

软件工程

实验二：软件设计与建模

成华睿 221850073

2024 年 11 月 1 日

目录

一 软件的主要功能	1
1 主要功能	1
2 功能模块	2
二 软件架构 UML 图	2
1 结构性图（一）：类图	2
2 结构性图（二）：组件图	3
3 行为性图（一）：活动图	5
4 行为性图（二）：用例图	6
5 交互性图（一）：时序图	6
6 交互性图（二）：序列图	7
三 UI 界面及功能描述	8
1 界面 1：主页与任务管理	8
2 界面 2：偏好设置	9
3 界面 3：日程一览	10
四 大模型的使用	11
1 UML 图代码生成	11
2 UI 界面设计工具	11
3 思考与总结	12

一 软件的主要功能

1 主要功能

根据实验一所得需求分析报告，现设计任务单管理系统，主要功能如下：

- DDL 提醒
- 任务归类和标签设计

- 任务时间线
- 日程与日常任务区分
- UI 风格选择
- 使用教程

2 功能模块

设计功能模块，包括提醒与通知模块、任务管理模块、任务属性模块、用户界面模块、帮助与支持模块、用户交互模块等。各模块将在下一章节中详细描述。

二 软件架构 UML 图

1 结构性图 (一): 类图

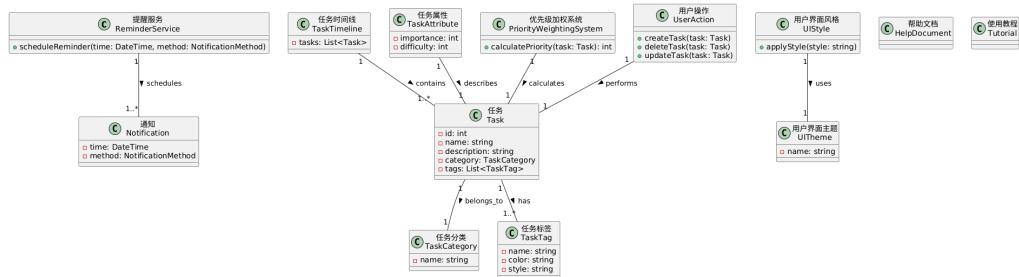


图 1.1: 类图

1. 提醒服务 (ReminderService)

- 负责调度提醒任务，包含一个方法 `scheduleReminder`，用于设置提醒的时间和通知方式。

2. 通知 (Notification)

- 包含通知的时间和通知方法，用于在特定时间提醒用户。

3. 任务时间线 (TaskTimeline)

- 管理任务列表，按时间顺序组织任务，帮助用户跟踪任务进度。

4. 任务属性 (TaskAttribute)

- 定义任务的属性，如重要性和难度，这些属性用于描述任务的紧急程度和完成难度。

5. 优先级加权系统 (PriorityWeightingSystem)

- 计算任务的优先级，帮助用户确定哪些任务应该优先完成。

6. 任务 (Task)

- 核心类，包含任务的基本信息，如 ID、名称、描述、类别和标签。任务可以属于某个类别，并有多个标签。

7. 任务分类 (TaskCategory)

- 用于对任务进行分类，每个类别有其名称。

8. 任务标签 (TaskTag)

- 用于给任务添加标签，每个标签有名称、颜色和样式。

9. 用户操作 (UserAction)

- 包含用户可以对任务执行的操作，如创建、删除和更新任务。

10. 用户界面风格 (UIStyle)

- 允许用户选择界面风格，通过 `applyStyle` 方法应用所选风格。

11. 帮助文档 (HelpDocument)

- 提供使用教程，帮助用户了解如何使用系统。

12. 用户界面主题 (UITheme)

- 定义用户界面的主题，包括名称等属性。

这些类和它们之间的关系共同构成了任务单管理系统的架构，支持系统的主要功能。

2 结构性图 (二): 组件图

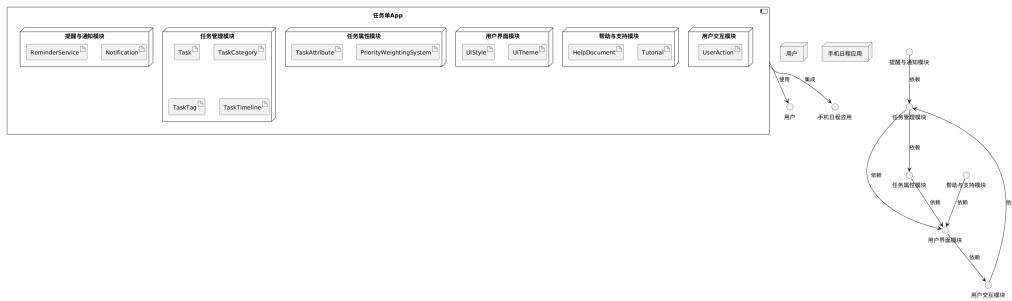


图 2.1: 组件图

根据提供的组件图，以下是组件图的简要说明：

组件图说明

组件图展示了任务单 App 的模块化结构，将系统分解为多个功能模块，每个模块负责特定的功能。这些模块通过明确的依赖关系相互协作，共同构成整个应用程序。

1. 提醒与通知模块 (ReminderService, Notification)

- 负责任务的提醒和通知功能，确保用户不会错过重要的截止日期。

2. 任务管理模块 (**Task**, **TaskCategory**, **TaskTag**, **TaskTimeline**)

- 提供任务的创建、分类、标签化和时间线管理功能，帮助用户组织和跟踪任务。

3. 任务属性模块 (**TaskAttribute**, **PriorityWeightingSystem**)

- 定义任务的属性，如重要性和难度，并计算任务的优先级，以便用户可以更有效地管理任务。

4. 用户界面模块 (**UIStyle**, **UITheme**)

- 允许用户自定义应用的界面风格和主题，提升用户体验。

5. 帮助与支持模块 (**HelpDocument**, **Tutorial**)

- 提供用户指南和教程，帮助用户了解如何使用应用的各项功能。

6. 用户交互模块 (**UserAction**)

- 包含用户与应用交互的操作，如创建、删除和更新任务。

依赖关系

- 用户和手机日程应用是系统的外部用户和系统，它们使用任务单 App 提供的功能。
- 提醒与通知模块依赖于任务管理模块来获取需要提醒的任务信息。
- 任务管理模块依赖于任务属性模块来获取和设置任务的属性和优先级。
- 用户界面模块依赖于帮助与支持模块来提供用户界面的样式和教程。
- 用户交互模块依赖于所有其他模块，因为它是用户与应用交互的接口，需要调用其他模块的功能来响应用户操作。

通过这种模块化设计，任务单 App 能够提供灵活、可扩展的功能，同时保持代码的清晰和易于维护。

3 行为性图 (一): 活动图

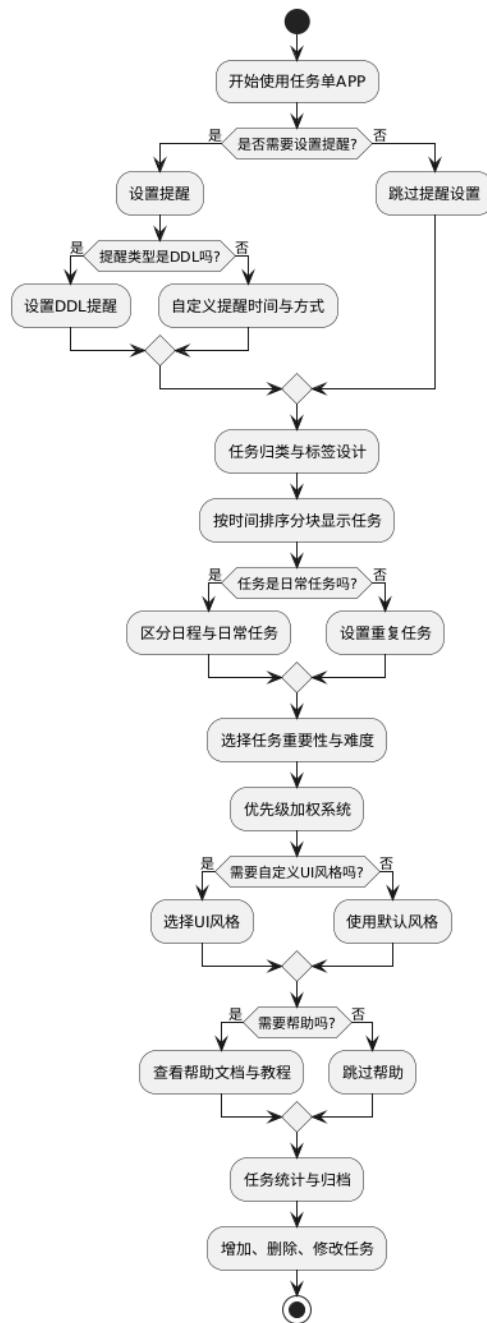


图 3.1: 活动图

活动图概述了用户在使用任务单 App 时的操作流程，从启动应用到设置提醒、任务管理、UI 风格选择，以及获取帮助等关键步骤。它展示了用户如何通过一系列决策点和操作来有效地管理任务，包括任务的分类、优先级设置、重复性标记和风格定制。整个流程旨在提供一个直观且用户友好的界面，帮助用户高效地组织和跟踪任务。

4 行为性图 (二): 用例图

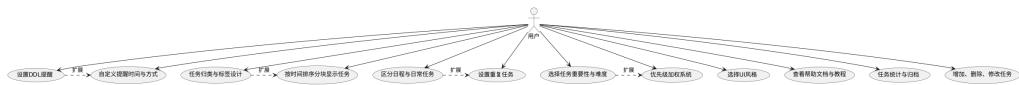


图 4.1: 用例图

用例图展示了任务单 App 的主要功能和用户如何与这些功能交互。中心用户角色与多个用例相连，表示用户可以执行的操作，涵盖了用户在任务管理过程中可能需要的所有操作，从提醒设置到任务的创建和修改，以及获取帮助和自定义界面风格。用例图清晰地描绘了用户与系统之间的交互关系，帮助理解系统的功能范围和用户的操作流程。

5 交互性图 (一): 时序图

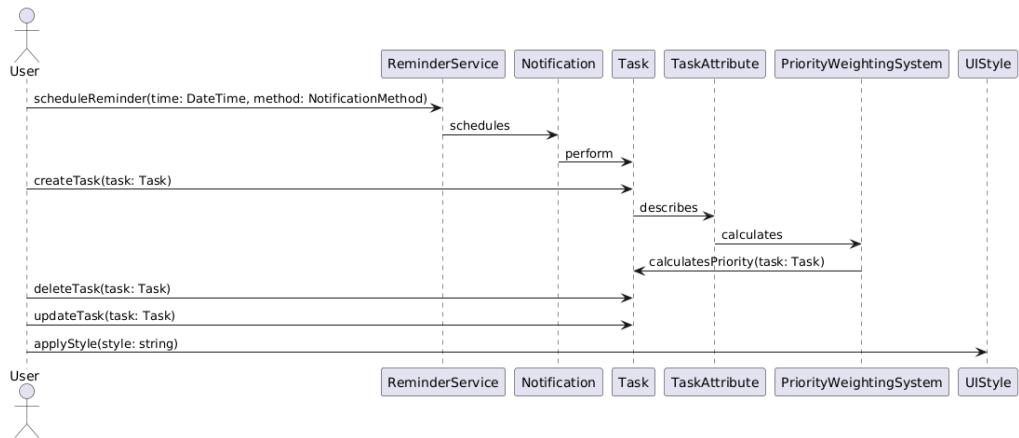


图 5.1: 时序图

时序图展示了用户与任务单 App 中各个组件之间的交互流程。用户可以执行以下操作：

1. **scheduleReminder**: 用户设置提醒，指定时间和通知方式。
 - 用户调用 ReminderService 的 scheduleReminder 方法，传递提醒时间和通知方法。
2. **createTask**: 用户创建任务。
 - 用户调用 Task 的 createTask 方法，创建一个新的任务实例。
3. **deleteTask**: 用户删除任务。
 - 用户调用 Task 的 deleteTask 方法，删除指定的任务。
4. **updateTask**: 用户更新任务。
 - 用户调用 Task 的 updateTask 方法，更新指定任务的信息。
5. **applyStyle**: 用户应用界面风格。

- 用户调用 `UIStyle` 的 `applyStyle` 方法，应用指定的界面风格。

在这些操作中，`Task` 组件负责描述任务的属性，并通过 `PriorityWeightingSystem` 计算任务的优先级。`Notification` 组件负责执行提醒服务，而 `UIStyle` 组件负责应用用户界面风格。整个流程展示了用户如何通过一系列操作来管理任务和定制界面。

6 交互性图 (二): 序列图

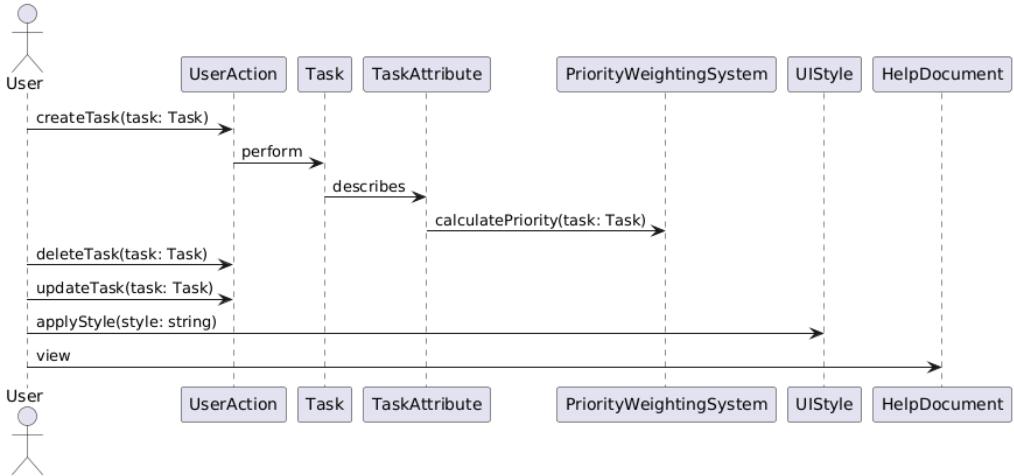


图 6.1: 序列图

序列图展示了用户与任务单 App 中各个组件之间的交互流程，具体包括以下操作：

1. `createTask`: 用户创建任务。

- 用户通过 `UserAction` 组件发起创建任务的请求。
- `UserAction` 调用 `Task` 组件的 `perform` 方法来执行任务创建。
- `Task` 组件描述任务属性，并通过 `TaskAttribute` 组件计算任务的优先级。

2. `deleteTask`: 用户删除任务。

- 用户通过 `UserAction` 组件发起删除任务的请求。

3. `updateTask`: 用户更新任务。

- 用户通过 `UserAction` 组件发起更新任务的请求。

4. `applyStyle`: 用户应用界面风格。

- 用户通过 `UserAction` 组件发起应用风格的请求。
- `UIStyle` 组件负责应用用户选择的界面风格。

5. `view`: 用户查看帮助文档。

- 用户通过 `UserAction` 组件发起查看帮助文档的请求。



图 1.1: 主页与任务管理

- HelpDocument 组件提供帮助信息供用户查看。

整个序列图展示了用户如何通过 UserAction 组件与系统中的其他组件交互，以完成任务的创建、删除、更新，界面风格的应用，以及帮助文档的查看。这些操作体现了用户在任务管理过程中的主要活动。

三 UI 界面及功能描述

根据 UML 图中的模块，设计了三个主要的 UI 界面，分别是主页与任务管理、偏好设置和日程一览。这些界面提供了用户与任务单 App 交互的主要功能，包括任务的创建、分类、标签化、提醒设置、优先级计算、界面风格选择和帮助文档查看等。下面将分别介绍这三个界面的功能和设计。

1 界面 1：主页与任务管理

相关模块 任务管理模块、用户界面模块、提醒与通知模块

功能描述

- 主界面展示了用户的任务列表，包括不同类别的任务，如工作、个人、家庭和学习。



图 2.1: 偏好设置

- 用户可以通过点击“+”按钮来创建新任务。
- 新任务的插入界面允许用户输入任务名称、设置重复性、输入日期、选择重要性和提醒方式。
- 用户可以对任务进行个性化设置，如设置提醒和调整任务的优先级。

2 界面 2：偏好设置

相关模块 用户界面模块、任务管理模块、帮助与支持模块

功能描述

- 用户可以在此界面设置个人偏好，包括任务分类。
- 提供了三个分类选项：个人事务、工作任务和学习计划，用户可以勾选适用的分类。



图 3.1: 日程一览

- 设置完成后，用户可以点击“应用”按钮保存偏好设置。
- 界面底部提供了帮助步骤，指导用户如何创建任务和分类任务。

3 界面 3：日程一览

相关模块 任务管理模块、用户界面模块、提醒与通知模块

功能描述

- 提供了一个日历视图，用户可以按周、月、年查看任务。
- 用户可以快速浏览特定日期的任务，如今天的任务列表。
- 任务按日期分组，用户可以点击日期查看当天的任务详情。
- 提供了导航按钮，允许用户在不同时间段之间切换。
- 用户可以查看和管理工作任务，如检查电子邮件、与团队开会、编写项目报告和更新日历。

UML类图 - 任务单App

```

plantuml

' 帮助与支持模块 (Help and Support Module)
class HelpDocument as "帮助文档\nHelpDocument" {
}

class Tutorial as "使用教程\nTutorial" {
}

' 用户交互模块 (User Interaction Module)
class UserAction as "用户操作\nUserAction" {
    +createTask(task: Task as "任务\nTask")
    +deleteTask(task: Task as "任务\nTask")
    +updateTask(task: Task as "任务\nTask")
}

' 关系
ReminderService "1" -- "1..*" Notification : schedules >
Task "1" -- "1..*" TaskTag : has >
Task "1" -- "1" TaskCategory : belongs_to >
TaskTimeline "1" -- "1..*" Task : contains >
TaskAttribute "1" -- "1" Task : describes >
PriorityWeightingSystem "1" -- "1" Task : calculates >
UITheme "1" -- "1" UITheme : uses >
UserAction "1" -- "1" Task : performs >

@enduml

```

图 1.1: 生成的类图 puml 代码

四 大模型的使用

1 UML 图代码生成

我使用了 Kimi 辅助 UML 图绘制。

使用场景

在设计任务单 App 时，使用 UML 图代码生成工具可以帮助快速生成代码，减少手动编写代码的工作量。比如，在生成类图时，我给出我预先确定好的类，希望能得到相应的 puml 代码。

生成结果

优缺点分析

可以看出，效果差强人意，生成的代码不符合预期，需要手动修改。

修改优化后的结果

手动修改得到的代码，最终呈现效果如 1 所示。

2 UI 界面设计工具

我使用了 Creati 辅助 UI 界面设计。

使用场景

在设计 UI 界面时，Creati 平台提供 AI 建议，可以帮助用户快速设计出符合用户体验的界面。

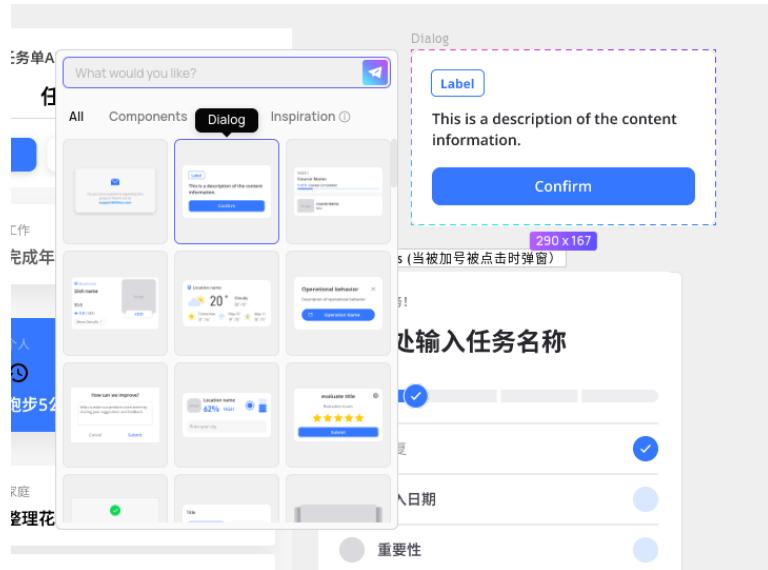


图 2.1: Creati 提供的风格建议

生成结果

如图所示，当我设计 Frame 时，Creati 提供了多种风格的建议，直接生成模板样式。

优缺点分析

只能提供模板，但内容需要自己填充，不够智能。元素与元素之间的关系需要自己调整。

修改优化后的结果

经过手动调整，最终得到了符合预期的界面设计。

3 思考与总结

大模型工具可以帮助软件设计者更好地理解和构建复杂的系统，提供了一种直观的方式来展示系统的结构和行为。通过使用大模型工具，可以更好地组织和管理系统的各个组件，减少设计和开发过程中的错误和冗余。然而，大模型工具也存在一些局限性，如生成的代码质量不高、界面设计不够智能等问题。因此，在使用大模型工具时，需要结合实际需求和自身经验，灵活运用这些工具，以提高软件设计和开发的效率和质量。