

# 系统设计说明书

福州大学 2022 级软件工程实践“研途无忧”小组

2024 年 10 月 28 日

# 目录

<b>第一章 引言</b>	<b>3</b>
1.1 编写目的 . . . . .	3
1.2 项目背景 . . . . .	3
1.3 参考材料 . . . . .	3
<b>第二章 系统总体设计</b>	<b>4</b>
2.1 整体架构 . . . . .	4
2.2 整体功能架构 . . . . .	4
2.3 整体技术架构 . . . . .	4
2.4 设计目标 . . . . .	5
2.4.1 总体原则 . . . . .	5
2.4.2 实用性和先进性 . . . . .	5
2.4.3 标准化、开放性、兼容性 . . . . .	5
2.4.4 高可靠性、稳定性 . . . . .	6
2.4.5 易用性 . . . . .	6
2.4.6 灵活性和可扩展性 . . . . .	6
2.4.7 经济性和投资保护 . . . . .	6
<b>第三章 系统功能模块详细设计</b>	<b>7</b>
3.1 个人办公 . . . . .	7
3.1.1 通知公告 . . . . .	7
3.1.2 待办事宜 . . . . .	7
<b>第四章 性能设计</b>	<b>8</b>
4.1 响应时间 . . . . .	8
4.2 并发用户数 . . . . .	8
<b>第五章 数据库设计</b>	<b>9</b>
<b>第六章 接口设计</b>	<b>10</b>
6.1 登录接口 . . . . .	10
6.1.1 6.1.1 调用说明 . . . . .	10

6.1.2	6.1.2 请求报文	10
6.1.3	6.1.3 应答报文	10
6.2	查询所有数据列接口	11
6.2.1	6.2.1 调用说明	11
6.2.2	6.2.2 请求报文	11
6.2.3	6.2.3 应答报文	11
6.3	修改指定数据接口	11
6.3.1	6.3.1 调用说明	11
6.3.2	6.3.2 请求报文	11
6.3.3	6.3.3 应答报文	12
6.4	删除指定数据接口	12
6.4.1	6.4.1 调用说明	12
6.4.2	6.4.2 请求报文	12
6.4.3	6.4.3 应答报文	12
6.5	获取待办事项接口	13
6.5.1	6.5.1 调用说明	13
6.5.2	6.5.2 请求报文	13
6.5.3	6.5.3 应答报文	13
6.6	修改用户信息接口	13
6.6.1	6.6.1 调用说明	13
6.6.2	6.6.2 请求报文	14
6.6.3	6.6.3 应答报文	14
<b>第七章</b>	<b>系统出错处理设计</b>	<b>15</b>
7.1	出错信息	15
7.2	补救措施	15
7.3	系统维护设计	15
<b>第八章</b>	<b>系统处理规定</b>	<b>16</b>
8.1	输入输出要求	16
8.2	数据管理能力要求	16
8.3	故障处理要求	16
8.4	其他专门要求	17

# 第一章 引言

## 1.1 编写目的

本系统设计说明书旨在明确“福小研”软件的系统设计方案，为项目的开发、测试和维护提供指导。通过本说明书，可以使项目组成员对系统的整体架构、功能模块、技术选型等有清晰的了解，为后续工作的开展奠定基础。

## 1.2 项目背景

- **软件名称：**福小研
- **项目任务提出者：**福州大学计算机与大数据学院教学办
- **项目任务：**软件工程实践
- **项目开发者：**福州大学 2022 级软件工程实践“研途无忧”小组

本项目旨在为考研学子提供全面的备考支持，包括智能学习规划、资源推荐、心理支持等功能。项目需求基于对实际用户的问卷调查和小组讨论，与其他软件或组织机构无关。

## 1.3 参考材料

1. 《计算机软件需求规格说明规范》(GB/T 9385-2008)
2. 福州大学考研群体需求分析调查报告

## 第二章 系统总体设计

### 2.1 整体架构

系统采用客户端-服务器（C/S）架构，基于 **uni-app** 和 **uniCloud** 云开发平台。整体架构包括以下部分：

- **客户端**：使用 uni-app 框架，基于 Vue3 和 JavaScript 进行开发，生成跨平台移动应用，支持 iOS、Android 和小程序等多端。
- **服务器端**：使用 uniCloud 云开发平台，处理业务逻辑、数据存储和接口提供。
- **数据库**：基于 uniCloud 提供的云数据库，存储用户数据、学习资源、社区内容等信息。

### 2.2 整体功能架构

系统功能主要分为以下模块：

- **个人办公**：通知公告、待办事宜管理。
- **学习管理**：日历打卡、学习计划、自习室、学情分析。
- **资源中心**：课程推荐、题库、考研资讯。
- **社区互动**：论坛交流、问题求助、活动打卡。
- **智能助手**：AI 问答、智能解答、个性化服务。

### 2.3 整体技术架构

系统技术架构包括：

- **前端技术**：
  - 使用 **uni-app** 框架，基于 **Vue3** 和 **JavaScript** 开发，支持多端发布。
  - 采用 **HTML** 和 **CSS** 进行页面布局和样式设计，提升用户体验。
- **后端技术**：

- 使用 **uniCloud** 云开发平台，提供云函数、云数据库、云存储等服务。
- 业务逻辑在云函数中实现，前端直接调用云函数接口。
- **数据库：**
  - 采用 **uniCloud 云数据库**，存储结构化数据，支持高并发和大数据量。
- **人工智能：**
  - 集成 **OpenAI** 的 **GPT-4** API，实现 AI 问答和智能解答功能。

## 2.4 设计目标

系统设计目标如下：

- **满足用户需求：**提供全面的考研备考支持，提升用户学习效率。
- **高可用性：**系统稳定运行，支持大量用户同时在线。
- **安全性：**保护用户数据安全，防止信息泄露。
- **可扩展性：**系统架构设计支持功能扩展和升级。
- **易维护性：**代码规范，文档齐全，便于后期维护。

### 2.4.1 总体原则

#### 2.5.1 总体原则

系统设计遵循以下原则：

- **用户至上：**以用户需求为导向，提供良好的用户体验。
- **技术先进：**采用先进技术，提升系统性能和竞争力。
- **安全可靠：**确保系统的安全性和可靠性。
- **标准规范：**遵循相关行业标准 and 规范。

### 2.4.2 实用性和先进性

#### 2.5.2 实用性和先进性

系统功能设计以实用性为基础，采用先进的技术手段，如 uni-app 框架、云开发平台、人工智能等，提升用户体验和系统性能。

### 2.4.3 标准化、开放性、兼容性

#### 2.5.3 标准化、开放性、兼容性

系统遵循国际标准，采用开放的技术架构，确保与其他系统的兼容性和可扩展性。使用 uni-app 框架可以兼容多端，满足不同用户的使用需求。

#### **2.4.4 高可靠性、稳定性**

##### **2.5.4 高可靠性、稳定性**

依托 uniCloud 云服务的高可靠性，通过云函数和云数据库的稳定性能，保证系统的高可用性和稳定性。

#### **2.4.5 易用性**

##### **2.5.5 易用性**

界面友好，操作简便，采用 Vue3 的组件化开发方式，提高开发效率和用户交互体验。

#### **2.4.6 灵活性和可扩展性**

##### **2.5.6 灵活性和可扩展性**

模块化设计，支持功能的灵活组合和扩展，云开发平台便于后期功能的快速上线和升级。

#### **2.4.7 经济性和投资保护**

##### **2.5.7 经济性和投资保护**

利用云开发平台减少服务器投入和运维成本，合理控制开发和维护成本，保护投资，实现资源的高效利用。

## 第三章 系统功能模块详细设计

### 3.1 个人办公

#### 3.1.1 通知公告

##### 3.1.1 通知公告

功能描述：提供系统通知和公告的发布与展示，用户可以查看最新的通知信息。

实现方案：

- 后端在 uniCloud 中创建云函数，提供通知公告的 CRUD 接口。
- 前端使用 uni-app 实现通知列表和详细内容的展示，支持推送通知功能。

#### 3.1.2 待办事宜

##### 3.1.2 待办事宜

功能描述：用户可以添加、查看、编辑和删除待办事项，帮助管理学习和生活任务。

实现方案：

- 待办事项数据存储在 uniCloud 云数据库中，关联用户 ID。
- 前端提供待办事项的管理界面，使用 Vue3 组件实现，支持提醒功能。



## 第四章 性能设计

### 4.1 响应时间

#### 4.1 响应时间

系统应保证关键操作的响应时间不超过 1 秒，一般操作的响应时间不超过 3 秒，确保用户操作的流畅性。

### 4.2 并发用户数

#### 4.2 并发用户数

系统设计支持同时在线用户数达到 1000 人，并利用 uniCloud 的弹性伸缩能力，满足高并发需求。

## 第五章 数据库设计

数据库采用 uniCloud 云数据库，主要数据表（集合）设计如下：

- **users**：存储用户信息（用户名、密码、个人简介、头像、福币等）。
- **tasks**：存储待办事宜（任务内容、开始时间、结束时间、提醒设置等）。
- **announcements**：存储通知公告（标题、内容、发布时间等）。
- **study\_\_plans**：存储学习计划（学习目标、科目、进度等）。
- **resources**：存储学习资源（课程、题库、资料链接等）。
- **posts**：存储社区帖子（帖子内容、作者、评论、点赞等）。
- **comments**：存储帖子评论（评论内容、作者、回复对象等）。

## 第六章 接口设计

### 6.1 登录接口

#### 6.1.1 调用说明

用于用户登录验证，获取访问令牌。

#### 6.1.2 请求报文

请求方式	POST
请求 URL	/login
请求头	Content-Type: application/json
请求体	<pre>{   "username": "user123",   "password": "pass123" }</pre>

#### 6.1.3 应答报文

状态码	200 OK
响应头	Content-Type: application/json
响应体	<pre>{   "token": "jwt-token",   "user": {     "id": "user-id",     "username": "user123"   } }</pre>

## 6.2 查询所有数据列接口

### 6.2.1 6.2.1 调用说明

获取指定数据表的所有字段信息。

### 6.2.2 6.2.2 请求报文

请求方式	GET
请求 URL	/api/metadata/collection_name
请求头	Authorization: Bearer jwt-token

### 6.2.3 6.2.3 应答报文

状态码	200 OK
响应头	Content-Type: application/json
响应体	<pre>{   "collection": "collection_name",   "fields": ["field1", "field2", "field3"] }</pre>

## 6.3 修改指定数据接口

### 6.3.1 6.3.1 调用说明

修改指定 ID 的数据记录。

### 6.3.2 6.3.2 请求报文

请求方式	PUT
请求 URL	/api/collection_name/id
请求头	<ul style="list-style-type: none"><li>Content-Type: application/json</li><li>Authorization: Bearer jwt-token</li></ul>

请求体	<pre>{   "field": "new_value" }</pre>
-----	---

6.3.3 6.3.3 应答报文

状态码	200 OK
响应头	Content-Type: application/json
响应体	<pre>{   "message": "Update successful" }</pre>

6.4 删除指定数据接口

6.4.1 6.4.1 调用说明

删除指定 ID 的数据记录。

6.4.2 6.4.2 请求报文

请求方式	DELETE
请求 URL	/api/collection_name/id
请求头	Authorization: Bearer jwt-token

6.4.3 6.4.3 应答报文

状态码	200 OK
响应头	Content-Type: application/json
响应体	<pre>{   "message": "Delete successful" }</pre>

## 6.5 获取待办事项接口

### 6.5.1 6.5.1 调用说明

获取当前用户的所有待办事项。

### 6.5.2 6.5.2 请求报文

请求方式	GET
请求 URL	/api/tasks
请求头	Authorization: Bearer jwt-token

### 6.5.3 6.5.3 应答报文

状态码	200 OK
响应头	Content-Type: application/json
响应体	<pre>{   "tasks": [     {       "id": "task1",       "content": "完成项目报告",       "due_date": "2023-11-30"     },     {       "id": "task2",       "content": "复习数学",       "due_date": "2023-12-05"     }   ] }</pre>

## 6.6 修改用户信息接口

### 6.6.1 6.6.1 调用说明

修改当前用户的个人信息。

6.6.2 6.6.2 请求报文

请求方式	PUT
请求 URL	/api/users/me
请求头	<ul style="list-style-type: none"><li>Content-Type: application/json</li><li>Authorization: Bearer jwt-token</li></ul>
请求体	<pre>{   "bio": "新的个人简介",   "avatar": "base64-encoded-image" }</pre>

6.6.3 6.6.3 应答报文

状态码	200 OK
响应头	Content-Type: application/json
响应体	<pre>{   "message": "Profile updated" }</pre>

## 第七章 系统出错处理设计

### 7.1 出错信息

系统在发生错误时，返回规范的错误码和信息，方便用户和开发者理解。常见错误信息如下：

错误码	说明
400 Bad Request	请求参数错误，服务器无法理解请求。
401 Unauthorized	未授权访问，需要用户登录。
403 Forbidden	无权限操作，用户权限不足。
404 Not Found	资源不存在，无法找到请求的内容。
500 Internal Server Error	服务器内部错误，无法完成请求。

### 7.2 补救措施

针对不同错误，系统提供相应的提示和引导：

- **请求参数错误**：提示用户检查输入，确保必填项已填写，格式正确。
- **未授权访问**：引导用户登录或注册，获取访问权限。
- **无权限操作**：提示用户权限不足，如需操作请联系管理员。
- **资源不存在**：提示用户内容已被删除或链接错误，提供返回主页的选项。
- **服务器错误**：提示用户稍后重试，或联系客服获取帮助。

### 7.3 系统维护设计

系统提供日志记录和监控机制，方便排查问题和维护：

- **日志管理**：记录系统操作日志和错误日志，包括用户操作、接口调用、错误信息等。
- **监控报警**：实时监控系统状态，如 CPU、内存、网络等，出现异常及时报警。
- **备份恢复**：定期备份数据库和重要文件，提供数据恢复方案，防止数据丢失。
- **版本管理**：采用代码版本控制工具，如 Git，便于代码的管理和回溯。



## 第八章 系统处理规定

### 8.1 输入输出要求

#### 8.1 输入输出要求

系统应支持多种数据格式的输入和输出，确保数据的完整性和一致性。主要要求如下：

- **输入数据**：支持文本、数字、日期、图片、文件等多种类型的数据输入。
- **输出数据**：以 JSON 格式为主，便于数据的解析和展示；支持图片、文件的下载。
- **数据校验**：对用户输入的数据进行校验，防止错误数据或恶意数据的提交。
- **字符编码**：采用 UTF-8 编码，支持多语言字符集。

### 8.2 数据管理能力要求

#### 8.2 数据管理能力要求

系统应具备高效的数据存储和检索能力，满足以下要求：

- **数据存储**：采用 uniCloud 云数据库，支持自动扩容和高并发访问。
- **数据检索**：支持复杂查询、分页、排序等功能，提供快速的数据访问。
- **数据安全**：对敏感数据进行加密存储，防止未经授权的访问。
- **数据备份**：定期对数据库进行备份，防止数据丢失。

### 8.3 故障处理要求

#### 8.3 故障处理要求

系统应具备故障自动切换和恢复能力，保证服务的连续性：

- **故障检测**：实时监控系统状态，及时发现故障。
- **自动切换**：当主服务器发生故障时，能自动切换到备用服务器。
- **故障恢复**：提供快速的数据恢复和服务重启机制，减少故障对用户的影响。
- **应急预案**：制定详细的故障处理和应急响应方案，定期演练。

## 8.4 其他专门要求

### 8.4 其他专门要求

- **安全要求：**遵循数据安全和隐私保护法规，使用 HTTPS 加密传输，用户敏感信息加密存储。
- **性能要求：**满足高并发、高可用性的需求，利用云函数的弹性伸缩特性。
- **兼容性要求：**兼容主流操作系统和设备，uni-app 框架支持多端发布。
- **可维护性：**代码结构清晰，注释规范，便于团队协作和后期维护。
- **可扩展性：**采用模块化设计，方便功能的增加和系统的升级。