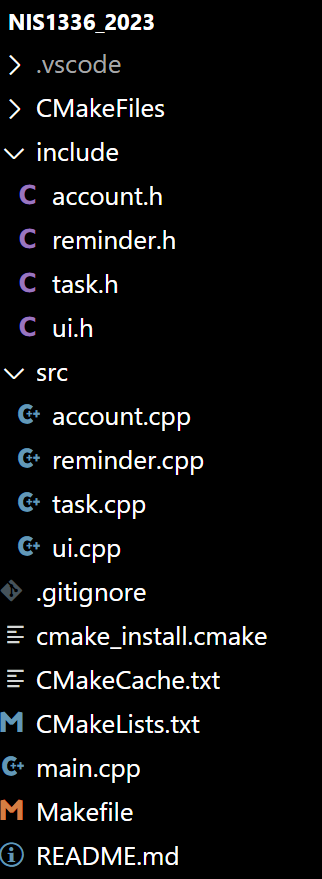
NIS1336课程作业设计文档

小组成员：崔轶哲、池承运、刘潇峰、陈楠

1. 分工
2. 崔轶哲 全部task ，部分ui
3. 池承运 main函数，frame，部分ui
4. 刘潇峰 account
5. 陈楠 reminder
6. 模块与类的设计

课程作业主题分为四个模块：account，reminder，task，ui。具体文件列表如图1所示：

其中各部分的功能分别为：

**account**：与User类基本一致

**reminder**：①通过文件名（UID）获取当前的任务列表，并以vector存储。②通过函数reminderbyDuetime()以任务截止时间输出，通过函数reminderbyId()以任务ID顺序输出。③scan()函数，通过后台循环，在文件锁（采用mutex锁）匹配的情况下，以1min的周期检查是否文件解锁，在解锁的情况下进行扫描操作，如果任务的提醒时间已到，则提醒。④通过compareTime, getCurrentTime函数来获取当前系统时间，并与任务的提醒时间比较。

**task**：①设置task类，以完成各种task类的操作（将在后面进一步讲明）。②addTask函数实现在任务列表（vector<Task>）中添加任务；deleteTask函数实现在任务列表（vector<Task>）中删除任务（调用erase）；loadTaskFromFile函数，为避免多线程之间的矛盾，该函数在完成操作前，先锁住文件，然后从文件中读出任务列表，以vector<Task>形式存储。操作完成后解锁文件；saveTaskToFile函数与loadTaskFromFile函数类似；printTasks函数以setw函数依次输出Id，Name，Start Time等信息；getTaskByDate函数通过将任务列表里的任务的截止时间信息中的日期字段与当前系统时间的日期字段相比较来输出当前日期需要完成的任务；sort\_by\_Id函数利用id\_compare **图1：文件列表**

函数将任务列表中的任务根据id进行升序排序；sort\_by\_reminderTime函数利用time\_compare函数将任务列表中的任务根据reminder\_time进行升序排序；lockFile与unlockFile函数应用<mutex>对文件进行上锁与解锁。

**ui**：①run函数：通过用户的输入而进行进程的选择，是前台的控制函数 ②login函数，该函数返回userlogin函数的结果。③由showHelp（打印help），showTask（打印任务列表），ui\_addTask（输出基本的提示，等待用户输入，然后调用task中的“addTask”函数，）, ui\_deleteTask（输出基本的提示，等待用户输入，然后调用task中的“deleteTask”函数，如输入有误则返回错误提示）, showTaskByDate（通过比较任务的duetime与输入的日期进行比较，返回输入日期当天的任务列表）, undoTask（调用setReminded函数，标记为未提醒，在该程序中，reminded与done使用同一个字段，即只要没完成逾期任务，在每次刷新时，系统都将提醒一遍）, doTask（调用setReminded，提前将任务的reminded字段修改为“true”即“已提醒”） 等函数完成ui界面所需要的基本功能。④ isValidTime，isValidDate等函数则实现了检验输入的合法性。

**main：**整体功能的调用与执行。

各部分的调用关系将在流程图中展示。

课程作业中设计的类有：User，Reminder，Task，UI，Priority，Category

**User**：①两个构造函数，分别从文件中加载用户信息与创建新的用户。②encrytPassword函数应用MD5进行对密码的hash加密；checkPassword函数将由用户输入的密码产生的hash码与原来存储的hash码进行对比，一致则成功登录。③saveUserToFile函数将用户列表导入到文件中；loadUserFromFile函数将文件中的用户列表导入到vector中。④userLogin函数中，注册与登录使用同一套代码，如果该账户尚未被注册过，则自动注册，如果在用户列表中查找到一致的，且用户名和密码均正确，则登录。如果注册的新账号和原有账户列表中的账号用户名相同，不会产生冲突，即该系统中允许重名。

**Reminder**：①两个构造函数，分别从文件中读取任务列表存入vector中与从以vector存储的任务列表中直接读取任务列表。②reminderbyId函数与reminderbyDuetime函数分别按照任务列表的Id或截止时间升序排序，并按此顺序输出任务列表。③通过文件名（UID）获取当前的任务列表，并以vector存储。④通过函数reminderbyDuetime()以任务截止时间输出，通过函数reminderbyId()以任务ID顺序输出。⑤scan()函数，通过后台循环，在文件锁（采用mutex锁）匹配的情况下，以1s的周期检查是否文件解锁，在解锁的情况下进行扫描操作，如果任务的提醒时间已到，则提醒。⑥通过compareTime, getCurrentTime函数来获取当前系统时间，并与任务的提醒时间比较。

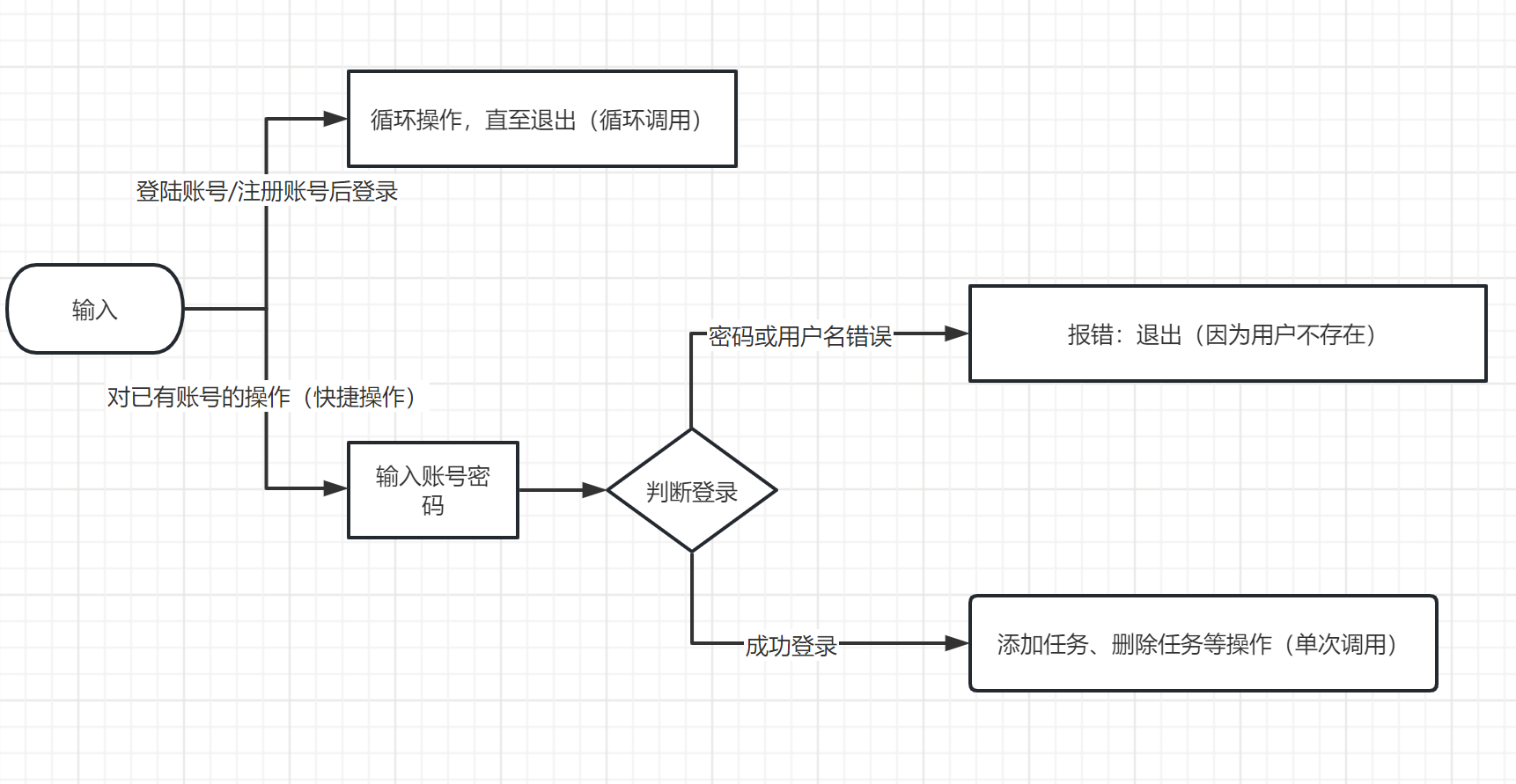
**Task**：①两个构造函数，分别从文件中直接读取任务列表（从文件中加载任务列表）和在添加新任务时调用。②Task类的私有成员包括s\_nextId, m\_id，m\_name, m\_startTime, m\_priority, m\_category, m\_reminderTime, m\_reminded。③一系列与Task类的私有成员对应的函数，如getName与setName，isreminded等等。

**UI**: 与ui部分一致

**Priority**：任务的优先级 HIGH, MEDIUM,LOW

**Category**：任务的类别 LEARNING, LIFE, ENTERTAINMENT

1. 流程图：



**图2：UI界面用户操作流程图**

1. 关键技术问题说明:
   1. **Multithreading**：采用多线程的方法， 防止后台在等待用户进行新的输入之前一直在等待用户输入，而无法在后台进行刷新（任务的扫描）。
   2. **Mutex**：互斥锁实现多线程同步的核心思想是：有线程访问进程空间中的公共资源时，该线程执行“加锁”操作（将资源“锁”起来），阻止其它线程访问。访问完成后，该线程负责完成“解锁”操作，将资源让给其它线程。当有多个线程想访问资源时，谁最先完成“加锁”操作，谁就最先访问资源。由此防止后台运行的scan（）与前台用户的输入同时进行，而导致程序运行出现问题。