

AI旅行规划软件设计文档

1. 项目概述

1.1 项目名称

AI Travel Planner - 智能旅行规划助手

1.2 项目描述

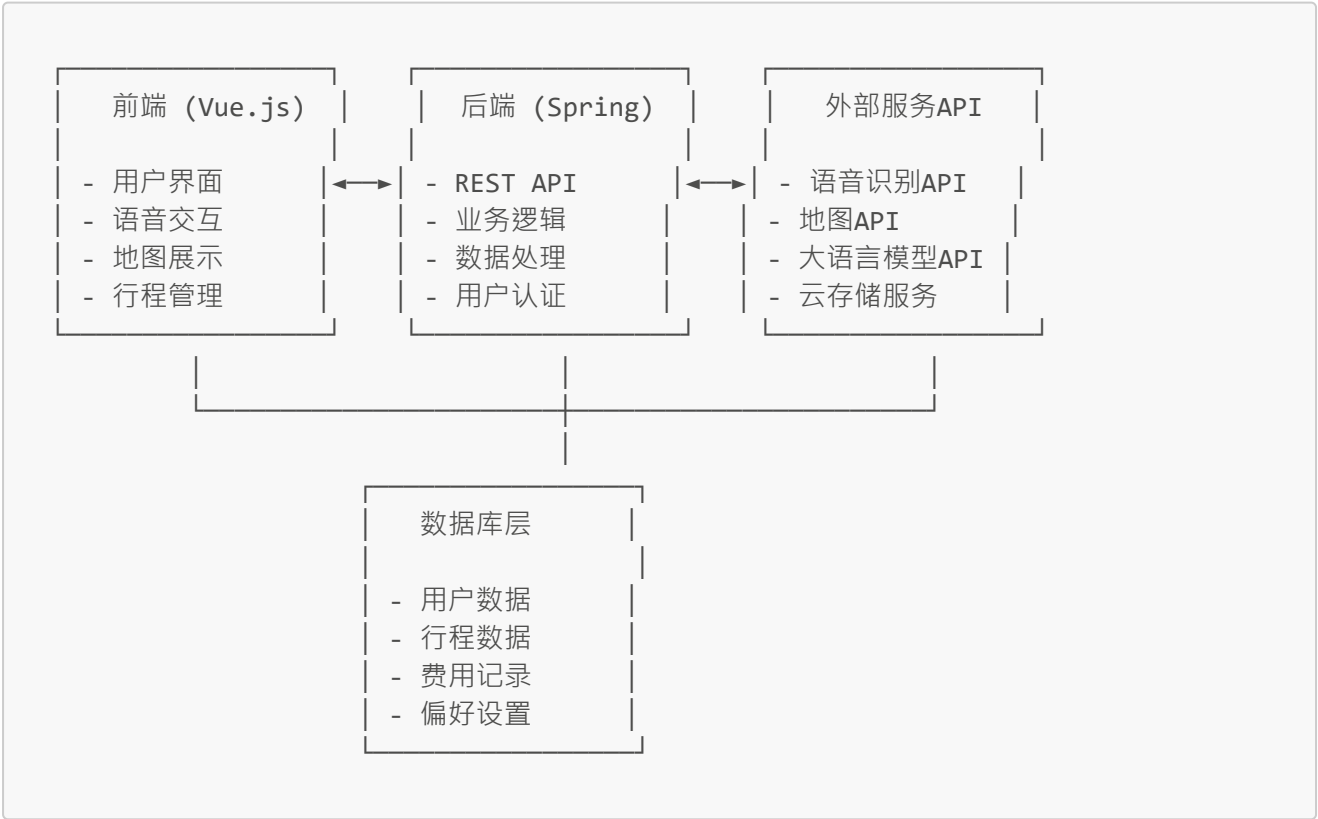
基于AI技术的智能旅行规划软件，通过语音识别和自然语言处理技术，为用户提供个性化的旅行路线规划、费用预算管理和实时旅行辅助服务。

1.3 项目目标

- 简化旅行规划过程
- 提供个性化的旅行建议
- 实现智能费用预算管理
- 支持多设备数据同步

2. 系统架构

2.1 整体架构



2.2 技术架构

- **前端:** Vue 3 + TypeScript + Vite + Element Plus

- 后端: Spring Boot + Spring Security + Spring Data JPA
- 数据库: PostgreSQL (主数据库) + Redis (缓存)
- 部署: Docker + Docker Compose
- **CI/CD**: GitHub Actions

3. 核心功能模块

3.1 用户管理模块

- **用户注册/登录**: 支持邮箱注册、第三方登录
- **用户资料管理**: 个人信息、旅行偏好设置
- **权限管理**: 基于角色的访问控制

3.2 智能行程规划模块

- **语音输入**: 集成科大讯飞语音识别API
- **自然语言处理**: 解析用户需求 (目的地、日期、预算、偏好等)
- **AI行程生成**: 调用大语言模型API生成个性化行程
- **行程优化**: 基于交通、时间、费用等因素优化路线

3.3 地图与导航模块

- **地图展示**: 集成高德地图API
- **位置服务**: 获取用户当前位置
- **路线规划**: 提供多种交通方式的路线选择
- **实时导航**: 支持步行、驾车、公交导航

3.4 费用预算管理模块

- **预算分析**: AI自动分析和预估旅行费用
- **费用记录**: 支持语音录入消费记录
- **预算跟踪**: 实时监控预算使用情况
- **费用统计**: 提供详细的费用分析报告

3.5 数据同步模块

- **云端存储**: 行程数据、用户偏好云端同步
- **多设备支持**: 支持手机、平板、电脑多设备访问
- **离线缓存**: 关键数据本地缓存, 支持离线查看

4. 数据库设计

4.1 用户表 (users)

```
CREATE TABLE users (  
    id BIGSERIAL PRIMARY KEY,  
    username VARCHAR(50) UNIQUE NOT NULL,  
    email VARCHAR(100) UNIQUE NOT NULL,
```

```
password_hash VARCHAR(255) NOT NULL,
avatar_url VARCHAR(255),
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

4.2 用户偏好表 (user_preferences)

```
CREATE TABLE user_preferences (
    id BIGSERIAL PRIMARY KEY,
    user_id BIGINT REFERENCES users(id),
    travel_style VARCHAR(50), -- 旅行风格：休闲、冒险、文化等
    budget_level VARCHAR(20), -- 预算等级：经济、中等、豪华
    accommodation_type VARCHAR(50), -- 住宿偏好
    transportation_preference VARCHAR(50), -- 交通偏好
    food_preference TEXT[], -- 美食偏好
    interests TEXT[], -- 兴趣爱好
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

4.3 旅行计划表 (travel_plans)

```
CREATE TABLE travel_plans (
    id BIGSERIAL PRIMARY KEY,
    user_id BIGINT REFERENCES users(id),
    title VARCHAR(200) NOT NULL,
    destination VARCHAR(100) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    total_budget DECIMAL(10,2),
    traveler_count INTEGER DEFAULT 1,
    status VARCHAR(20) DEFAULT 'draft', -- draft, confirmed, completed
    ai_generated_content JSONB, -- AI生成的详细行程
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

4.4 行程详情表 (itinerary_items)

```
CREATE TABLE itinerary_items (
    id BIGSERIAL PRIMARY KEY,
    travel_plan_id BIGINT REFERENCES travel_plans(id),
    day_number INTEGER NOT NULL,
    start_time TIME,
```

```

end_time TIME,
activity_type VARCHAR(50), -- 景点、餐厅、住宿、交通等
title VARCHAR(200) NOT NULL,
description TEXT,
location_name VARCHAR(200),
latitude DECIMAL(10, 8),
longitude DECIMAL(11, 8),
estimated_cost DECIMAL(8,2),
booking_info JSONB,
created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

4.5 费用记录表 (expense_records)

```

CREATE TABLE expense_records (
  id BIGSERIAL PRIMARY KEY,
  travel_plan_id BIGINT REFERENCES travel_plans(id),
  category VARCHAR(50) NOT NULL, -- 交通、住宿、餐饮、门票等
  amount DECIMAL(8,2) NOT NULL,
  currency VARCHAR(3) DEFAULT 'CNY',
  description TEXT,
  expense_date DATE NOT NULL,
  location VARCHAR(200),
  receipt_url VARCHAR(255),
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

5. API设计

5.1 用户认证API

POST /api/auth/register	# 用户注册
POST /api/auth/login	# 用户登录
POST /api/auth/logout	# 用户登出
GET /api/auth/profile	# 获取用户信息
PUT /api/auth/profile	# 更新用户信息

5.2 行程规划API

POST /api/plans	# 创建旅行计划
GET /api/plans	# 获取用户的旅行计划列表
GET /api/plans/{id}	# 获取特定旅行计划详情
PUT /api/plans/{id}	# 更新旅行计划
DELETE /api/plans/{id}	# 删除旅行计划

POST /api/plans/generate	# AI生成行程
POST /api/plans/{id}/optimize	# 优化行程

5.3 语音处理API

POST /api/voice/recognize	# 语音识别
POST /api/voice/synthesize	# 语音合成

5.4 地图服务API

GET /api/map/search	# 地点搜索
GET /api/map/route	# 路线规划
GET /api/map/nearby	# 附近推荐

5.5 费用管理API

POST /api/expenses	# 添加费用记录
GET /api/expenses/{planId}	# 获取计划的费用记录
PUT /api/expenses/{id}	# 更新费用记录
DELETE /api/expenses/{id}	# 删除费用记录
GET /api/expenses/{planId}/stats	# 获取费用统计

6. 外部服务集成

6.1 语音识别服务

- **服务商:** 科大讯飞
- **功能:** 实时语音转文字、语音合成
- **集成方式:** REST API + WebSocket

6.2 地图服务

- **服务商:** 高德地图
- **功能:** 地图展示、地点搜索、路线规划、导航
- **集成方式:** JavaScript SDK + REST API

6.3 大语言模型服务

- **服务商:** 阿里云百炼平台 (或其他LLM服务)
- **功能:** 行程规划、预算分析、智能推荐
- **集成方式:** REST API

6.4 云存储服务

- **服务商:** 阿里云OSS (或AWS S3)
- **功能:** 文件存储、图片上传
- **集成方式:** SDK

7. 安全设计

7.1 认证与授权

- JWT Token认证
- 基于角色的访问控制(RBAC)
- API密钥管理

7.2 数据安全

- 密码加密存储(BCrypt)
- HTTPS传输加密
- 敏感数据脱敏

7.3 API安全

- 请求频率限制
- 参数验证
- SQL注入防护

8. 性能优化

8.1 前端优化

- 组件懒加载
- 图片懒加载
- 代码分割
- CDN加速

8.2 后端优化

- Redis缓存
- 数据库索引优化
- 连接池配置
- 异步处理

8.3 数据库优化

- 查询优化
- 索引设计
- 分页查询
- 读写分离

9. 部署方案

9.1 Docker化部署

```
# docker-compose.yml
version: '3.8'
services:
  frontend:
    build: ./frontend
    ports:
      - "80:80"

  backend:
    build: ./backend
    ports:
      - "8080:8080"
    depends_on:
      - database
      - redis

  database:
    image: postgres:13
    environment:
      POSTGRES_DB: travel_planner
      POSTGRES_USER: ${DB_USER}
      POSTGRES_PASSWORD: ${DB_PASSWORD}

  redis:
    image: redis:6-alpine
```

9.2 CI/CD流程

1. 代码提交到GitHub
2. GitHub Actions自动构建
3. 运行测试用例
4. 构建Docker镜像
5. 推送到阿里云镜像仓库
6. 自动部署到生产环境

10. 开发计划

10.1 第一阶段 (基础功能)

- 用户注册登录系统
- 基础的行程创建和管理
- 简单的地图集成

10.2 第二阶段 (AI功能)

- 语音识别集成
- AI行程生成

- 智能推荐系统

10.3 第三阶段 (高级功能)

- 费用管理系统
- 实时导航
- 数据分析和报告

10.4 第四阶段 (优化完善)

- 性能优化
- 用户体验改进
- 移动端适配

11. 风险评估

11.1 技术风险

- 外部API服务稳定性
- 大语言模型响应时间
- 语音识别准确率

11.2 业务风险

- 用户接受度
- 数据隐私合规
- 成本控制

11.3 风险缓解措施

- 多服务商备选方案
- 本地缓存机制
- 渐进式功能发布

12. 测试策略

12.1 单元测试

- 后端业务逻辑测试
- 前端组件测试
- API接口测试

12.2 集成测试

- 前后端集成测试
- 外部服务集成测试
- 数据库集成测试

12.3 端到端测试

- 用户流程测试
- 性能测试
- 安全测试

本文档将随着项目开发进度持续更新和完善