

温控计费系统开发建议书

班级\_小组：2021219110\_110f

组长：李根赢

组员1：徐振铭

组员2：黄正轩

组员3：陈欣然

日期：2023-09-27

**目录**

[1. 第一章：系统建设背景 3](#__RefHeading___Toc464_3551849109)

[1.1 系统建设目标 3](#__RefHeading___Toc466_3551849109)

[1.2 业务背景 4](#__RefHeading___Toc468_3551849109)

[2. 第二章：系统建设方案 5](#__RefHeading___Toc470_3551849109)

[2.1 系统框架 5](#__RefHeading___Toc472_3551849109)

[2.2 技术路线和特点 7](#__RefHeading___Toc474_3551849109)

[2.3 技术路线可行性 8](#__RefHeading___Toc476_3551849109)

[3. 第三章：工程能力说明 9](#__RefHeading___Toc478_3551849109)

[3.1 小组及组员能力介绍 9](#__RefHeading___Toc480_3551849109)

[3.2 成功案例介绍 9](#__RefHeading___Toc482_3551849109)

[3.3 完成任务可行性 11](#__RefHeading___Toc484_3551849109)

# 第一章：系统建设背景

## 系统建设目标

本项目旨在为波普特廉价酒店设计和实现一个自助计费式中央温控系统。该系统主要服务于两个用户群体：酒店客户和酒店的管理人员以及空调管理员。客户可以通过此系统自主设定房间的温度和风速，同时实时查看空调使用所产生的费用。退房时，酒店将为客户提供详细的空调使用账单。

该系统的目标包括：

* 响应节能和环保理念，通过自助温控系统，客户可以更加精确地控制空调的使用，从而降低酒店的能源消耗，减少环境负担，提高环保意识。
* 提供个性化服务，通过允许客户自主控制房间温度和风速，以提高客户满意度和入住体验。
* 设计和实现一个可靠的中央温控系统，确保客户能方便地进行自助操作，实现实时计费，同时为管理人员提供空调使用情况的监控和报表生成功能。

通过实现上述目标，本系统将有助于提高酒店的能源效率，优化资源分配，提升客户满意度，从而增强酒店的综合运营效益。

## 业务背景

波普特廉价酒店是一家以提供经济实惠住宿为主打的酒店，位于帝都三环内的大学城附近。由于酒店的舒适住宿环境和相对较低的价格，吸引了广泛的客户群体，包括附近的学生、学生的亲属、旅行者和商务出差者等。酒店共有100间客房，分布在多层楼上，每间客房都配备了基本的设施，如床、浴室、电视以及新引入的自助中央温控系统。

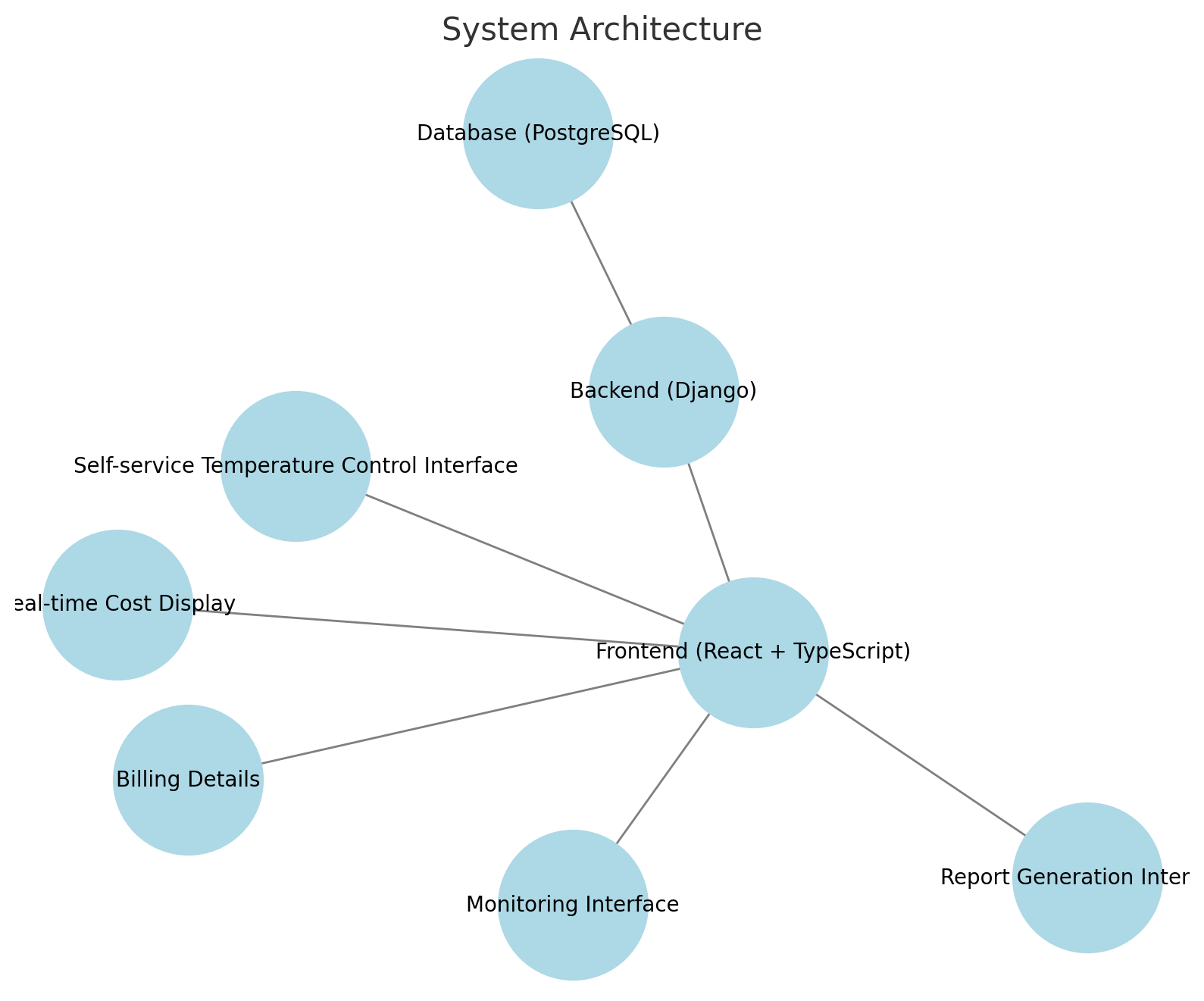
酒店的主要业务流程包括客房预订、入住和退房服务。客人可以通过网上预订客房，或直接前来办理入住手续。入住后，客人可以自助操作房间内的中央温控系统，根据个人需求调整房间的温度和风速，以确保舒适的住宿体验。在客人离开时，他们需要结算住宿费用，通常包括房费和空调费用，但目前空调费用的计费方式较为模糊，缺乏透明度。

为了提高酒店的环保意识，降低能源消耗，并提供更个性化的服务，酒店管理团队计划引入一个自助计费式中央温控系统。该系统将允许客人更精确地控制空调的使用，同时实时显示使用空调所产生的费用。此外，酒店管理人员也希望能够监控客房空调的使用情况，以便更好地管理和优化资源分配。为了实现这些目标，酒店决定进行系统的建设，以提供更为环保、个性化和高效的服务。

# 第二章：系统建设方案

## 系统框架

暂定采用基于 Django + PostgreSQL + React（TypeScript）的系统整体框架，但具体的技术选择应根据项目团队的技能和项目需求进行调整。系统数据流框图如下:



1. **前端 (React + TypeScript)：**

**前端将是系统的用户界面，负责与客户交互以及展示相关信息。前端应该包括以下组件：**

* + **自助温控界面**：允许客户自主调整房间温度和风速。
  + **实时费用显示**：显示客户当前使用空调所产生的费用。
  + **账单详情**：客户可以查看在退房时生成的详细账单。
  + **监控界面**：管理员可以查看房间的空调使用情况。
  + **报表生成界面**：管理人员可以生成酒店及房间的空调使用统计报表。

1. **后端 (Django)：**

**后端将处理前端请求，管理数据和业务逻辑。主要功能包括：**

* + **用户认证和授权**：确保只有授权用户能够访问系统。
  + **温控系统逻辑**：处理客户设置温度和风速的请求，实时计算费用，并生成账单。
  + **监控和报表逻辑**：允许管理员查看房间的使用情况，并生成统计报表。
  + **数据库交互**：与PostgreSQL数据库交互以存储用户信息、房间状态、费用记录等数据。
  + **API接口：**向前端提供数据接口，包括自助温控、费用显示、监控和报表等。

1. **数据库 (PostgreSQL)：**

**PostgreSQL提供了强大的数据存储和查询功能，适合处理系统中的复杂数据需求。**

* + 数据库将存储关键数据，包括用户信息、客房信息、费用记录等。
  + 数据库模型应该包括用户表、客房表、费用记录表等。

1. **通信 (axios)：**
   * **前后端通信**：通过HTTP请求和响应进行通信，使用axios库处理HTTP请求与后端API通信。后端和数据库之间通过Django的ORM（对象关系映射）进行交互。

## 技术路线和特点

* **前端技术**：

前端采用React框架，结合TypeScript增强代码可维护性，通过React Router实现前端路由导航，状态管理库Redux用于确保数据流的一致性和可追踪性，同时使用Axios和Fetch等库处理HTTP请求与后端API通信。

* **后端技术**：

后端部分使用Django框架，基于Python，提供全功能的Web应用支持，包括认证、路由、ORM等，Django REST framework用于构建API，管理用户身份验证和授权，以及与前端的数据交互。

* **数据存储**：

数据存储方面选择PostgreSQL，一种强大的关系型数据库，用于存储系统的关键数据，并通过Django ORM实现数据操作。这个技术组合提供了开发效率、安全性和性能的平衡，适用于构建稳健的自助温控系统。

* **通信技术**：

Axios库提供了基于Promise的HTTP客户端，简化了前后端之间的HTTP通信。

## 技术路线可行性

首先，这些技术在实际应用中被广泛采用，拥有成熟的技术生态系统和强大的社区支持，因此我们可以轻松获取学习和解决问题所需的资源。其次，这个技术堆栈提供了全栈支持，从前端到后端再到数据库的整合，减少了技术栈的混合和不匹配问题，使得我们能够在一个一致的开发环境中工作。此外，Django和React都具备高度可扩展性，适应不断变化的需求，允许我们在项目中添加新功能和模块。（特点）同时需要注意的是组内并不是每一位同学都之前接触过Django+react框架，对于初学者来说，学习React和Django可能需要一些时间，此外，前后端的集成和性能优化也可能是挑战，但通过充分利用开发社区和在线资源，相信我们可以逐步克服这些难点，确保选择的技术路线能够成功应用于课程作业项目中。

# 第三章：工程能力说明

## 小组及组员能力介绍

1. 李根赢：编码能力较强，C++忘差不多了，easyx绘制UI界面有一些基础、python的话，使用过Django、matlib。
2. 徐振铭：后端开发能力：理解并掌握后端开发语言和框架，数据库设计和优化，保证数据的安全和高效的访问。算法设计：解决复杂的算法问题，有数据结构和算法基础。文档书写能力：擅长书写文档，如需求文档，设计文档，用户手册等，文档结构清晰，内容详实，易于理解。
3. 黄正轩：书写文档和编码能力较好，学习过Latex, C++，Python, Java，对基础算法熟悉。负责过项目网站的开发，熟悉Vue.js。负责过数据库设计和接口的开发，对SQL有相应的理解。
4. 陈欣然：擅长交互界面设计，设计易于用户使用，擅长文档书写，如需求文档，设计文档，内容详实，易于理解。

## 成功案例介绍

以下是我们成员在过去项目中的成功案例，以证明我们具备实现该温控计费系统的能力和经验。

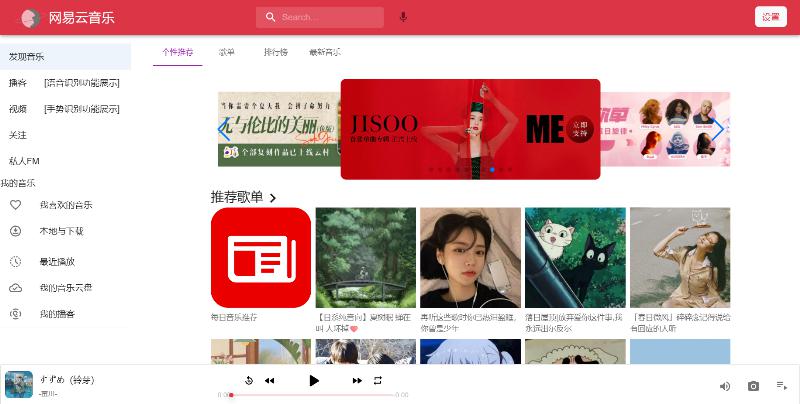
* **音乐播放系统**

**技术栈:**

* 前端:React、Material UI
* 后端:Django、Django REST Framework
* 数据库:PostgreSQL
* 部署:Nginx、Gunicorn

**项目特点:**

* 实现了音乐资源的上传和管理,音乐元数据的提取,以构建起音乐库
* 应用推荐算法,根据用户喜好生成个性化推荐歌单
* 支持歌词显示,并实现音乐的连续无缝播放
* 前端界面简洁优雅,用户体验良好
* 后端API完整,数据结构合理
* 成功部署上云服务器,确保系统稳定运行

 该项目让我们积累了构建音乐类Web应用的经验,尤其是掌握了Python/Django后端开发的技能,以及使用React构建复杂前端的能力。这为我们基于类似技术栈开发温控计费系统提供了坚实基础。我们有信心能够应用这些经验,完成当前项目的开发任务。

## 完成任务可行性

经过分析,我们认为温控计费系统开发任务完全可行。理由如下:

1. **技术上可行**

* 本团队对React、Django等技术栈有成功运用经验
* 4人分工明确,均具备所需技能
* 会使用Git进行版本控制,确保代码质量

1. **时间上可行**

* 明确系统功能优先级,确保核心功能顺利完成
* 采用敏捷开发,每2周一个Sprint
* Sprint内详细规划每周任务, tracking进度

1. **风险控制**

* 评估可能的技术风险,并提前研究解决方案
* 设置代码review流程,减少bug
* 测试覆盖率不低于90%,确保系统稳定

1. **资源保障**

* 充分利用网络资源进行学习
* 寻求老师的指导
* 项目期间全体团队成员投入充足时间

我们小组角色划分明确，采用敏捷开发方法，确保有效交付产出。根据项目小组现有成员能力与课程作业要求评估，我们有信心在2-3次作业验收内交付一个功能完整、质量优良的温控计费系统,完成课程作业要求。