# 旅行记录本软件

# 需求分析与概要设计

## 项目说明

## 项目目标：

我们的项目简介主要为以下几个模块：

1. 地图管理：在用户进行旅游之前，可以在该软件进行旅游路线的规划，用户在软件中输入准备前往的地方，软件会形成路线，
2. 待办管理：用户还可以在软件上输入一些待办事项，在之后的过程中，用户随时可以查看待办事项，当待办事项完成后，用户可以消除该待办事项
3. 日志管理：在用户进行旅游时，可以向该软件传输一些图片，并且给图片增加注释，记录用户在旅游过程中所遇到的事情。用户还可以通过联网将自己的旅游经历分享到平台上，也可以浏览平台上其他用户的动态并添加至收藏或并下载。
4. 回忆板块：在用户完成旅游之后，这些资料会保存在本地数据库里，用户可以打开软件的回忆板块，就可以回顾以往的旅游记忆。

## 软硬件环境需求

1. **硬件配置：（只支持PC端）**

* 处理器:Intel(R) core(TM) i7
* 显示适配器:NVIDIA GeForce GTX
* 网络适配器:WAN Miniport

1. **软件配置：**

* 开发工具：Visual Studio、Visual Studio Code
* 测试工具：QTP
* 操作系统：Windows10
* 数据库：MySQL
* 驱动程序：所有硬件驱动

## 使用的关键技术：

1. 系统面临的技术难点主要集中在WEB API模块的设计、网络平台开发、数据库管理、系统维护三个方面，系统的开发采用了经典的MVC结构方式，这种方式非常适合系统程序的开发
2. 旅游信息管理又是一项重要繁重的工作，近年来，出现了大量的旅行管理应用。如何巧妙的实现系统的功能需求同时又实现一些新鲜的设计是本次课程设计的一个难点
3. 数据库的连接与管理

说明：信息管理系统的实现会产生和使用较为大量的数据，在保证任何情况下数据库的正常使用和数据安全是另一技术难点

1. 对不同优先级别、类型的日志的设计与实现

说明：根据行程的要求，在生成日志时会有不同的优先级和类型，对这些不同的日志进行外观，功能和级别的划分设计与实现是应用功能的最基本要求

1. C#语言在系统开发时的使用、编程技巧和算法设计

说明：C#的语言风格较为自由且对同一种功能的实现方式较多，因此需要选择简洁高效的实现方式

1. 团队开发的代码管理

说明：由于初次进行团队开发，每个人的代码风格和习惯都有较大差别，因此在项目中形成一个统一且较为友好的代码规范也是本次项目的难题所在

## 需求分析

## 系统用例

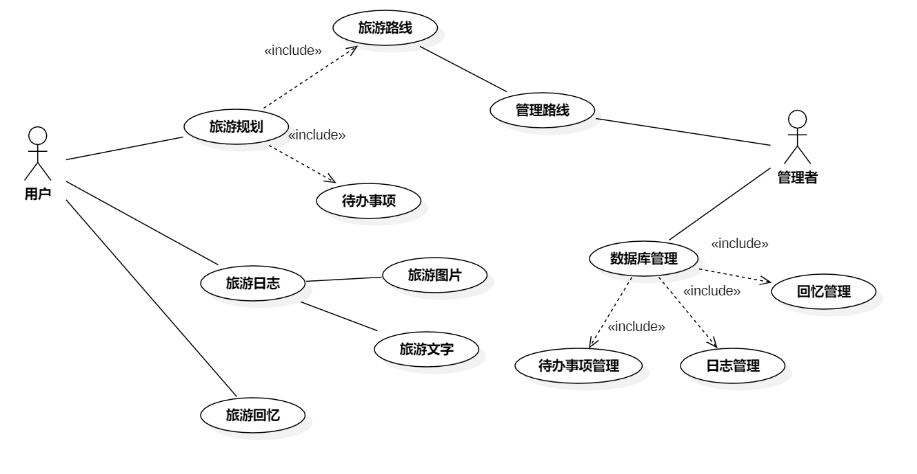


图 1 系统用例图

1. 在用户进行旅游之前，可以在该软件进行旅游路线的规划，用户在软件 中输入准备前往的地方，软件会形成路线，用户还可以在软件上输入一些待办事项。
2. 在用户进行旅游时，可以向该软件传输一些图片，并且可以添加一些感想，记录用户在旅游过程中所遇到的事情，这些图片与文字形成用户此次旅游的日志。
3. 在用户完成旅游之后，这些资料会保存在数据库里，用户可以打开软件的回忆板块，就可以回顾以往的旅游记忆。
4. 而管理者可以通过系统管理旅游的最佳路线，管理用户保存在系统数据库的各种数据。

## 业务流程

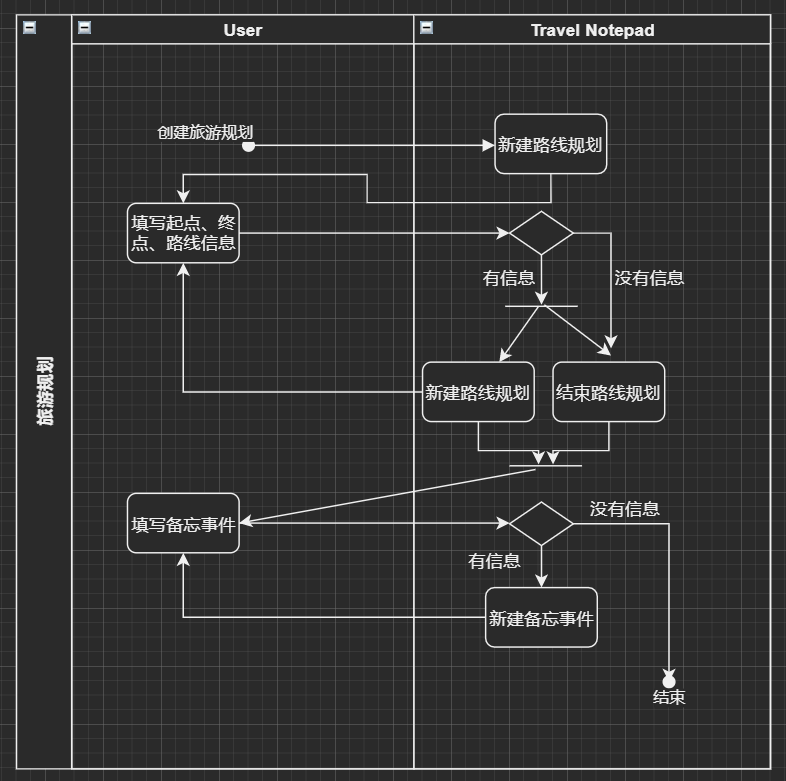


图3旅游规划用例活动图

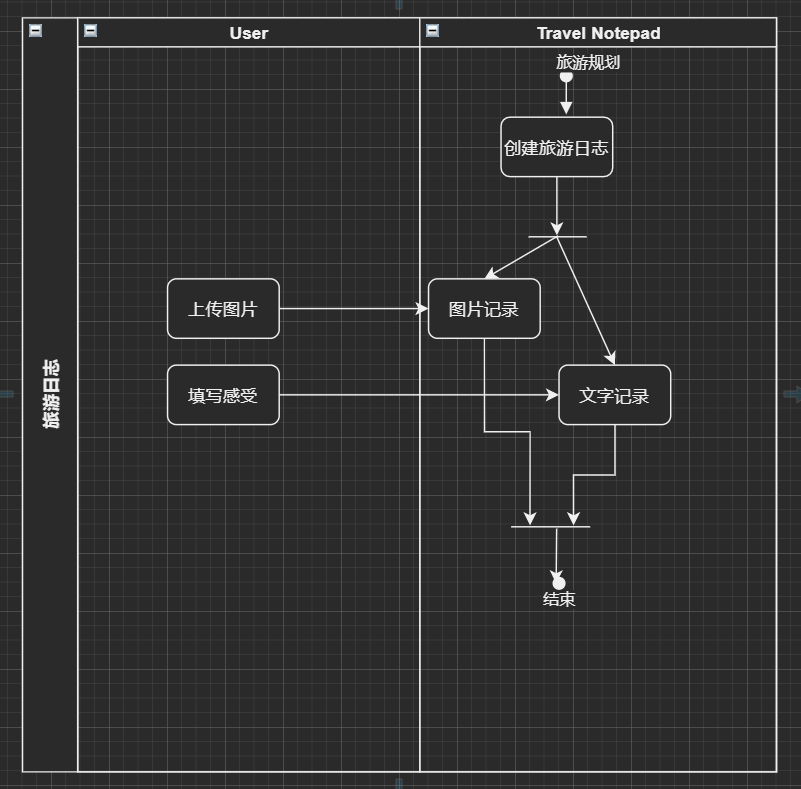


图2：旅游日志用例活动图

## 概要设计

## 功能模块设计

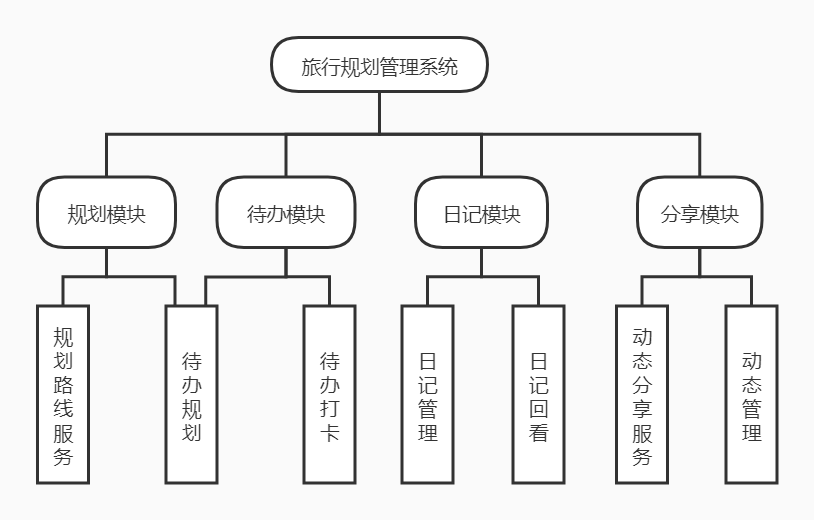


图4:功能模块图

1. 路线规划服务：

输入：用户对于路线行程的规划，即某个时间段所要游玩的地方。

功能概述：对用户的输入进行归纳处理，呈现一个易于用户查看和编辑的行程路线。

1. 待办模块：

输入：用户在旅游过程中想要做的一些事情

功能概述：基于这些事件生成待办列表，用户可以通过“打卡”来改变待办事件的状态，表示这个待办事件已完成。

1. 日记管理：

输入：用户在旅游过程中的所见所闻，可以针对图片做注释。

功能概述：将用户的输入保存到日记中，并提供对以往的日记进行修改等功能。

1. 日记回看：

功能概述：用户可以通过回忆板块，回顾以往的旅游记忆。

1. 动态分享服务：

输入：用户自己的旅游日志或者一些旅游心得等等。

功能概述：将用户的分享内容发布到分享平台上。

1. 动态管理：

功能概述：对所有用户分享的动态进行管理，包括向用户提供浏览、收藏、删除等功能。

## 核心类图

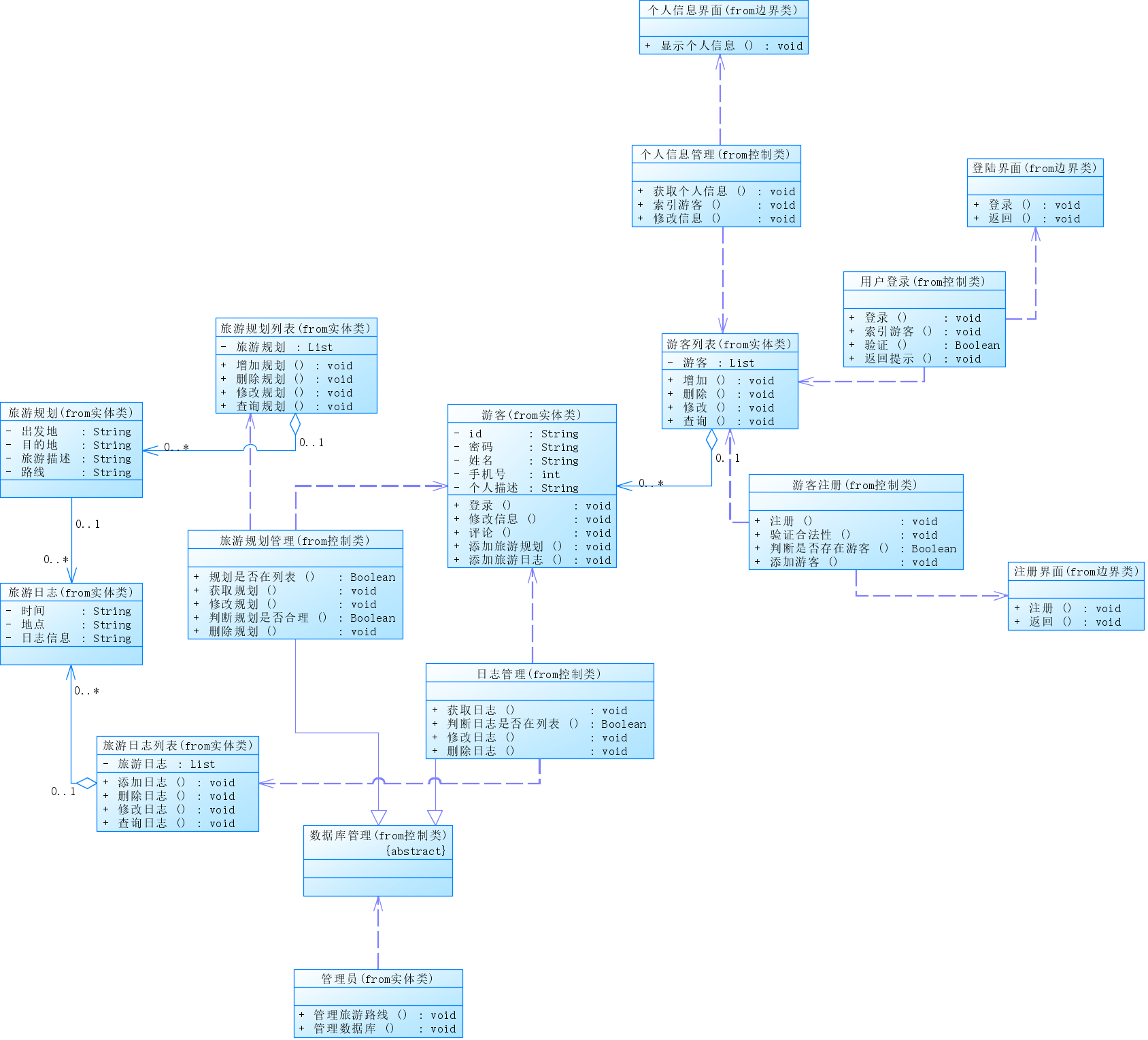


图5：核心类图

1. **类图描述**

本类图实现的功能有，游客添加旅游规划，对旅游添加日志，修改个人信息，登录，注册和管理员对日志、旅游计划进行修改，删除等。以边界类、控制类和实体类进行组织。

1. **类描述**

* 类名：游客列表

1. 整体说明：负责用于存储游客
2. 属性说明：游客：List，存储游客类，用于聚合游客类
3. 操作说明：+增加游客：；+删除游客：；+修改游客：；+查询游客：；

关系说明：聚合游客类

* 类名：游客

1. 整体说明：游客实体类，用于表示游客
2. 属性说明：id：string，游客账号；密码：string，游客登陆密码；姓名：string，游客姓名；手机号：int，游客手机
3. 个人描述：string，游客信息描述；
4. 操作说明：

+登录：游客登录；+修改信息：游客修改个人信息；+添加旅游规划：增加旅游规划；+添加日志：为每次旅游规划添加日志；

* 类名：旅游规划列表
  1. 整体说明：负责用于存储游客旅游规划
  2. 属性说明：旅游规划：List，存储旅游规划类，用于聚合旅游规划类
  3. 操作说明：+增加旅游规划：；+删除旅游规划：；+修改旅游规划：；+查询旅游规划：；
  4. 关系说明：聚合旅游规划类
* 类名：旅游规划

1. 整体说明：旅游规划实体类，用于表示旅游规划
2. 属性说明：出发地：string，游客出发位置；目的地：string，游客到达位置；旅游描述：string，游客的旅游行程安排等内容；路线：string，路线选择；

* 类名：旅游日志列表
  1. 整体说明：负责用于存储游客旅游日志
  2. 属性说明：旅游日志：List，存储旅游日志类，用于聚合旅游日志类
  3. 操作说明：+增加旅游日志：；+删除旅游日志：；+修改旅游日志：；+查询旅游日志：；
  4. 关系说明：聚合旅游日志类
* 类名：旅游日志

1. 整体说明：旅游日志实体类，用于表示旅游日志
2. 属性说明：时间：string，日志时间；地点：string，日志位置；日志描述：string，日志的内容；

* 类名：管理员
  1. 整体说明：最高权限用户，拥有所有权利
  2. 操作说明：+管理数据库：对游客和游客信息的管理；
* 类名：数据库管理

1. 整体说明：用于对所有信息进行管理，虚类
2. 关系说明：泛化旅游日志管理类和旅游计划管理类

* 类名：日志管理

1. 整体说明：负责管理员用于管理游客旅游日志
2. 操作说明：+获取其旅游日志：；+删除旅游日志：；+修改旅游日志：；+判断旅游日志是否在日志列表：；

* 类名：旅游规划管理

1. 整体说明：负责管理员用于管理游客旅游规划
2. 操作说明：+获取旅游规划：；+删除旅游规划：；+修改旅游规划：；+判断旅游规划是否在列表：；+判断规划是否合理：；

* 类名：游客注册

1. 整体说明：控制类，用于注册界面的逻辑控制
2. 操作说明：+注册：；+验证合法行：；+判断是否已存在游客：；+添加新游客：；

* 类名：注册界面

1. 整体说明：界面类，用于游客注册时的交互
2. 操作说明：+注册：输入信息进行注册；+返回：关闭此界面并返回调用界面；

* 类名：用户登录

1. 整体说明：控制类，负责用于管理用户登录界面的逻辑控制
2. 操作说明： +登录：；+索引游客：；+验证：；+返回提示：；

* 类名：登录界面

1. 整体说明：界面类，用于用户登录时的交互
2. 操作说明：+登录：输入信息进行登录；+返回：关闭此界面返回调用界面；

* 类名：个人信息管理

1. 整体说明：控制类，负责游客用于管理自身信息
2. 属性说明：+获取个人信息：；+索引游客：；+修改信息：；

* 类名：个人信息界面

1. 整体说明：界面类，用于游客管理自身信息时的交互
2. 操作说明：+显示个人信息；：
3. **继承描述**

旅游规划管理和旅游日志管理由数据库管理泛化，对系统数据(游客行程信息、行程记录等)进行操作管理

## 界面设计



图6旅程列表

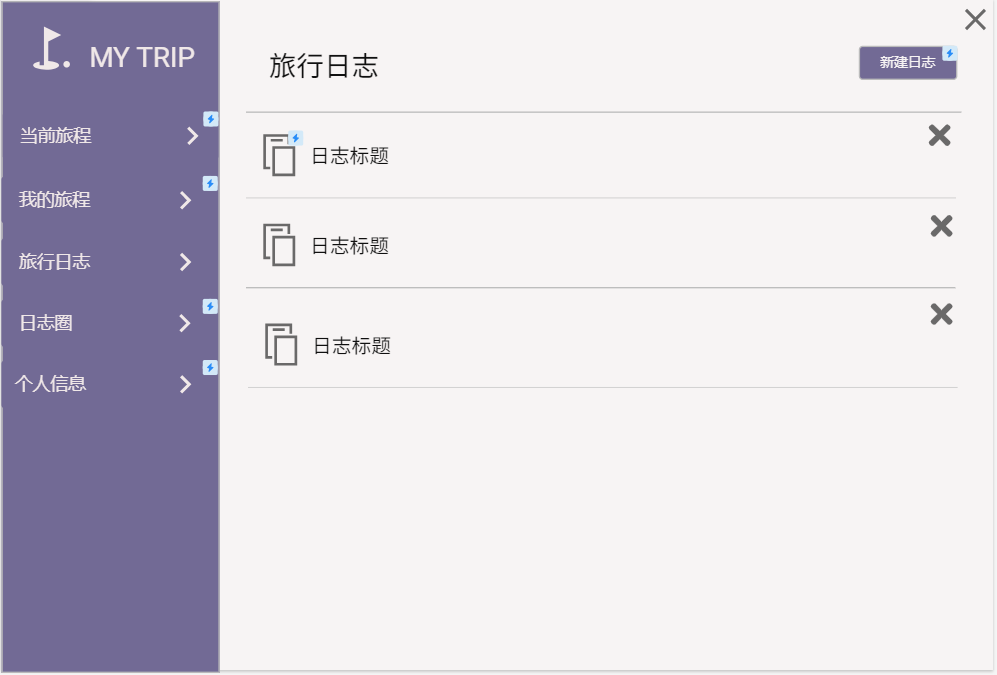


图7：日志列表



图8：旅游路线