

时光清浅日程软件 研发文档

项目 Git 访问地址:

<https://github.com/33281/20215898---.git>

时光清浅日程软件研发小组

组长: 习婉婷

组员: 孙欣欣

吕希彤

吉廉嘉

目 录

1 需求分析.....	1
1.1 引 言.....	1
1.1.1 编写目的.....	1
1.1.2 背景.....	2
1.1.3 定义.....	2
1.1.4 参考资料.....	2
1.1.5 人员安排.....	2
1.2 任务概述.....	3
1.2.1 目标.....	3
1.2.2 用户的特点.....	4
1.2.3 假定和约束.....	5
1.3 用例分析(或数据流程分析).....	5
1.3.1 系统 Actor 分析.....	5
1.3.2 系统用例描述.....	6
1.4 对象-关系建模.....	10
1.5 系统流程分析.....	11
1.6 系统开发及运行环境规定.....	15
1.7 需求小结.....	15
2 系统设计.....	16
2.1 引 言.....	16
2.1.1 编写目的.....	16
2.1.2 背景.....	17
2.1.3 定义.....	17
2.1.4 参考资料.....	18
2.1.5 人员安排.....	18
2.2 系统功能设计.....	19
2.2.1 功能模块设计.....	19
2.2.2 管理日程模块设计.....	19
2.3 数据库设计.....	20
2.4 界面设计.....	21
2.7 设计过程中的需求变动情况说明.....	25
2.8 设计小结.....	26
3 系统实现.....	27
3.1 人员安排.....	27
3.2 基础日程表模块实现.....	27
3.2.1 基础日程表模块实现简介.....	27
3.2.2 基础日程表模块相关类实现.....	27
3.3.3 基础日程表模块模块文件及跳转关系.....	28
3.3.4 基础日程表模块程序流程.....	29
3.3.5 基础日程表模块实现界面.....	30

3.3.6 登录模块实现界面.....	31
3.4 实现过程中的需求和设计变动情况说明	32
3.5 实现小结.....	32
4 系统测试.....	33
4.1 人员安排.....	33
4.2 测试用例设计.....	33
4.2.1 用户模块测试.....	33
4.2.2 用户模块测试报告	35
4.2.3 用户模块调试过程.....	37
4.3 模块测试.....	37
5 心得体会.....	38

1 需求分析

1.1 引言

1.1.1 编写目的

本需求说明书的编写旨在明确日程软件的开发目标、功能需求、用户交互方式及系统性能等关键要素，为软件开发团队提供清晰、一致的指导和参考。通过本说明书，我们希望能够确保软件开发过程中各方对项目的理解保持一致，提高软件开发的效率和质量。

下面，我们将首先分析潜在目标用户并尝试分析他们的需求。

1. 个人用户：

（1）日常生活管理：需要一款简洁易用的日历软件来记录和管理日常生活事件，如约会、生日、纪念日等。

（2）日程提醒：软件应提供定时提醒功能，确保用户不会错过重要事件。

（3）多设备同步：支持手机、平板、电脑等设备之间的数据同步，方便用户随时随地查看和更新日程。

2. 商务人士：

（1）会议安排：软件应支持快速创建、编辑和分享会议日程，方便商务人士安排和管理会议。

（2）团队协作：支持邀请团队成员加入日程，实时查看团队成员的日程安排，提高团队协作效率。

3. 教育机构：

（1）课程管理：支持导入、编辑和发布课程表，方便教育机构安排和管理课程。

（2）考试安排：支持创建和发布考试日程，提醒学生按时参加考试。

通过本需求说明书，我们希望能够确保日历软件能够满足不同目标用户的实际需求，提高用户的使用体验和满意度。同时，也为软件开发团队提供了明确、一致的指导和参考，有助于提高软件开发的效率和质量。

1.1.2 背景

随着生活节奏的加快和人们对时间管理的需求增加，日程管理成为了越来越多人生活中的重要组成部分。传统的纸质日历已经不能满足人们对灵活性和便捷性的需求，因此习婉婷提出开发决定开发一款全新的日程软件，以满足现代用户对日程管理、提高效率的需求。

时光清浅日程软件(Schedule Management Software)是由习婉婷，孙欣欣，吕希彤，吉廉嘉自主研发的用于帮助用户轻松管理日程、提高用户效率的软件产品。我们的目标用户包括白领人士、自由职业者、学生、家庭主妇等一切希望有效地管理工作和个人生活的人群。技术上计划暂时部署在东北大学校内。实现后我们将结合实际体验，把开发成果部署在全网。

1.1.3 定义

日程：指个人或组织在特定时间段内所安排的一系列活动或任务的清单。可涵盖日常生活、工作、学习等各个方面的计划。

提醒：一种通知或指示，用于引起某人对特定事件、活动或任务的注意，通常是为了确保该事件、活动或任务不会被忘记或错过。

1.1.4 参考资料

《软件工程及应用》，张斌、郭军主编，东北大学出版社，2007 年 5 月第 1 版

《软件工程：实践者的研究方法（本科教学版）》，（美）Roger S. Pressman、Bruce R. Maxin 著，郑人杰、马素霞等译，机械工业出版社，2017 年 1 月第 8 版

GB8567-88 计算机软件文档编制规范，国家标准化管理委员会，1988 年

1.1.5 人员安排

时光清浅日程软件需求分析小组

组长：习婉婷

组员：孙欣欣

吕希彤

吉廉嘉

1.2 任务概述

1.2.1 目标

本软件产品是一项独立的软件，而且全部内容自含。

时光清浅日程软件(Schedule Management Software)是一款自主研发的、旨在帮助用户更有效地管理时间，优化日常生活和工作的安排。该软件旨在为用户提供一种全新的时间管理方式，通过智能化的提醒、分类、共享等功能，帮助用户更好地掌控自己的时间，提高工作和生活效率。

日程软件的应用目标主要包括以下几个方面：

（1）提供个性化的日程安排：根据用户的需求和习惯，提供个性化的日程管理方案，包括添加、编辑、删除日程事件，设置提醒等功能。

（2）实现跨平台同步：支持多种设备（如手机、平板、电脑等）的同步使用，确保用户在不同场景下都能方便地查看和修改日程安排。

（3）提供智能提醒服务：根据用户的日程安排，提供及时的提醒服务，确保用户不会错过重要的会议、约会等事件。

（4）支持团队协作和共享：提供团队协作和共享功能，方便团队成员之间共享日程安排，提高团队协同效率。

日程软件的作用范围主要包括以下几个方面：

（1）个人时间管理：帮助个人用户有效地管理时间，优化日程安排，提高工作和生活效率。

（2）团队协作：支持团队成员之间的日程共享和协作，提高团队协同效率。

（3）商务活动安排：为商务人士提供便捷的日程安排和管理功能，方便他们安排会议、出差等商务活动。

（4）教育培训：在教育领域，日程软件可以帮助学生和教师更好地安排学习和教学计划，提高学习效果。

1.2.2 用户的特点

此部分旨在介绍本软件的最终用户特点、对操作人员与维护人员要求及预期使用频度。

一、最终用户特点

1. 个人用户：

年龄分布广泛，从年轻人到中老年人。

大多具备基本的计算机操作技能，如浏览网页、使用办公软件等。

对于时间管理有较高的需求，希望通过软件提高生活和工作效率。

可能需要跨平台同步功能，以便在不同设备上查看和编辑日程。

2. 商务人士：

主要为职场中的中层及以上管理者，经常需要安排会议、出差等商务活动。

对软件的易用性和功能丰富性有较高要求，追求高效、便捷的日程管理方式。

需要团队协作和共享功能，以便与团队成员共同管理日程。

3. 教育机构用户：

主要为学校、培训机构等教育机构的教师和管理人员。

需要软件支持课程表管理、考试安排等功能。

可能会使用软件的批量导入、导出功能来管理大量数据。

二、操作人员要求

操作人员应具备基本的计算机操作技能，能够熟练使用鼠标、键盘等输入设备。

对于软件的基本功能应有一定的了解，如添加、编辑、删除日程等。

对于不同用户群体，可能需要具备一些特定的技能或知识，如教育机构用户可能需要了解课程表管理相关知识。

三、维护人员要求

维护人员应具备较高的计算机技术水平，熟悉软件开发和测试流程。

应熟悉软件所使用的开发语言和工具，以便进行代码修改和调试。

需要具备一定的网络和安全知识，以确保软件的稳定运行和数据安全。

应具备良好的沟通能力和团队合作精神，以便与开发人员和用户进行有效的沟通。

四、预期使用频度

个人用户：根据个人的工作和生活需求，使用频度可能有所不同。但一般来说，他们可能会每天使用软件来查看和编辑日程安排。

商务人士：由于经常需要安排会议、出差等商务活动，他们可能会更频繁地使用软件来管理日程。预计每天至少使用数次。

教育机构用户：这类用户可能会根据工作需求定期使用软件来管理课程表、考试安排等事务。使用频度可能因具体需求而异，但一般来说，每周或每月至少使用数次。

1.2.3 假定和约束

此部分旨在列出进行本软件开发工作的假定和约束，例如经费限制、开发期限等。

人力：5 人

开发期限：4 月

开发语言：C++

经费限制：20 万

其他限制：应符合国家相关法律和学校相关政策

1.3 用例分析(或数据流程分析)

1.3.1 系统 Actor 分析

该软件的 Actor 为用户及后台管理员，下面进行详细分析。

(1) 用户

用户是软件的使用者，经过登陆页面进行登录后，可查看当前日程，管理日程，导出日程。

管理日程可以包括：新日程创建，修改原有日程，日程删除，对日程优先级进行排序，设置日程提醒。

(2) 后台管理员

后台管理员是系统的维护人员，可以查看用户创建日程内容，进行系统

维护，包括数据备份及系统恢复。

1.3.2 系统用例描述

1. 用户的用例图如图 1.3.1 所示。

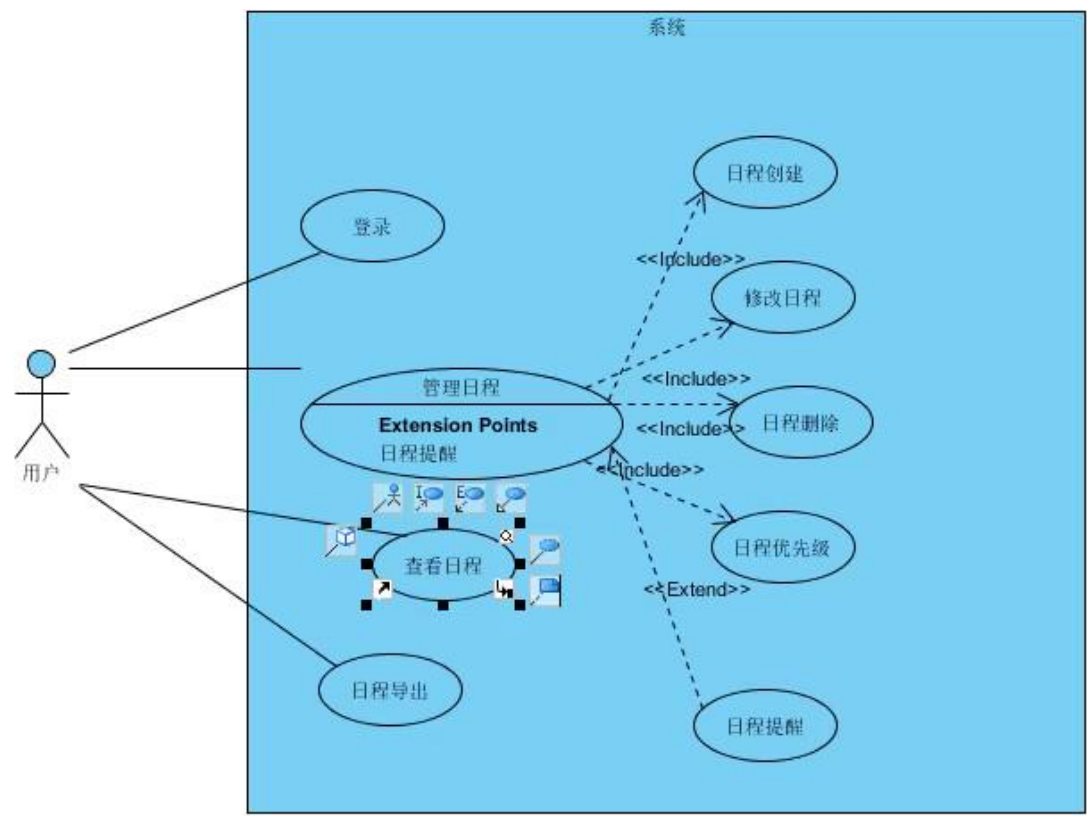


图 1.3.1 用户用例图

用户用例图中的用例全部详细描述如下：

(1) 查看日程

表 1.3.1 查看日程用例说明

用例名称	查看日程
用例描述	用户想要查看已有日程
主执行者	用户
触发条件	用户登录本 APP
后置条件	用户的 APP 页面显示已有日程表
基本事件流	1. 用户选择“查看日程”选项。 2. 日程软件显示用户的日程表，包括所有已保存的事件。 3. 用户可以选择按日期、类型或其他条件筛

	选事件（可选）
异常事件流	无可查看日程

（2）管理日程

管理日程可以包括：新日程创建，修改原有日程，日程删除，对日程优先级进行排序，设置日程提醒。

表 1.3.2 管理日程用例说明

用例名称	管理日程
用例描述	进行日程的管理
主执行者	用户
触发条件	用户登录本 APP
后置条件	日程信息修改完成，系统录入。
基本事件流	1.用户进入管理日程 2.选择要进行管理方向：日程创建、日程修改、日程删除、日程优先级排序。 3.输入具体内容，并进行相应操作 4.选择“提交” 5.选择日程提醒（可选） 6.选择“提交”
异常事件流	录入数据错误

（3）日程创建

表 1.3.3 日程创建用例说明

用例名称	日程创建
用例描述	用户要将新日程添加到日程表
主执行者	用户
触发条件	用户登录并选择进行日程管理
后置条件	新日程创建完毕
基本事件流	1.用户选择“添加新事件”选项。 2.用户输入事件的标题、日期、时间、地点和描述（可选）。 3.用户可以选择设置提醒（可选）。

	4.用户点击“保存”按钮。
	5.日程软件保存事件，并在用户的日程表中显示新事件。
异常事件流	输入数据异常

(4) 日程修改

表 1.3.4 日程修改用例说明

用例名称	日程修改
用例描述	用户要将已有日程进行修改
主执行者	用户
触发条件	用户登录并选择进行日程管理
后置条件	日程修改完毕
基本事件流	1.用户选择“日程修改” 2.用户选择“编辑事件”选项。 3.日程软件系统显示事件的当前详细信息。 4.用户修改事件的标题、日期、时间、地点和描述（可选）。 5.用户可以选择修改提醒设置（可选）。 用户点击“保存”按钮。 6.日程软件系统保存修改后的事件，并在用户的日程表中更新显示。
异常事件流	系统中没有用户所要修改的日程信息（未创建或已删除）

(5) 日程删除

表 1.3.5 日程删除用例说明

用例名称	日程删除
用例描述	用户要将已有日程从日程表中删除
主执行者	用户
触发条件	用户登录并选择进行日程管理
后置条件	日程删除完毕，日程表中该日程消失

基本事件流	<ol style="list-style-type: none"> 1.用户选择要删除的日程 2.日程软件询问用户是否确认删除该事件。 3.用户确认删除。 4.日程软件系统从用户的日程表中删除选定的事件。
异常事件流	系统中没有用户所要删除的日程信息（未创建或已删除）

（6）日程优先级

表 1.3.6 日程优先级用例说明

用例名称	日程优先级
用例描述	用户要对已有日程进行优先级排序
主执行者	用户
触发条件	用户登录并选择进行日程管理
后置条件	日程优先级排序完成，后续日程将以此优先级顺序表示。
基本事件流	<ol style="list-style-type: none"> 1.用户选择“日程优先级排序”选项。 2.用户输入所想进行优先级排序的日程名， 3.输入优先级 1-n。 3.用户点击“保存”按钮。
异常事件流	系统中没有相应的日程信息（未创建或已删除）

（7）日程提醒

表 1.3.7 日程提醒用例说明

用例名称	日程提醒
用例描述	用户想要 APP 对其进行日程提醒
主执行者	用户
触发条件	用户登陆并选择日程管理
后置条件	相应日程在指定时间 APP 对用户进行提醒
基本事件流	<ol style="list-style-type: none"> 1. 用户选择“添加日程提醒”选项。 2.用户输入要添加提醒的日程名称

异常事件流	3.用户选择想要被提醒的日期及时间。
	4.用户选择被提醒的方式（推送通知、邮件、短信）。
	5.点击“保存”按钮。
	6.系统内对相应日程添加上提醒
	7.在设定的提醒时间到达时，系统向用户发送提醒通知（如：推送通知、邮件、短信等）
	系统中没有用户所要导出的日程信息（未创建或已删除）

（8）日程导出

表 1.3.8 日程导出用例说明

用例名称	日程导出
用例描述	进行已有日程的导出
主执行者	用户
触发条件	用户此前已添加某些日程，并选择导出
后置条件	导出的日程以文件形式导出至指定路径
基本事件流	1. 用户选择“日程导出”选项。 2.用户输入导出的指定路径 3.用户选择所要导出内容 3.用户点击“确认”按钮。 4.指定路径出现日程文件
异常事件流	系统中没有用户所要导出的日程信息（未创建或已删除）

1.4 对象-关系建模

在进行对象-关系建模前，我们首先使用传统需求分析建模方法中的数据模型（E-R 图）对本系统进行初步分析。时光清浅日程软件的 E-R 图如图 1.4.1 所示。

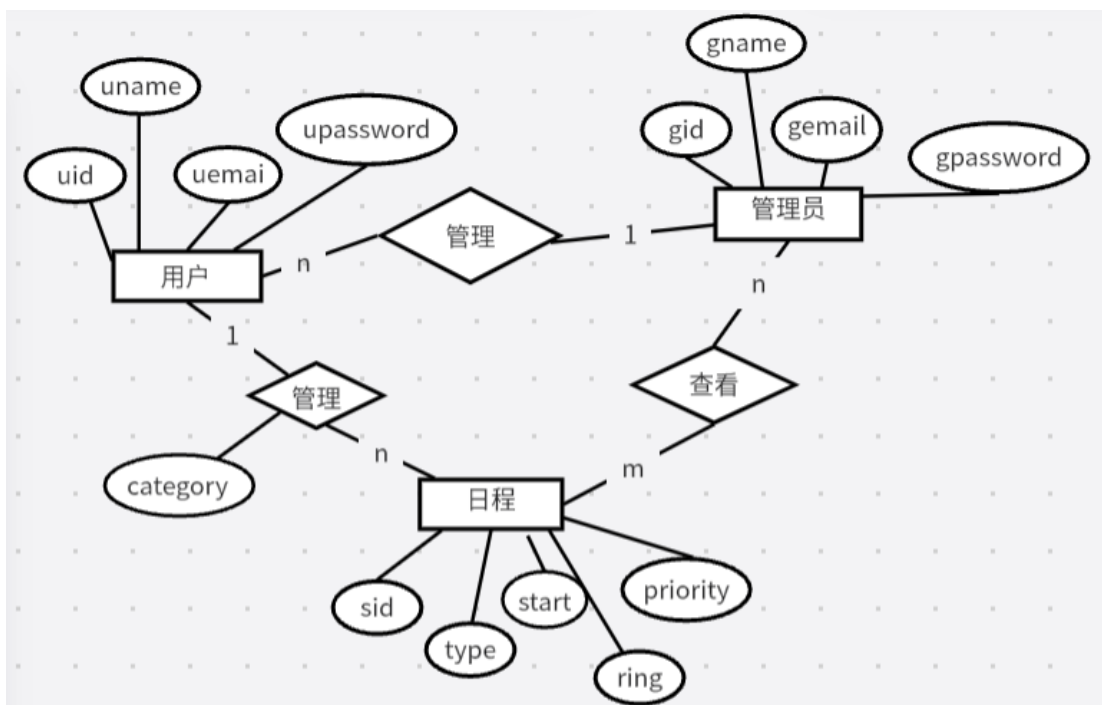


图 1.4.1 时光清浅日程软件的 E-R 图

时光清浅日程软件系统对象-关系建模（类图）如图 1.4.2 所示。

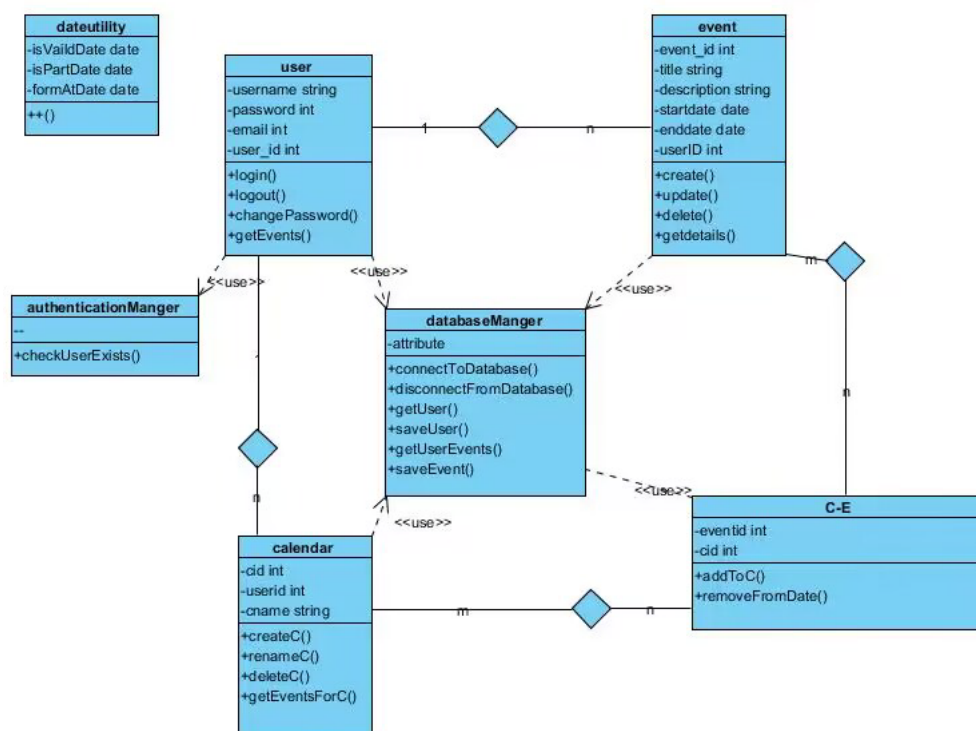


图 1.4.2 时光清浅日程软件系统对象-关系图

1.5 系统流程分析

(1) 登录、查看、退出流程

用户的登录、注册模块主要包括了对用户的账号、邮箱和密码的操作，用户的隐私的保护。用户可以登录自己的帐号以使得用户可以操作系统的更多的功能。

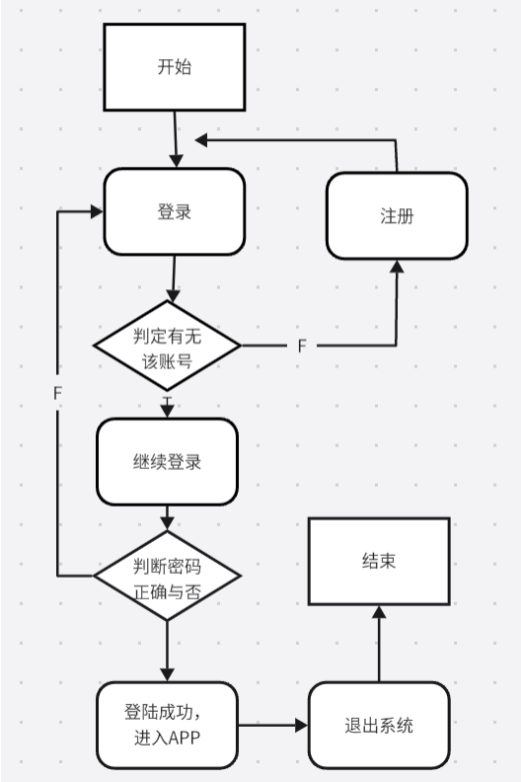


图 1.5.1 登录、查看、退出流程的时序图

(2) 用户的查看日程流程

用户的查看日程流程包括查看总体日程表，通过选择日期、类型或其它条件对日程表中内容进行筛选，以及对筛选后日程的查看。

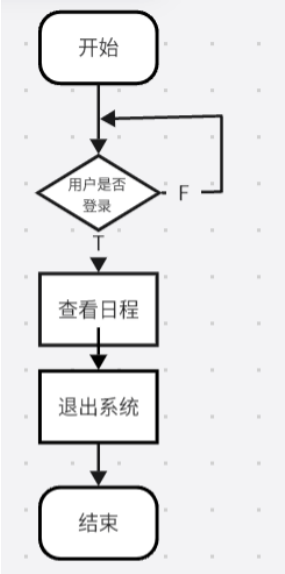


图 1.5.2 用户查看日程流程的时序图

(3) 用户的管理日程流程

用户选择进行日程管理，可以创建日程，修改日程，删除日程，修改日程优先级，设置日程提醒。

创建日程包括填写日程名字、选择日程时间、填写相关描述。

修改日程包括输入所要修改日程的名字，选择想要修改的日程的相应信息（名字、时间、相关描述与内容），进行修改。

删除日程包括输入所要删除的日程名字，选择“确认”，日程删除完毕。

日程优先级排序包括选择要进行排序的日程，添加优先级序号 1-n，优先级排序成功。

日程提醒包括用户输入要添加提醒的日程名称，选择想要被提醒的日期及时间，选择被提醒的方式。

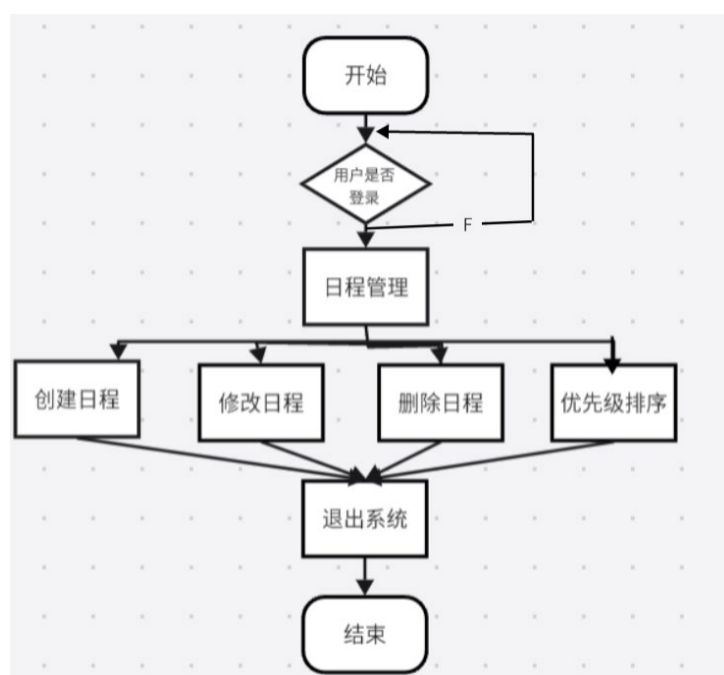


图 1.5.3 用户管理日程流程的时序图

(4) 用户的导出日程流程

用户的导出流程图可以包括输入导出文件的指定路径，输入所想导出的日程名字，指定路径中出现指定日程文件。

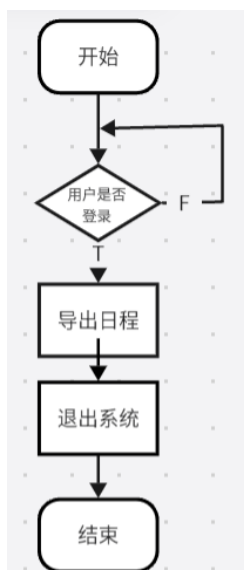


图 1.5.4 用户导出日程流程的时序图

(5) 管理员登录以维护流程图

管理员登录后，可以查看用户创建的日程信息，并进行软件的系统维护，包括数据备份和系统恢复。

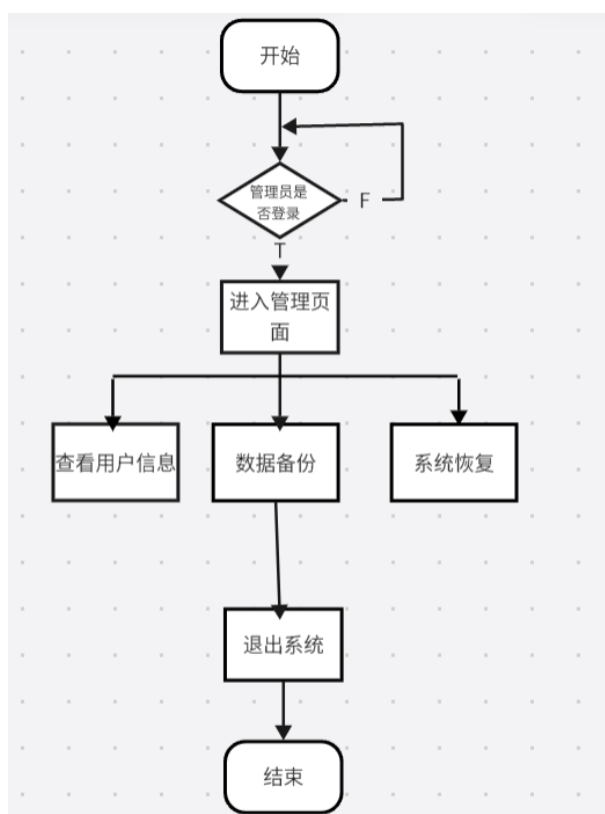


图 1.5.5 管理员登录以维护流程的时序图

1.6 系统开发及运行环境规定

系统运行所需的最低硬件环境如表 1.6.1 所示，软件开发平台如表 1.6.2 所示。

表 1.6.1 系统运行硬件环境表

环境	配置
CPU	Intel 二核 2.0GHz 及以上
内存	2G 或以上
硬盘	40G 或以上
操作系统	Windows 11

表 1.6.2 软件开发平台表

平台软件	说明
Windows 11	系统开发平台
MySQL	数据库系统平台
SpringBoot	后端框架
Qt	前端框架

1.7 需求小结

时光清浅日程软件 (Schedule Management Software) 用于帮助用户管理日程、提高用户效率及专注度。软件提供了用户注册与登录、添加、查看、编辑、删除和管理日程安排等基本功能。系统还支持通过云盘进行多设备间的日程同步、数据备份与恢复等保障数据一致性与安全性的功能。除此之外，我们在市场已有日程管理软件基础上，创新性地为软件添加专注模式等其他功能，帮助用户更高效地完成日程。

2 系统设计

2.1 引言

2.1.1 编写目的

编写这份系统设计说明书的目的在于为开发团队、利益相关者、测试人员、用户以及可能的未来维护者提供一个全面、详细且一致的参考文档，以确保软件的开发、测试、部署和维护过程能够顺利进行。以下是编写该系统设计说明书的主要目的：

（1）明确需求和目标：

通过系统设计说明书，可以清晰地定义软件的核心功能、特性和业务需求，确保所有相关人员对项目的目标和期望有共同的理解。

（2）指导开发工作：

详细说明软件架构、模块划分、数据模型、接口定义等，为开发人员提供具体的开发指导，减少误解和返工。标准化开发流程，提高开发效率和代码质量。

（3）促进团队协作：

当多个开发人员或团队共同参与项目时，系统设计说明书可以作为沟通的桥梁，确保每个人对项目的理解和执行方式保持一，便于团队成员之间的任务分配和进度管理。

（4）支持测试工作：

提供测试所需的信息，如测试场景、测试数据、预期结果等，帮助测试人员制定测试计划和执行测试用例。确保软件的质量和稳定性，降低上线风险。

（5）用户理解和使用：

对于最终用户来说，虽然系统设计说明书可能不是直接使用的文档，但它可以作为用户手册或帮助文档的基础，确保用户能够正确理解和使用软件。在软件升级或功能变更时，用户可以通过阅读更新后的系统设计说明书了解新的功能和改进。

(6) 维护和支持:

在软件的生命周期内,可能会遇到各种问题和需求变更。系统设计说明书为维护人员提供了必要的信息,以便他们能够快速定位和解决问题。对于未来可能的扩展或新功能开发,系统设计说明书可以作为参考和依据,确保系统的稳定性和可扩展性。

(7) 项目管理和决策支持:

为项目管理人员提供项目进展、风险评估和决策支持所需的信息。

在项目评审、验收和审计过程中,系统设计说明书可以作为重要的参考文件。

(8) 知识传承和文档化:

项目的设计思路、决策过程和关键技术点,为团队成员提供学习和参考的资料。在人员流动或项目交接时,确保项目的连续性和稳定性。

2.1.2 背景

随着生活节奏的加快和人们对时间管理的需求增加,日程管理成为了越来越多人生活中的重要组成部分。传统的纸质日历已经不能满足人们对灵活性和便捷性的需求,因此习婉婷提出开发决定开发一款全新的日程软件,以满足现代用户对日程管理、提高效率的需求。

时光清浅日程软件(Schedule Management Software)是由习婉婷,,吕希彤,吉廉嘉自主研发的用于帮助用户轻松管理日程、提高用户效率的软件产品。我们的目标用户包括白领人士、自由职业者、学生、家庭主妇等一切希望有效地管理工作和个人生活的人群。技术上计划暂时部署在东北大学校内。实现后我们将结合实际体验,把开发成果部署在全网。

2.1.3 定义

日程: 指个人或组织在特定时间段内所安排的一系列活动或任务的清单。可涵盖日常生活、工作、学习等各个方面的计划。

提醒: 一种通知或指示,用于引起某人对特定事件、活动或任务的注意,通常是为了确保该事件、活动或任务不会被忘记或错过。

2.1.4 参考资料

《软件工程及应用》，张斌、郭军主编，东北大学出版社，2007 年 5 月第 1 版

《软件工程：实践者的研究方法（本科教学版）》，（美）Roger S. Pressman、Bruce R. Maxin 著，郑人杰、马素霞等译，机械工业出版社，2017 年 1 月第 8 版

GB8567-88 计算机软件文档编制规范，国家标准化管理委员会，1988 年

2.1.5 人员安排

时光清浅日程软件系统设计小组

组长：习婉婷

组员：孙欣欣

吕希彤

吉廉嘉

本章详细人员安排如表 2.1.1 所示。

表 2.1.1 软件开发过程中具体人员安排

Role	Personnel	Responsibility
开发工程师	习婉婷，吉廉嘉，吕希彤，孙欣欣	<ul style="list-style-type: none">● 开发完成软件代码● 完成软件的单元开发，集成开发及系统开发● 完成软件的发布文档
项目软件负责人	习婉婷	<ul style="list-style-type: none">● 完成各类软件开发过程中的文档● 参加软件设计缺陷评审● 确保软件进度的按时完成● 批准软件的结构设计● 批准软件测试报告● 批准软件发布
项目经理 (PM) *	杨金柱老师	<ul style="list-style-type: none">● 批准软件开发设计计划● 确保软件进度的按时完成● 召开软件设计缺陷评审

质量负责人 (QA)	吕希彤	<ul style="list-style-type: none"> ● 参加软件设计缺陷评审 ● 参与软件设计开发计划，测试计划和测试报告的评审
测试工程师	习婉婷，吉廉嘉， 吕希彤，孙欣欣	<ul style="list-style-type: none"> ● 按照验证计划要求，按时完成测试 ● 完成相应的测试报告

2.2 系统功能设计

2.2.1 功能模块设计

根据分析工作进行系统的功能分配，画出系统的功能模块图（如果使用面向对象系统设计方法，可以画包图）。功能层次图示例：

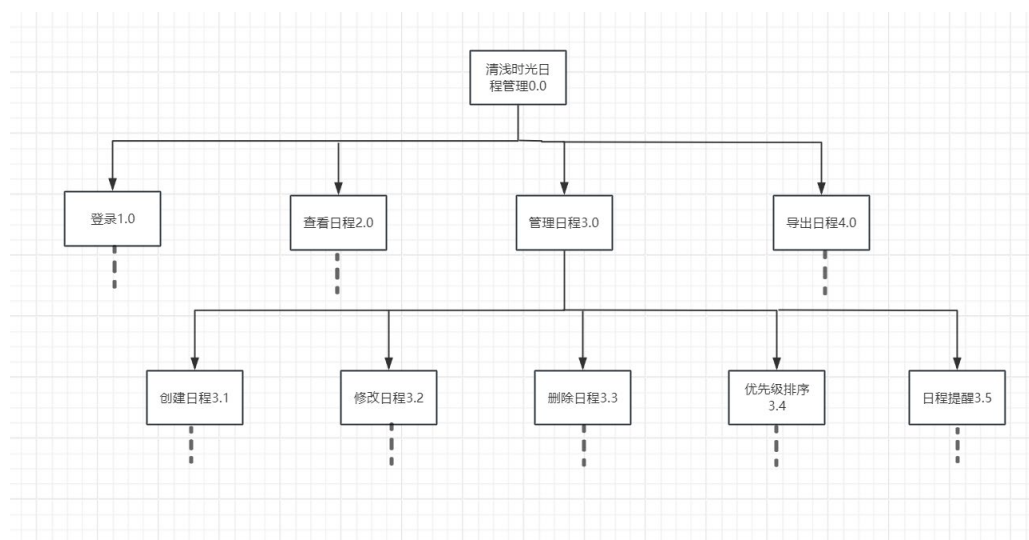


图 2.2.1 功能模块图

2.2.2 管理日程模块设计

管理日程模块：包含创建日程、删除日程、修改日程、日程优先级排序日程提醒的功能

创建日程子模块：包含填写日程名字、选择日程时间、填写相关描述功能。

删除日程子模块：包含输入所要删除的日程名字，选择是否“确认”删除功能。

修改日程子模块：包含输入所要修改日程的名字，选择想要修改的日程的相应信息（名字、时间、相关描述与内容）的功能。

日程优先级排序子模块：包含选择要进行排序的日程，添加优先级序号 1-n 功能。

日程提醒子模块：包含用户输入要添加提醒的日程名称，选择想要被提醒的日期及时间，选择被提醒的方式的功能。

2.3 数据库设计

根据系统需求分析结果尽量画出 E-R 图（E-R 图不是必须的），并根据这些工作设计出实现系统功能所需要的数据库表的结构。并以表格的形式尽可能列出每一个数据库表的表结构，即字段名、类型、长度、小数位数、是否主键、索引、是否允许为空、说明等。数据库设计表如表 2.3.1 所示。

字段	类型	可为空	描述
用户信息表 USER			
uid	Varchar	N	账号，主键
uname	Varchar	Y	姓名
upassword	Varchar	n	密码
uemail	Varchar	Y	邮件
nickname	varchar	Y	昵称
enabled	varchar	Y	启用状态（0：禁用；1：启用）
admin_status	varchar	Y	是否为管理员（0：不是；1：是）
audit_status	varchar	Y	审核状态（1：待审核，2：审核通过）
日程记录表 SCHEDULE			
sid	Varchar	N	主键
user_id	Varchar	N	用户 id，外键
type	varchar	Y	日程类型

start	Datetime	Y	建立时间
Ring	int	Y	是否提醒
Priority	int	Y	优先级
用户管理日程表 u-s			
uid	Varchar	N	用户 id, 外键
sid	Varchar	N	日程 id, 外键
category	varchar	N	操作类型

表 2.3.1 数据库设计表

2.4 界面设计

此部分旨在说明界面转换关系。此软件中的界面跳转描述如下：

打开日程软件后，首先进入登录页面，输入对应的账号密码，若为新用户则自动注册；成功登录后将跳转至登录后主界面，除显示今日日程外还有五个功能按钮，分别对应新增日程、修改日程、删除日程、设置日程优先级和用户自我管理；分别点击对应按钮后，将分别跳转至对应界面。

除此之外，为保证本软件的易用性，界面设计时我们考虑了自身使用习惯，降低了软件使用的繁琐程度，具体表现为：若当日已有日程则可进行日程管理，日程管理时跳转至日程管理界面，可以删除、修改、添加日程。

用户管理界面可跳转到退出登录、导出日程界面。

界面详细设计结果如下：

（1）登录界面

时光清浅日程软件登录界面如图 2.4.1 所示。左侧蓝色字体在绿色背景的映衬下，很好的诠释了软件名称中“清浅”二字。登录可采用三种方式，其中首选推荐为手机号登录。若用户为从来没有注册过的新用户则自动注册，将其手机号、密码等登录信息记录在数据库中。



图 2.4.1 软件登录界面图

(2) 登录后主界面（若暂无日程）

时光清浅日程软件登录后主界面（若暂无日程）如图 2.4.2 所示。成功登录后的主界面上方首先将展示今日日程，若暂无日程则显示“暂无”，若有日程则按照优先级顺序从高到低顺序排序。下方四个按钮分别对应“新增日程”、“修改日程”、“删除日程”、“设置日程优先级”的四个主要功能。右上角的用户按钮点击后可进行退出登录等登录管理。



图 2.4.2 软件登录后的主界面（当日暂无日程）图

（3）添加日程界面

时光清浅日程软件添加日程界面如图 2.4.3 所示。在主界面点击“添加日程”按钮后，可以进行新增日程操作。新增日程时需写明“标题”（即日程名称）、“日期”、“时间”和“地点”等一系列必填的关键字信息。除此之外，还可以点击“设置提醒”或“设置优先级”按钮进行高级日程设置。



图 2.4.3 添加日程界面图

（4）登录后主界面（若已有日程）

时光清浅日程软件登录后主界面（若已有日程）如图 2.4.4 所示。若已有日程则显示日程标题、时间、地点等必选关键字信息和提醒时间、优先级等可选关键字信息。点击“日程管理”按钮，可以对这些信息进行修改。



图 2.4.4 登录后主界面（若已有日程）界面图

（5）日程管理界面

时光清浅日程软件日程管理界面如图 2.4.5 所示。进入日程管理时，可以删除、修改或新增日程。



图 2.4.5 日程管理界面图

(6) 用户管理界面

时光清浅日程软件用户管理界面如图 2.4.6 所示。用户管理界面显示用户名、账号、已专注时长、已有日程数、用户积分等基本信息。除此之外，可以进行修改用户名、修改头像、退出登录、导出所有日程等功能。



图 2.4.5 用户管理界面图

2.7 设计过程中的需求变动情况说明

此部分旨在说明如果设计过程中发现相关需求有变动，对系统需求变动情况进行说明。

在设计过程中，我们发现在初始需求分析中设计的“专小果”专注学习树等功能较为繁杂，通过调查这些功能也不是用户真正需要的，故删除这些除基础功能外的需求，更换为使用专注积分在商城中兑换商品的功能，以达到吸引用户的同时增强软件易用性的目的。

2.8 设计小结

系统设计部分主要进行了系统功能设计、数据库设计和界面设计。也通过设计过程对我们软件的需求有了更明确的规划，也对需求进行了调整。

3 系统实现

3.1 人员安排

时光清浅日程软件系统实现小组

组长：习婉婷

组员：孙欣欣

吕希彤

吉廉嘉

3.2 基础日程表模块实现

3.2.1 基础日程表模块实现简介

此部分旨在说明实现该模块的基本方法和过程。

每个用户在登入自己的账号后，首先显示系统时间所在的年月日程，用户也可以根据导览选择查看不同月、年的日程。用户在编辑日程时，对于每一个日程表中的已有项目都可以进行修改，查看详情、删除和设置提醒操作。其中，在修改日程表时，若修改部分有变化且合法即可修改成功，修改成功后显示新的日程内容，否则将保持显示旧的日程数据。在查看详情时，可以编辑时间、地点、事件优先级等等。在设置提醒时，可以选择提醒的提前时间，比如事件发生的前三天、前一天、前一小时等等。

需要在此说明的是，由于本小组人手不足以及技术能力不强等种种原因，目前仅对日程表的基础框架进行实现。且此版本为第一次迭代开发过程，本次开发采用演进系列模型中的增量模型作为过程模型，更多功能的实现可在后续迭代中添加。

3.2.2 基础日程表模块相关类实现

基础日程表模块涉及类的 UML 类图如图 3.2.1 所示。

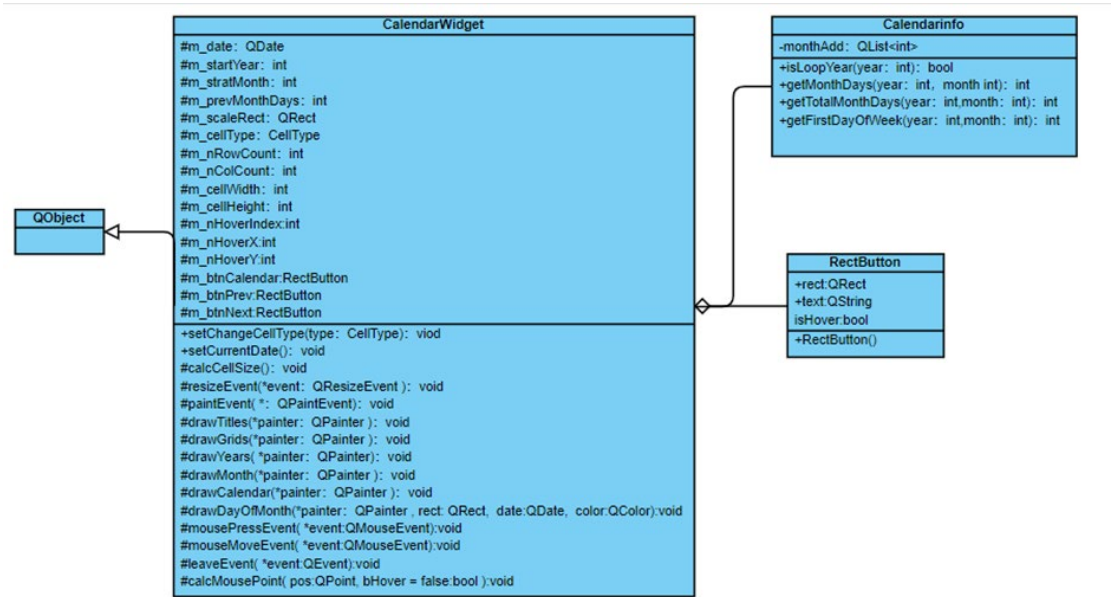


图 3.2.1 基础日程表模块 UML 类图

该模块使用的数据表及字段的描述如表 3.2.1 所示。

表 3.2.1 基础日程表模块数据表及字段描述

字段	关于该字段描述
year	用于表示年份
month	用于表示月份
countDay 和 loopDay	用于计算指定年月对应到该月共多少天 每月前面的天数+loopday 也用于计算指定年月的某一天对应星期几
week	用于表示星期
celltype	用于判断显示的是年份月份选择窗体还是当前日历显示窗体

3.3.3 基础日程表模块模块文件及跳转关系

此部分旨在详细列出该模块用到的文件及其跳转关系。

(1) 本模块涉及代码文件列表如表 3.3.2 所示。

表 3.3.2 基础日程表模块涉及代码文件列表

文件名	文件路径	文件说明
Calendar.pro	D:\qt-calendar	整体模块的工程文件
calendarinfo.h	D:\qt-calendar	日程表中信息类(年、月、日、星期)的定义

calendarinfo.cpp	D:\qt-calendar	日程表中信息类(年、月、日、星期)的实现
calendarwidget.h	D:\qt-calendar	日程表窗体的定义
calendarwidget.cpp	D:\qt-calendar	日程表窗体的实现
mainwindow.h	D:\qt-calendar	主窗体的定义
mainwindow.cpp	D:\qt-calendar	主窗体的实现
mainwindow.ui	D:\qt-calendar	UI 图形界面
mian.cpp	D:\qt-calendar	主函数

3.3.4 基础日程表模块程序流程

此部分旨在列出实现该模块的程序流程图并进行相应解释，基本流程如图 3.3.3 所示。

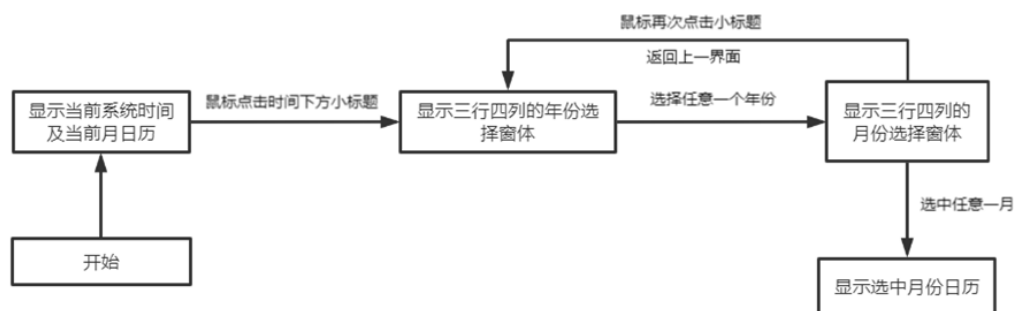


图 3.3.3 基础日程表模块程序流程图

1. 在开始运行后，就会显示出当月份的日历，左上方显示系统时间，其下方设有小标题，点击右侧两个箭头可以进行前后月份的切换的日历。
2. 然后是切换到月份和年份的选择界面，鼠标点击小标题后，即可切换到三行四列的月份选择窗体，再次点击小标题切换为年份选择窗体，其排列同样采取三行四列的形式。
3. 当第一次切换到月份选择界面时，若选择点击某月，则显示的日历为本年选中月份的日历；若暂不做月份选择，再次点击小标题，在出现的年份选择窗体中，任选一年，后在新出现的月份选择窗体中任选一月，则可显示选中的

****年**月的日历。年月日均选择好后即可显示今日日程并进行后续的日程管理等功能。

3.3.5 基础日程表模块实现界面

初始运行界面如图 3.3.4 所示。

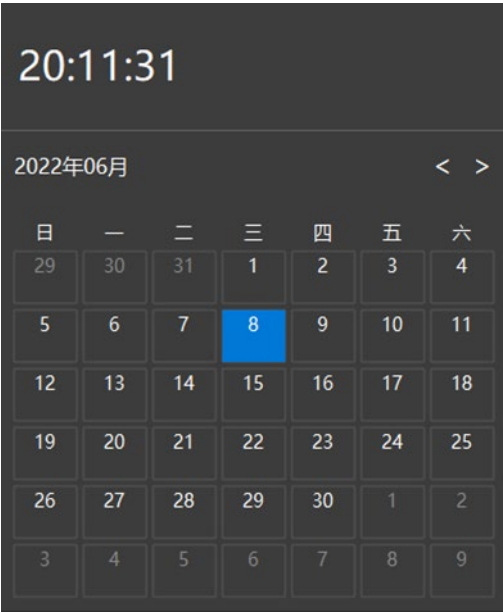


图 3.3.4 初始运行实现界面

选择某年内的不同月份的界面如图 3.3.5 所示。

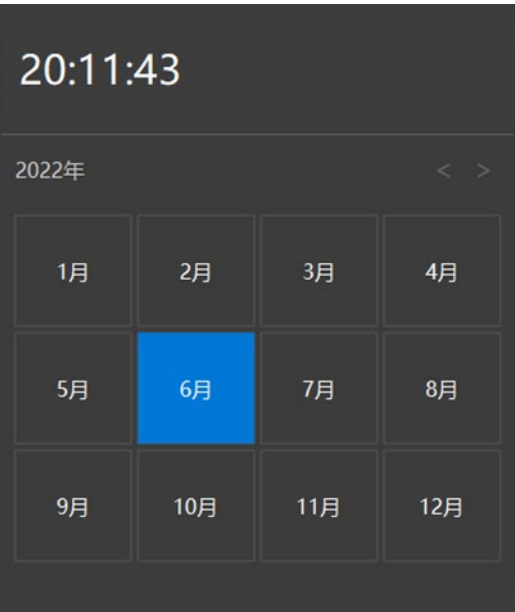


图 3.3.5 选择月份实现界面

选择不同年份的界面如图 3.3.6 所示。

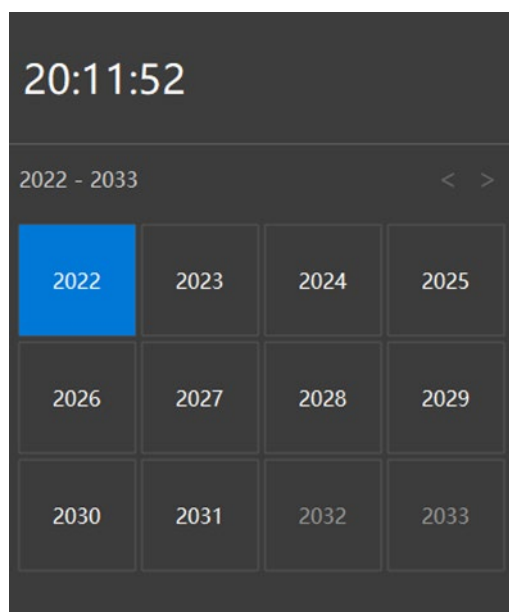


图 3.3.6 选择年份实现界面

3.3.6 登录模块实现界面

登录模块实现界面如图 3.3.7 所示。



图 3.3.7 登录模块实现界面

3.4 实现过程中的需求和设计变动情况说明

在实际实现过程中，发现界面设计难以实现，因此对基础界面修改为简洁模式。

3.5 实现小结

由于此软件开发周期较短，团队成员较少且存在较多技术障碍，系统的实现完成并不多。后续将首先对登录界面等进行实现并加入数据库数据。

4 系统测试

4.1 人员安排

时光清浅日程软件测试小组

组长：习婉婷

组员：孙欣欣

吕希彤

吉廉嘉

4.2 测试用例设计

4.2.1 用户模块测试

（1）白盒测试用例

白盒测试用例设计如表 4.2.1 所示。

表 4.2.1 白盒测试用例设计

用例名称	发布新消息测试用例
测试场景	计算两个日期之间的天数差异
测试数据	选择几个具有代表性的日期对 包括跨越不同月份、年份、闰年、非闰年的情况。
测试步骤	输入开始日期和结束日期。 调用日期计算函数。 检查返回的天数是否与预期一致。
测试场景	在指定日期添加日程事件。
测试数据	有效/无效的日期，事件标题和描述等
测试步骤	输入相关信息 调用添加事件函数 检查事件是否已成功添加到日程中 检查在无效输入（如非法日期）时，软件是否给出错误提示
测试场景	修改已存在的日程事件
测试数据	选择已存在的日程事件进行修改，包括修改日期、标题和描述等
测试步骤	选择要修改的日程事件。 输入新的日期、标题或描述。 调用修改事件函数。 检查事件是否已成功修改。
测试场景	删除已存在的日程事件

测试数据	已存在的日程
测试步骤	选择要删除的日程 调用删除事件函数 检查事件是否已成功删除
测试场景	检查软件在边界条件和异常输入下的行为
测试数据	包括日期边界值（如 1 月 1 日、12 月 31 日）、非法日期（如 2 月 30 日）、空标题、超长标题等。
测试步骤	输入边界条件和异常输入 观察软件的响应和错误处理机制 检查软件是否给出正确的错误提示或处理异常输入

（2）黑盒测试用例

黑盒测试用例设计如表 4.2.2 所示。

表 4.2.2 黑盒测试用例设计

用例名称	发布新消息测试用例
测试目的	验证日程软件的主要功能，如添加、查看、修改和删除日程事件，是否按预期工作。 检查软件在不同条件下的稳定性和易用性。
测试用例设计	
添加日程	
测试步骤	1. 打开日程软件 2. 选择添加新事件的选项 3. 输入事件的标题、日期、时间和描述 4. 确认输入无误后，上传数据
预期结果	事件应成功添加到日程中，并在指定的日期和时间显示 可以在软件的事件系列中看到新添加的事件
查看日程事件	
测试步骤	1. 打开日程软件。 2. 导航到事件查看页面或列表。 3. 检查所有已添加的事件是否都正确显示，包括标题、日期、时间和描述
预期结果	所有已添加的事件都应按日期顺序正确显示。 事件的详细信息应与添加时输入的信息一致
修改日程	
测试步骤	1. 打开日程软件。 2. 在事件列表中找到要修改的事件。 3. 选择修改选项。 4. 修改事件的标题、日期、时间或描述（如果需要）。 5. 确认修改后，上传数据。

预期结果	修改后的事件应成功更新，并在指定的日期和时间显示新的信息。 在软件的事件列表中，修改后的事件应显示新的详细信息。
删除日程	
测试步骤	1.打开日程软件。 2.在事件列表中找到要删除的事件。 3.选择删除选项。 4.确认删除操作。
预期结果	选定的事件应从日程中删除，不再显示。 在软件的事件列表中，应不再包含已删除的事件。
边界条件和输入 异常测试	
测试步骤	1.输入无效日期（如 2 月 30 日）并尝试添加事件。 2.尝试删除不存在的事件。 3.尝试添加或修改事件时，不输入任何信息或输入空字符串。
预期结果	软件应给出明确的错误提示，如“无效日期”或“事件不存在”。 对于空输入或无效输入，软件应拒绝保存或提交操作，并给出相应的错误提示

4.2.2 用户模块测试报告

登录模块测试报告如表 4.2.3 所示。

表 4.2.3 登录模块测试报告

用例	功能	错误描述			统计项				
		应达目标	实际结果	出错原因	出错类型	测试者	测试日期	改正日期	状态
UseCase0001	检查密码错误	resultEntity 中 message 属性被赋值为“密码错误”	符合目标			习婉婷	2024-4-28		通过
UseCase0001	检查账号错误	网页提示“账号不存在”	系统抛出异常	未对数据库查询结果判空	代码逻辑错误	习婉婷	2023-4-28	2023-4-28	改正通过

UseCase0001	检查空用户名	网页提示“用户名为空”	符合目标			习婉婷	2023-4-28		通过
UseCase0001	检查空密码	网页提示“密码为空”	符合目标			习婉婷	2023-4-28		通过
UseCase0001	检查身份错误	网页提示“密码错误”	符合目标			习婉婷	2023-4-28		通过
UseCase0002	检查保存功能	resultEntity 中 message 属性被赋值为“fail”	符合目标			孙欣欣	2023-4-28		通过
UseCase0003	添加日程功能	新事件成功添加至日程中,并在指定的日期和时间显示	符合目标			孙欣欣	2023-4-28		通过
UseCase0003	日程添加失败	网页提示“请重新添加日程”	符合目标			孙欣欣	2023-4-28		通过
UseCase0004	日程查看功能	所有已添加事件按日期顺序显示	符合目标			吉廉嘉	2023-4-28		通过
UseCase0005	修改日程	修改后的时间应成功更新	符合目标			吉廉嘉	2023-4-28		改正通过

	程 功 能								过
UseCase0005	修 改 日 程 失 败	提示“日程更新失败！请重新操作。”	符合目标			吉廉嘉	2023-4-28		通过
UseCase0006	删 除 日 程 功 能	选定的事件应从日程中删除，不再显示	符合目标			吕希彤	2023-4-28		通过
UseCase0006	删 除 日 程 失 败	提示“删除失败，请重新操作！”	符合目标			吕希彤	2023-4-28		通过

4.2.3 用户模块调试过程

针对出错测试用例 UseCase0001，检查 IDE 的控制台输出，发现系统抛出了空指针异常，根据提示信息定位到对应的方法体内，发现出错位于对数据库查询类返回的对象进行操作的语句中。分析得知此处未对返回的对象判空，加上判空语句后通过测试。

测试用例 UseCase0005 出错时，测试数据是“2025-2-30”，该日期非法，但仍成功添加日程，后修改代码，添加了输入验证逻辑，确保用户输入的日期是有效的。使用日期库或函数来检查日期的有效性。同时，当检测到无效的日期输入时，软件应给出明确的错误提示，如“输入的日期无效，请重新输入”。

4.3 模块测试

由于本组实现的模块暂未设计出测试用例，也未能察觉到需要设计的用例，

因此不再赘述。

5 心得体会

在本次软件工程开发项目中，我们从 0 开始，进行一个项目的研发。从问题定义、可行性分析，到需求分析、软件架构设计和详细设计，再到最后的系统实现以及系统测试，通过一步步的摸索我们组四个人对软件的开发流程有了更加深刻的理解，对软件工程这门课也有了更加深刻的认识。

首先必须要承认是我们此次作业无论是在系统实现还是在系统测试部分完成地都不够完善。团队成员总结原因为软件开发周期较短且在前期问题定义、可行性分析、需求分析等准备阶段耗时较长。除此之外，本团队成员均没有相关开发经验，软件开发过程中存在较大的技术风险，大部分需要边学边做。下面对整个开发过程进行总结。

在软件开发的前期，在确定了我们的开发方向是做一款日程管理 APP 后，我们组内成员通过上网搜索网上已有类似软件的相关资料，探索件功能的定义。在人员安排上，我们充分考虑了不同人优势及劣势，进行分工。在软件开发语言的选择上，我们选择了较为熟悉的 C++ 语言进行开发，可以通过封装、继承和多态等特性，更有效地组织和管理代码，实现更复杂的功能。在工具的选择上，我们也尽量选择了我们较为熟悉的 Visual Studio Code、QT、PostgreSQL 等，尽量减少我们学习相关语言、工具使用的时间在整体开发过程中的占比，提高开发效率。在过程模型的选择上，我们一开始选择了快速原型模型，在开发过程中，考虑到实际因素，改为增量模型，并用甘特图展示软件开发周期（进度计划）。

在需求分析阶段，通过深入了解用户需求，明确日程软件的开发目标、功能需求、用户交互方式及系统性能等关键要素，为软件开发团队提供清晰、一致的指导和参考。并通过文档的编写，进行用例图绘制以及相应的用例描述，进行对象行为模型、对象关系模型构建等，确保软件开发过程中各方对项目的理解保持一致，提高软件开发的效率和质量。

在系统设计阶段，我们从系统功能模块设计开始，逐步进行类设计、数据库设计、接口及过程设计、界面设计、系统流程分析，为开发团队、利益相关者、测试人员、用户以及可能的未来维护者提供一个全面、详细且一致的参考，以确

保软件的开发、测试、部署和维护过程能够顺利进行。

在系统实现阶段，囿于我们组四个人代码能力不足，我们最后并未完成我们之前规划的所有功能，只完成了非常基础的部分实现，也因此，我们的系统测试阶段也只停留在的报告中的测试用例的规划阶段。但在这一过程中，我们都收获良多，不仅加深了我们对课程知识的理解，更提高了我们的团队协作能力，沟通能力，问题解决能力……为我们以后学习和工作打下良好基础。