

# 深圳大学考试答题纸

(以论文、报告形式考核专用)

二〇二二~二〇二三 学年度第 一 学期

课程编号 1500720002 课程名称 软件工程 主讲教师 卢亚辉 评分

学 号 2021150039 姓名 金杨展 专业年级 计算机科学与技术 2021 级

教师评语:

题目:

系统设计--提交系统设计报告

分组开发软件项目，以 6 人为小组，第六组（天生一队）。

参与人员与分工明细

学号	姓名	主要负责内容
2021150039	金杨展	报告的撰写，智能统计分析模块
2020115009	刘涛	任务与日程模块及对应文档
2021150044	邓明涛	消息与通知、“我的”模块及对应文档
2021110246	蔡昕蓓	任务与日程模块及对应文档
2021150204	李林樵	任务与日程模块及对应文档
2021150061	许展源	文件管理模块及对应文档

本次提交报告:

3) 系统设计--提交系统设计报告

# 引言

TSYD-OA 任务系统项目以最小化的办公 OA 任务系统为基础框架辅以智能分析决策方向进行项目设计，在 likeadmin 框架的基础上开发一款基于小程序的轻量智能企业信息系统。项目致力于追求任务系统的最小化轻量设计，强化补齐任务文件管理、智能分析的短板，为中小企业用户带来清爽、直观、完整、智能的任务系统体验。我们要打造一个稳定、高效、易用的任务管理平台，以满足企业日常运营中对任务协同和信息管理的需求。

在需求分析、原型设计的基础上，本报告将基于面向对象设计的方法，深入介绍 TSYD-OA 任务系统的设计理念、体系结构以及各功能模块的详细设计，通过系统设计的原则、体系结构的规划以及数据库的设计，共三个章节来阐述。

在第一章中，我们将阐述总体设计的思路，包括系统设计的原则、系统体系结构的设计以及系统功能架构的规划。这一章的目的在于为读者提供对整个系统的宏观把握，理解系统的设计方向和基本框架。

第二章将详细讨论各功能模块的设计，包括系统管理模块、任务管理模块、文件管理模块、用户管理模块以及系统退出管理模块。每个模块都会进行功能结构设计、类图设计以及时序图设计，以确保系统的功能得到清晰而完整的表达。

在第三章，我们将深入数据库设计，包括数据库概念结构设计、数据库逻辑结构设计以及数据表的创建。数据库是系统的底层支持，设计合理的数据库结构是保障系统高效运行的基础。

最后，在第四章，我们将宏观地对整个系统的部署作出设计，设计出部署图，补充实现细节。

通过本报告的阅读，读者将全面了解 TSYD-OA 任务系统的设计思路和各功能模块的实现细节，为系统的开发和应用提供了清晰而详尽的指南。

## 目录

<b>1 总体设计.....</b>	<b>3</b>
1.1 面向对象设计方法.....	3
1.2 系统设计的原则.....	4
1.3 系统体系结构设计.....	5
1.4 系统功能架构设计.....	6
<b>2 功能模块设计.....</b>	<b>9</b>
2.1 系统管理模块.....	9
2.2 任务管理模块.....	11
2.3 文件管理模块.....	14
2.4 用户管理模块.....	16
2.5 系统消息模块.....	18
<b>3 数据库设计.....</b>	<b>20</b>
3.1 数据库概念结构设计.....	20
3.2 数据库逻辑结构设计.....	25
<b>4 系统实现及部署设计.....</b>	<b>27</b>
4.1 系统实现设计.....	27
4.2 系统部署设计.....	27

# 1 总体设计

## 1.1 面向对象设计方法

项目采用最常用的**面向对象设计**方法进行系统设计。

面向对象分析和面向对象设计是同一套理论，但存在一定的分工区别，面向对象设计是将面向对象分析阶段完成的分析模型转换为软件设计的过程。在面向对象设计阶段，软件设计人员需要在面向对象分析的基础上，完善软件设计，在原有分析模型的基础上补充与软件实现相关的信息。为了准确表达面向对象设计阶段的细节，软件设计人员可以同样使用 UML 中的：

- **类图**建模软件体系架构；
- **时序图**建模对象之间的消息交互；
- **状态图**来建模对象在运行过程中的状态。

同时，当目标软件的体系架构完成以后，软件设计人员还可以对软件实现方面的内容进行设计，例如对数据库的设计，通过组件图和部署图来建模软件实现及部署的相关细节。与此同时，在面向对象设计阶段，需求分析人员除了完成软件设计以外，还必须从软件设计质量和效率方面来优化软件设计。

## 1.2 系统设计的原则

在 TSYP-0A 任务系统的设计过程中，我们遵循一系列详细而全面的设计原则，以确保系统设计与实际需求相契合，并最大程度地发挥项目的独特优势。在系统设计时，应遵循以下设计原则：

### （一）实用性原则。

TSYP-0A 系统通过简洁的界面和直观的交互设计，使得任务规划和管理变得异常便捷，切实解决企业任务系统繁琐性的问题。同时，引入智能分析决策，系统具备对任务进行智能分析的能力，为用户提供更智能、更高效的决策支持。因此，实用性原则是本系统的一个重要设计原则。

### （二）先进性原则。

在强调实用性的同时，我们还要考虑系统的先进性。本系统的设计过程采用目前较为先进和成熟的技术和理念，使系统在性价比上拥有一定的优势。具体有：利用小程序平台的先进性，TSYP-0A 系统在移动端提供出色的用户体验，使得企业信息管理得以轻松进行。引入最新的智能决策技术，系统能够通过数据分析和机器学习，为用户提供个性化、智能化的任务分析和推荐。

### （三）规范性原则。

系统中的所有设计都遵循现有的国家标准、行业标准，符合软件工程中的设计原则与规范。所有项目文档的书写都遵照国标的要求，并保持统一规范。

#### （四）安全性原则。

系统提供安全机制以防止非法授权操作，为各子系统提供严格灵活的用户和权限划分，对于数据库管理，建立起完备的数据备份恢复机制。对关键的信息的操作提供日志记录。通过严格的用户权限划分，系统确保只有授权用户能够访问敏感信息和执行关键操作。

#### （五）可扩展性原则。

我们充分考虑了接口设计灵活性以及模块化设计：考虑前后端接口的灵活性，确保系统具备未来扩展的可能性，以适应业务的快速变化。同时，采用模块化设计，使系统的各功能模块能够独立扩展，从而更好地适应企业信息管理的不断变化。出于对后续升级工作的考虑，系统的所有功能模块的设计都预留了扩展接口。随着用户需求的不断变化和增加，信息处理量也会随之增加，这就要求 TSYP-OA 任务系统能够实现快速便捷地扩展，完成系统的平滑升级。

#### （六）可维护和可管理性原则

这要求系统能够实现对设备的便捷管理和时时监控，时时掌握设备的情况，遇到问题能进行报警处理、会话控制，以及通过性能分析，实现系统的优化、故障的隔离及故障的恢复，尽可能缩短维护时间，提高系统可维护性，延迟可系统使用时间。

#### （七）系统模块化设计原则

模块化即以功能块为单位进行程序设计，实现其求解算法的方法。模块化的目的是为了降低程序复杂度，使程序设计、调试和维护操作简单化。因此，系统采用模块化设计原则，增强了系统的灵活性和可扩展性。任务系统的业务如果需要微调以适应不同企业的需求而发生变化时，系统的功能模块也可根据实际变化情况进行快速调整。

### 1.3 系统体系结构设计

系统体系结构设计是对系统整体架构的规划和设计，包括系统的层次结构、模块划分、组件关系方面。

我们的项目 TSYP-OA 系统采用 **B/S 三层架构**，将系统的整个业务应用划分为**表示层、业务逻辑层和数据访问层**，这样有利于系统的开发、维护、部署和扩展。B/S 结构的基本原则是将计算机应用任务分解成多个子任务，由多台计算机分工完成，即采用“功能分布”原则。**客户端**完成数据处理，数据表示以及用户接口功能；**服务器端**完成 DBMS 的核心功能。这种客户请求服务、服务器提供服务的处理方式是一种新型的计算机应用模式。

下面是系统各层的总体设计：

➤ **表现层 (Presentation Layer) :**

小程序前端界面，基于 uniapp 框架开发，提供用户友好的交互体验。

通过微信小程序实现用户的登录、任务查看、文件管理功能。

➤ **业务逻辑层 (Business Logic Layer) :**

包含系统的核心业务逻辑，处理用户请求、任务管理、文件操作功能。

利用云函数和后端服务器实现业务逻辑的处理，确保系统的稳定性和可扩展性。

➤ **数据访问层 (Data Access Layer) :**

负责与数据库交互，进行数据的读取和存储。

使用云数据库存储用户信息、任务数据、文件信息关键数据。

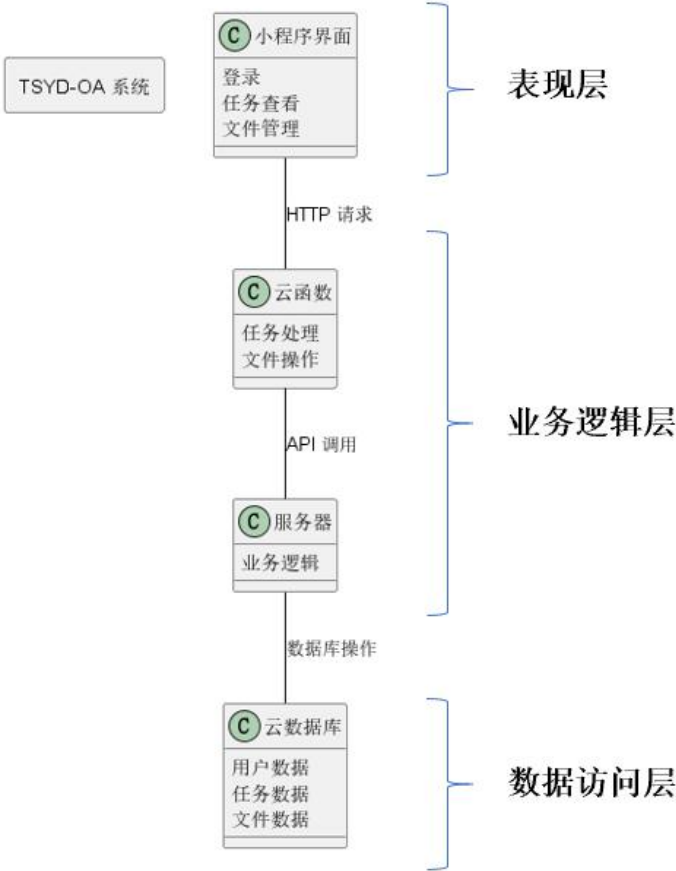


图 1 TSYD-OA 系统采用 B/S 三层架构示意图

B/S 结构的优点：

- 1、**交互性强。**在 B/S 模式中，客户端有一套完整的应用程序，在出错提示、在线帮助方面都有强大的功能，充分的发挥客户端的处理能力；
- 2、**安全性高。**由于 B/S 是配对的点对点的结构模式，采用适用于局域网、安全性比较好的网络协议，具有较高的安全性；
- 3、**网络通信量少。**B/S 的网络通信量只包括客户端与服务器之间的通信量。因而，B/S 具有较少的网络通信量，具有较强的数据处理能力，对应了轻量化的需求；

**4、数据处理能力强。**由于 B/S 的三层逻辑结构处理数据时，处理速度较快，更适合处理大量数据。

#### 1.4 系统功能架构设计

系统功能架构设计旨在明确系统的各项功能，并将其划分为不同的模块，以便更好地实现系统的功能特性。TSYD-OA 任务系统一共分为两个子系统，用户任务管理子系统和后台管理子系统。其中，小程序端的用户任务管理子系统分为四个模块，Web 端的后台管理子系统为系统管理模块（在 likeadmin 基础上设计）。

我们的项目 TSYD-OA 系统的**用户任务管理子系统**分为以下四个**功能模块**。

##### （一）系统消息模块：

- ◆ **功能模块描述：** 该模块主要用于展示用户接收到的系统消息通知，方便用户在一进入小程序就能一目了然知道哪些任务（日程）已收到提交、哪些任务（日程）即将到期。
- ◆ **系统消息：** 系统信息通知包括系统通知、任务提醒、日程安排提醒、文件上传成功提醒，有助于员工及时了解重要信息，保持团队协作的效率。
- ◆ **操作方式：** 首页展示不同种类的系统信息，可以区分已读/未读进行分栏显示，用户可以通过该模块首页进入查看某类系统消息（如任务提醒系统）的详细内容。

##### （二）任务模块：

- ◆ **功能模块描述：** 该模块是日程和任务管理的入口，根据前文提到的两种原型实现方式，可能是日历或任务列表。为员工提供了统一的时间管理和任务完成的入口，方便用户查看和规划工作日程。
- ◆ **任务创建与分配：** 允许用户创建新任务，并将任务分配给特定成员。
- ◆ **任务查看和更新：** 提供任务列表，支持用户查看和更新任务状态、优先级信息。
- ◆ **任务完成与反馈：** 实现任务的完成操作，并允许用户提供任务反馈。
- ◆ **操作方式：** 用户可以在该模块中添加、编辑、删除日程或任务，设定提醒，查看详细信息。

##### （三）文件模块：

- ◆ **功能模块描述：** 该模块用于管理和浏览与项目、任务相关的文件和文档，可能包括上传、下载、分享功能。提供了一个集中管理文件的地方，方便团队成员协作时共享和查找所需的文件。

- ◆ 文件上传和下载： 允许用户上传和下载与任务相关的文件。
- ◆ 文件管理： 提供文件的查看、删除管理功能。
- ◆ 操作方式： 用户可以通过该模块上传、下载、分享文件，创建文件夹进行分类管理。

#### （四）用户管理模块：

- ◆ 功能模块描述： 该模块主要是用户的个人中心，包含个人信息、设置、权限相关功能。  
用户可以通过该模块查看和编辑个人信息，包括极简化的任务总结。
- ◆ 用户登录和退出： 提供用户登录和退出功能，确保系统安全性。
- ◆ 用户信息管理： 允许用户查看和编辑个人信息，包括密码修改、头像上传。
- ◆ 权限控制： 实现对用户权限的精细管理，确保信息安全。
- ◆ 操作方式： 用户可以在该模块中修改个人信息、更改密码，以满足个性化需求。也可以在账户总结中清晰看到已完成、未完成的任务的数量。

小程序端的用户任务管理子系统分为四个模块，Web 端的后台管理子系统为系统管理模块，设计模块功能架构示意图如下。

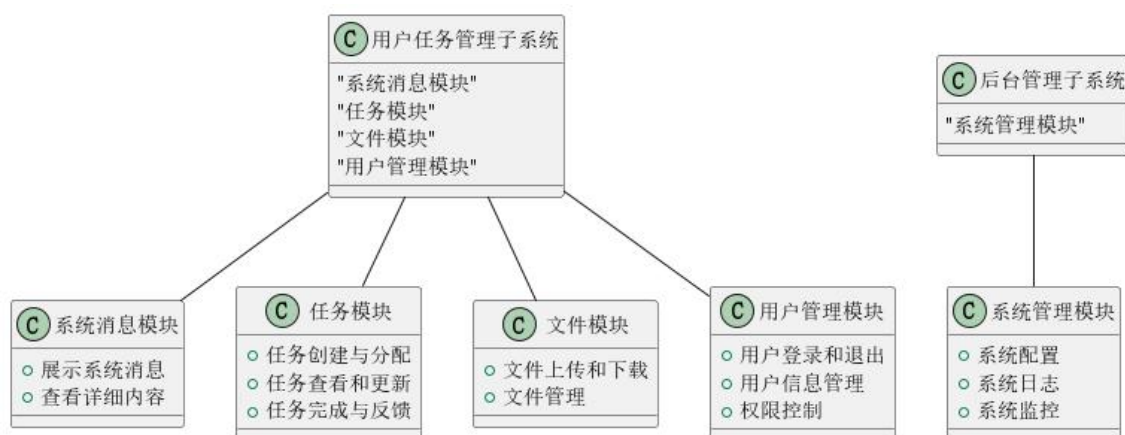


图 2 TSYD-0A 模块功能架构示意图

除了关注模块划分，我们也关注**模块间交互**，这是设计组件关系的基础。系统的不同功能模块之间存在紧密的交互，以实现协同工作和信息流通。

- （一）**用户管理模块与任务管理模块交互**，确保任务的创建和分配与特定用户相关联。
- （二）**任务管理模块与文件管理模块关联**，确保任务附带的文件能够被正确管理和使用。
- （三）**系统管理模块与用户管理模块进行交互**，以实现全面的系统管理和监控。系统管理模块能够与用户管理模块协同工作，实现对用户权限的灵活控制。管理员可以通过系统管理模块为特定用户分配任务、文件的访问权限。
- （四）**系统管理模块与任务管理模块进行交互**，系统管理模块能够接收任务管理模块反



馈的任务执行状态信息。这包括任务的进度、完成情况。管理员可以通过系统管理模块查看任务的实时执行情况，从而进行及时调整和监控。

（五）**系统管理模块与文件管理模块进行交互**，系统管理模块与文件管理模块协同工作，实现对文件的权限管理。管理员可以通过系统管理模块设置文件的访问权限，确保敏感信息的安全性。

通过上述功能架构设计，TSYD-OA 系统将具备完备的任务管理、文件管理、用户管理功能，以满足企业任务系统的需求。同时，模块间的良好交互确保系统的高效运作和用户体验。

## 2 功能模块设计

### 2.1 系统管理模块

#### 2.1.1 功能结构设计

系统管理模块作为 TSYD-OA 任务系统的 Web 端的后台管理子系统主要模块，基于成熟的 likeadmin 进行基本的完善设计，旨在提供对系统整体运行的管理和监控功能，以确保系统平稳运行并提供良好的用户体验。以下是系统管理模块的功能结构设计：

- **用户管理：** 实现用户的注册（新增）、注销基本操作，同时支持用户信息的查看和修改。
- **部门管理：** 实现部门的新增、删除基本操作，同时支持部门用户信息的查看和修改。
- **权限管理：** 精细化的权限控制，确保用户只能访问其具备权限的功能和数据。
- **系统设置：** 提供系统基本参数的配置，包括界面主题、语言设置，以满足用户个性化需求。
- **日志管理：** 记录关键操作日志，便于系统管理员追踪用户行为和系统运行状况。
- **通知与消息：** 实现系统向用户发送通知和消息，确保及时的沟通和信息传递。

#### 2.1.2 类图设计

系统管理模块的类图主要包括以下核心类：

**User（用户）：** 包含用户的基本信息，用户名、密码等。

**Permission（权限）：** 定义系统中各功能模块的权限，与用户进行关联，实现权限控制。

**SystemSetting（系统设置）：** 包括系统的基本设置信息，界面主题、语言。

**Log（日志）：** 记录系统的关键操作日志，包括用户操作、系统事件。

**Notification（通知）：** 存储系统向用户发送的通知和消息。

由于 likeadmin 有较好实现，项目开发不涉及其他更多类的修改。

以下是系统管理模块的简化类图设计：

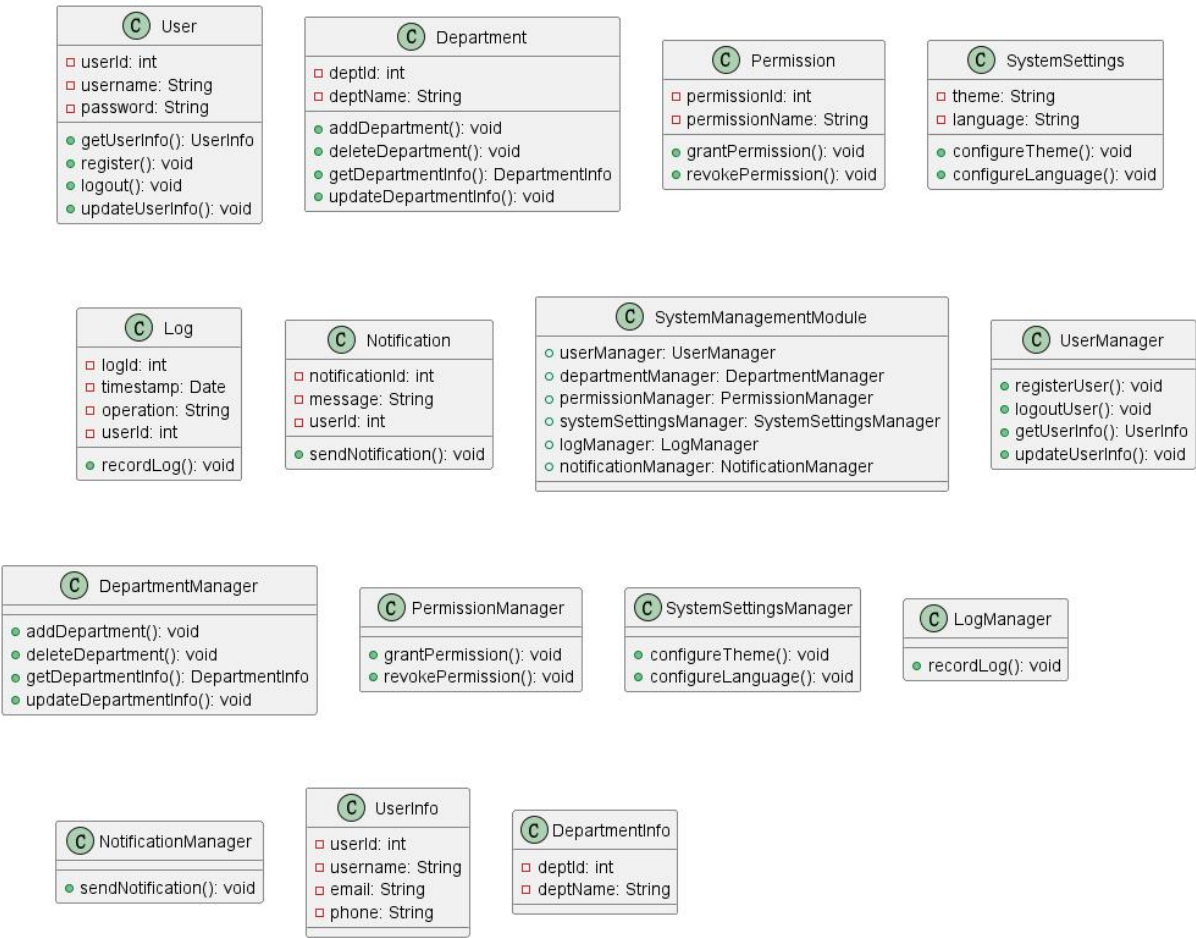


图 3 系统管理模块类图

### 2.1.3 时序图设计

系统管理模块的时序图描述了不同功能之间的交互过程，以下是系统管理模块中用户登录的简化时序图：

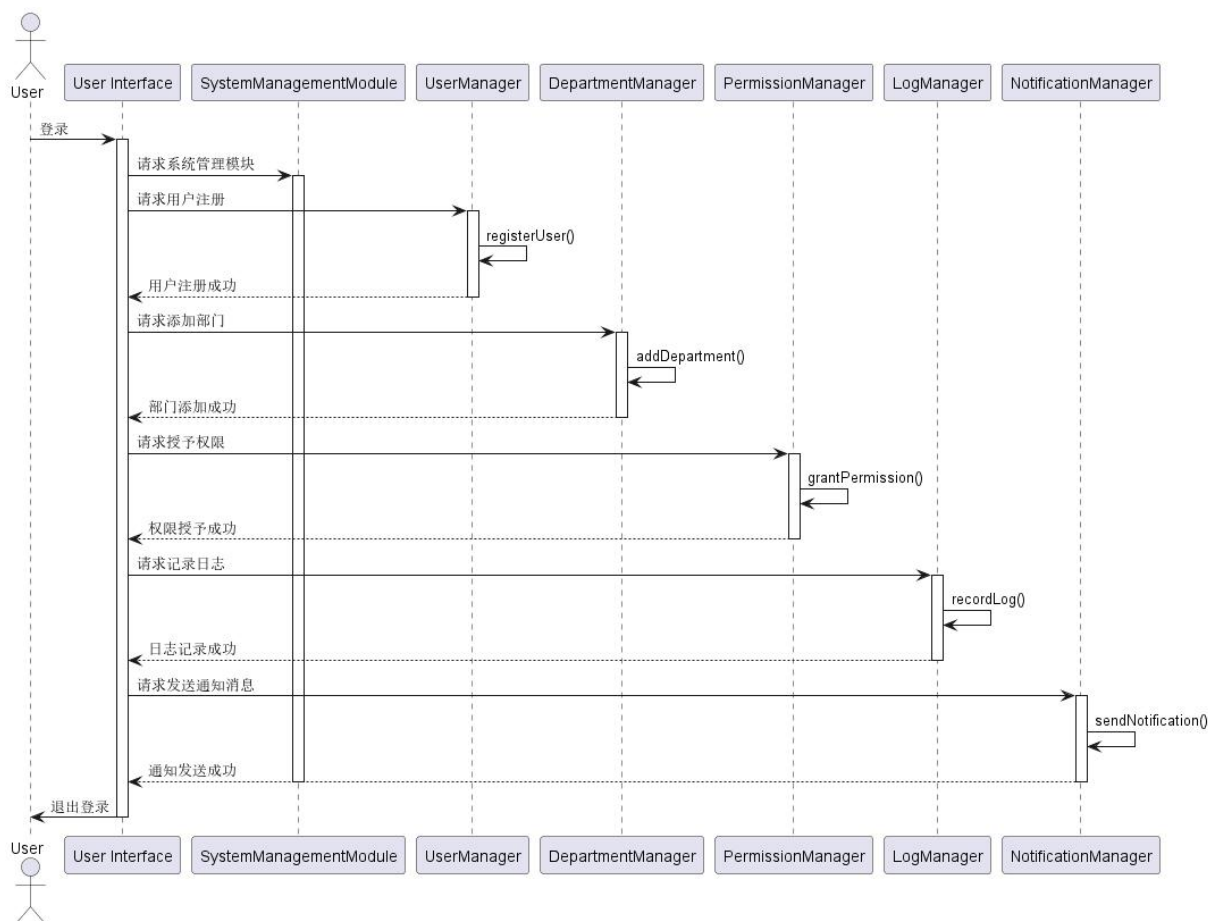


图 4 系统管理模块时序图

在时序图中，前后表示不同步骤的顺序，箭头表示消息的流向，描述了用户登录、注册、添加部门、授予权限、配置系统参数、记录日志和发送通知消息交互过程。以登录为例，用户通过发送登录请求，系统验证用户信息，然后向用户发送用户信息，完成登录过程。

## 2.2 任务管理模块

### 2.2.1 功能结构设计

任务管理模块是 TSYP-OA 任务系统的核心功能之一，旨在为用户提供高效的任务管理和协作功能。以下是任务管理模块的功能结构设计：

- **任务创建与分配：** 允许用户创建新任务并分配给特定成员或团队。
- **任务状态管理：** 提供任务状态跟踪，包括任务进行中、已完成、延期状态。
- **文件管理：** 支持任务相关文件的上传、下载和共享，确保团队成员能够方便地访问所需文件。
- **智能提醒：** 设置任务截止日期和重要程度，系统将自动发送提醒，确保任务按时完成。
- **任务进度追踪：** 提供任务进度可视化，让用户清晰了解任务完成情况。

- **评论与讨论：** 允许团队成员在任务中进行评论和讨论，促进协作，智能分析的结果也可以放在这里。

### 2.2.2 类图设计

任务管理模块的类图主要包括以下核心类：

**Task（任务）：** 包含任务的基本信息，任务名称、描述、截止日期。

**User（用户）：** 任务的创建者和参与者，与任务进行关联。

**File（文件）：** 任务相关的文件，与任务和用户进行关联。

**Comment（评论）：** 用户在任务中的评论，与任务和用户进行关联。

以下是任务管理模块的简化类图设计：

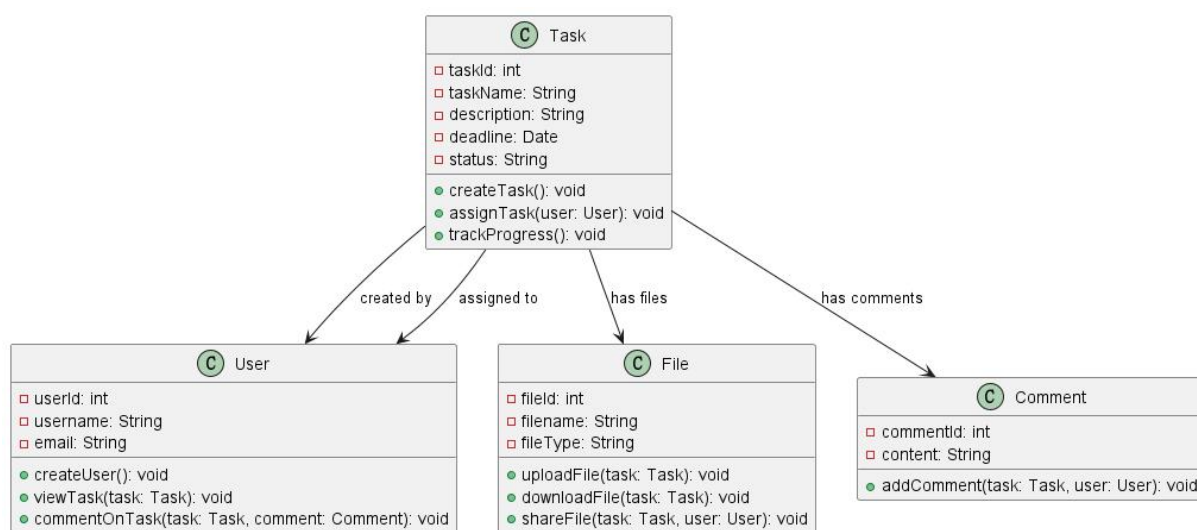


图 5 任务管理模块类图

### 2.2.3 时序图设计

任务管理模块的时序图描述了用户创建任务的过程，以下是任务创建与分配的简化时序图：

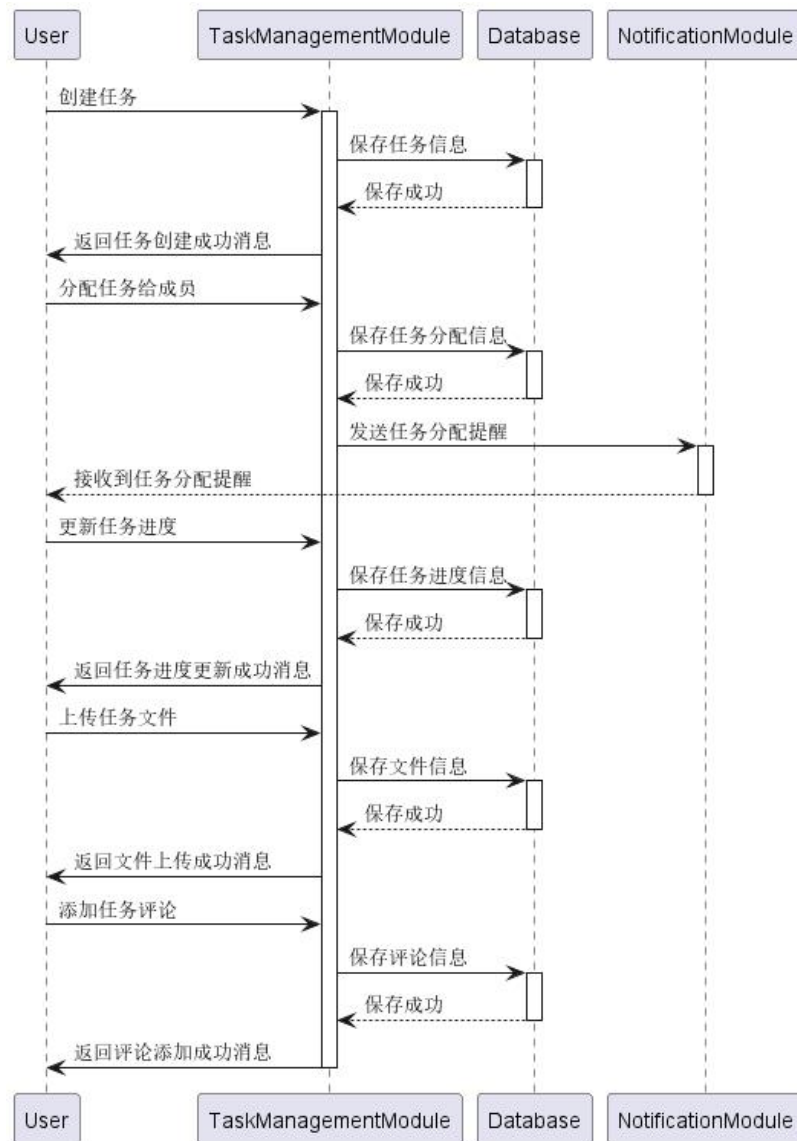


图 6 任务管理模块时序图

在时序图中，描述了用户创建任务、分配任务、更新任务进度、上传文件和添加评论（或智能分析）的过程。用户通过发送创建任务的请求，系统保存任务的详细信息，然后用户可以分配任务给特定成员或团队。这个过程保证了任务的有效创建和分配。

#### 2.2.4 任务状态图设计

状态图（State Diagram）是一种描述对象在其生命周期中各种状态变化及相应事件触发的图表。在任务管理模块的情境下，我们可以为任务对象设计状态图，表示任务在不同阶段的状态及状态之间的转换。

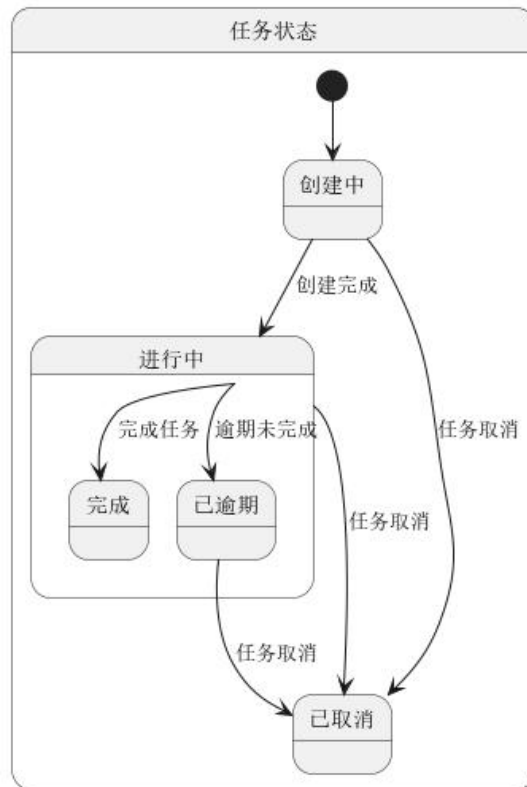


图 7 任务管理模块任务状态图

状态图解释：

- ✧ 初始状态为“创建中”，表示任务创建中。
- ✧ 从“创建中”状态，任务可以进入“进行中”状态，表示任务创建完成，正在进行中。
- ✧ 在“进行中”状态，任务可以进入三个不同的子状态：
- ✧ “完成”：任务成功完成。
- ✧ “已逾期”：任务未在截止日期前完成。
- ✧ “已取消”：任务被取消。
- ✧ 在“已逾期”状态，任务可以进入“已取消”状态，表示逾期未完成的任务被取消。
- ✧ “已取消”和“完成”状态为最终状态，表示任务的最终状态要么是成功完成，要么是被取消。

## 2.3 文件管理模块

### 2.3.1 功能结构设计

文件管理模块旨在提供高效的任务文件上传、下载和共享功能，确保团队成员能够方便地访问所需文件。以下是文件管理模块的功能结构设计：

- **文件上传与下载：** 允许用户上传任务相关文件，并支持其他成员下载。
- **文件共享与权限管理：** 提供文件共享功能，同时确保对文件的访问权限进行灵活管理。

- **文件夹管理：** 允许用户创建文件夹进行文件分类和组织。

### 2.3.2 类图设计

文件管理模块的类图主要包括以下核心类：

**File（文件）：** 包含文件的基本信息，文件名、大小、上传者。

**User（用户）：** 与文件进行关联，表示文件的上传者或具有文件访问权限的用户。

**Folder（文件夹）：** 用于组织文件，可以包含其他文件或文件夹。

**Permission（权限）：** 管理文件的访问权限，与文件和用户进行关联。

以下是文件管理模块的简化类图设计：

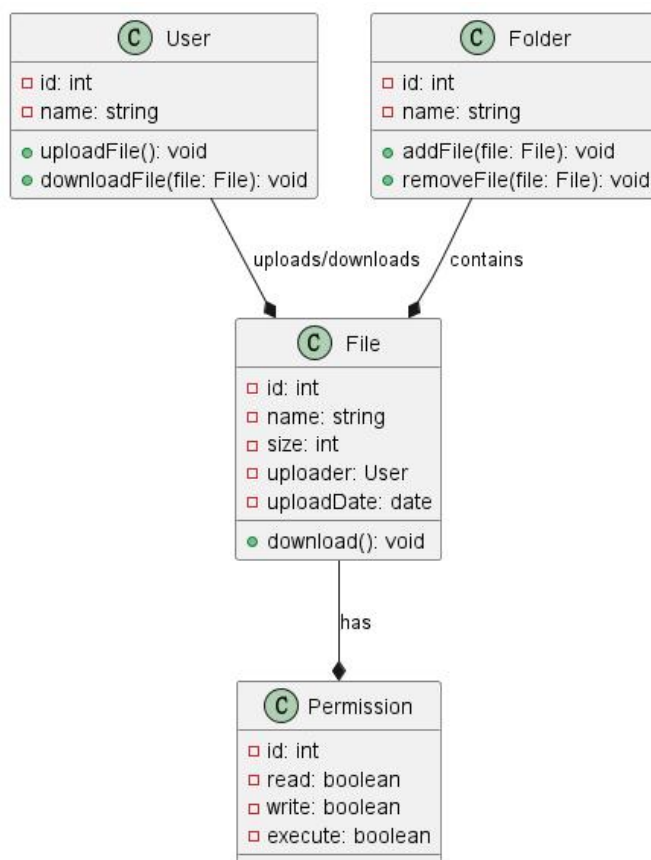


图 8 文件管理模块类图

### 2.3.3 时序图设计

文件管理模块的时序图描述了用户上传文件的过程，以下是文件上传与下载的简化时序图：

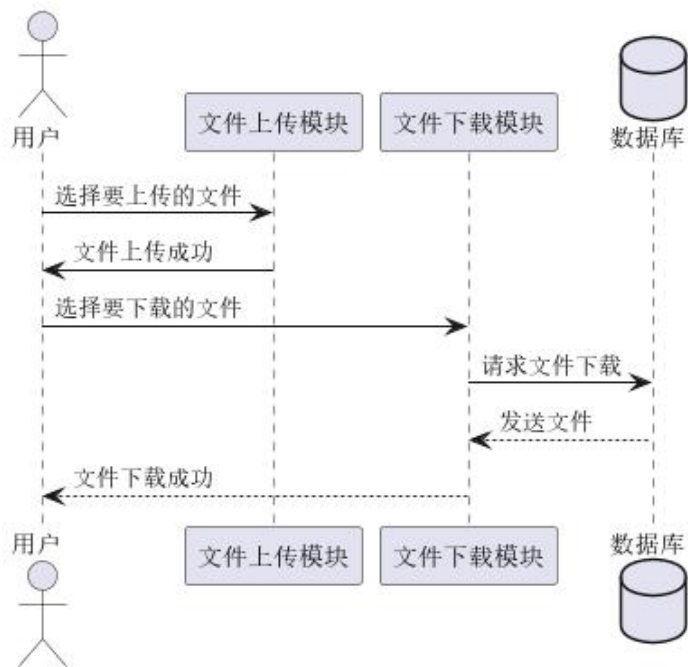


图 9 文件管理模块时序图

在时序图中，用户通过发送上传文件的请求，系统保存文件的详细信息，然后其他成员可以通过发送下载文件的请求获取文件。这个过程确保了文件的有效上传和下载。

需要注意的是，通常情况下，服务器并不直接将文件保存在数据库中。相反，服务器上会使用文件系统或云存储来存储文件，而数据库则通常用于存储文件的元数据（文件名、路径、上传时间信息）。

文件系统是一种直接将文件以二进制形式保存在服务器文件系统中的方法。每个文件都有一个唯一的路径，通过这个路径可以在文件系统中找到和检索文件。这种方式适用于较小的文件和中小型系统。

## 2.4 用户管理模块

### 2.4.1 功能结构设计

用户管理模块旨在提供对系统中用户的灵活管理和权限控制。以下是用户管理模块的功能结构设计：

- **用户信息管理：** 允许用户查看和编辑个人信息，用户名、密码。
- **角色与权限管理：** 管理用户角色和权限，确保不同用户有不同的系统访问权限。
- **部门管理：** 允许管理部门，并将部门成员添加到任务中。



2.4.2 类图设计

用户管理模块的类图主要包括以下核心类：

**User（用户）：** 包含用户的基本信息，用户名、密码、角色。

**Role（角色）：** 定义系统中的不同角色，普通员工、部门领导。

**Department（部门）：** 用于组织用户，可以包含多个成员。

**Permission（权限）：** 管理用户的访问权限。

以下是用户管理模块的简化类图设计：

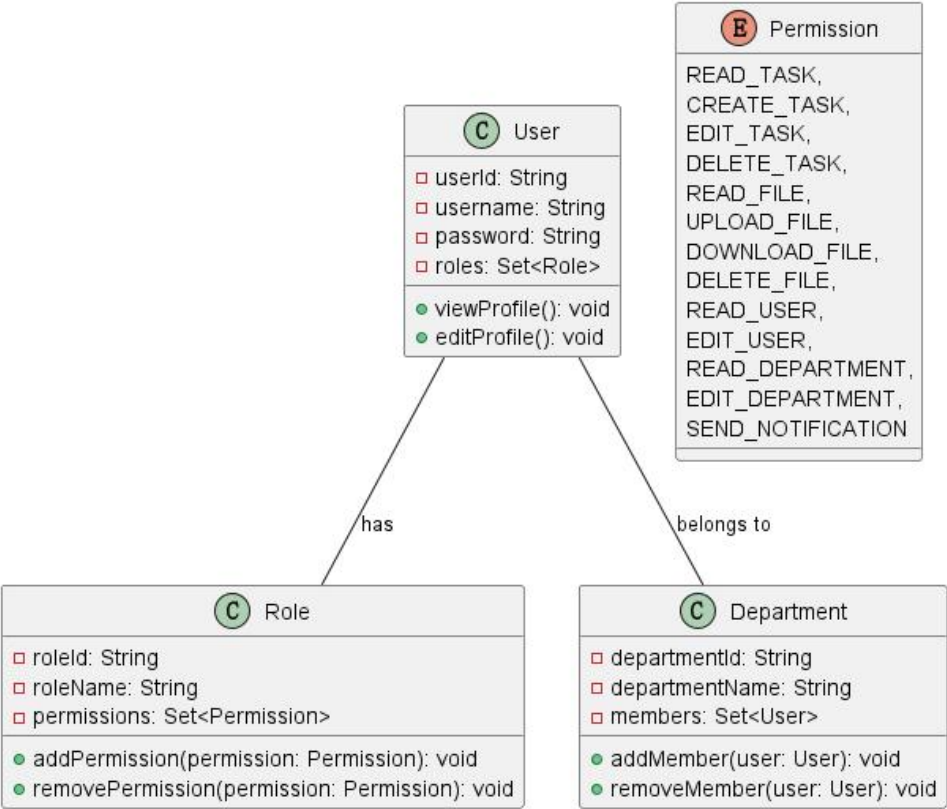


图 10 用户管理模块类图

2.4.3 时序图设计

用户管理模块的时序图描述了用户注册和登录的过程，以下是用户注册与登录的简化时序图：

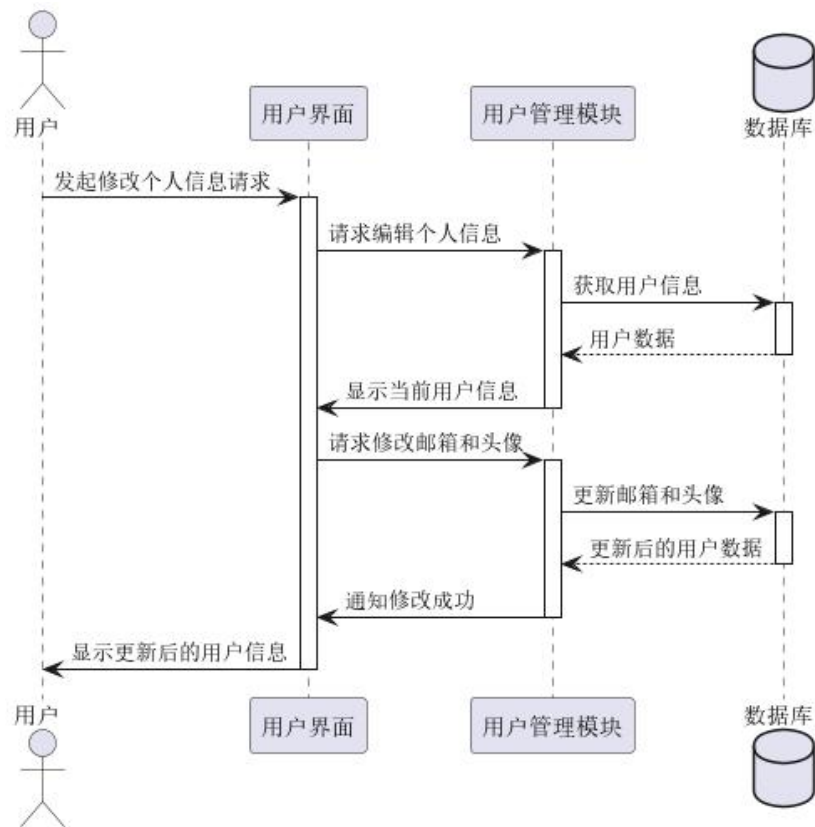


图 11 用户管理模块时序图

上面的时序图描述了用户管理模块中用户修改邮箱和头像的过程。在此时序图中，用户通过用户界面发起修改个人信息的请求，用户管理模块收到请求后从数据库中获取用户信息，显示当前用户信息。用户界面再次请求修改邮箱和头像，用户管理模块更新数据库中的信息，并通知用户界面修改成功，最终显示更新后的用户信息。

## 2.5 系统消息模块

### 2.5.1 功能结构设计

该模块主要用于展示用户接收到的系统消息通知，方便用户在一进入小程序就能一目了然知道哪些任务已收到提交、哪些任务即将到期。系统消息通知包括系统通知、任务提醒、文件上传成功提醒，有助于员工及时了解重要信息，保持团队协作的效率。功能结构设计：

- **部门通知与消息：** 提供系统通知和消息功能，使用户及时了解部门动态。
- **任务提醒：** 提供系统通知和消息功能，使用户及时了解任务提交和即将到期的动态。
- **文件上传成功提醒：** 提供文件上传成功提醒，使用户及时了解文件上传动态。

### 2.5.2 类图设计

系统消息模块的类图主要包括以下核心类：

**SystemMessage（系统消息）：** 通用系统通知和消息功能。

下分 3 个子类：

DepartmentNotification（部门通知）

TaskReminder（任务提醒）

FileUploadNotification（文件上传提醒）

以下是系统消息模块的简化类图设计：

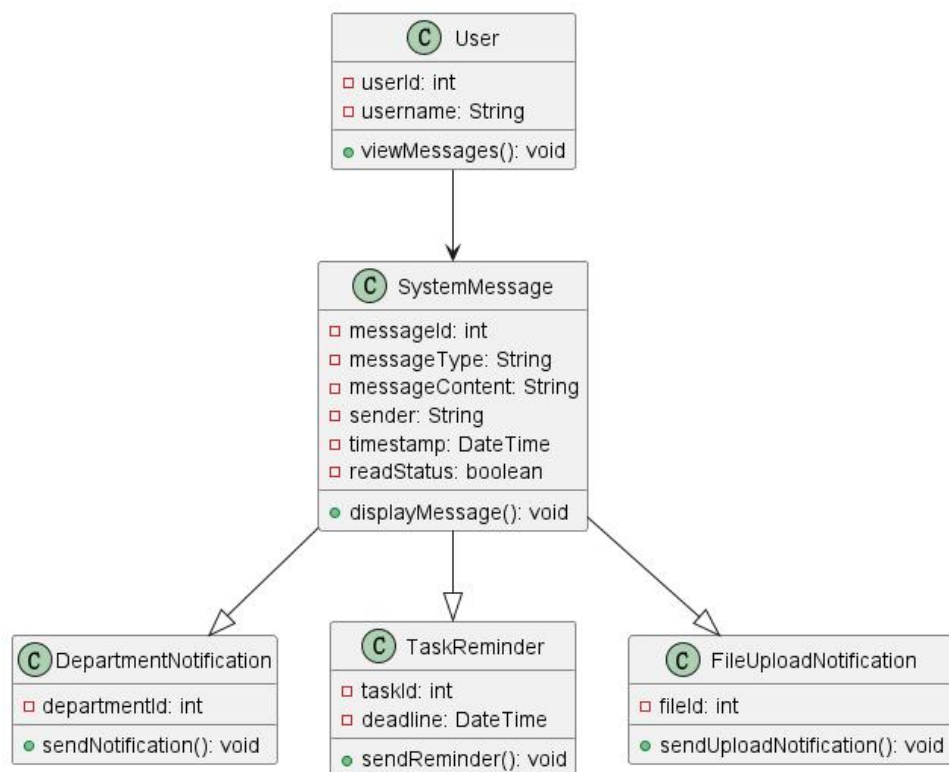


图 12 系统消息模块类图

### 2.5.3 时序图设计

用户查看系统消息的时序图：

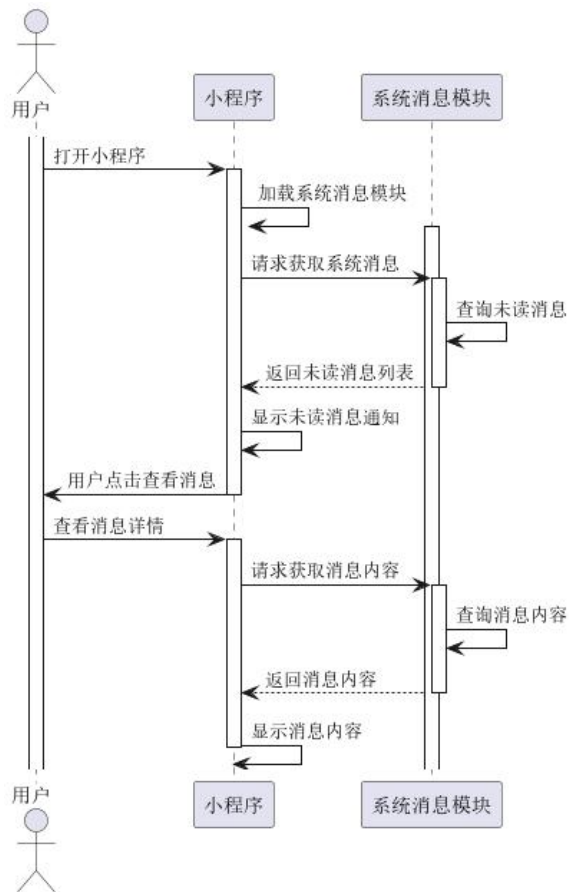


图 13 系统消息模块时序图

上述是用户查看系统消息的时序图示例，其中包含了用户打开小程序、加载系统消息模块、请求获取系统消息、查看消息通知、查看消息详情过程。

## 3 数据库设计

### 3.1 数据库概念结构设计

数据库概念结构设计是在数据库设计过程中，将用户需求抽象为概念模型的过程。

#### 3.1.1 实体属性图

数据库的概念结构设计通过实体属性图展现系统中的主要实体及它们之间的关系。以下是 TSYD-0A 任务系统的实体属性图，具体描述了系统中涉及的关键实体及其属性。

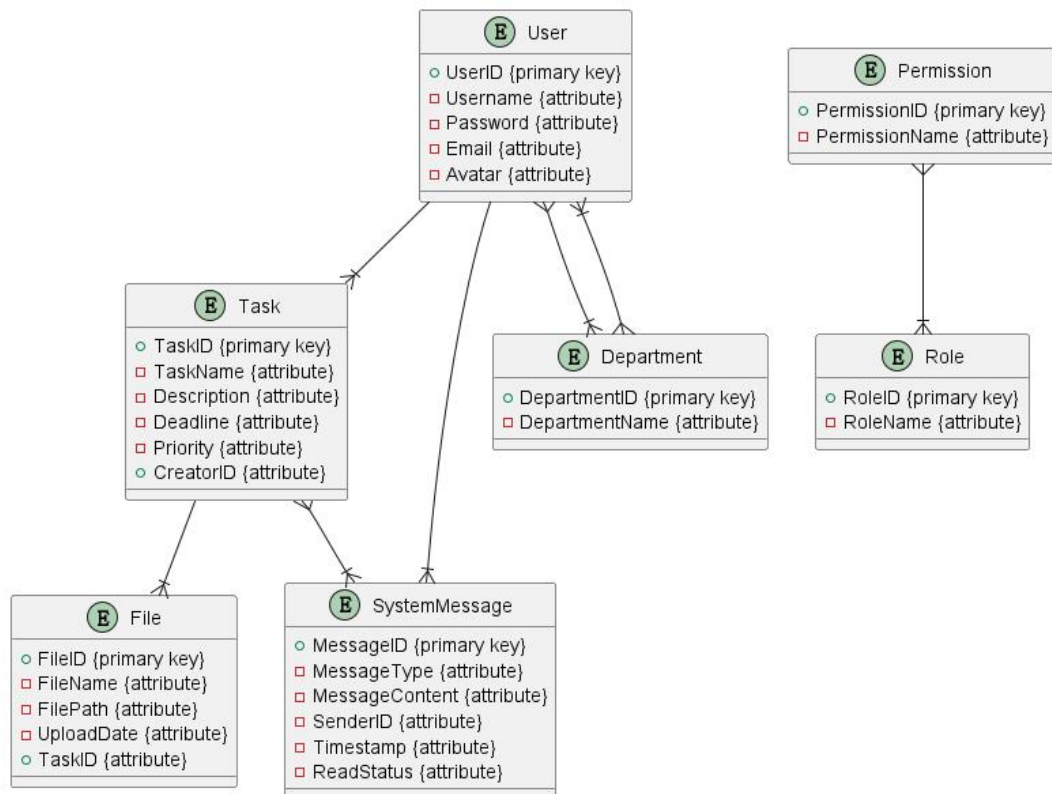


图 14 TSXD-0A 任务系统实体属性图

在上述实体属性图中，有七个核心实体：User（用户）、Task（任务）、File（文件）。它们之间通过关系进行连接，例如 Task 中包含了 assigneeId，与 User 中的 userId 形成关联。此外，File 实体与 User 实体之间也存在关联，用于记录文件的上传者信息。这一实体属性图为后续数据库逻辑结构设计提供了基础，确保系统中的数据能够被有效地组织和管理。

### 3.1.2 局部 E-R 图

局部 E-R 图是数据库概念结构设计的进一步细化，它呈现了某一特定部分的实体和关系，有助于更详细地理解系统中数据的结构。

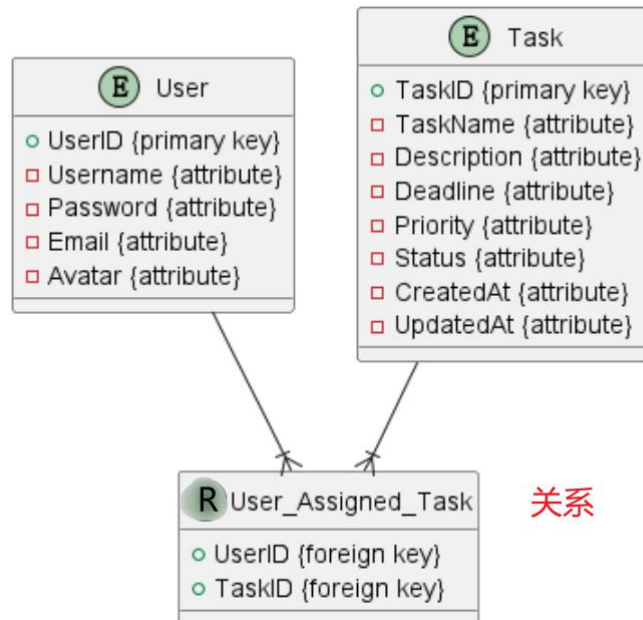


图 15 TSJD-OA 用户和任务的局部 ER 图

这是一个包含用户和任务的局部 ER 图，展示了它们之间的关系。以下是对图中元素的详细解释：

➤ **User（用户）：**

**UserID（用户 ID）** 是员工用户实体的主键，用于唯一标识每个员工用户。

**Username（用户名）** 是用户的属性，表示用户的登录名。

（非完整用户表，具体在 3.2.2 都有定义）

➤ **Task（任务）：**

**TaskID（任务 ID）** 是任务实体的主键，用于唯一标识每个任务。

**TaskName（任务名称）** 是任务的属性，表示任务的名称。

**Description（描述）** 是任务的属性，用于详细描述任务的内容。

**Deadline（截止日期）** 是任务的属性，表示任务的完成截止日期。

**Status（状态）** 是任务的属性，表示任务的当前状态（未开始、进行中、已完成、已取消）。

**CreatedAt（创建时间）** 是任务的属性，表示任务创建的时间。

**UpdatedAt（更新时间）** 是任务的属性，表示任务最后一次更新的时间。

➤ **User\_Assigned\_Task（用户分配的任务关系）：**

**UserID 和 TaskID** 分别是用户和任务的外键，建立了用户和任务之间的关系。

这个关系表示一个用户可以被分配给多个任务，同时一个任务可以被分配给多个用户，形成了**多对多**的关系。

这个 ER 图描述了用户和任务之间的关系，以及任务的一些重要属性。ER 图的目的是通过

图形方式清晰地表达实体之间的关系，如 Task 中的 assigneeId 与 User 中的 userId 形成外键关系，有助于维护数据的完整性和一致性。局部 E-R 图为后续的数据库逻辑结构设计奠定了基础。

### 3.1.3 总体 E-R 图

在 TSYD-OA 系统的数据库设计中，总体 E-R 图描述了系统中各个实体之间的关系，为系统数据库的整体架构提供了清晰的视图。以下是总体 E-R 图的主要元素：

#### （一）实体（Entity）解释：

##### 1) 任务（Task）实体：

属性包括任务编号、任务名称、任务描述、截止日期。

与用户实体存在“负责人”关系，表示每个任务都由特定的用户负责。

##### 2) 文件（File）实体：

包括文件编号、文件名称、文件类型、上传时间属性。

与用户实体存在“上传者”关系，表示每个文件都由特定的用户上传。

##### 3) 用户（User）实体：

包括用户 ID、用户名、密码属性。

与任务实体存在“负责人”关系，表示每个用户可以负责多个任务。

与文件实体存在“上传者”关系，表示每个用户可以上传多个文件。

##### 4) 权限（Permission）实体：

包括权限 ID、权限名称属性。

与用户实体存在“拥有权限”关系，表示每个用户可以拥有多种权限。

##### 5) 部门（Department）实体：

包括部门 ID、部门名称属性。

与用户实体存在“所属部门”关系，表示每个用户属于特定的部门。

##### 6) 系统消息（System Message）实体：

包括消息 ID、消息内容、接收者属性。

与用户实体存在“接收者”关系，表示每条系统消息涉及特定的接收者。

#### （二）关系（Relationship）解释：

1) **User\_Responsible\_For\_Task（用户负责任务关系）：** 表示用户与任务之间的关系，一个用户可以负责多个任务，一个任务只有一个负责人。

2) **User\_Uploads\_File（用户上传文件关系）：** 表示用户与文件之间的关系，一个用户可以上传多个文件，一个文件只有一个上传者。

- 3) **User\_Has\_Permission (用户拥有权限关系)**：表示用户与权限之间的关系，一个用户可以拥有多个权限，一个权限可以被多个用户拥有。
- 4) **User\_Belongs\_To\_Department (用户所属部门关系)**：表示用户与部门之间的关系，一个用户可以属于一个部门，一个部门可以有多个用户。
- 5) **Task\_Uploaded\_To\_File (任务上传到文件关系)**：表示任务与文件之间的关系，一个任务可以上传到多个文件，一个文件可以包含多个任务。
- 6) **Message\_Sent\_To\_User (消息发送给用户关系)**：表示消息与用户之间的关系，一条消息可以发送给多个用户，一个用户可以接收多条消息。

总体 E-R 图如下，注意属性有部分简化。

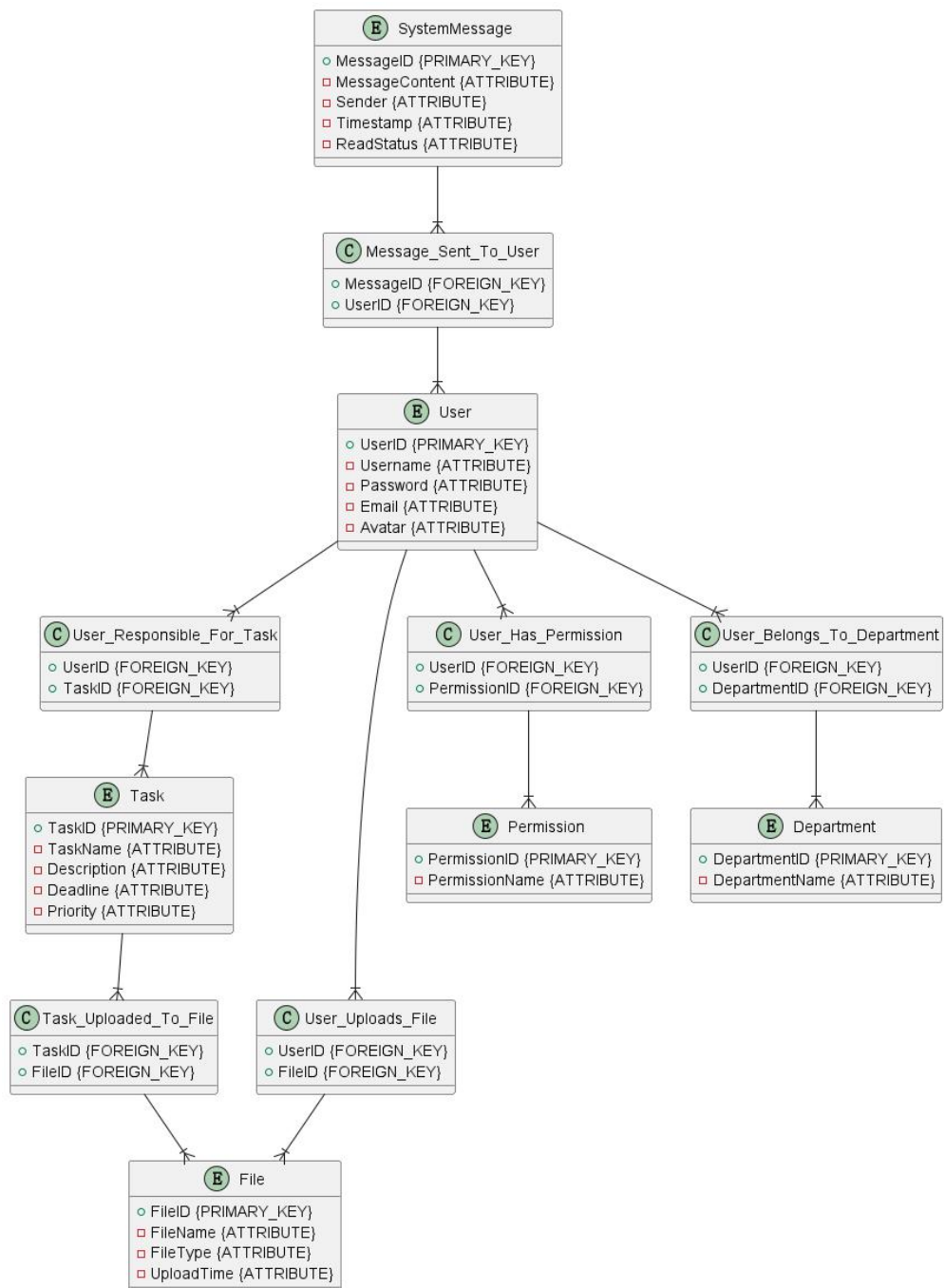


图 16 TSYD-OA 总体 ER 图



### 3.2 数据库逻辑结构设计

数据库逻辑设计将 E-R 图转换为关系模型，包括实体规范化，关系模型创建工作。

#### 3.2.1 实体规范化

关系规范化分析：

关系规范化是数据库设计中的一个关键步骤，通过将关系划分成更小、更简洁的关系，以减少数据冗余、提高数据一致性和维护性。在进行关系规范化分析时，需要确保数据库中的关系模型达到第三范式（3NF）或 BC 范式（Boyce-Codd 范式）的要求，以减少数据冗余和提高数据的一致性。

以 `User_Responsible_For_Task`（用户负责任务关系）为例，其他类同：

原始关系：

{ UserID, TaskID }

规范化：

第一范式（1NF）：已满足，每个属性原子性。

第二范式（2NF）：已满足，无部分依赖。

第三范式（3NF）：已满足，无传递依赖。

BC 范式（Boyce-Codd 范式）：主属性都是候选键，都完全依赖于主属性，没有部分依赖。

检查其他所有关系，显然，所有关系都达到 BC 范式的要求。

#### 3.2.2 关系模型创建

以下是对每个表的属性。

表 1 Task（任务）表

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>TaskID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	任务唯一标识
<i>TaskName</i>	<i>varchar(255)</i>	255	<i>No</i>	<i>No</i>	任务名称
<i>Description</i>	<i>text</i>		<i>Yes</i>	<i>No</i>	任务描述
<i>Deadline</i>	<i>datetime</i>		<i>No</i>	<i>No</i>	任务截止日期

表 2 File（文件）表

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>FileID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	文件唯一标识
<i>FileName</i>	<i>varchar(255)</i>	255	<i>No</i>	<i>No</i>	文件名称
<i>FileType</i>	<i>varchar(50)</i>	50	<i>No</i>	<i>No</i>	文件类型
<i>UploadTime</i>	<i>datetime</i>		<i>No</i>	<i>No</i>	文件上传时间

表 3 User（用户）表

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>UserID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	用户唯一标识
<i>Username</i>	<i>varchar(50)</i>	<i>50</i>	<i>No</i>	<i>No</i>	用户名
<i>Password</i>	<i>varchar(255)</i>	<i>255</i>	<i>No</i>	<i>No</i>	密码
<i>Email</i>	<i>varchar(255)</i>	<i>255</i>	<i>No</i>	<i>No</i>	邮箱
<i>Avatar</i>	<i>varchar(255)</i>	<i>255</i>	<i>Yes</i>	<i>No</i>	头像

表 4 Permission（权限）表

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>PermissionID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	权限唯一标识
<i>PermissionName</i>	<i>varchar(50)</i>	<i>50</i>	<i>No</i>	<i>No</i>	权限名称

表 5 Department（部门）表

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>DepartmentID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	部门唯一标识
<i>DepartmentName</i>	<i>varchar(50)</i>	<i>50</i>	<i>No</i>	<i>No</i>	部门名称

表 6 SystemMessage（系统消息）表

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>MessageID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	消息唯一标识
<i>MessageContent</i>	<i>text</i>		<i>No</i>	<i>No</i>	消息内容
<i>Sender</i>	<i>varchar(50)</i>	<i>50</i>	<i>No</i>	<i>No</i>	发送者
<i>Timestamp</i>	<i>datetime</i>		<i>No</i>	<i>No</i>	时间戳
<i>ReadStatus</i>	<i>boolean</i>		<i>No</i>	<i>No</i>	已读状态

表 7 User\_Responsible\_For\_Task（用户负责任务关系）

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>UserID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	用户唯一标识
<i>TaskID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	任务唯一标识

表 8 User\_Uploads\_File（用户上传文件关系）

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>UserID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	用户唯一标识
<i>FileID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	文件唯一标识

表 9 User\_Has\_Permission（用户拥有权限关系）

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>UserID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	用户唯一标识
<i>PermissionID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	权限唯一标识

表 10 User\_Belongs\_To\_Department（用户所属部门关系）

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>UserID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	用户唯一标识
<i>DepartmentID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	部门唯一标识

表 11 Task\_Uploaded\_To\_File（任务上传到文件关系）

字段名称	数据类型	长度	是否为空	是否为主键	说明
<i>TaskID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	任务唯一标识
<i>FileID</i>	<i>int</i>		<i>No</i>	<i>Yes</i>	文件唯一标识

表 12 Message\_Sent\_To\_User（消息发送给用户关系）

字段名称	数据类型	长度	是否为空	是否为主键	说明
MessageID	int		No	Yes	消息唯一标识
UserID	int		No	Yes	用户唯一标识

## 4 系统实现及部署设计

这部分从宏观角度简单介绍系统部署上的设计。

### 4.1 系统实现设计

#### 4.1.1 开发环境

编程语言： 选择适合小程序开发的编程语言，JavaScript、TypeScript，适合后端开发的编程语言 Java。

开发工具： 使用小程序开发工具，微信开发者工具、HBuilder X。

服务端部署工具： 宝塔面板（nginx）

数据库： MySQL、Redis

#### 4.1.2 小程序前端实现

在小程序端实现 TSYP-OA 系统的前端，确保用户可以通过小程序界面方便地进行任务管理、消息查看操作。主要功能包括：

登录页面： 提供用户登录功能，确保用户可以安全登录系统。

文件页面： 包含文件列表，允许上传下载，提供直观的用户界面。

任务管理： 允许用户查看任务列表、添加新任务、编辑任务详情。

消息通知： 展示系统消息，确保用户能及时了解重要信息。

具体小程序前端实现在系统原型报告中有详细描述；Web 端的后台管理子系统前端页面已有，无需设计。

#### 4.1.3 后端实现

在服务器端实现 TSYP-OA 系统的后端，处理前端请求，管理数据库，实现系统的核心逻辑。主要功能包括：

用户管理模块： 提供用户注册、登录、信息查看、编辑功能。

任务管理模块： 处理任务的增删改查操作，确保任务信息的完整性。

消息通知模块： 处理系统消息的发送与接收，保证消息的及时传递。

权限控制： 管理用户角色、权限，确保不同用户有不同的系统访问权限。

部门管理： 允许管理部门，并将部门成员添加到任务中。

### 4.2 系统部署设计

系统部署设计为以下步骤：

- 1) 创建服务器实例，并配置相关环境。
- 2) 数据库部署：部署选定的数据库系统，创建数据库和相应的表结构。确保数据库的安全性和可靠性。
- 3) 后端部署：将后端代码部署到服务器，配置相应的运行环境，确保后端服务可以响应前端请求。
- 4) 小程序部署：通过开发者工具将小程序部署到平台，确保用户可以访问并使用 TSYD-OA 小程序系统。

部署图是软件工程中用于表示系统架构及其组件在物理层面上的布局的图表。它展示了不同的硬件节点、软件组件以及它们之间的关系。TSYD-OA 系统的部署图如下。

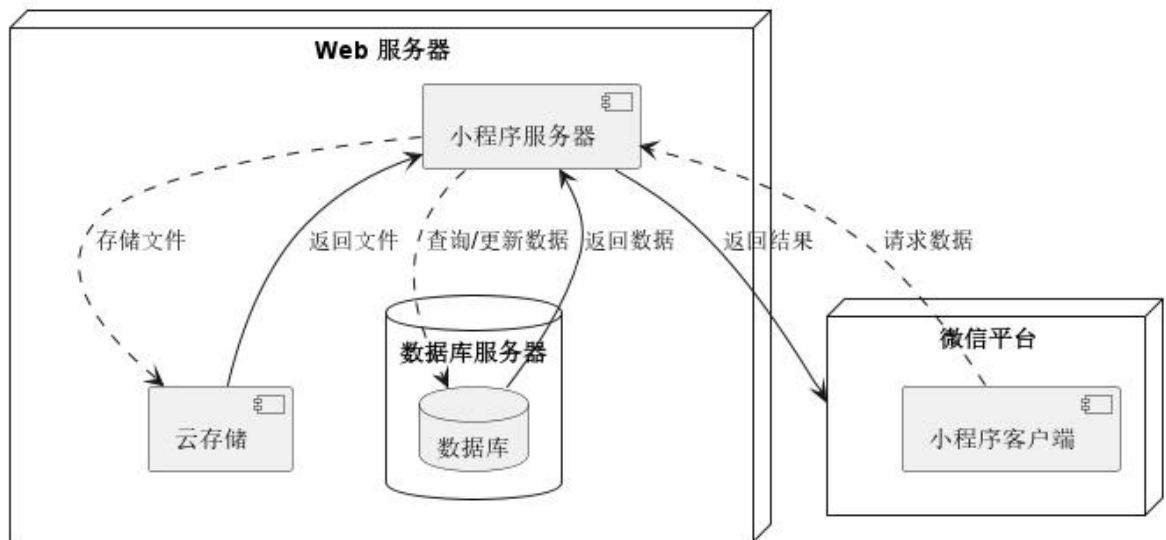


图 17 TSYD-OA 系统部署图

该部署图展示了系统的物理部署结构。Web 服务器包括小程序服务器、数据库服务器和云存储，部署在同一台服务器。小程序客户端通过请求数据与小程序服务器通信，小程序服务器通过查询/更新数据库和存储文件与数据库服务器和云存储通信。小程序服务器返回处理结果给小程序客户端，同时从云存储获取文件和从数据库获取数据。