**旅游自助系统的设计与实现**

摘 要

随着《Java企业级应用开发》课程的结束，同学们组队并选择一个课题进行课程设计。经过对各个题目的对比分析，我们组最终选择了旅游自助系统这个课题进行开发。旅游自助系统是一个基于互联网的旅游服务平台，它可以为用户提供个性化的旅游规划、预订、导航、咨询等功能，让用户享受更便捷、更灵活、更丰富的旅游体验。这个课题的亮点是设计灵活、功能广泛，在我们的能力范围内，并且有极强的横向扩展能力。即便课程设计项目结束，我们也能将其继续维护，并使其成为一个功能完善的可用项目。

关于本课题，我们考虑了三个开发阶段。

第一阶段，我们会基于前后端分离式设计，借助Spring Cloud、Vue3等技术，完成系统的开发和构建，使其达到功能完备的状态。我们采用微服务架构，将系统分为多个模块，并定义统一的RESTful API进行通信。

第二阶段，我们结合分布式思想，将服务扩展，提升其并发能力和容灾能力。我们将使用Nacos进行服务注册和发现，使用RabbitMQ进行消息队列处理，使用Redis进行缓存优化，使用Hystrix进行熔断降级。

第三阶段，我们将利用Transformer和BERT等框架和模型，对用户的行为、偏好、反馈、评价等数据进行挖掘和建模，训练出能够根据用户的需求和情境提供智能推荐、咨询、预订等服务的AI助手。我们的AI助手计划覆盖80%的用户场景，进一步提高用户的满意度和留存率。

考虑到课程设计时间有限，我们将在保证第一阶段顺利完成的情况下，尽可能多地完成第二阶段甚至第三阶段，对于没能完成的部分，我们将考虑实现可配置的预览功能，并将其完善到可供演示的程度。

**关键词**：前后端分离；微服务；分布式；人工智能；AI

目 录

目录

[1 绪论 3](#_Toc133545745)

[2 相关技术 3](#_Toc133545746)

[2.1微服务 3](#_Toc133545747)

[3 系统需求分析 3](#_Toc133545748)

[4 系统设计与实现 3](#_Toc133545749)

[5 系统测试与评估 3](#_Toc133545750)

[6 结论与展望 3](#_Toc133545751)

[参考文献 4](#_Toc133545752)

[附 录 4](#_Toc133545753)

1 绪论

介绍论文的研究背景、意义、目标、内容和方法，以及国内外相关研究的现状和不足。

2 相关技术

介绍旅游自助系统涉及到的相关技术，如Java企业级应用开发技术、微服务技术、分布式技术、预训练模型技术等，说明它们的原理、特点和优势。

## 2.1 前后端分离

前后端分离是一种架构模式，它将一个应用的前端代码和后端代码分开写，前后端之间通过JSON格式进行数据交互，前端可以更方便地利用现代化的框架和技术，实现更丰富和流畅的界面和交互。

相比前后端不分离的架构模式，前后端分离不但实现了前端和后端的解耦，使代码易于维护和复用，还提高了开发效率，使前端后端可以并行开发，只需要约定好接口文档，无需等待对方的开发进度便能完成开发任务。

## 2.2 Vue3

Vue是一个用于构建用户界面的JavaScript框架，它基于标准的HTML、CSS和JavaScript，并提供了一种声明式和基于组件的编程模型，帮助你高效地开发用户界面，无论是简单的还是复杂的。Vue是一个渐进式的框架，它允许用户根据你的需求逐步地使用它，无需一次性地引入所有的功能。Vue的核心库只关注视图层，而且易于学习和集成到其他库或现有项目中。

相对于JSP，Vue拥有更高效的开发效率，并能组件化开发，实现页面的模块化和复用，提高代码质量和可维护性，也可以方便地扩展功能和适应需求变化。

## 2.2微服务

微服务是一种软件架构风格，它是以专注于单一责任与功能的小型功能区块为基础，利用模块化的方式组合出复杂的大型应用程序，各功能区块使用与语言无关，而且复杂的服务背后是使用简单 URI 来开放接口，任何服务，任何细粒都能被开放。

相比传统的单体式架构系统，微服务系统解决了其维护难，扩展性差，稳定性低的痛点，并且开发便捷，分工明确，不关注每个服务所使用的技术甚至语言，可以让小组不同成员更便捷地合作。

## 2.3 Nacos

Nacos是一个易于使用的开源平台，用于动态服务发现、服务配置和服务管理。它提供了一系列简单而有用的功能，让用户能够很方便地实现动态服务发现、服务配置、服务元数据和流量管理。

相对传统的Eureka，Nacos不仅支持加权的动态DNS服务发现，还集成了动态配置管理，让用户能够在所有环境中集中、外部化和动态地管理所有应用和服务的配置。动态配置可以避免配置更新时重新部署应用和服务，也可以提高配置的可维护性和安全性。

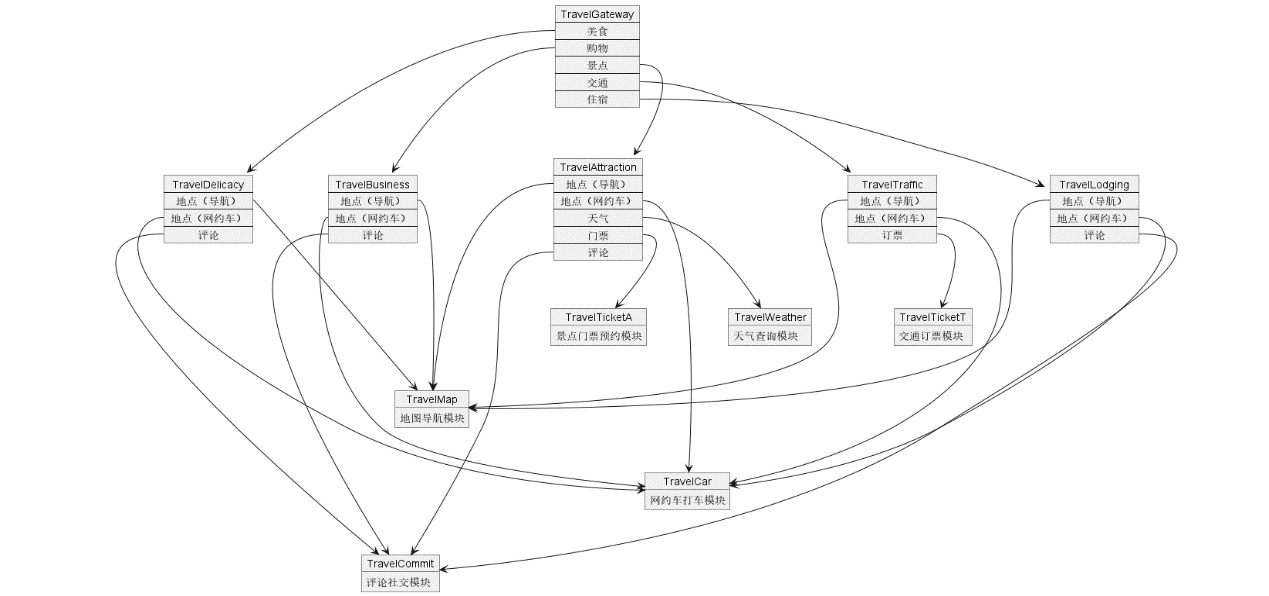
## 2.4 Redis

## 2.5 RabbitMQ

3 系统需求分析

分析旅游自助系统的功能需求和非功能需求，确定系统的用户群体、使用场景、业务流程和性能指标等。

4 系统设计与实现

详细描述旅游自助系统的架构设计、模块划分、接口定义、数据库设计等，以及各个模块的具体实现过程和技术细节。

1系统架构设计

5 系统测试与评估

介绍旅游自助系统的测试方法、测试环境、测试用例和测试结果，以及对系统功能、性能、可用性等方面的评估方法和评估结果。

# 6 结论与展望

总结旅游自助系统的主要工作和创新点，分析系统存在的问题和不足，提出改进方向和未来工作计划。

# 参考文献

列出论文引用的文献，按照规范格式排列。

# 附 录

附上一些辅助性的材料，如系统截图、源代码、数据集等。