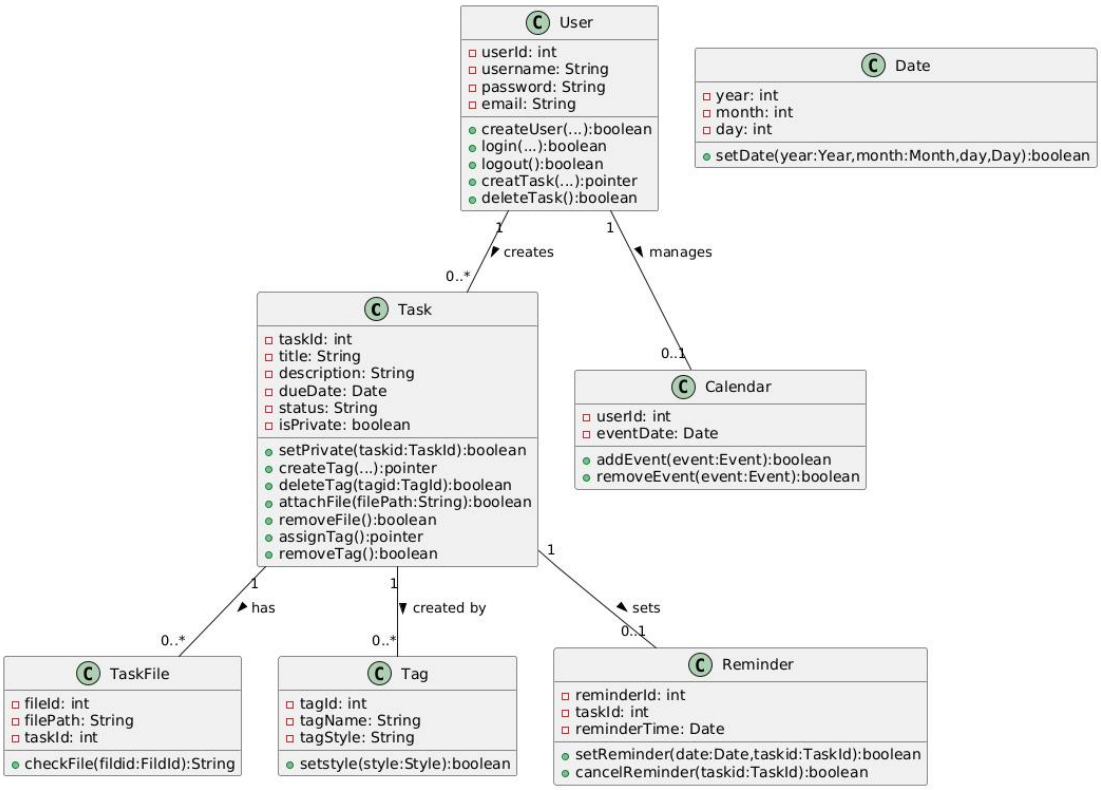
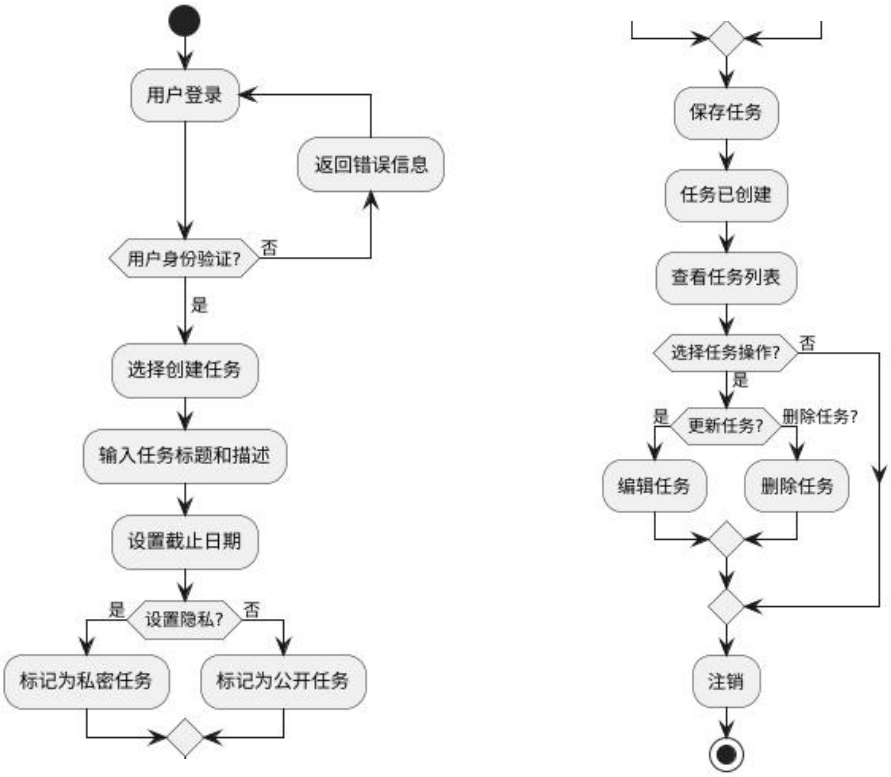
软工实验 3 实验报告

徐庶 221220041

一、项目内容

本次实验需要根据实验二所设计的 UML 图，将设计的软件实现为代码，实验一中我选择的软件是 ToDoList，现将 UML 类图和活动图粘贴如下：





二、实验过程

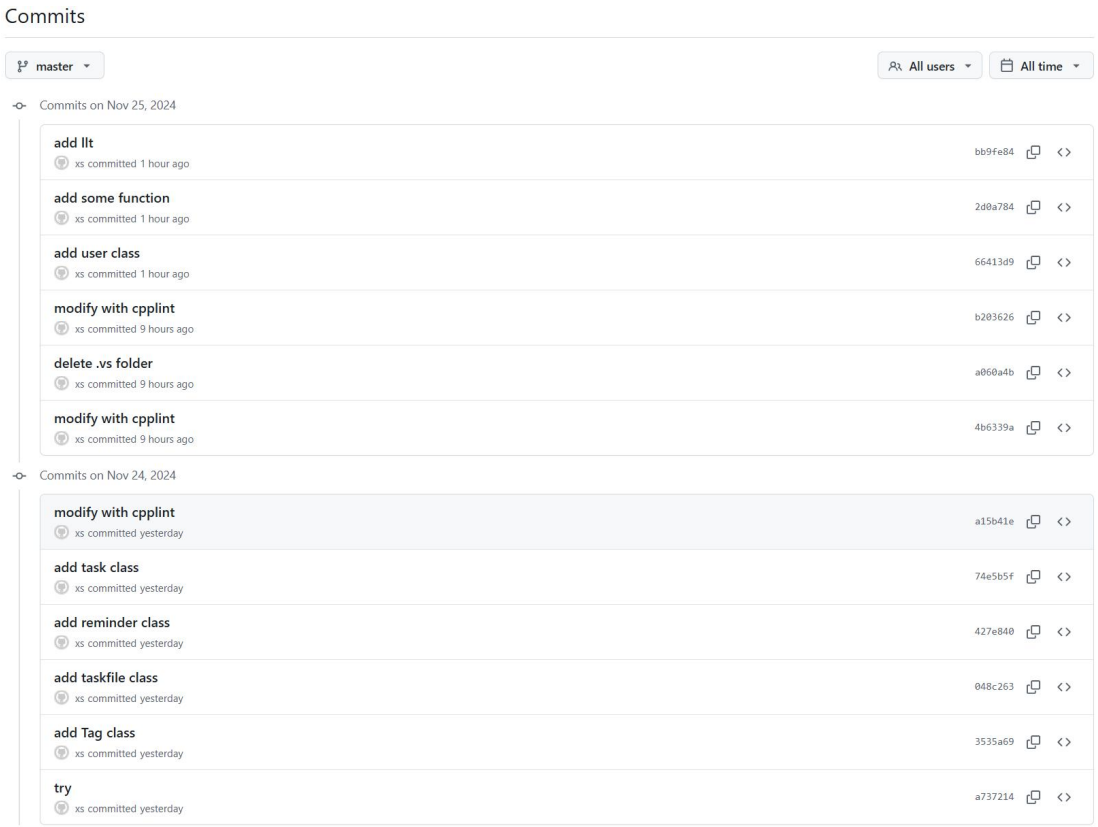
我选择使用 C++编程语言，使用 ChatGPT 大模型辅助开发，使用 github 远程仓库进行代码管理。

首先我将实验二中 UML 类图的源代码喂给大模型并让它帮助我据此生成第一版的 C++代码。大模型的思路与我相同，将 UML 类图的每个类进行实现，再进行耦合。但是显然大模型并不能完美胜任此工作，即大模型生成的内容不符合我的预期，它直接在一个文件中写完了这些内容。于是，我再详细地告诉大模型我的需求：将类的定义和实现分开来写，即一个类分为头文件（.h）和源文件（.cpp）两个文件来写。据此我获得了第二版的 C++代码，共有五个类，十个文件。此时已然框架已经搭建完毕，下面就可以在其中不断丰富。

大模型生成的代码在细节上不是很完美，比如在 UML 代码中显然给出了各个类的成员变量都是私有成员，但是大模型都一视同仁地将类中所有的元素设置为了 public。最难受的是因为这样，大模型生成的源代码中很多在类外直接调用类的成员变量，但是我手动设置为 private 后显然这是不合法的，我需要写很多 get 和 set 方法并且修改。再比如有些函数的返回值是布尔值，其中隐含的逻辑是若函数的语义（本来目的）执行成功则返回真，否则为假，但是大模型并没有 get 到。在不断的实现过程中，我学习并使用了 std 库中的 tm 类型，这是一个 struct，用来表示时间，可以存储很多时间量，因此可以完美替代类图中对 Date 类的需求。但是 vs 对于 std::asctime()函数有意见，认为它不安全从而导致编译不通过，解决方案见错误 C4996 asctime。最终，大模型帮助我生成的代码在 150 行左右，我将代码扩充至了 543 行。

在修改过程中，我选择按类的划分来写，每完成一个类就提交一次代码。其中由于类之间存在依赖关系，如 task 类依赖三个类，而 user 类又依赖 task 类，所以提交的顺序基本固定。此外，每做一次重要的改动也会提交代码。下图为 github 上的 commit 过程。仓库链接：

G221220041/toDoList



可以看到提交过程中有些提交的内容为“modify with cpplint”，我使用 cpplint 对我的代码风格进行了检查，除了没有在各个文件最开始标明著作权外其它均符合 cpplint 对代码风格的要求。还记得暑假实习的时候被代码风格检查（code check）支配的恐惧，比如一行代码最后面不能有空格，每行字符不能超过 80 个，命名要用小驼峰法等等。最无法理解的是，单个文件代码超过 2000 行会有大文件警告，需要找主管消除警告否则只能拆成两个文件来写。这也难怪他们说，尽管有开源社区，但是公司外的人单是代码合入那一步就要捣鼓一年。所以，有了这些奇怪的经历，我对 cpplint 的代码检查就见怪不怪了。

三、实验结果

我让大模型根据 UML 活动图的源代码帮我生成了测试用例（LLT），并自己完善写入 test.cpp。比如大模型认为登录失败应该直接返回，但是这样就覆盖不到其他的函数，所以我将逻辑改为循环登录直到成功。下图为整体测试流程的控制台输出，可以看到与 UML 活动图展示的流程基本相同。

```
please give me your username and password:
xs 123456
Wrong password
Login failed! Please try again
please give me your username and password:
xs 123
User xs logged in
User logged in successfully!

Creating task...
Task Created: Study for exam with ID 1
User xs creating task Study for exam
Task ID: 1, Title: Study for exam, Description: Prepare for upcoming exam, Due Date: Sun Oct 15 00:00:00 2024
, Status: active, Private: 0

Do you want to mark the task as private? (y/n): y
Task 1 marked as private
Creating tag for task 1 with name: Important
Tag created: Important with ID 1
Get task id
TaskFile created: /files/study_guide.pdf with ID 1
Get task file path
Attaching file /files/study_guide.pdf to task 1
Create reminder for task 1
Reminder Created with ID 1

Do you want to update the task? (y/n): y
Editing task...
Set task title to Revised Study Plan
Set task description to Prepare for final exam.
Task ID: 1, Title: Revised Study Plan, Description: Prepare for final exam., Due Date: Sun Oct 15 00:00:00 2024
, Status: active, Private: 1

Do you want to delete the task? (y/n): y
Deleting task...
Get task title
User xs deleting task Revised Study Plan
User xs logged out

User logged out successfully!
```

四、心得体会

本次实验完成了 ToDoList 软件的框架实现与构建，学习使用大模型辅助软件开发和 Git 管理本地及远程仓库。大模型对开发的帮助是巨大的，我们可以通过大模型生成琐碎的框架代码，然后在其中进行完善和写入复杂的逻辑。但是实验报告的书写不允许使用大模型的辅助是痛苦的，这也是我本次实验报告篇幅不长的重要原因。Git 是一个很高效的代码管理工具。通过数据库对 undo 和 redo 的学习，对 Git 的 cherry-pick 有了更深入的理解。但是代码提交的频率是一个值得思考的问题，过低容易丢失代码，过高，哈哈，那就慢慢一条一条 cherry-pick 吧。