

中国人民大学信息学院

# 服务计算

旅游代理系统 报告

小组成员：陶友贤（2017100955）、赵旺（2017104089）、陈成（2017000613）

Github: <https://github.com/taoyouxian/TravelAgencySys>

# 目录

- 1 分工安排.....2
- 2 服务说明.....2
  - 2.1 系统运行工具.....2
  - 2.2 企业端.....3
    - 2.2.1 FlightServiceI .....3
    - 2.2.2 SeatServiceI .....4
    - 2.2.3 HotelServiceI .....5
    - 2.2.4 RoomServiceI.....6
    - 2.2.5 CarServiceI.....8
    - 2.2.6 AttractionServiceI .....9
  - 2.3 用户端.....10
    - 2.3.1 BEPL 流程.....11
- 3 系统特色.....13
- 4 系统演示.....13
- 5 报告总结.....17

# 1 分工安排

任务名称	人员
系统架构设计、数据库设计	陶友贤
用户端业务逻辑与后台接口	陶友贤
企业端业务逻辑与后台接口	陈成、赵旺
界面设计	陶友贤、赵旺
WSDL、BEPL	陶友贤、陈成（参与）
Github 信息待补充	陶友贤、陈成、赵旺
PPT 汇报	陶友贤
报告	赵旺、陶友贤（补充）

Github 目录：

```

TravelAgencySys/
├── docx/
│   ├── db
│   ├── images
│   ├── ppt
│   ├── report
│   ├── wsdl
│   └── bepl
├── images/
│   ├── visualization_rowobject
│   ├── visualization_metafilelocation
│   └── visualization_metatable
├── src/
│   └── SpringMVC<Tomcat 8080>
├── web/
│   ├── Front
│   └── Back<Tomcat 9080>
├── Others
└── Wiki<Not completed>

```

# 2 服务说明

## 2.1 系统运行工具

- Eclipse for JAVA EE (Eclipse neon)
- JDK 1.8
- Apache Tomcat 8.5（2 个服务器，端口分别设置成 8080，9080）

- Apache ODE
- Axis2
- SQL Server 2008R2（单一数据库，未及时设计成多个数据库）
- IIS（跨域，前后台分离）

## 2.2 企业端

- 提供 WSDL 文档
- 实现对航空公司、酒店、租车公司和景点管理公司数据信息的维护，主要是实现增删改查功能

### 2.2.1 FlightServiceI

地址: <http://10.77.40.27:9080/axis2/services/FlightServiceI?wsdl>

功能描述:

#### *Available Operations*

- EditFlight
- DeleteFlight
- NewFlight
- getFlight

#### (1) getFlight

输入: 排序字段 (aOrderFields), 页大小 (aPageSize), 页起始 (aPageIndex), 参数 (aSqlPs)

输出: 分页的航班信息

#### (2) NewFlight

输入:

```
f_code, f_caption, f_start_addr, f_arrive_addr, f_start_time,  
f_arrive_time, f_duration, f_duration_info, f_price
```

输出: 新增成功

#### (3) EditFlight

输入:

```
f_code, f_caption, f_start_addr, f_arrive_addr, f_start_time,  
f_arrive_time, f_duration, f_duration_info, f_price, id
```

输出：更新成功

(4) DeleteFlight

输入：航班号 (id)

输出：删除成功

数据库描述：（[详细描述信息见附录](#)）

	列名	数据类型
🔑	f_id	int
	f_code	varchar(10)
	f_caption	varchar(20)
	f_info	varchar(30)
	f_start_addr	varchar(50)
	f_arrive_addr	varchar(50)
	f_duration	varchar(10)
	f_duration_info	varchar(20)
	f_start_time	datetime
	f_arrive_time	datetime
	f_price	int

## 2.2.2 SeatServiceI

地址：<http://10.77.40.27:9080/axis2/services/SeatServiceI?wsdl>

功能描述：

### Available Operations

- NewSeat
- EditSeat
- getSeat
- DeleteSeat

(1) getSeat

输入：排序字段 (aOrderFields)，页大小 (aPageSize)，页起始 (aPageIndex)，参数 (aSqlPs)

输出：分页的舱位信息

(2) NewSeat

输入：

s\_f\_id, s\_num, s\_type, s\_daily\_price

输出：新增成功

### (3) EditSeat

输入：

s\_f\_id, s\_num, s\_type, s\_daily\_price, id

输出：更新成功

### (4) DeleteSeat

输入：舱位号 (id)

输出：删除成功

## ● 数据库描述：

	列名	数据类型
🔑	s_id	int
	s_f_id	int
	s_num	int
	s_type	int
	s_daily_price	int

## 2.2.3 HotelServiceI

地址：<http://10.77.40.27:9080/axis2/services/HotelServiceI?wsdl>

## ● 功能描述：

### Available Operations

- getHotel
- NewHotel
- EditHotel
- DeleteHotel

### (1) getHotel

输入：排序字段 (aOrderFields)，页大小 (aPageSize)，页起始 (aPageIndex)，参数 (aSqlPs)

输出：分页的酒店信息

### (2) NewHotel

输入：

h\_name, h\_code, h\_start\_time, h\_end\_time, h\_city, h\_t\_id, h\_s\_i

d, h\_p\_id, h\_b\_id, h\_a\_id, h\_price

输出：新增成功

### (3) EditHotel

输入：

h\_name, h\_code, h\_start\_time, h\_end\_time, h\_city, h\_t\_id, h\_s\_i  
d, h\_p\_id, h\_b\_id, h\_a\_id, h\_price, id

输出：更新成功

### (4) DeleteHotel

输入：酒店号 (id)

输出：删除成功

### ● 数据库描述：

	列名	数据类型
🔑	h_id	int
	h_t_id	int
	h_s_id	int
	h_p_id	int
	h_b_id	int
	h_a_id	int
	h_name	varchar(20)
	h_code	varchar(20)
	h_start_time	datetime
	h_end_time	datetime
	h_city	varchar(20)
	h_addr	varchar(50)
	h_price	int
	h_company	varchar(30)
	h_tel	varchar(15)
	h_photo	varchar(50)

## 2.2.4 RoomServiceI

地址：<http://10.77.40.27:9080/axis2/services/RoomServiceI?wsdl>

### ● 功能描述：

### Available Operations

- EditRoom
- getRoom
- NewRoom
- DeleteRoom

#### (1) getRoom

输入：排序字段（aOrderFields），页大小（aPageSize），页起始（aPageIndex），参数（aSqlPs）

输出：分页的酒店信息

#### (2) NewRoom

输入：

```
r_h_id, h_code, r_code, r_type, r_bed, r_cancel, r_eat, r_wifi, r_daily_price
```

输出：新增成功

#### (3) EditRoom

输入：

```
r_h_id, h_code, r_code, r_type, r_bed, r_cancel, r_eat, r_wifi, r_daily_price, id
```

输出：更新成功

#### (4) DeleteRoom

输入：房间号（id）

输出：删除成功

### ● 数据库描述：

	列名	数据类型
🔑	r_id	int
	r_h_id	int
	r_code	varchar(10)
	r_type	varchar(30)
	r_bed	varchar(20)
	r_cancel	int
	r_eat	int
	r_wifi	int
	r_daily_price	int



## 2.2.5 CarServiceI

地址: <http://10.77.40.27:9080/axis2/services/CarServiceI?wsdl>

- 功能描述:

### *Available Operations*

- NewCar
- EditCar
- getCar
- DeleteCar

#### (1) getCar

输入: 排序字段 (aOrderFields), 页大小 (aPageSize), 页起始 (aPageIndex), 参数 (aSqlPs)

输出: 分页的酒店信息

#### (2) NewCar

输入:

```
c_lease_addr, c_return_addr, c_lease_time, c_return_time, c_recommen  
d_com, c_s_id, c_pre_id, c_pri_id, c_ty_id, c_tr_id, c_daily_price
```

输出: 新增成功

#### (3) EditCar

输入:

```
c_lease_addr, c_return_addr, c_lease_time, c_return_time, c_recommen  
d_com, c_s_id, c_pre_id, c_pri_id, c_ty_id, c_tr_id, c_daily_price, id
```

输出: 更新成功

#### (4) DeleteCar

输入: 租车号 (id)

输出: 删除成功

- 数据库描述:

	列名	数据类型
🔑	c_id	int
	c_s_id	int
	c_pre_id	int
	c_pri_id	int
	c_ty_id	int
	c_tr_id	int
	c_lease_addr	varchar(50)
	c_return_addr	varchar(50)
	c_lease_time	datetime
	c_name	varchar(30)
	c_addr	varchar(50)
	c_city	varchar(20)
	c_return_time	datetime
	c_recommend_com	varchar(30)
	c_com_tel	varchar(15)
	c_pic	varchar(50)
	c_daily_price	int

## 2.2.6 AttractionServiceI

地址: <http://10.77.40.27:9080/axis2/services/AttractionServiceI?wsdl>

### ● 功能描述:

#### *Available Operations*

- NewAttraction
- DeleteAttraction
- getAttraction
- EditAttraction

#### (1) getAttraction

输入: 排序字段 (aOrderFields), 页大小 (aPageSize), 页起始 (aPageIndex), 参数 (aSqlPs)

输出: 分页的景点信息

#### (2) NewAttraction

输入:

a\_location, a\_tour\_type, a\_tour

输出: 新增成功

#### (3) EditAttraction

输入：

a\_location, a\_tour\_type, a\_tour, id

输出：更新成功

(4) DeleteAttraction

输入：景点号 (id)

输出：删除成功

● 数据库描述：

	列名	数据类型
🔑	a_id	int
	a_location	varchar(10)
	a_spot	varchar(100)
	a_tour_type	int
	a_tour	varchar(20)
	a_pic	varchar(10)

## 2.3 用户端

- 用户自定义 ( {机票} × {酒店} × {租车} × {景点} )
- 根据预算做系统推荐 (根据用户条件, 最节约预算)

2.3.1 BEPL 流程

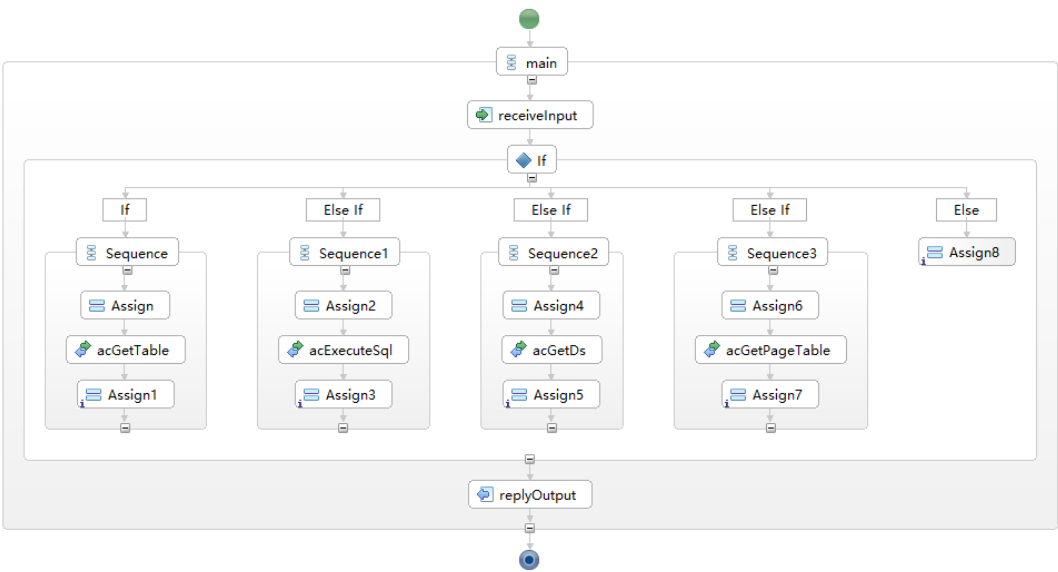


图 1

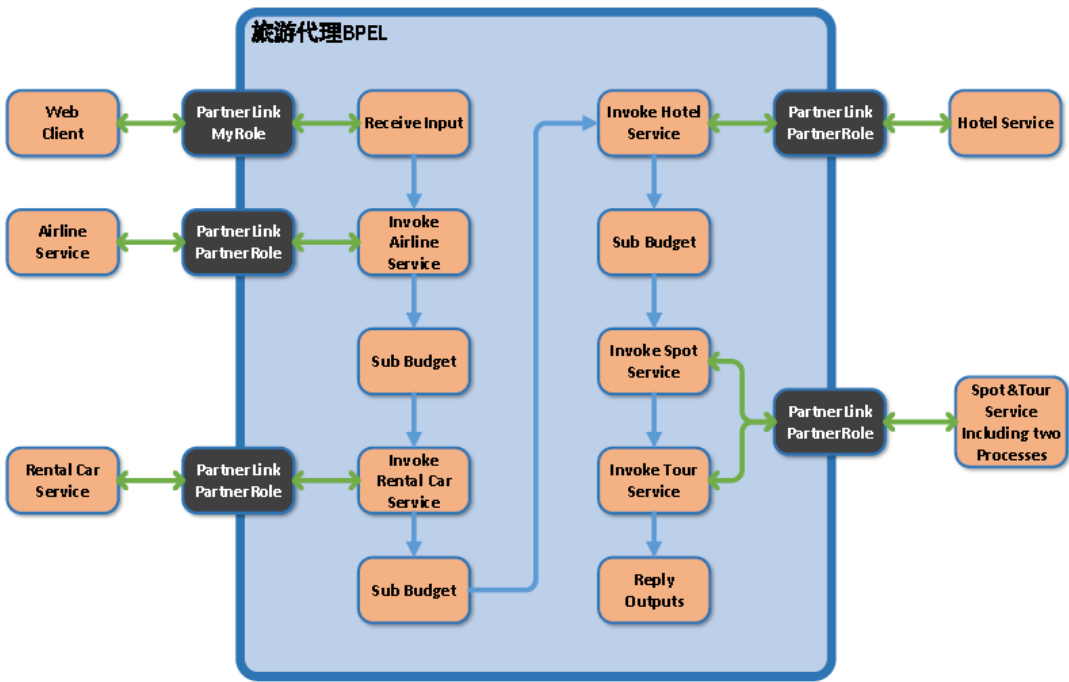


图 2

如上图中所示，本系统的 BEPL 流程有两个，但是图 2 在生成 Client 代码调用中出现了问题，因此只有流程图画出来了，后面会把 Bug 解决，也是一个挑战对自己的过程。

(1) 根据业务接口的 BEPL 流程

- 设计思路：

partnerLink	partnerLinkType	role
Client（客户交互端）	tns:TravelProcess	myRole="TravelProcessProvider"
acGetTable（提供表信息查询服务）	tns:acGetTablePL	partnerRole="acGetTableProvider"
acExecuteSql（提供表操作服务）	tns:acExecuteSqlPL	partnerRole="acExecuteSqlProvider"
acGetDs（提供多表信息查询服务）	tns:acGetDsPL	partnerRole="acGetDsProvider"
acGetPageTable（提供分页信息服务）	tns:acGetPageTablePL	partnerRole="acGetPageTableProvider"

## （2） 根据各个服务部分的 BEPL 流程

### ● 设计思路

- 1) 接收客户端输入；
- 2) 同步调用 airlineService 服务，查询符合条件的机票，airlineService 服务返回最佳机票信息；
- 3) 从总预算中扣除机票费用；
- 4) 同步调用 rentalCarService 服务，查询符合用户条件的租车信息，并且价格不大于剩余预算，rentalCarService 服务返回最佳租车信息；
- 5) 从剩余预算中扣除租车费用；
- 6) 同步调用 hotelService 服务，查询符合用户条件的旅馆，并且价格不大于剩余预算，hotelService 服务会返回最佳旅馆信息；
- 7) 从剩余预算中扣除旅馆费用；
- 8) 同步调用 spot&tourService 服务，查询符合用户条件的景点和旅游信息，并且价格不大于剩余预算，spot&tourService 服务会返回最佳的 spot&tourService。
- 9) 若所有服务请求都满足条件，则最后返回查询结果，否则，返回空，即无满足条件的服务组合。

备注：由于出现了部分 Bug，只能通过 WSDL 代替了这部分的 BEPL，地址：

<http://10.77.40.27:9080/axis2/services/TourServiceI?wsdl>

### 3 系统特色

- 通过 Bootstrap 框架设计前端,通过 BaiDu 模板技术 Ajax 从后台获取的 Json 数据与相应模块结合,更便捷地进行代码开发
- 仿照携程、艺龙等多个官方网站进行开发,使得系统更加接近于实际的项目实战
- 系统只允许非登陆用户进行在线查询
- 系统允许登录用户构建旅游方案,每个方案可以选择多个机票方案、酒店方案等,如此实现笛卡尔积的最终方案,按照预算大小进行排序
- 系统给非登陆用户进行整个旅游方案的推荐(包括机票、酒店、租车、景点),给登录用户按照自己制定的旅游方案进行推荐
- 系统给用户进行系统所有用户的旅游方案
- 

### 4 系统演示

- 企业端

**RucTour Travel Agent Service System**

Search ChangePwd Exit

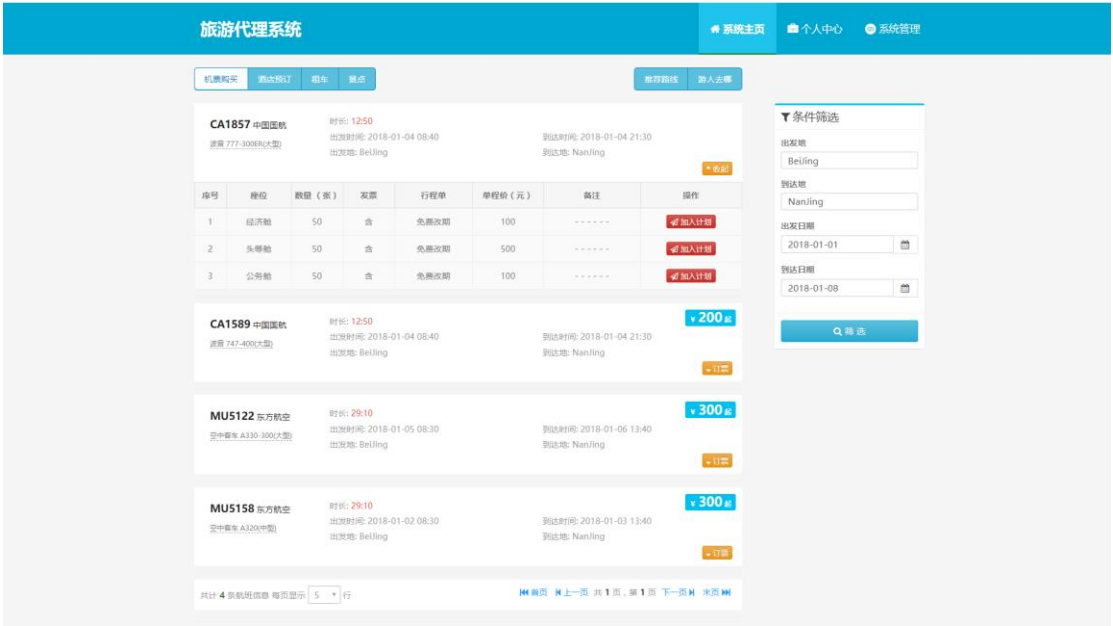
**Flight Service** + Add Flight

Code	Caption	Airline	Time	Price	Duration	Info	Operation
HU7603	海南航空	Beijing 上海	2018-01-05 08:30:00.0 -> 2018-01-06 13:40:00.0	400	29:10	1 days arrival	Seat Update Delete
FM9106	上海航空	Beijing 上海	2018-01-05 08:30:00.0 -> 2018-01-06 13:40:00.0	400	29:10	1 days arrival	Seat Update Delete
MU5122	东方航空	Beijing NanJing	2018-01-05 08:30:00.0 -> 2018-01-06 13:40:00.0	300	29:10	1 days arrival	Seat Update Delete
CA1589	中国国航	Beijing NanJing	2018-01-04 08:40:00.0 -> 2018-01-04 21:30:00.0	200	12:50	0 days arrival	Seat Update Delete
HU7601	海南航空	Beijing 上海	2018-01-03 08:30:00.0 -> 2018-01-04 13:40:00.0	400	29:10	1 days arrival	Seat Update Delete
MU5120	东方航空	Beijing 上海	2018-01-02 08:30:00.0 -> 2018-01-03 13:40:00.0	400	29:10	1 days arrival	Seat Update Delete
MU5158	东方航空	Beijing NanJing	2018-01-02 08:30:00.0 -> 2018-01-03 13:40:00.0	300	29:10	1 days arrival	Seat Update Delete
CA1857	中国国航	Beijing NanJing	2018-01-04 08:40:00.0 -> 2018-01-04 21:30:00.0	200	12:50	0 days arrival	Seat Update Delete

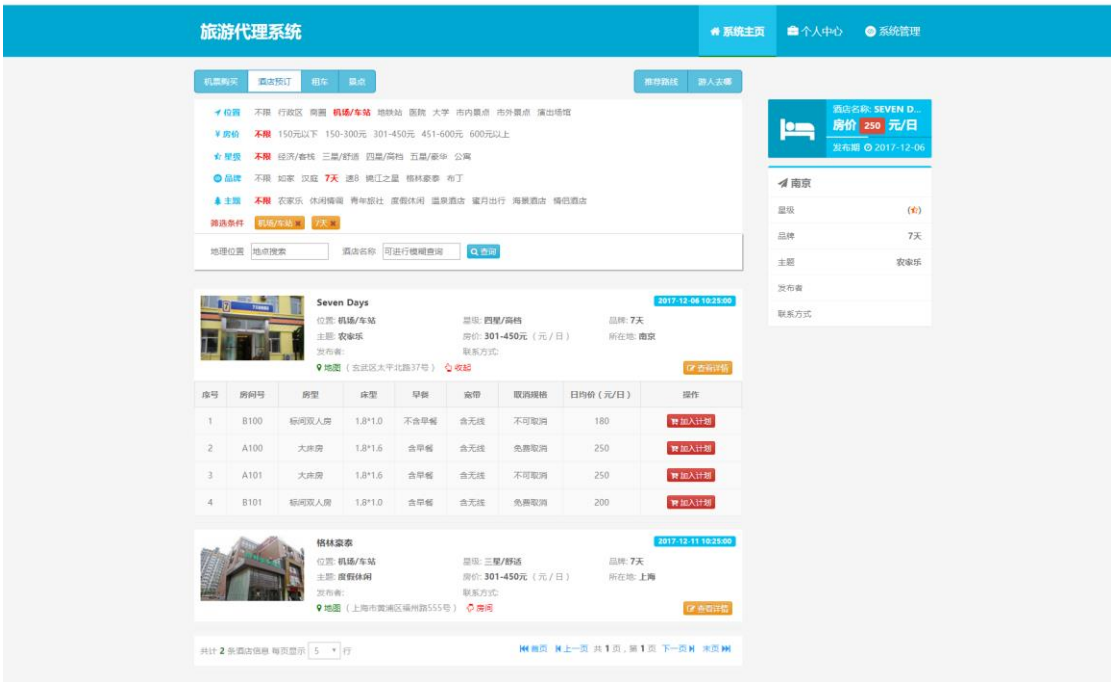
All 8 messages, Every page shows 10 rows

First page Pre page Total 1 page, No 1 Page Next page Last page

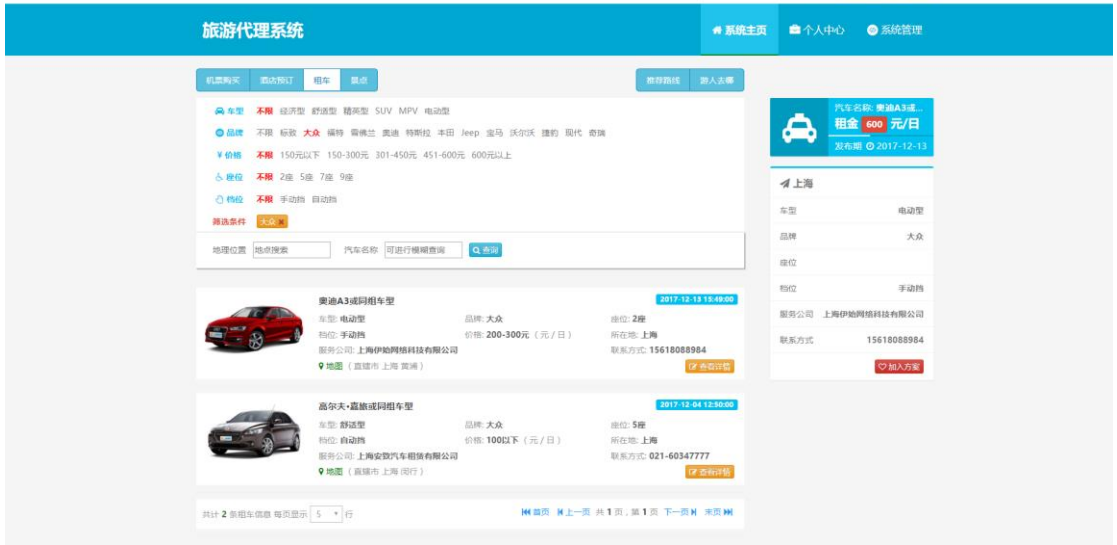
- 用户端——机票



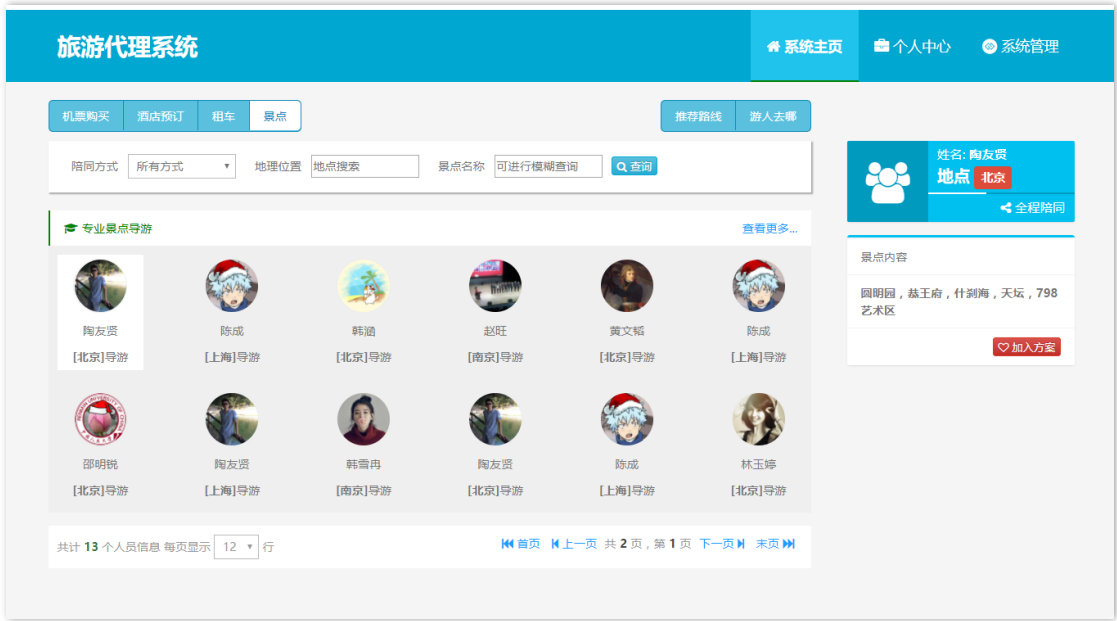
● 用户端——酒店



● 用户端——租车

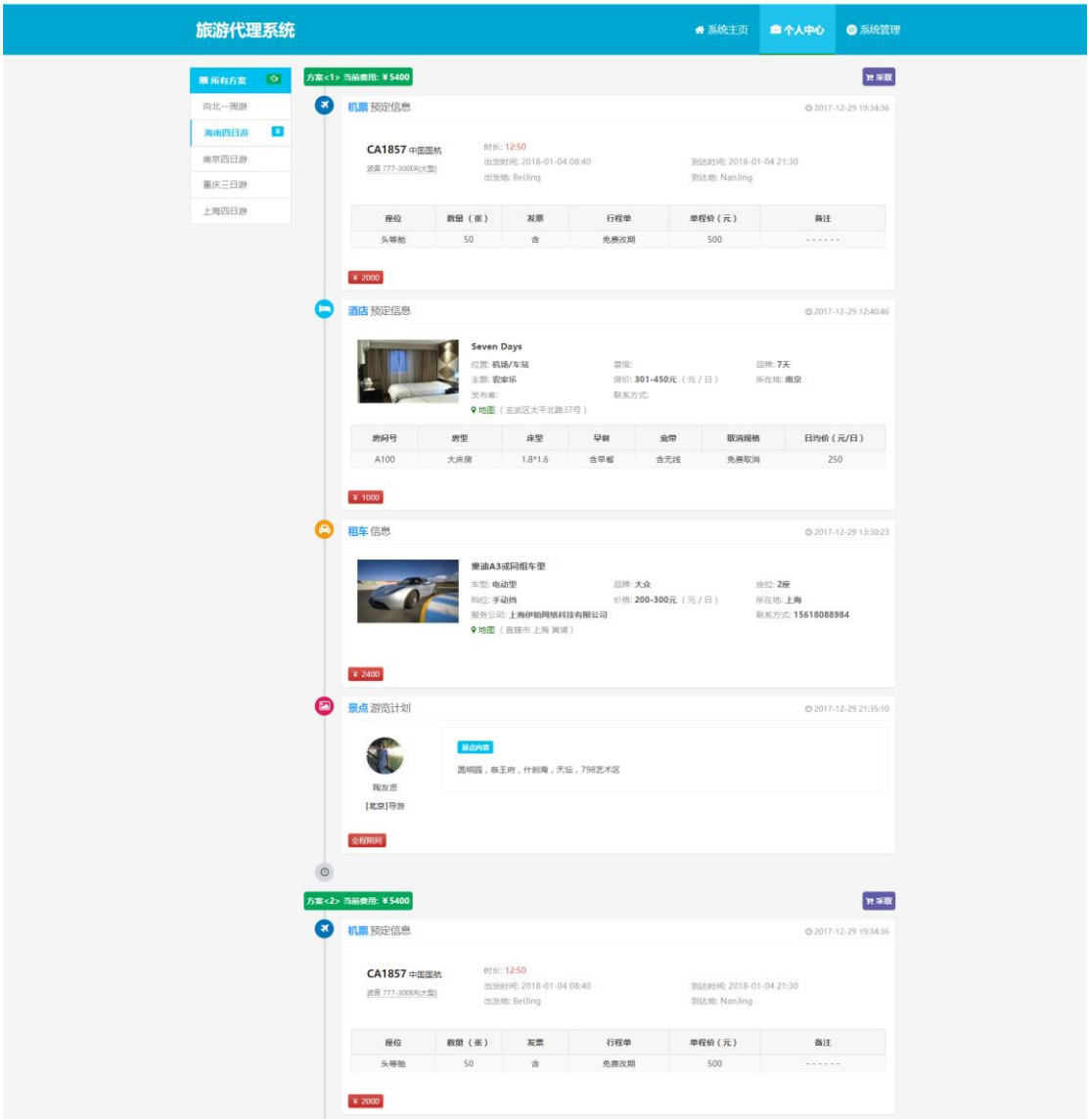


● 用户端——景点



● 用户端——个人中心





● 用户端——加入方案



## 5 报告总结

首先，从最开始的教学到后来的系统任务，可以说我们真的学习到了很多，从最开始对这类服务的不熟悉，在于实验室同学的沟通中慢慢了解需求、WebService 等，真正动手去做系统相关的开放，才真正了解自己的优缺点，学习到了 BEPL 处理多个服务的方式，写程序的过程中都是不断挑战自己的过程，也希望自己在这个过程中是不断享受的。

部分数据库、系统内容说明参照 github，在后面过程中会继续补充一些知识点，也会在博客中提及。