

# 行迹精灵需求分析与概要设计

## 1. 项目说明

本项目开发了一款名为“行迹精灵”的旅行软件，旨在为用户提供全面的旅行规划和管理服务。通过集成路线规划、地图导航、天气预报追踪、待办管理、待办打卡和日志生成等功能，行迹精灵致力于简化旅行规划过程，提升旅行体验。

### 1.1 项目目标

- 为用户提供一个集成化的旅行规划平台，使他们能够轻松地输入旅行愿望和计划，并得自动化的路线规划。
- 通过地图导航和天气预报追踪功能，提供实时、实用的工具，帮助用户提前规划旅程路线，提升旅行幸福感。
- 通过待办管理和打卡功能，提供一个有效的方式来跟踪旅行计划和成就。
- 通过日志生成和日志管理功能，提供一个记录和回顾旅行经历的方式，增加旅行乐趣并促进分享与交流。

### 1.2 软硬件环境需求

为了保证“行迹精灵”的顺利运行和最佳性能，我们建议遵循以下软硬件环境：

#### (1) 软件环境：

- 操作系统：Windows11操作系统。
- 数据库：MySQL数据库
- 浏览器：支持Google Chrome、Mozilla Firefox、Safari等主流浏览器。
- 网络：稳定的互联网连接，支持HTTP/HTTPS协议。
- 驱动程序：支持所有硬件驱动
- 开发工具：Visual Studio2022

#### (2) 硬件环境：

- 处理器：Intel Core i5或相当以上的处理器；
- 内存：8GB RAM或更高；
- 硬盘：200GB SSD或更高容量，以便存储软件数据和日志文件；
- 显示器：1080p分辨率或更高，以获得更好的视觉体验。

## 1.3 使用的关键技术

### (1) Winform技术

为了保证各个功能模块拥有单独的操作的界面，我们采用Winform程序来搭建桌面应用程序的用户界面，并协调各界面之间的跳转。同时为了增强用户的使用体验，我们要对Winform界面的各个控件进行美化。在搭建过程中，我们需要处理多种控件的布局 and 交互，以及与服务端的数据同步问题。

### (2) Web API设计

为了实现应用程序的服务端逻辑，我们设计并实现了Web API模块。该模块使用ASP.NET Core来构建，以提供可靠的数据服务和业务逻辑支持。Web API模块的设计考虑了安全性、扩展性和性能优化。通过合理的路由设计和中间件的使用，我们确保了API的高效运行和安全访问。同时，使用Swagger等工具来生成API文档，并利用postman进行API功能的调试，提高了开发和维护的便利性。

### (3) 数据库的连接与管理

在软件开发过程中，数据库的连接与管理是关键环节。为了存储和管理应用数据，我们使用了MySQL数据库。通过C#中的ADO.NET技术和Entity Framework (EF) 框架，我们实现了与数据库的连接和数据操作。EF框架使得数据访问更加简单和直观，极大地提高了开发效率。

数据库的设计和优化是关键，需要确保数据的一致性、完整性和效率。同时，在不同的开发和测试环境中保持数据同步也是一个挑战。软件功能包括用户注册、信息修改和日志记录，这些信息都需要在数据库中进行管理和维护。使用过程中，保证数据的一致性也是一个重要的挑战。

### (4) C#开发语言的使用

C#是一种面向对象的编程语言，是.NET框架的主要开发语言之一，它在.NET框架下具有广泛的应用。C#提供了强大的类型系统、内存管理和跨平台支持，因此我们选用C#作为开发的主要语言。在项目开发过程中，我们需要处理复杂的业务逻辑和多线程编程，这对我们的编码能力和对C#特性的理解提出了较高要求。

### (5) API的调用

为了实现与其他系统的集成和数据交换，我们选择调用了第三方API服务，如在地图导航和点亮城市的功能模块中，我们选择调用高德地图的API服务。API的调用通常涉及到HTTP请求、数据序列化和反序列化，以及异常处理。API的稳定性、响应速度和文档完整性是我们选择API时需要考虑的关键因素。

### (6) 图片文件上传技术

图片文件上传的关键技术包括使用HttpClient类进行文件上传和下载，通过配置基本地址和SSL证书验证来创建HttpClient，利用MultipartFormDataContent类构建请求内容并添加文件流发送POST请求，同时通过GET请求获取文件内容并转化为字节数组或图片对象。在服务器端，通过ASP.NET Core中的路由与控制器设置文件上传、下载和删除的API端点，利用MultipartRequestHelper解析multipart/form-data请求，使用FileHelper处理文件流并验证文件扩展名和签名，确保文件安全。文件安全性通过验证文件扩展名和检查文件签名来防止文件伪造，错误处理与日志通过模型状态检查和异常处理来捕获和记录在上传、下载和删除过程中的问题。文件存储则通过从配置文件中读取存储路径，并使用FileStream和System.IO在服务器指定目录中创建、读取、写入和删除文件。

## 2. 需求分析

### 2.1 系统用例

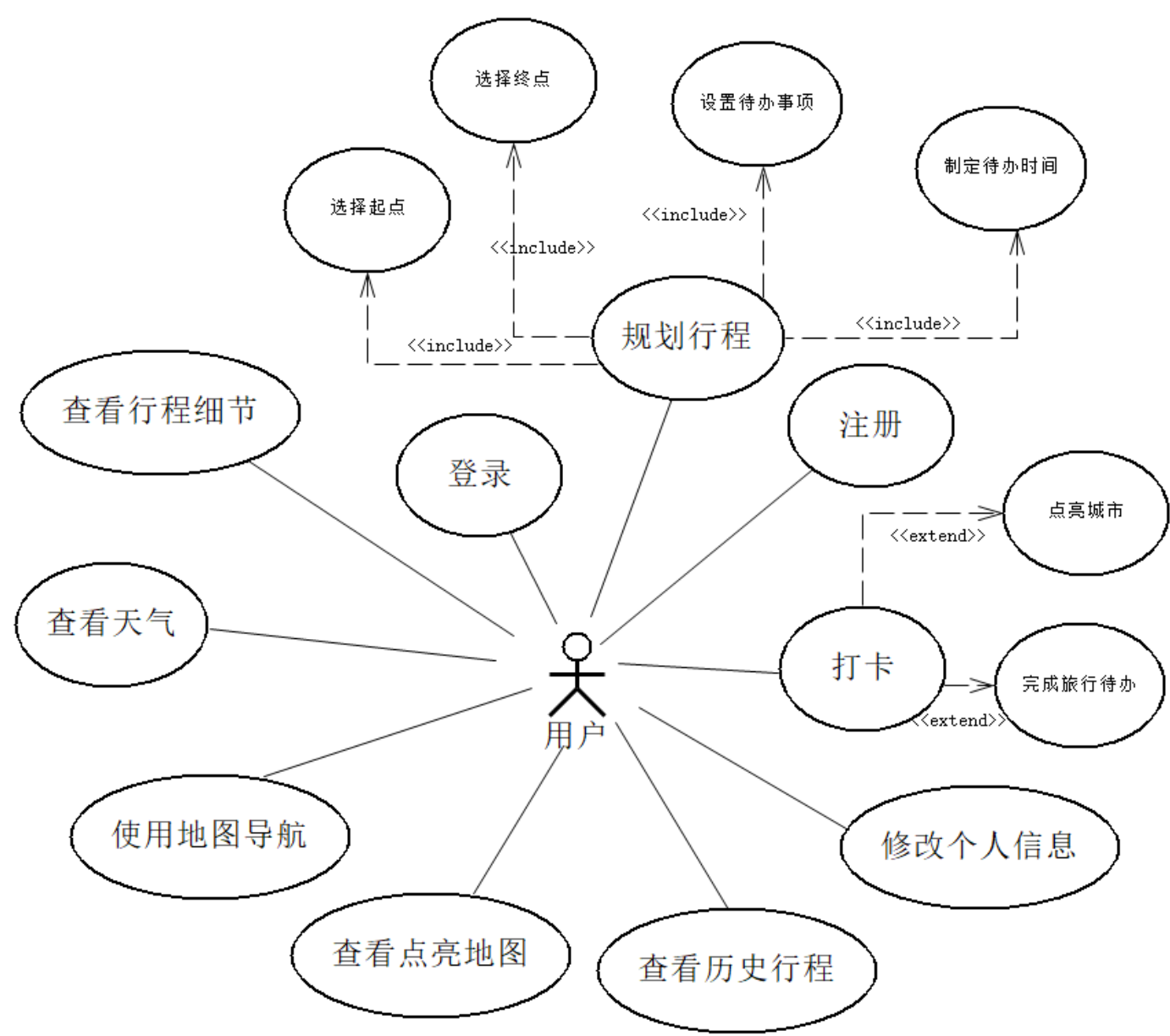


图2-1 系统用例图

(1) 规划行程：

参与者：用户

基本事件流：用户在规划行程界面，可以选择本次行程的起点、终点，写下相应的待办事项，制定相应的完成时间来新建行程。

(2) 查看行程细节：

参与者：用户

基本事件流：用户点击“行程”按钮可以查看每个行程的细节信息。

### (3) 查看天气：

参与者：用户

基本事件流：用户可以在天气页面查看当前天气以及今后几天天气预报。

### (4) 使用地图导航：

参与者：用户

基本事件流：用户可以在导航页面输入想去的目的地，跟随导航到达目的地。

### (5) 打卡：

参与者：用户

基本事件流：用户在打卡界面，可以选择点亮相应的城市，也可选择添加相应的碎片旅行记录。

### (6) 查看点亮地图：

参与者：用户

基本事件流：用户在点亮地图页面可以查看自己已点亮的所有地区的地图。

### (7) 查看历史行程：

参与者：用户

基本事件流：用户点击“行程”按钮可以查看历史行程。

### (8) 注册、登录、修改个人信息：

参与者：用户

基本事件流：用户点击“注册”按钮可以注册个人信息（姓名、密码）系统随机生成用户ID。

用户点击“登录”按钮可以登录系统。

用户点击“个人信息修改”按钮可以修改个人信息（姓名，密码）。

## 2.2 业务流程

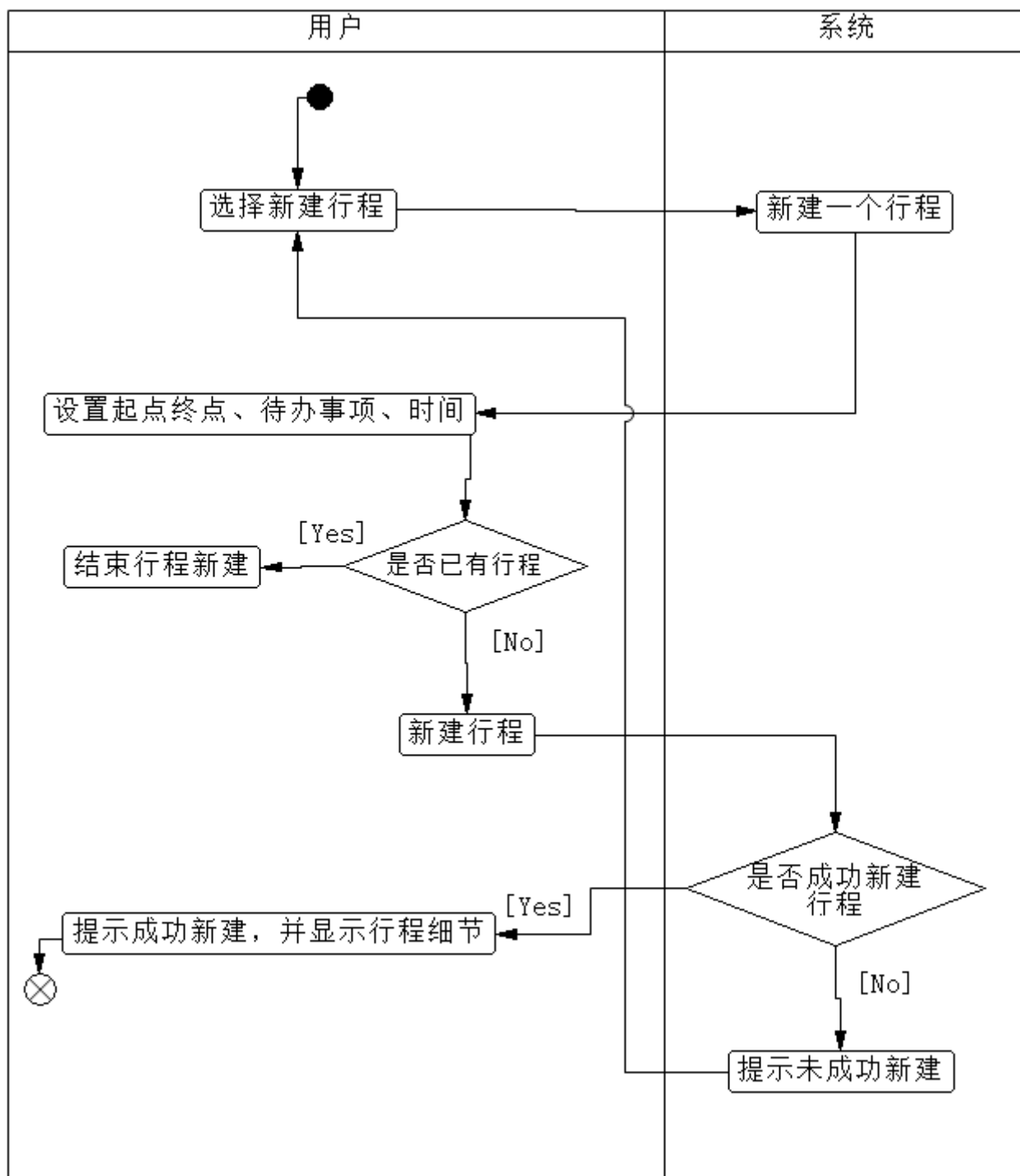


图2-2 规划行程活动图

## 3. 概要设计

### 3.1 功能模块设计

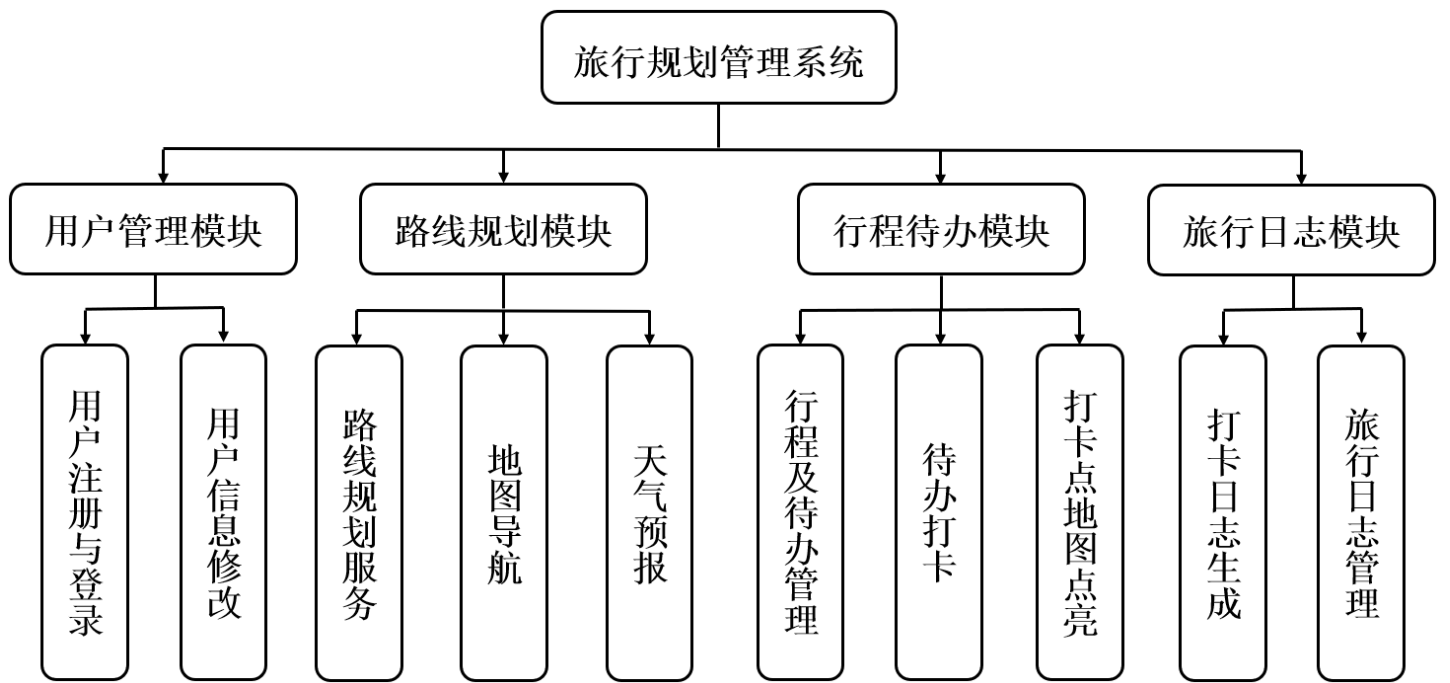


图3-1 功能模块图

## 用户注册

**输入：**新用户的名称、设置密码

**输出：**新用户的用户ID，密码被加密

**功能概述：**用户通过输入并确认自己用户名称与密码实现注册，系统将自动为其分配独特的用户ID号，并将原密码进行加密，加强隐私保护。

## 路线规划服务

**输入：**用户的待办行程地，即客户心仪的各个旅游点等

**输出：**用户的旅游路线规划

**功能概述：**用户通过输入并确认待办行程地点获得路线规划，并将具体路线以页面进行可视化展示。

## 地图导航

**功能概述：**用户可通过路线规划模块的内置地图获取到实时导航服务，以保障用户的出行便利。

## 天气预报

**功能概述：**路线规划模块中的天气板块将对旅行行程地点进行实时的天气预报，帮助用户避开较差或恶劣的天气气候，以保障用户的安全与出行的愉悦。

## 待办管理

**输入：**用户准备进行的旅游计划

**输出：**待办列表增加

**功能概述：**基于待办行程事件生成待办列表，用户可以对其进行增加、修改、删除，或是通过“打卡”使待办事件被完成并从列表移除。

## 待办打卡

**输入：**用户未完成的待办事件

**输出：**待办事件完成，打卡点于地图“点亮”（即标注），转入日志生成

**功能概述：**用户通过“打卡”来改变待办事件的状态，若“已打卡”则表示这个待办事件已完成，本软件将自动为已打卡地点于系统内置地图中进行“点亮”标注操作，然后自动跳转至新日志编辑生成页面。

## 日志生成

**输入：**用户打卡时的地点天气信息以及旅行感想如心情、所见所闻等

**输出：**日志回忆列表增加

**功能概述：**用户在打卡时通过选择输入自己旅行时的感想等等，依靠系统生成旅行日志列表。

## 日志管理

**功能概述：**用户后续可以在日记回忆页面对之前生成的日志进行查看、修改或删除。

## 3.2 核心类图

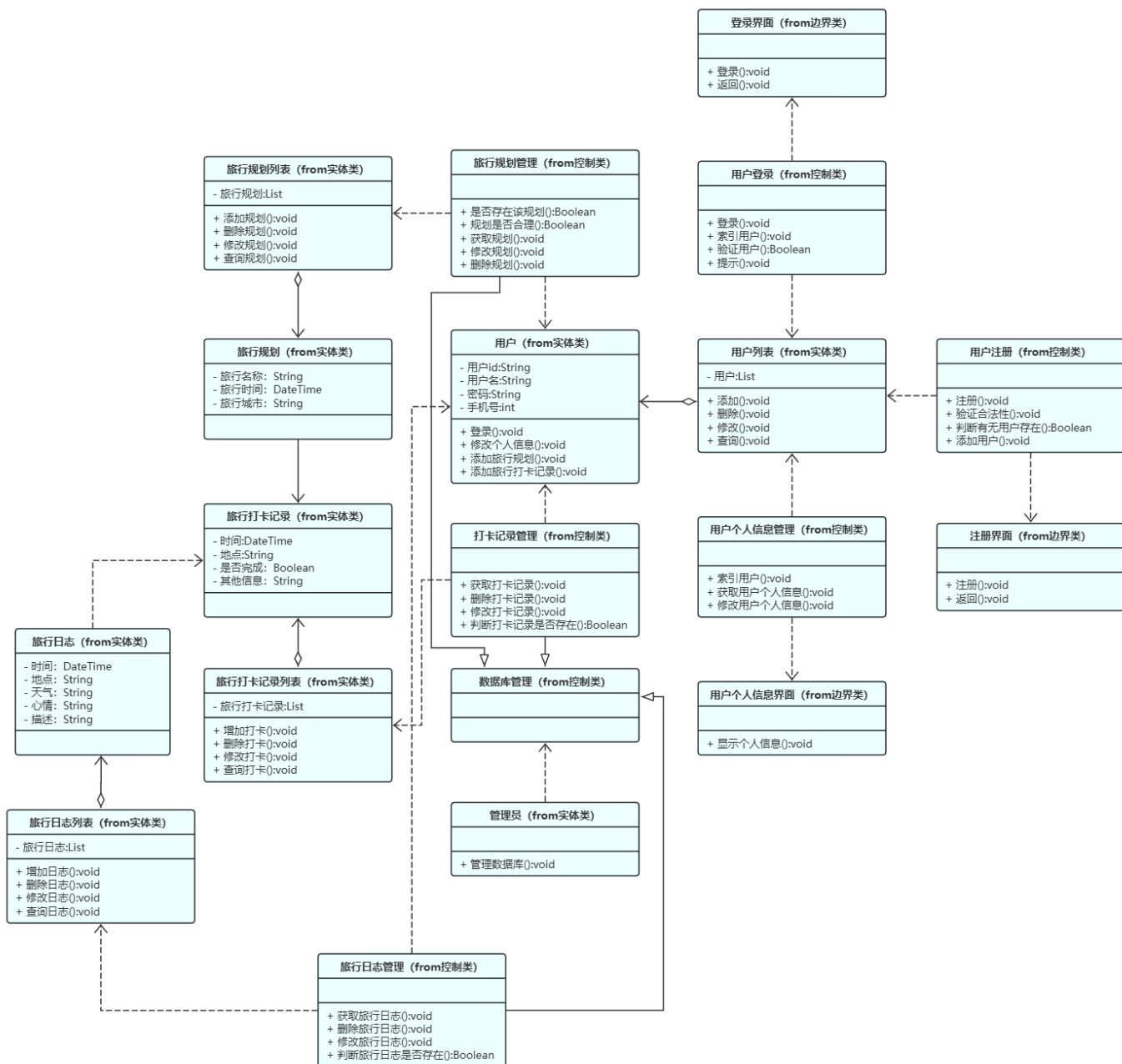


图3-2 核心类图

类的描述：

类名：用户

①整体说明：用户实体类，用于表示用户

②属性说明：id: string，用户账号信息；密码: string，用户登录密码；姓名: string，用户姓名；手机号: int，用户手机

③操作说明：+登录：用户登录；+修改信息：用户修改个人信息；+添加旅行规划：增加旅行规划；+添加旅行打卡记录：为每次旅行规划添加打卡记录；

类名：用户列表

①整体说明：负责用于存储用户

②属性说明：用户: List，存储用户类，用于聚合用户类



③操作说明：+添加用户：；+删除用户：；+修改用户：；+查询用户：；

④关系说明：聚合游客类

### 类名：旅行规划

①整体说明：旅行规划实体类，用于表示旅行规划

②属性说明：旅行城市：string，用户旅行的目的地；旅行规划名称：string，记录用户此次出行的名称；旅行时间：DateTime，用户出行时间

### 类名：旅行规划列表

①整体说明：负责用于存储用户的旅行规划

②属性说明：旅行规划：List，存储旅行规划类，用于聚合旅行规划类

③操作说明：+增加旅行规划：；+删除旅行规划：；+修改旅行规划：；+查询旅行规划：；

④关系说明：聚合旅行规划类

### 类名：旅游打卡记录

①整体说明：旅行打卡记录类，用于表示旅游打卡记录

②属性说明：时间：DateTime，记录时间；地点：string，记录位置；是否完成：Boolean，记录打卡是否完成；其他信息：string，包括打卡详情信息，用于记录到日志中

### 类名：旅行打卡记录列表

①整体说明：负责用于存储用户旅行打卡记录

②属性说明：旅游打卡记录：List，存储旅行打卡记录类，用于聚合旅行打卡记录类

③操作说明：+增加旅行打卡记录：；+删除旅行打卡记录：；+修改旅行打卡记录：；+查询旅行打卡记录：；

④关系说明：聚合旅游打卡记录类

### 类名：旅行日志

①整体说明：实体类，用于整合一次旅行的所有打卡记录

②属性说明：时间：DateTime，记录该条日志的时间；地点：string，记录该条日志的地点；天气：string，记录创建该条日志时的天气；心情：string，记录创建该条日志时的心情；描述：string，记录该日志时的具体心情

### 类名：旅行日志列表

①整体说明：负责用于存储旅行日志

②属性说明：旅行日志：List，存储旅行日志类，用于聚合旅行日志类

③操作说明：+添加旅行日志：；+删除旅行日志：；+修改旅行日志：；+查询旅行日志：；

④关系说明：聚合旅行日志类

## 类名：旅行日志管理

①整体说明：负责管理员用于管理用户旅行日志信息

②操作说明：+获取旅行日志信息：；+删除旅行日志信息：；+修改旅行日志信息：；+判断旅行日志是否在列表：；

## 类名：旅行打卡记录管理

①整体说明：负责管理员用于管理用户旅行打卡记录

②操作说明：+获取其旅行打卡记录：；+删除旅行打卡记录：；+修改旅行打卡记录：；+判断旅行打卡记录是否在列表中：；

## 类名：旅行规划管理

①整体说明：负责管理员用于管理用户旅行规划

②操作说明：+获取旅行规划：；+删除旅行规划：；+修改旅行规划：；+判断旅行规划是否在列表：；+判断规划是否合理：；

## 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：泛化地图点亮、旅行打卡记录管理类和旅行规划管理类

## 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：用户注册

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：；+验证合法性：；+判断是否已存在用户：；+添加新用户：；

## 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：；+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：用户个人信息管理

- 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：用户个人信息界面

- 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；：

4. 页面设计

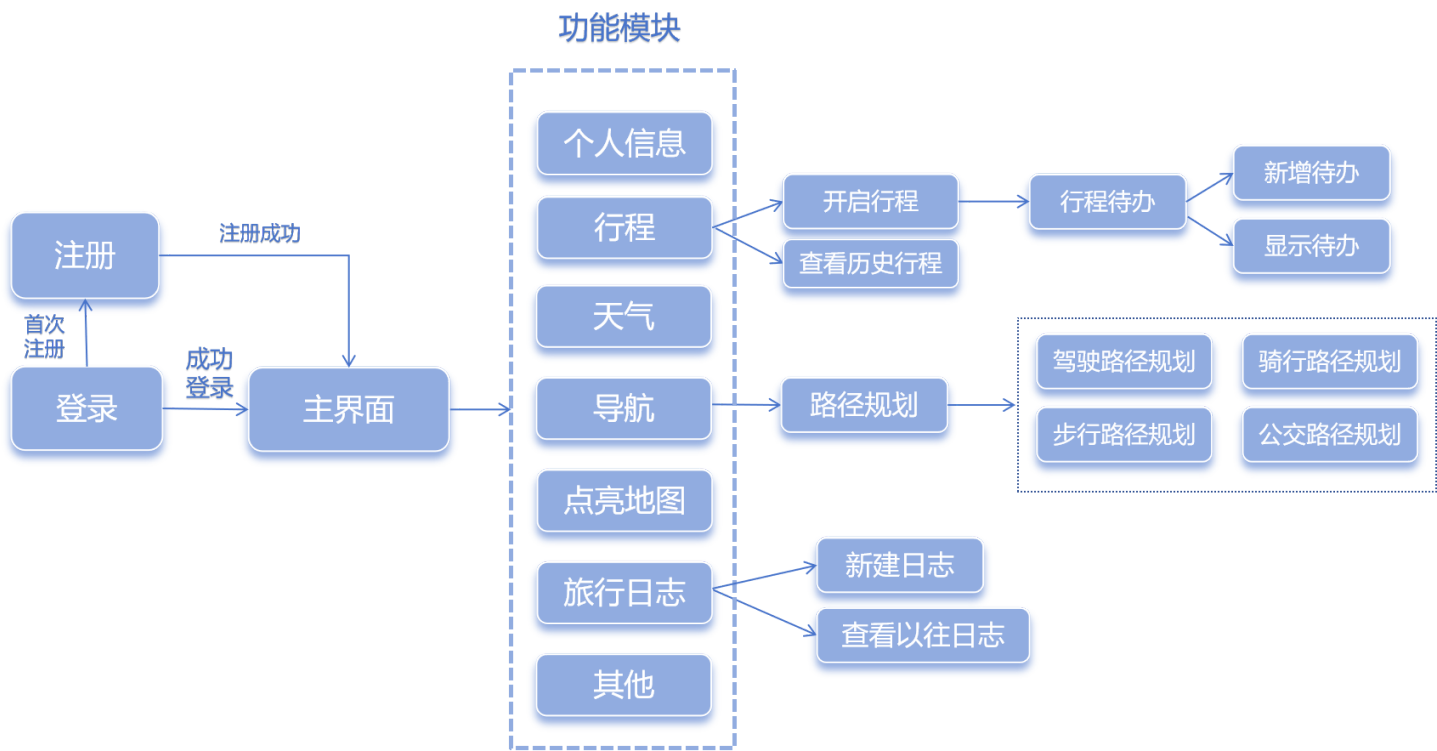


图4-1 页面流程图



图4-2 系统界面1



图4-3 系统界面2



图4-4 系统界面3

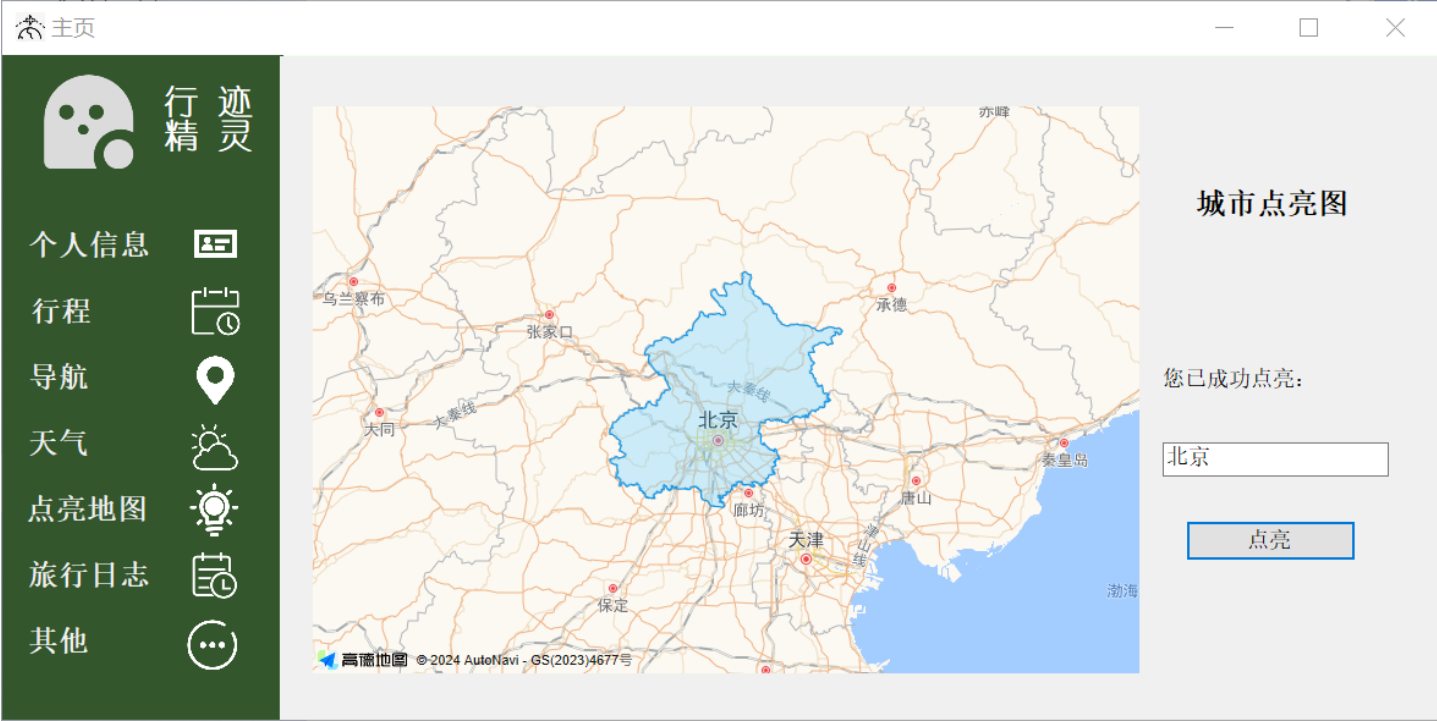


图4-5 系统界面4



图4-6 系统界面5



图4-7 系统界面6



图4-8 系统界面7