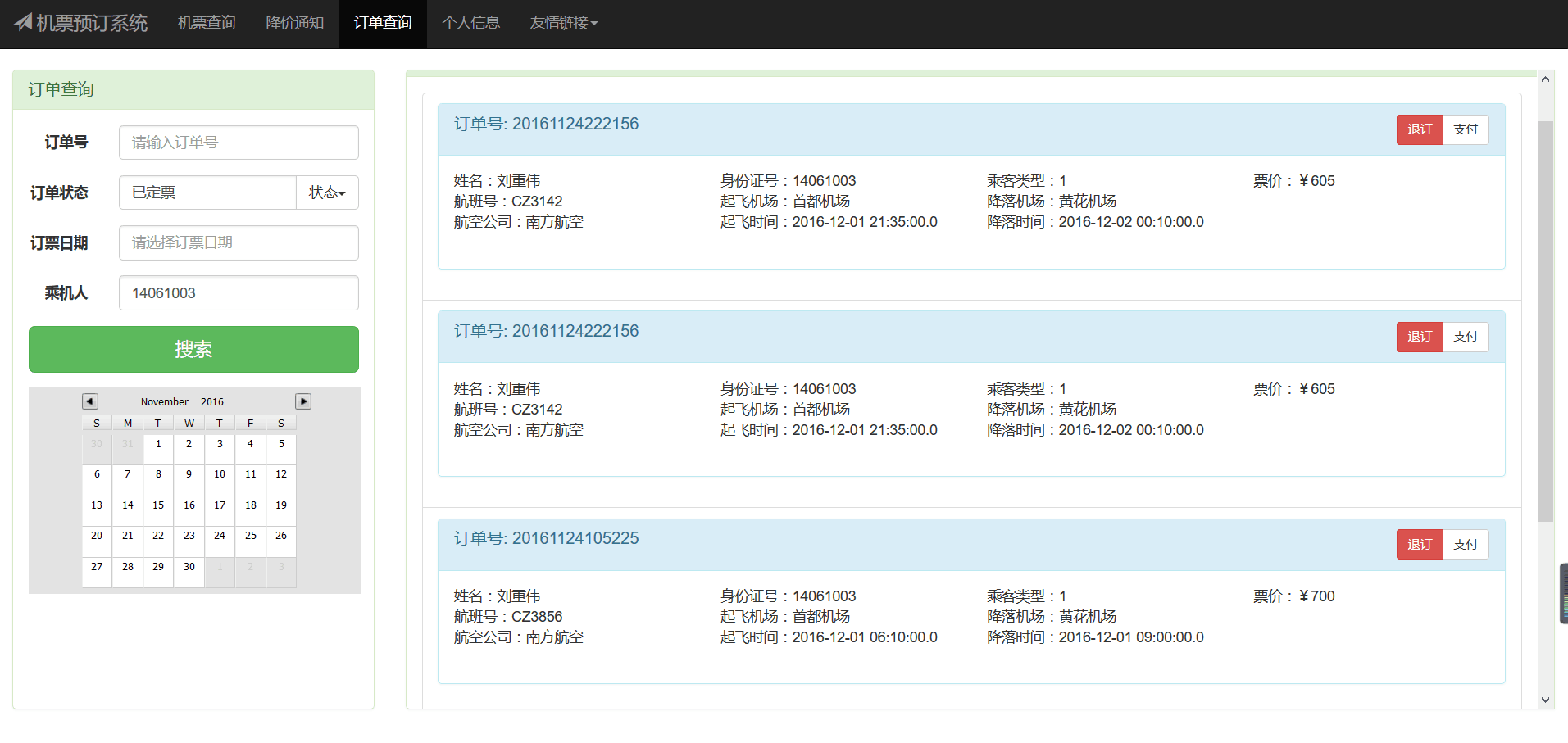


图表 6-6确认支付提示信息

页面介绍：

用户点击“确认购买”之后会弹出确认提示框，用户可以选择“是”或者“否”。

**6.7订单管理页面**

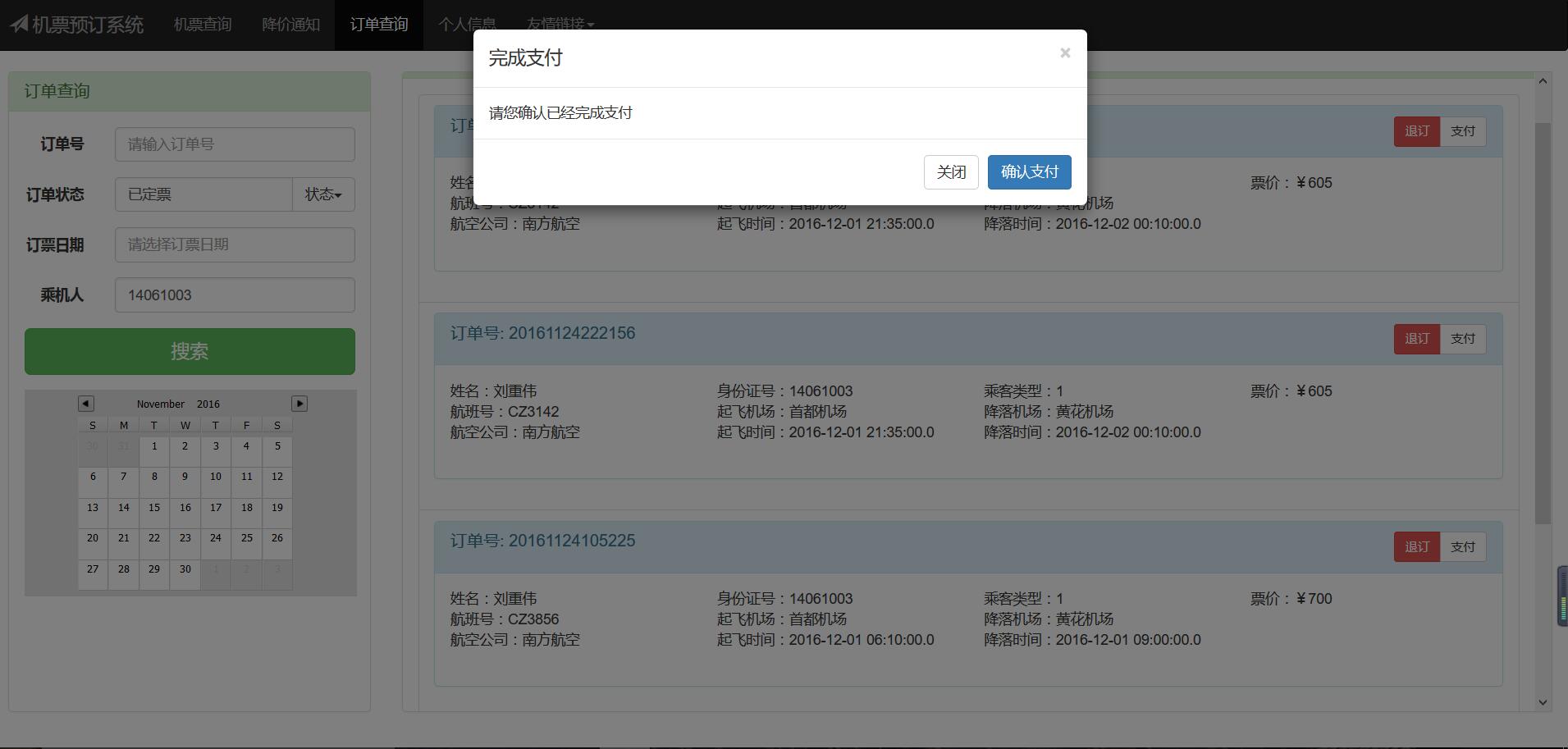


图表 6-7订单管理页面

页面介绍：

此页为用户订单管理页面。页面左侧是输入表单，用户可以输入订单号、订单状态、订票日期和购票人信息以检索相应的订单。提交信息之后，系统会检索数据库，如果有符合条件的订单会在页面右侧显示出来。订单信息背景色有红、蓝、绿三种，分别表示已退订单、已订但未支付订单和已经支付订单。对于已订但未支付订单可以选择退订或支付。

**6.8订单支付提示**



图表 6-8订单支付提示

页面介绍：

此页展示用户在订单信息页点击“支付”之后的提示信息。

**6.9订单改变效果**

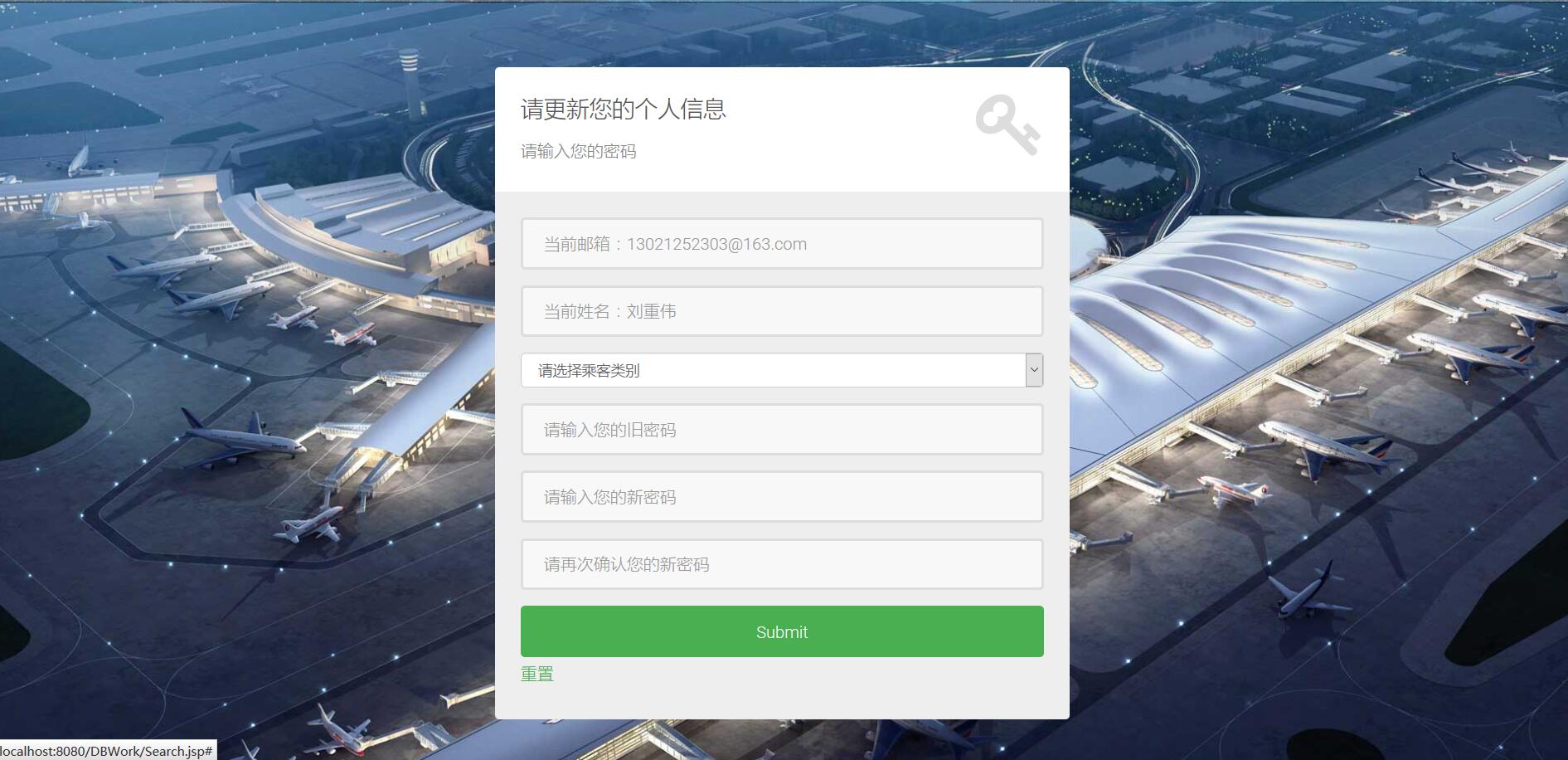


图表 6-9订单改变效果

页面介绍：

此页展示用户完成支付之后订单状态的改变情况。可以看出订单信息背景色由蓝色变为绿色。

**6.10用户信息管理页面**

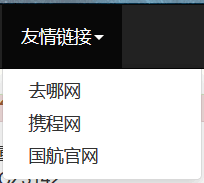


图表 6-10用户信息管理页面

页面介绍：

此页面为用户信息管理页面。页面会在相应条目上显示用户当前信息，用户既可以查看也可以修改个人信息。修改个人信息同样需要检查输入格式，如果格式不正确会给出错误提示。用户修改密码需要先填入原来密码，如果不正确的话，系统不予更新，并重置该页面。用户也可以自己点击最下面“重置”按钮对页面进行重置。

**6.11友情链接**

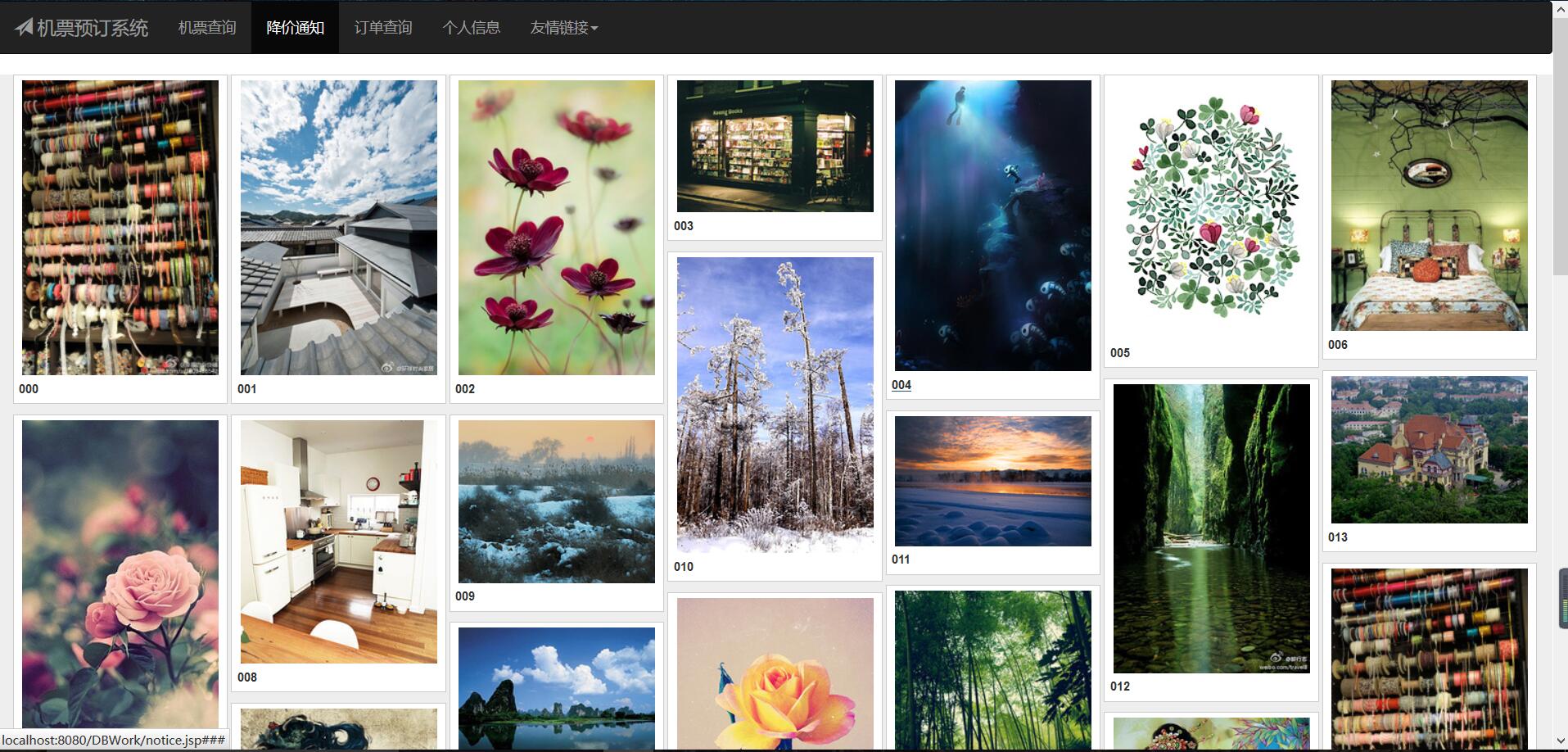


图表 6-11友情链接

页面介绍：

此页为位于导航栏，用户选择点击便会转到相应网站。

**6.12机票推荐**



图表 6-12机票推荐页面

页面介绍：

此页为机票推荐页。系统会根据用户的出行习惯以及当前航班降价情况进行分析，给用户提供一些有价值的推荐信息。推荐信息会以图片方式呈现给用户。

**七、总结**

通过本次课设，我们掌握了B/S架构的数据库应用的基本设计方法和设计流程，对数据库的设计有了更清楚的认识，用实践巩固了理论知识。

但是毕竟是第一次设计数据库应用，刚开始完全不知道从何入手，不知道用什么工具，也不知道怎样设计才能保证在接下来工作中少走弯路。后来经过查询资料才渐渐有了一些头绪，了解了一个B/S结构的数据库应用的基本框架，决定采用java和jsp进行开发。有了大体思路之后我们进行了前后端分工，但是由于缺乏沟通，初期两个人的思路不甚相同，导致所做的工作难以结合在一起，做了不少无用功。没办法只好重新进行调整，边交流边进行开发，遇到问题一起思考解决办法，进度也快了不少。

由于事时间原因，我们的系统还有很多不完善的地方，有不少计划的功能没有做出来，今后还要慢慢完善。还有就是，由于我们花了不少时间在前端设计上，导致后端数据库设计不算完善，这也算一个值得吸取的教训吧。