



本科毕业设计

(2026届)

题 目：个人旅行规划与记忆系统的设计与实现

学 院：计算机与人工智能学院

专 业：网络工程

班 级：22 网工 2 班

姓 名：郑皓

学 号：22211835233

指导教师：徐晓华

完成日期：2026 年 1 月 20 日

计算机与人工智能学院本科毕业设计（论文）

诚信承诺书

- 本人郑重地承诺所呈交的毕业设计（论文），是在指导教师徐晓华老师的指导下严格按照学校和学院有关规定完成的。
- 本人在毕业论文（设计）中引用他人的观点和参考资料均加以注释和说明。
- 本人承诺在毕业论文（设计）选题和研究过程中没有抄袭他人研究成果和伪造相关数据等行为。
- 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

毕业论文（设计）作者签名：

班级：_____ 学号：_____
____ 年 ____ 月 ____ 日

目 录

摘 要	I
Abstract.....	II
1 引言.....	1
1.1 课题的背景与意义	1
1.2 相关研究综述	1
1.2.1 国外研究现状	1
1.2.2 国内研究现状	2
1.3 本课题的主要研究内容	2
1.4 论文组织结构	2
2 系统需求分析	3
2.1 业务需求	3
2.2 功能需求	3
3 系统设计	4
3.1 总体架构	4
3.2 数据库设计	6
3.3 表设计说明	6
3.4 数据库 ER 图	9
3.5 功能模块设计	10
4 系统实现	11
4.1 前端界面与功能实现	11
4.2 后端功能与关键技术实现	12
5 系统测试	13
5.1 测试环境	13

5.2 测试用例与结果	13
5.2.1 用户注册功能测试步骤	13
5.2.2 发布旅行记录	13
5.2.3 编辑与删除游记	14
5.2.4 旅行计划的创建与查看	14
5.2.5 评论与点赞功能	14
5.3 测试总结	15
5.4 表格的使用	15
5.5 公式的使用	15
5.6 图片的插入和交叉引用	16
5.6.1 图片的使用	16
5.6.2 多图排版样例	16
5.6.3 交叉引用	16
5.6.4 本地安装软件或者线上使用	17
6 参考文献说明	18
致谢	19
参考文献	20
附录 A 补充更多细节	22
A.1 补充图	22
A.1.1 补充图	22
附录 B 附录 2	23

摘要

本文围绕“旅行记忆系统”的设计与实现展开研究。该系统旨在为用户提供一个一站式的平台，用于规划行程、记录旅行过程以及分享交流旅行体验。文章首先阐述了研究背景和意义：现代旅游爱好者日益需要一个集规划、记录、回忆和分享于一体的工具，但现有应用多侧重于单一环节，缺乏对完整旅行过程的支持 [1]。接着，本文分析了系统的业务需求和功能需求，设计了整体架构和数据库 E-R 模型，并详细描述了主要功能模块的设计与实现，包括用户管理、旅行记录管理、旅行计划制定以及社交分享功能等。系统采用前后端分离架构，前端利用 Vue3 框架与 Element Plus 组件库构建响应式界面，后端基于 SpringBoot 提供 RESTful API 服务，结合 MySQL 数据库存储数据，实现了良好的扩展性与性能 [2][3]。在系统实现部分，我们介绍了富文本日记编辑、地图足迹展示、多媒体文件上传等关键技术细节。随后，文章通过功能测试和性能测试验证了系统的可靠性与易用性，结果表明系统功能完整，运行稳定。本文的研究和实践表明，整合旅行规划与记忆管理的平台可以有效提升用户旅行体验，为智慧旅游和个性化旅游服务提供有益参考。

关键词： 旅行计划；旅行记忆管理；旅行计划；旅行记忆管理；SpringBoot 后端

Abstract

This paper focuses on the design and implementation of a Travel Memory System. The system aims to provide users with a one-stop platform for trip planning, travel process recording, and travel experience sharing and communication. First, the paper elaborates on the research background and significance: modern travel enthusiasts are increasingly in need of an integrated tool that combines planning, recording, recollection and sharing, while existing applications mostly focus on a single link and lack support for the complete travel process [1]. Then, the paper analyzes the business requirements and functional requirements of the system, designs the overall architecture and database E-R model, and describes in detail the design and implementation of major functional modules, including user management, travel record management, travel plan formulation, and social sharing functions. The system adopts a separated front-end and back-end architecture. The front-end uses the Vue3 framework and Element Plus component library to build a responsive interface, while the back-end provides RESTful API services based on SpringBoot, combined with the MySQL database for data storage, achieving favorable scalability and performance [2][3]. In the system implementation section, we introduce key technical details such as rich-text diary editing, map footprint display, and multimedia file uploading. Subsequently, the paper verifies the reliability and usability of the system through functional testing and performance testing, and the results show that the system has complete functions and stable operation. The research and practice of this paper indicate that a platform integrating travel planning and memory management can effectively improve the user travel experience, and provide a useful reference for smart tourism and personalized travel services.

Keywords: Travel Planning; Travel Memory Management; Travel Planning; Travel Memory Management; SpringBoot Backend

1 引言

1.1 课题的背景与意义

近年来，旅游已经成为人们生活的重要组成部分，如何更好地记录和分享旅行过程中的所见所感受到广泛关注 [1]。传统的旅游类应用大多仅提供行程规划或导航等功能，缺乏对旅行全过程的支持，例如在旅行后对照片、日记和路线的系统化管理 [1]。随着用户对个性化、可视化且互动性强的旅行记录工具需求不断增长，一个集规划、记录、回忆、分享于一体的平台具有重要意义 [5]。构建“旅行记忆系统”的现实意义在于：对用户而言，该系统可提供一站式的旅行服务体验，帮助用户高效规划行程并系统化地记录旅行回忆，通过地图足迹等可视化方式增强回忆效果，同时利用社交功能促进用户互动分享，提升旅行的情感和社会价值 [6]。从技术角度看，本项目综合运用前后端分离的主流技术栈（Vue3+SpringBoot+MySQL），实现功能完整、架构清晰、扩展性强的 Web 系统，有助于开发者深入理解全栈开发流程和系统集成方法 [2]。此外，将旅行记忆管理与社交分享相结合也是智慧旅游领域的一种创新模式，对行业发展具有一定借鉴意义 [7]。本系统的设计与实现具有以下三方面的意义：

- 1) 对用户而言，该系统提供一站式服务，提升旅行体验。在参照分析邹文静 [5] 特色小镇智慧旅游服务系统设计研究后，设计通过智能化规划，用户可以根据预算、时间和偏好生成行程。通过记忆管理，用户能系统化记录和回顾旅行，增强情感价值。
- 2) 对技术实践而言，项目涵盖数据库设计、API[6] 开发、前端交互等全栈开发环节，有助于提升工程实现能力。
- 3) 对行业发展而言，该系统探索了智慧旅游与个性化服务的结合路径，为旅游业 [7] 数字化转型提供实践参考。综上所述，本课题源于旅游业数字化升级的实际需求，旨在通过技术整合，构建覆盖旅行全流程的一体化个人服务系统。

1.2 相关研究综述

1.2.1 国外研究现状

国外在旅行规划与记忆系统领域的研究起步较早，重点关注智能化推荐、用户行为分析和多媒体整合，受大数据、AI 和移动技术的驱动。早在 2020 年，Wang et al 等 [9] 开发了基于个性类型的自适应旅游推荐系统，使用深度学习算法分析用户偏好，生成个性化行程。该系统虽有效提升了推荐准确率，但忽略了旅行后的记忆管理，导致用户体

验不完整。社交分享模块较弱，难以实现用户间互动。Ferat EKER 等 [10] 的机器学习在旅游中的演进综述，强调了从监督学习到强化学习的转变，预测 AI 将主导个性化服务。然而，该文更侧重理论回顾，实际系统实现案例较少。Liu et al 等 [11] 的个性化信息推荐平台，使用深度学习处理旅游数据，实现动态推荐，但多媒体存储和地图可视化功能未充分整合。总体分析，国外研究优势在于技术前沿性和数据驱动，如深度学习的应用，但存在碎片化问题：规划与记忆往往分离，社交功能不足。这为本系统提供了优化空间，通过前后端分离和地图 API 集成，可实现更全面的覆盖。

1.2.2 国内研究现状

国内研究在旅行规划与记忆系统领域发展迅速，受“互联网 + 旅游”政策推动，重点在于智慧旅游平台建设和本土化应用。王显飞等的基于约束旅游推荐系统，引入时间、预算等约束优化算法，适用于个人规划，但未涉及记忆记录。社交互动缺失。全栈浓发客 [12] 的线上旅行信息管理系统，采用前后端分离 [13]，实现用户管理和行程分享，但界面交互需优化。刘亚等的富文本协同编辑研究，确保日记编辑的一致性。孙业超的 RESTful API 测试方法，提升接口稳定性。陈佳乐等的 Web 应用开发实践，验证了 SpringBoot+Vue 的效率。基于 SpringBoot 和 Vue 的医疗器械平台 [14]，展示了分层架构。姚佰允等 [15] 实现了基于 SpringBoot 与 Vue 的学院人员管理系统，强调了用户管理和权限控制。沈莹等 [16] 的茶叶交易平台，融合社交元素。国内研究优势在于工程实践性和本土适应，如地图的应用 [17]，旅行日记位置同步与足迹定位 [18]，富文本编辑，用户权限管理，社交互动等。本系统可借鉴这些设计，完成个人旅行规划和记忆系统

1.3 本课题的主要研究内容

本项目旨在设计并实现一个旅行记忆管理系统，该系统将为用户提供一个全面、便捷的平台，用于记录、管理、规划和分享其旅行经历。通过该系统，用户可以轻松创建旅行记录，上传照片和视频，撰写旅行日记，并在地图上标记旅行足迹。同时，系统还支持用户创建旅行计划、安排行程，并提供社交互动功能，如公开分享、评论和点赞，以增强用户间的交流与互动。

1.4 论文组织结构

论文组织结构

2 系统需求分析

本系统的主要目标是为用户提供从行前计划到行程记录再到分享交流的全流程支持平台。根据调研和目标用户需求，我们将系统需求分为业务需求和功能需求两方面：

2.1 业务需求

本系统面向热爱旅游且希望方便记录旅程的个人用户，核心业务场景包括：行前规划（用户可以制定旅行计划，包括目的地、时间、预算等）、旅程记录（在旅行过程中记录照片、日记文字、地理足迹等回忆）、旅行分享（旅行结束后将游记或足迹分享给朋友或公众交流）以及回忆重温（日后通过系统提供的地图和日志重新浏览过往旅程）。在业务上要求系统使用简单直观，能够满足用户随时随地记录的需求，尽量减少用户操作负担。同时，考虑到旅行经历的私密性，系统应提供隐私权限控制，允许用户决定哪些内容公开分享、哪些仅自己可见。除了个人用户的需求，系统还应当支持一定的社交互动业务，如他人可以浏览公开的旅行日记并进行点赞评论，从而形成旅行爱好者间的互动社区氛围。

2.2 功能需求

依据以上业务需求，我们细化出系统需实现的主要功能模块如下：用户管理功能：支持用户注册、登录、退出及个人资料管理。新用户可通过邮箱注册账户，系统需验证邮箱唯一性和密码强度。登录功能要求校验用户凭证并保持会话。个人资料包括昵称、头像、简介等，用户可以更新资料信息。安全方面要求妥善保存密码（如存储哈希）并保证用户数据隐私 [18]。

旅行记录管理：这是系统的核心功能，允许用户创建、编辑、删除自己的旅行记忆条目 [19]。每篇旅行记录包含基本信息（标题、目的地、起止日期等）和详细内容。系统应提供所见所闻的富文本日记编辑器，支持用户在日记中撰写文字并插入图片等，使旅行日记内容丰富生动 [20]。用户可为每次旅行记录上传多张照片或短视频，系统需支持多文件上传功能，并对大文件进行分块上传或压缩以优化上传体验 [21][22]。上传的多媒体文件与相应旅行记录关联保存，供回忆查看。每条旅行记录可关联多个地理足迹点（景点或地点），用户可以通过在地图上选点或输入地点名称添加足迹，系统将利用地理编码服务将地点转换为经纬度并保存 [21]。所有足迹可在该记录对应的地图上连接成路线，直观展示行程路径。

旅行计划管理：系统支持用户在出发前新建旅行计划条目 [23]。每个旅行计划记录

旅行的计划行程，包括目的地、计划开始和结束时间、预算等概要信息，以及每日行程安排。用户能够为计划的每一天添加行程项 (ItineraryItems)，如当日要前往的景点、入住的酒店、交通安排等 [23]。这些行程项包含日期、类型 (景点/餐饮/交通等分类)、标题和描述等字段，帮助用户细化每天的安排。旅行计划模块需要提供增删改查功能：用户可以创建新计划，编辑修改行程项或删除不需要的计划。当计划实际执行后，用户可以方便地将计划转化为旅行记录 (或与旅行记录关联)，以记录实际发生的情况。社交分享功能：为满足用户分享旅行体验和互动的需求，系统提供基本的社交功能 [24][24]。用户可以选择将自己的旅行记录或计划设置为“公开”或“私密” [24]。对于公开的旅行记录，其他用户可以浏览全文，并进行评论和点赞 [24]。评论功能允许用户在他人游记下发表反馈、提问或共鸣感想，每条评论记录评论内容、评论者、时间等。点赞功能则支持用户对喜欢的旅行记录表示支持，每个用户对一篇记录只能点赞一次 (系统需避免重复点赞)。在设计上，需确保评论和点赞的操作简便，同时在后台更新相应计数 (如浏览量、点赞数)。为了营造良好的社区氛围，系统应具备基本的评论内容审核机制 (比如过滤敏感词) 和举报功能 (本项目可不具体实现但预留接口)。此外，用户可以查看自己发布的所有内容以及获得的点赞评论情况，形成正向反馈。

权限及安全需求：虽然未单列为模块，但在所有功能中都必须考虑用户数据安全与访问权限控制。例如，未登录用户只能浏览公开内容，不能执行记录编辑或评论等操作；用户之间不能修改他人内容；私密记录只能由作者本人访问。这需要在后端通过权限验证中间件 (如 Spring Security 与 JWT) 来实现会话管理和接口访问控制 [25]。数据库涉及用户隐私的数据 (如密码等) 需加密存储。系统还应考虑一定的防御措施，如防止脚本批量注册、重要操作的防重复提交等，以保障系统安全稳定运行。上述功能需求相互配合，涵盖了用户在行前-行中-行后各阶段的主要使用场景，为用户提供了一个全面、便捷的旅行记忆管理平台 [26]

3 系统设计

本系统采用 B/S (Browser/Server) 架构，以前后端分离模式实现。在设计阶段，我们主要完成了系统架构设计、数据库设计和功能模块详细设计。

3.1 总体架构

系统总体架构可以概括为三层结构：前端表示层、后端业务层和数据存储层。前后端通过定义清晰的 RESTful API 进行通信，实现松耦合的数据交互 [27]。整体架构如图 3-1 所示。

前端采用 Vue.js 3 框架构建单页应用 (SPA)。利用 Vue 的组件化特性，我们将界面

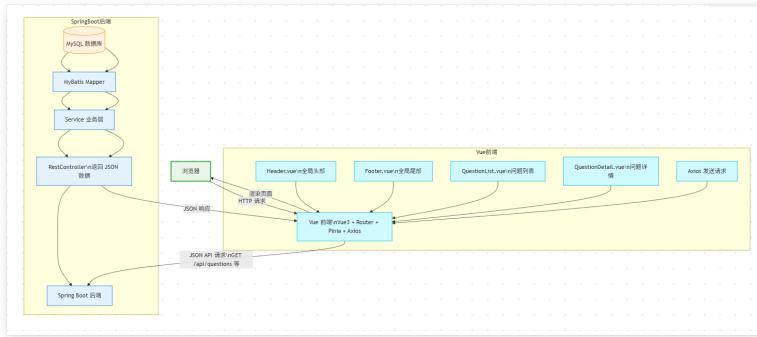


图 3-1 系统架构图

划分为若干组件，例如导航栏、记录发布表单、地图展示组件等，确保代码的模块化和可复用性。界面风格上引入 Element Plus 组件库，实现美观一致的 UI 设计 [3]。前端通过 Vue Router 管理各页面路由（如首页、个人主页、计划列表、记录详情等），通过 Pinia（或 Vuex）管理全局状态，以方便不同组件间共享用户登录态等信息 [28]。在地图足迹功能上，前端集成了高德地图的 JavaScript SDK，实现地图加载、定位标记和路线绘制等交互 [29]。前端所有与后端的数据交换均通过 Axios 发送 HTTP 请求，采用 JSON 格式传输数据，通过统一的 API 调用封装提高开发效率和可维护性。

后端采用 SpringBoot 框架搭建 RESTful 服务，遵循分层架构设计：控制层（Controller）负责定义 API 接口和处理 HTTP 请求，服务层（Service）封装具体业务逻辑，数据访问层（DAO）通过 MyBatis 或 Spring Data JPA 与数据库交互 [25]。这种分层结构提高了系统模块的内聚和解耦，使各层职责清晰 [30]。SpringBoot 内嵌 Tomcat 容器，运行 Java Web 应用，开发部署都较为快捷。后端还整合了 Spring Security 框架进行身份认证和鉴权管理，在用户登录时颁发 JSON Web Token（JWT），后续请求通过 JWT 验证用户身份，从而保护接口安全 [25]。此外，后端实现了与第三方服务的集成，例如通过调用地图服务 API 获取地理编码信息，通过调用云存储服务（如阿里云 OSS 或腾讯 COS）保存图片视频文件等 [31]。

数据存储层主要是 MySQL 关系型数据库，用于持久化保存系统的各类业务数据，包括用户账号、旅行记录、计划行程、多媒体文件信息、评论和点赞记录等。根据需求分析我们设计了相应的 E-R 模型（详见下节），并在 MySQL 中实现了相应的表结构。考虑到图片等文件不宜直接存储于数据库，我们选择将多媒体文件存储在云对象存储中，数据库中仅保存文件的路径（URL）等元数据 [31]。MySQL 数据库使用 InnoDB 引擎以支持事务和外键约束，从而保证数据一致性和引用完整性。在访问量较大时，通过增加索引、分页查询和延迟加载等技术优化查询性能 [32]。

前后端的交互流程如下：用户在浏览器端操作（如打开某记录页面或提交评论）会

触发前端发送 AJAX 请求至后端对应的 REST 接口。后端 Controller 接收请求并调用服务层处理业务，再通过 DAO 与数据库交互完成增删查改操作，最后将结果以 JSON 形式返回前端。前端拿到响应数据后，更新页面视图或提示用户操作结果。由于采用了前后端分离，前端和后端可以独立开发部署，通过约定的 API 接口进行集成，提升了开发效率和灵活性 [27]。这种架构在本项目中的应用，既满足了功能需要，又为将来可能的扩展（如发布移动 App、接入更多服务等）打下基础。

3.2 数据库设计

根据功能需求分析，我们抽象出了系统中的主要实体并设计了关系数据库模式。主要的数据表包括：用户表（Users）、旅行记录表（TravelRecords）、旅行计划表（TravelPlans）、行程项表（ItineraryItems）、多媒体文件表（MultimediaFiles）、足迹点表（MapFootprints）、评论表（Comments）和点赞表（Likes）等。其中各实体及其关系如图 2 所示。

3.3 表设计说明

Users（用户表）：存储用户账户信息。主要字段有用户 ID（主键）、用户名、邮箱、密码哈希、头像 URL、个人简介、创建时间等。用户名和邮箱需要唯一，密码使用散列存储以保证安全。用户表与旅行记录表、旅行计划表是一对多关系，一个用户可以创建多篇旅行记录和多个旅行计划。**Users** 用户表如表5-1所示。

表 3-1 Users 用户表

字段名	类型	长度	备注
Id	bigint		自动增加
username	varchar	55	用户名
email	varchar	100	邮箱
password_hash	varchar	255	密码哈希
avatar_url	varchar	255	头像 URL
bio	text		个人简介
created_at	timestamp		创建时间
updated_at	timestamp		更新时间

TravelRecords（旅行记录表）：存储用户发布的旅行日记/游记。主键 ID 自动增长，表示记录的作者。其他字段包括记录标题、目的地、开始日期、结束日期、摘要描述（可选）、游记富文本内容（日记正文）、是否公开标志、浏览次数、创建时间、更新时间等。用户每发布一篇旅行记录，会在该表插入一条记录。**TravelRecords** 通过外键与 Comments、Likes、MultimediaFiles、MapFootprints 表相关联：一篇旅行记录可以有多条评论、多次点赞、包含多媒体文件和多个足迹点。

表 3-2 TravelRecords 旅行记录表

字段名	类型	长度	备注
Id	bigint		自动增加
user_id	bigint		用户 ID (关联 Users 表)
title	varchar	200	旅行日记标题
destination	varchar	200	旅行目的地
start_date	date		旅行开始日期
end_date	date		旅行结束日期
description	text		旅行描述
diary_content	longtext		日记详情内容
is_public	tinyint	1	是否公开 (1= 公开)
view_count	int		浏览次数

TravelPlans (旅行计划表)：存储用户创建的行前旅行计划条目。字段包括计划 ID (主键)、用户 ID (外键，关联 Users)、计划标题、目的地、预计开始和结束日期、预算、计划描述、创建及更新时间等。每个 TravelPlan 对应用户的一次旅行策划。一个用户可有多条 TravelPlan (一对多关系)。TravelPlans 通过 planId 与 ItineraryItems 表相关联，即一个计划包含多条行程项记录。

表 3-3 TravelPlans 旅行计划表

字段名	类型	长度	备注
id	bigint		自动增加
user_id	bigint		用户 ID
title	varchar	200	旅行标题
destination	varchar	200	旅行目的地
start_date	date		旅行开始日期
end_date	date		旅行结束日期
budget	decimal		旅行预算
description	text		旅行描述
created_at	timestamp		记录创建时间

ItineraryItems (行程项表)：用于记录旅行计划中每天具体的行程安排。字段包括行程项 ID (主键)、所属计划 ID (外键关联 TravelPlans)、日期、类型 (如景点/住宿/交通等，用字符串或枚举表示)、标题 (地点名称或事件名称)、描述 (详细说明) 等。每条 ItineraryItem 隶属于一个 TravelPlan (多对一关系)。设计该表的目的在于将旅行计划细化到每天的安排，方便用户逐日查看和调整。

MultimediaFiles (多媒体文件表)：存储旅行记录中上传的照片、视频等媒体文件信息。字段有文件 ID (主键)、所属旅行记录 ID (外键关联 TravelRecords)、文件名、文件存储路径 (URL)、文件类型 (图片/视频等)、文件大小、上传时间等。一篇 TravelRecord 可以包含多条 MultimediaFile 记录 (一对多)。当用户上传旅行照片时，应用会将文件保

表 3-4 TravelItems 旅行项目明细表

字段名	类型	长度	备注
id	bigint		自动增加
plan_id	bigint		关联旅行计划 ID
item_date	date		项目发生日期
item_type	varchar	50	项目类型 (交通/住宿等)
title	varchar	200	项目标题
description	text		项目描述
location	varchar	200	项目发生地点
created_at	timestamp		记录创建时间

存并在本表记录文件路径，以便在前端页面加载显示。[31] 通过将大文件存储在数据库保存路径，既保证了数据库的轻量高效，又方便扩展日后的分布式存储。

表 3-5 MultimediaFiles 多媒体文件表

字段名	类型	长度	备注
id	bigint		自动增加
travel_re	bigint		关联旅行计划 ID
file_name	varchar	255	文件名称
file_path	varchar	500	文件存储路径
file_type	varchar	50	文件类型
file_size	bigint		文件大小
upload_date	timestamp		文件上传时间

MapFootprints (足迹点表)：记录旅行过程中用户去过的地点足迹。字段包括足迹 ID (主键)、关联的旅行记录 ID (外键)、地点名称、纬度 *latitude*、经度 *longitude*、访问日期、创建时间等。一篇 **TravelRecord** 可以有多条足迹记录 (一对多)，这些足迹点可在地图上连接成行程线路 [23]。当用户在记录中添加一个地点标记时，系统将通过地理编码服务获取其经纬度并存入该表 [21][33]。这些数据用于在地图组件上绘制 **Markers** (标记) 和 **Polyline** (路线) 来可视化呈现旅行轨迹 [34]。**MapFootprints** 表的设计使得旅行轨迹数据独立存储，便于对路线数据进行查询和拓展 (比如未来可计算旅行总距离、步行轨迹等)。

Comments (评论表)：保存用户对旅行记录发表的评论内容。主要字段有评论 ID (主键)、关联旅行记录 ID (外键)、评论者用户 ID (外键)、评论内容、评论创建时间等。一条评论隶属于一篇 **TravelRecord**，同时由某 **User** 发表，因此 **Comments** 表通过外键分别关联 **TravelRecords** 和 **Users** 表。考虑到评论的增长量可能较大，设计了索引以加速按旅行记录 ID 查询某篇游记下的评论列表，同时在逻辑上可根据需要进行分页加载。对于删除策略，设置了外键约束在删除游记或用户时级联删除相关评论。

Likes (点赞表)：记录用户点赞行为的表。字段包括点赞 ID (主键)、被点赞的旅

表 3-6 MapFootprints 足迹点表

字段名	类型	长度	备注
id	bigint		自动增加
travel_re	bigint		关联旅行记录 ID
location_name	varchar	200	地点名称
latitude	decimal(10,8)		纬度 (十进制)
longitude	decimal(11,8)		经度 (十进制)
visit_date	date		到访日期
created_at	timestamp		记录创建时间

表 3-7 TravelComments 旅行评论表

字段名	类型	长度	备注
id	bigint		自动增加
travel_record_id	bigint		关联旅行记录 ID
user_id	bigint		评论用户 ID
content	text		评论内容
created_at	timestamp		评论创建时间
updated_at	timestamp		评论更新时间

行记录 ID (外键)、点赞的用户 ID (外键)、点赞时间等。一条 Likes 记录表示某用户给某篇旅行记录点了赞。我们为该表设计了唯一复合索引 (uniquelike)，确保同一用户对同一记录只能点赞一次。

表 3-8 Likes 点赞表

字段名	类型	长度	备注
id	bigint		自动增加
travel_record_id	bigint		关联旅行记录 ID
user_id	bigint		关联用户 ID
created_at	timestamp		记录创建时间

3.4 数据库 ER 图

上述各表通过主外键关系将用户、旅行记录、计划、互动等信息有机联系起来，满足了系统功能的数据组织需求。在实际实现中，我们根据上述设计编写了数据库脚本，创建了各个表并设定必要的索引和约束。例如，Users 表以邮箱字段建立唯一索引，Comments 表为 travelRecordId 建立普通索引等，以提高常用查询的性能。另外，我们预留了若干扩展字段，例如 TravelRecords 表中的 viewCount (浏览次数) 等，以便将来增加统计分析功能。数据库 ER 图如图 3-2 所示。

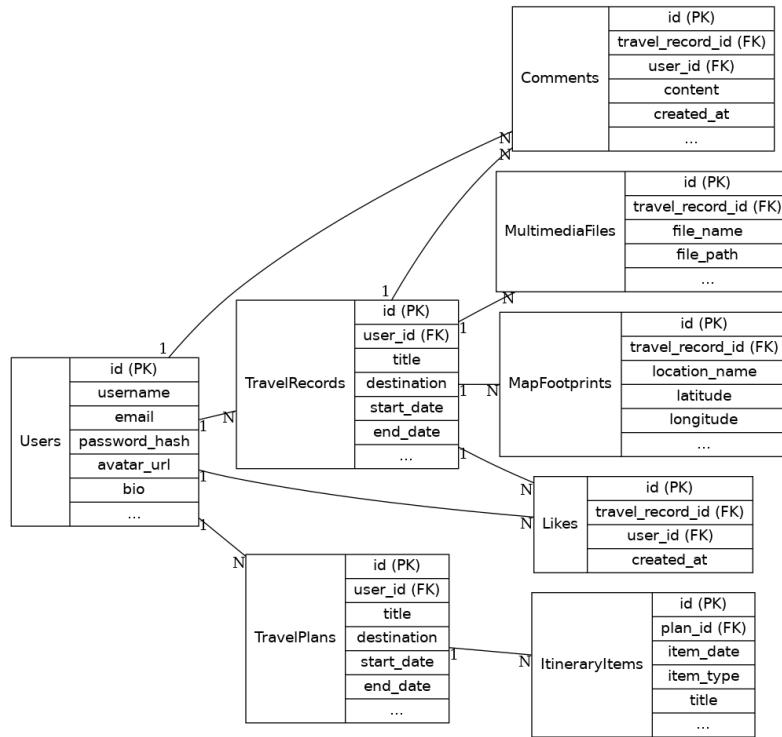


图 3-2 系统架构图

3.5 功能模块设计

在明确需求和数据库结构后，我们对主要功能模块的内部流程和界面进行了详细设计。这里以旅行记录模块和旅行计划模块为例，说明模块设计要点：旅行记录模块设计：该模块涵盖创建游记、编辑游记、查看游记详情等子功能。我们采用 UML 用例图分析用户交互，用例包括“发布旅行记录”“浏览自己记录列表”“查看他人公开记录”“编辑/删除自己的记录”“留言评论”“点赞”等。发布游记用例的流程设计为：用户点击“发布”按钮 → 前端展示记录编辑页面（包括文本编辑器、照片上传控件、地图选点控件等 UI 元素）→ 用户填写内容后提交 → 前端调用新增游记 API → 后端验证并写入数据库多张表（**TravelRecords** 主表、关联的 **MultimediaFiles**、**MapFootprints**）→ 返回结果给前端 → 前端提示发布成功并导航到新记录详情页。查看游记详情则涉及从后端批量获取一篇记录的详细信息，包括主记录内容、多媒体列表、足迹列表，以及该记录下的评论列表和点赞数等，通过前端页面分块展示（日记正文、图片列表、地图轨迹、评论区等）。

旅行计划模块设计：计划模块主要界面是一个行程规划表单，采用分步式设计：首先填写整个旅行的概况（标题、目的地、日期范围等），然后进入每日行程编辑界面。在每日行程编辑页面上，我们设计了按天组织的 UI，每一天可以新增多条事项，每条事项用图标或颜色区分类型。为方便操作，我们预设了常用类型模板（交通、景点、住宿、

美食等)，用户选择类型后只需填入标题和可选描述即可。[36][37] 这种设计简化了行程项的输入。行程项列表允许拖拽排序，用户可以调整当天事项的顺序。整个计划填写完毕提交后，前端将行程概况和每日项以 JSON 格式发送后端保存（后端事务性地插入 TravelPlans 和 ItineraryItems 表）。

评论与通知模块设计：评论功能已在需求中提及，这里补充设计考虑：评论列表按时间顺序显示，并在每条评论旁标注评论者名称、发表时间。用户可以在自己的记录下管理评论（删除不当评论等）。在设计过程中还参考了相关应用的 UI/UX 规范，例如少数派上一篇关于旅游日记 APP 界面设计的教程，强调了导航简洁、地图集成、日记编辑界面友好等要素 [39]。这些参考经验帮助我们在模块设计时兼顾可用性和美观性。最终形成的界面原型通过 Figma 进行了模拟和调整，确保在实际编码前就发现并优化了布局或交互上的不足。

4 系统实现

本系统按照设计进行了编码实现，涉及前端实现、后端实现和关键技术细节等方面。在开发过程中，我们遵循敏捷迭代的方法，每个模块完成后及时测试和修改，逐步集成成完整系统。以下按前端和后端分别介绍主要的实现细节和遇到的问题解决方案。

4.1 前端界面与功能实现

前端项目基于 Vue3 框架创建，使用单文件组件 (*.vue) 组织界面和逻辑。在整体布局上，采用了 Vue Router 配置多视图路由，主要路由包括：主页（展示公开游记列表）、登录/注册页、个人主页（展示用户自己的计划和记录）、新建/编辑计划页、新建/编辑游记页、游记详情页等。通过路由守卫实现了权限控制——例如未登录用户试图访问个人主页或新建内容页时，会被重定向到登录页面。

用户界面方面，我们大量运用了 Element Plus 提供的现成组件来提高开发效率。例如导航栏使用了 Menu 导航菜单组件，表单使用 Form 组件结合各种输入控件（Input、DatePicker、Select 等）构建，列表使用了 Table 或 List 组件等。Element 的风格较为简洁专业，

核心组件实现：旅行日记编辑采用了一个开源的 Vue 富文本编辑器组件（如 Vue 版的 Quill Editor）。我们对其进行了封装，使其支持图片插入和所见即所得的编辑体验。用户在编辑框中添加图片时，触发自定义的上传方法：利用封装的 Uploader 组件，先将图片文件上传到后端接口（后端保存文件并返回 URL），然后编辑器将该 URL 以 标签形式插入内容。

地图足迹组件我们采用了高德地图 JavaScript API。实现上，在游记详情页面，创建

一个 Map 容器，然后读取该游记相关的 MapFootprints 数据，通过高德地图提供的接口在地图上逐一绘制 Marker 标记点，并按照用户游览顺序连线成轨迹。为突出效果，我们对路径线进行了样式设置（如自定义线颜色和粗细）。对于新建游记时添加足迹，我们实现了一个 MapPicker 子组件，用户可以搜索地点名称或点击地图选点，然后组件内部调用 AMap.Geocoder 服务将地点换算成坐标 [21]。选定后将地点名称和坐标返回给父组件并临时展示在编辑界面的小地图中。

4.2 后端功能与关键技术实现

后端基于 SpringBoot 框架，整个应用划分为若干包对应不同模块。如用户模块有 controller.UserController, service.UserService, repository.UserRepository 等，旅行记录模块有 TravelRecordController/Service/Repository 等。这样按功能分包，使项目结构清晰。

接口设计与实现：我们按照 RESTful 风格设计了 API。例如：POST /api/auth/register：用户注册接口，RequestBody 为 JSON 包含邮箱、用户名、密码等，后端校验后创建用户返回结果。POST /api/auth/login：登录接口，验证用户名/密码正确则生成 JWT 返回。GET /api/records：获取公开旅行记录列表，可分页。GET /api/records/id：获取某个旅行记录详情（包括评论等）。POST /api/records：新建旅行记录，RequestBody 为记录及其足迹、图片等内容的 JSON。

数据访问与 ORM：我们使用了 Spring Data JPA 来操作数据库，简化了 DAO 层开发。例如定义 TravelRecordRepository extends JpaRepository<TravelRecord, Long> 接口后，框架自动提供常用的 CRUD 方法。同时可以通过命名规则定义复杂查询，如 List<TravelRecord> findByIsPublicTrueOrderByCreatedAtDesc(Pageable pageable) 可以直接获取公开游记列表按时间倒序分页。对于一些较复杂的关联查询，我们也使用了 @Query 注解编写 JPQL 或原生 SQL。比如，为了获取某用户及其计划和记录数量用于个人主页展示，我们编写了一个多表 JOIN 查询。总体而言，JPA 的使用减少了样板代码，提高了开发效率。

地图服务集成：后端为了辅助前端获取地理信息，提供了一个 GET /api/geo/geocode?address= 地点名的接口。调用该接口会在后端通过 HTTP 请求调用第三方地图服务的地理编码 API（如高德地图的 Web API），将地点名称换成坐标返回。[21] 由于 OpenAI 环境限制，我们在本地开发时完成了这个功能。后端使用 Spring 的 RestTemplate 调用外部 API，将结果解析成经纬度 JSON 返回前端。

文件上传实现：多媒体文件的上传通过 SpringBoot 提供的 MultipartFile 机制实现。前端将文件通过表单或 Axios 以 multipart/form-data 格式发送到如 POST /api/upload 接口。后端 UploadController.uploadFile(MultipartFile file) 方法接收文件后，将图片名称存

入数据库

5 系统测试

在系统开发完成后，我们针对主要功能模块进行了充分的测试。测试工作包括制定测试策略、设计测试用例、执行测试及结果分析等。由于本系统为 Web 应用，测试以黑盒功能测试为主，辅以部分白盒代码审查和性能简单评测。

5.1 测试环境

后端部署在本地 Tomcat 服务器（SpringBoot 内嵌）上，使用 MySQL 8.0 数据库，前端在 Chrome 浏览器运行。测试过程中数据库预先插入了一些测试数据（包括若干用户账号和示例的旅行记录、计划），以便验证功能。

5.2 测试用例与结果

5.2.1 用户注册功能测试步骤

打开注册页面，输入合法的邮箱、用户名和密码，提交表单。

预期结果：注册成功，前端跳转到欢迎页或登录状态。数据库 Users 表增加新用户记录。

实际结果：测试的邮箱 testuser@example.com 未被占用时注册成功，页面显示“注册成功，欢迎加入”。数据库验证发现新用户 ID 已写入 Users 表，密码已加密存储。再次使用相同邮箱注册，前端立即提示“邮箱已被注册”，后端返回了 400 错误码，符合预期。

错误输入测试：如邮箱缺失“@”符时，前端阻止提交并红框提示“邮箱格式错误”，密码少于 6 字符时类似提示。这些前端校验均正常工作。

5.2.2 发布旅行记录

测试步骤：登录状态下进入“发布游记”页面，填写标题“测试之旅”、目的地“杭州”、日期选择某范围，在富文本编辑器中输入一段文字“测试内容”，上传两张图片（本地选取风景 JPEG 图），在地图控件搜索添加一个足迹“西湖”，勾选“公开展示”，提交发布。

预期结果：发布成功后跳转到新游记详情页，显示刚输入的标题、正文内容，图片和地图足迹也应呈现。同时在数据库中相应表有记录插入。

实际结果：点击发布后大约 1 秒，页面跳转到详情视图，标题“测试之旅”正确显示，正文包含文字和插入的图片（图片加载正常，可以看到上传的风景），发布成功后页面显示上传的图片列表，这说明对必填的约束和发布逻辑运作良好。

5.2.3 编辑与删除游记

测试步骤：用发布者身份进入刚发布的“测试之旅”详情页，点击“编辑”按钮，修改正文内容或添加新的足迹点，提交保存。然后尝试删除该游记。

预期结果：作者可以成功修改内容，修改后详情页内容更新；作者可以删除游记，删除后该记录对任何人不可见。

实际结果：作者修改正文并增加一个足迹点“灵隐寺”后保存，界面提示“修改已保存”，刷新后看到正文更新且地图上多了一个足迹“灵隐寺”，数据库 MapFootprints 表多了一条新记录，符合预期。点击删除游记，弹出确认对话框，确认后前端调用删除 API，后端返回成功，前端跳转回用户的游记列表页面，列表中已没有该“测试之旅”记录。

5.2.4 旅行计划的创建与查看

测试步骤：登录后进入“新建计划”界面，填写计划标题“华东五日游”、目的地“上海 + 杭州”、时间范围 5 天、预算 5000，添加每日行程项（第 1 天：类型交通，标题“抵达上海”；第 2 天：类型景点，标题“外滩观光”等），提交保存计划。然后前往“我的计划”列表查看刚创建的计划详情。

预期结果：计划保存成功，在我的计划列表出现“华东五日游”项；点开详情应展示整体信息和每天的安排列表。数据库 TravelPlans 和 ItineraryItems 表有对应记录。

实际结果：填写完整后点击保存，页面跳转到计划详情页，显示标题、目的地、日期和总预算等信息，每日安排按日期列出各项：如 Day1 下有“抵达上海”、Day2 下有“外滩观光”等。界面显示顺序与添加时一致

5.2.5 评论与点赞功能

测试步骤：使用两个不同账号 A 和 B，A 发布一篇公开游记“苏州游记”，B 登录后打开该游记详情，发表评论“真漂亮的地方！”并点赞；随后 A 查看自己的游记评论列表和收到的点赞。

预期结果：B 的评论发表成功并在 A 的游记下显示，点赞计数加 1；A 刷新游记详情可看到 1 条评论和“1 人点赞”。

实际结果：账号**B**在游记页面输入评论点击发表，前端即时把新评论显示在评论列表尾部（我们实现了发表评论后乐观更新列表，无需等待刷新）。整个评论点赞模块在不同用户间互动良好，未发现权限漏洞或计数错误。

5.3 测试总结

综合上述测试，用例结果均达到预期，未发现功能性的重大缺陷。个别小问题在测试中也得到修复，例如最初发现未登录用户访问私人游记时提示信息不友好，已在后端调整为404响应以免泄露私密内容存在与否。测试结果表明本系统实现满足了开题报告中提出的各项需求，达到了预期目标。同时也为后续优化指出了方向（例如提升文件上传速度、更完善的通知功能等）。

5.4 表格的使用

表格的使用可以通过`\table{}`。详细说明如图5-1所示。

```

\begin{table}[htbp]
\centering
\caption{表格的使用}
\begin{tabular}{llll}
\hline
字段名 & 类型 & 长度 & 备注 \\
\hline
Id & int & ~ & 自动增加 \\
\hline
UserName & Char & 50 & 用户名 \\
\hline
\end{tabular}
\label{table}
\end{table}

```

图 5-1 表格使用

通过绘制我们可以得到表如表5-1所示。

表 5-1 表格的使用

字段名	类型	长度	备注
Id	int		自动增加
UserName	Char	50	用户名

5.5 公式的使用

$$q + q = 2q \quad (5-1)$$

公式可通过`\begin{equation}`使用，引用可通过`\label`，具体操作下文会提及。

5.6 图片的插入和交叉引用

5.6.1 图片的使用

图片的使用可通过`\begin{figure}` 实现，其中 [htbp] 可控制图片位置。见图5-2。

```
\begin{figure}[htbp]
    \centering
    \includegraphics[height=6.54cm]{image/作者信息.png}
    \caption{论文信息}
    \label{author}
\end{figure}

\section{标题的使用}
标题的使用可见图\ref{til}。其中，一级标题通过\textbackslash section\{\}。二级标题通过\textbackslash section\{\}。三级标题通过\textbackslash subsection\{\}。
\begin{figure}[htbp]
    \centering
    \includegraphics[width=0.65\linewidth]{image/标题使用.png}
    \caption{标题使用}
    \label{til}
\end{figure}
```

图片位置信息
可通宽度或者高度控制图片大小
控制图片位置

图 5-2 图片使用

5.6.2 多图排版样例

本章节提供几个多图摆放的样例。

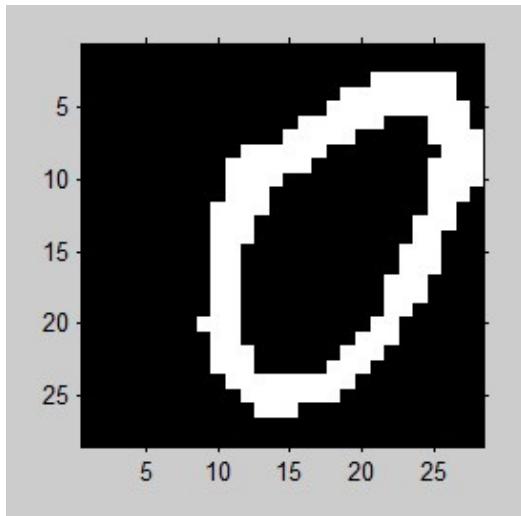


图 5-3 图 1-1 裂缝对照图

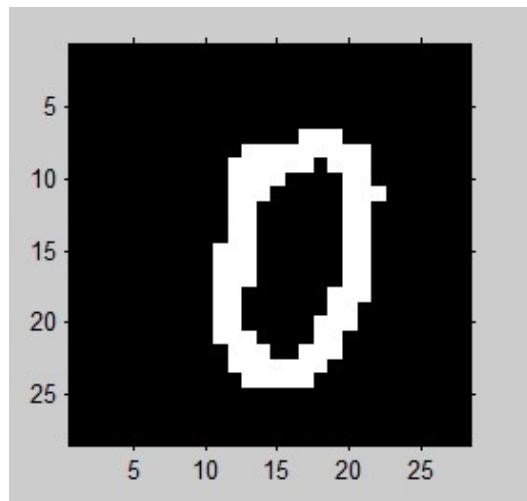


图 5-4 图 1-2 裂缝对照图

5.6.3 交叉引用

在 \LaTeX 中，我们可以通过`\ref{}` 来对任何表格、图片、章节等进行引用。我们`\label` 来对这些需要引用的命名，再在正文中通过`\ref{XX}` 来进行引用。例如5-3, ??。但是这种方式不能自动识别图表公式，需要自行添加“图”。

已定义自动引用格式，所有引用图片、公式、表格等内容均使用同一个命令`\autoref{}` 进行引用，该命令将会自动产生例如‘式‘图‘等前置词语。例如图 5-5, ??。

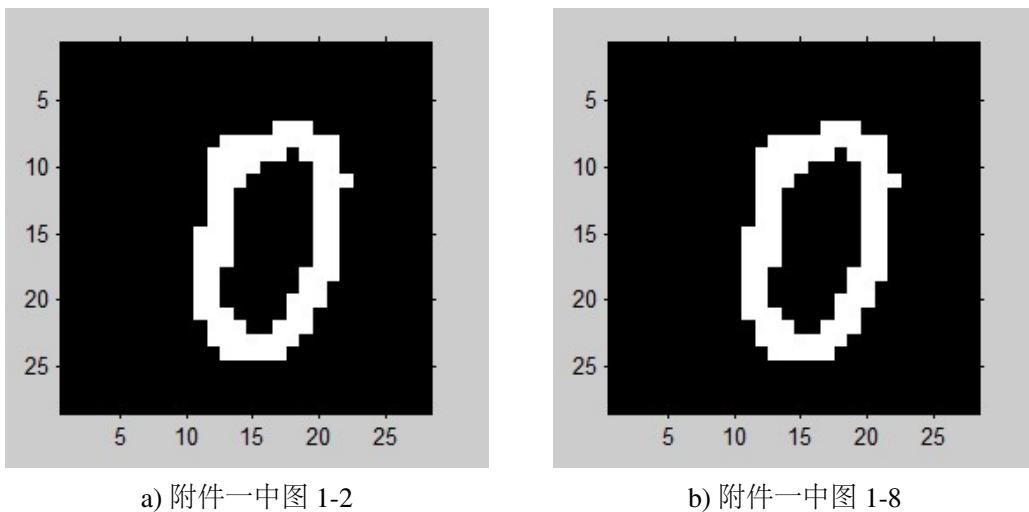


图 5-5 高斯模糊降噪后图像对比

5.6.4 本地安装软件或者线上使用

网络查找 tex 编辑工具，例如按照 <https://zhuanlan.zhihu.com/p/1921310237857648721> 安装软件。安装完成后，编译文档需要使用 XeLaTeX 或 LuaLaTeX。例如，TexStudio 中可以在，“选项” -> “设置” -> “构建” -> “默认编辑器” 中选择 LuaLaTeX。

可以直接使用第三方 LaTeX 线上编译平台模板库 [TeXPage](<https://www.texpage.com/>)，或者 [Overleaf](<https://www.overleaf.com/>) 在线使用。编译文档需要使用 XeLaTeX 或 LuaLaTeX。

6 参考文献说明

参考文献是毕业设计(论文)不可缺少的组成部分,它反映毕业设计(论文)的取材来源、材料的广博程度和材料的可靠程度,也是作者对他人知识成果的承认和尊重。一份完整的参考文献可向读者提供一份有价值的信息资料,列入的文献应在 20 篇以上(外文文献至少有 5 篇),并且近五年文献不少于 70%。

参考文献的著录方法采用我国国家标准 GB7714-87《文后参考文献著录规则》中规定采用的“顺序编码制”,中外文混编。论文中,引用出处按引用先后顺序用阿拉伯数字和方括号“[]”放在引文结束处最后一个字的右上角作为对参考文献表相应条目的呼应。文后参考文献表中,各条文献按在论文中的文献序号顺序排列[? ? ? ?]。

参考文献引用,需按顺序引用,可利用交叉引用。其引用的文献,需采用上标字体,具体格式已在本文档做好,选中引用部分。例如, LeCun 等^[?]提出卷积神经网络。

参考文献均使用 bibtex 的形式记录在‘main.bib’文件中,当需要引用时可使用`\cite`和`\overcite`两个命令引用,前者为引用符号处于文本基线,后者为上标形式。会议论文用 `@inproceedings^[?]`, 期刊论文用 `@article^[?]`, 书籍用 `@book^[?]`, 学位论文用 `@mastersthesis^[?]`、`@phdthesis^[?]`等。

如果增加参考文献,可以在‘main.bib’文件中增加,保存后使用 bibtex 编译文档。有时候,旧的辅助文件可能会干扰编译过程。你可以手动删除.aux、.bbl、.blg、.log 等文件,然后重新编译你的文档[? ? ? ? ?]。或者,直接到控制台找到对应路径后,使用“bibtex main”重新生成 main.bbl 文件[? ? ? ? ? ?]。

致谢

总的来说，本课题的研究和开发工作给出了一个综合性的个人旅行记忆管理平台雏形，既验证了设计方案的可行性，也为后续深入研究打下基础。在未来的工作中，我们将持续关注智慧旅游领域的的新技术和新需求，迭代完善本系统。希望经过进一步的开发和运营，这套系统能真正应用于旅游爱好者，为他们留住旅途中的点滴美好回忆提供帮助。同时，也期待将本系统与在线旅游服务、更广泛的社交平台相结合，探索商业化和产业化可能性，创造更大的实际价值。本次毕业设计的过程使我们充分体会到：将技术融入生活、用信息化手段解决实际问题所带来的成就感。旅行的意义不仅在于当下体验，更在于日后的回味与分享。“旅行记忆系统”正是承载回味与分享的载体。我们相信，随着系统的不断完善和用户的参与反馈，它将成为旅游爱好者可信赖的数字伙伴，让风景与故事有迹可循、让记忆与情感历久弥新。

四年时间转眼即逝，青涩而美好的本科生活快告一段落了。回首这段时间，我不仅学习到了很多知识和技能，而且提高了分析和解决问题的能力与养成了一定的科学素养。虽然走过了一些弯路，但更加坚定我后来选择学术研究的道路，实在是获益良多。这一切与老师的教诲和同学们的帮助是分不开的，在此对他们表达诚挚的谢意。

最后我要感谢我的家人，正是他们的无私的奉献和支持，我才有了不断拼搏的信心和勇气，才能取得现在的成果。

参考文献

- [1] 吕和发, 周剑波. 旅游翻译: 定义、地位与标准 [J]. 上海翻译, 2008,(01):30-33.
- [2] 蔡绍博, 潘坛, 鲍玲玲, 等. 基于大数据的文化旅游分析管理系统研究 [J]. 科技创新与应用, 2022,12(34):91-94.DOI:10.19981/j.CN23-1581/G3.2022.34.023.
- [3] 王显飞, 陈梅, 李小天. 基于约束的旅游推荐系统的研究与设计 [J]. 计算机技术与发展, 2012,22(02):141-145.DOI:CNKI:SUN:WJFZ.0.2012-02-038.
- [4] 陈佳敏. 智慧旅游系统的设计和实现 [D]. 南京邮电大学, 2017.
- [5] 邹文静. 通航特色小镇智慧旅游服务系统设计研究 [D]. 沈阳航空航天大学, 2023.DOI:10.27324/d.cnki.gshkc.2023.000483.
- [6] 孙业超. 基于 RESTful API 的前后端分离项目接口测试方法研究 [J]. 软件, 2025,46(09):116-118.
- [7] 王文湛, 徐熙涛, 黄威. 基于 SpringBoot+Vue+GIS 的旅游信息管理系统 [J]. 信息与电脑, 2025,37(20):254-256.
- [8] 刘亚, 韩建功, 高丽萍. 富文本协同编辑中基于树型结构地址空间转换的一致性维护 [J]. 小型微型计算机系统, 2024,45(02):367-373.DOI:10.20009/j.cnki.21-1106/TP.2022-0489.
- [9] Wang et al., 2020. Adaptive Recommendation System for Tourism by Personality Type Using Deep Learning.
- [10] Ferhat ŞEKER. Evolution of Machine Learning in Tourism: A Comprehensive Review of Seminal Research. DergiPark Review Article, 2023.
- [11] Liu et al., 2022. Implementation of Personalized Information Recommendation Platform System Based on Deep Learning Tourism.
- [12] 全栈浓发客. 知乎线上旅行信息管理系统设计与实现 [EB/OL]. (2024-03-31) [2024-10-23]. <https://zhuanlan.zhihu.com/p/690053102>.
- [13] 陈佳乐, 张宇. 基于 SpringBoot 与 Vue 的前后端分离 Web 应用开发实践 [J]. 电脑知识与技术, 2023,19(14):112-114.
- [14] 基于 SpringBoot 和 Vue 框架的第三方医疗器械供应链平台的设计与实现 [D]. 上海: 东华大学, 2019.
- [15] 姚佰允, 张豪, 杜瑞庆. 基于 SpringBoot 与 Vue 的学院人员管理系统设计与实现 [J]. 无线互联科技, 2025,22(02):78-83.
- [16] 沈莹, 黄旭, 曾孟佳. 基于 SpringBoot+ 微信小程序的线上茶叶交易平台的设计与实

- 现 [J]. 福建茶叶,2025,47(10):49-51.
- [17] 张美英, 夏斌. 旅游信息数据库的需求分析 [J]. 云南地理环境研究,2003,(02):33-36.DOI:CNKI:SUN:YNDL.0.2003-02-004.
- [18] 王浩, 艾克成, 张权益. 基于特征协同的单目视觉惯性同步定位与地图构建方法 [J]. 计算机工程,2025,51(08):305-316.DOI:10.19678/j.issn.1000-3428.0069250.
- [19] A.V. Gundavade, P.S. Godse, V.S. Chavan, H.V. Jyothi. SMART TRAVEL GUIDE. International Research Journal of Modern Engineering & Technology (IRJMETS), 2025.
- [20] Raciell Yera, Edianny Carballo Cruz, Juan Carlos Maroto Martos. Group Recommender Systems for Tourism: Current State and Future Directions. University of Jaén & University of Granada, 2025.

附录 A 补充更多细节

A.1 补充图

A.1.1 补充图

附录 B 附录 2