

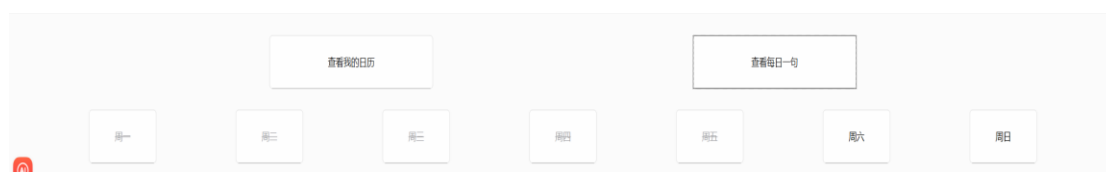
TriDoS 小组大作业报告

赵俊凯 张艺洋 龙俊宇

一、模块及其功能介绍

模块一：The Calendar 模块

整体功能：提供日历索引、星期几提示以及每日箴言

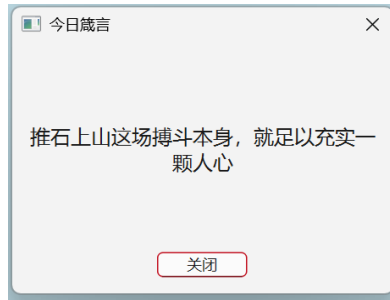


功能一：我的日历



点击“查看我的日历”，可以弹出一个日历窗口。这个日历会把所有涉及ddl的日期显示为黄色高亮，将光标移动到对应日期上去，可以看到当天的ddl实际情况。同时每个日期也作为一个索引，点击对应的日期，可以切换下方的“每日计划”到对应日期的界面，实现更为方便的跳转与事项查询。

功能二：每日箴言



点击“查看每日一句”，会弹出一个窗口，里面是精选的每日箴言，用于鼓舞用户。

功能三：激励式星期



七个图标显示每周的周一到周日，每当对应的星期几结束后，会在对应的图标上面显示一条横线以表示这一天已经结束。希望可以借此提醒用户当天是星期几，并且给予用户一点心理慰藉，周末马上就到了！

模块二：deadline 模块



功能一：设置 ddl

点击“设置 ddl”，可以创建一个新的 ddl 实例，并且设置它的标题、截止日期、描述

功能二：ddl 显示与提醒

已经被创立的 ddl 会显示在应用程序的对应窗口（如上），显示标题、截止日期、剩余时间，如果点击对应的 ddl 事项会弹出我们一开始写的描述，点击删除会将对应的 ddl 事项移除。

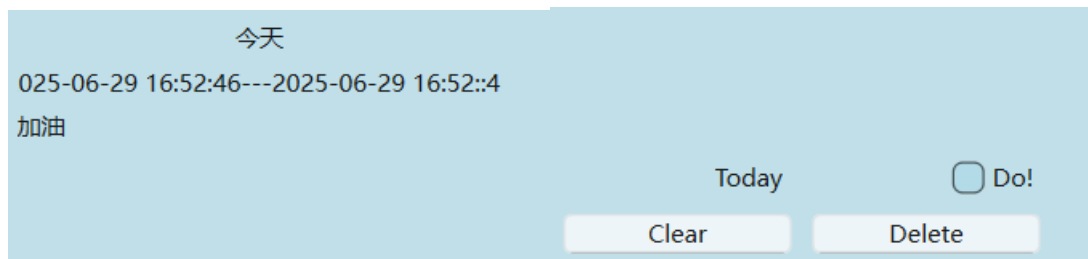
模块三：每日计划模块



功能一：首先用户界面可以看到当天的时间和日期，进度条展示的是当天已经过去了多少时间。

功能二：在日历中点击对应的日期之后，将会在界面呈现出当天的日程表，即在这一天内需要完成的任务。

功能三：点击日程表中的事件，可以在右侧的详情栏看到对应的事件内容，包括事件的主题，事件的持续时间，事件的描述。可以在详情页中通过下方的勾选来标记该日程是否已经完成。也可以通过修改和删除按钮对当日的时间进行调整。



功能四：支持创建习惯，鼓励用户养成良好的习惯，点击用户界面上方的日程表，则可以直接切换到“习惯列表”，在这里将呈现出用户创建的习惯项目，用户也可以进行打卡和修改，来帮助自己养成好的习惯，提升自己。



功能五：在每日计划的界面除了日程和习惯，还提供了一个番茄钟的外置挂件，用来满足倒计时功能



模块四：Longtimeproject 模块

整体功能：对于个人长期项目进行管理，可以实现添加子项目并对每个项目的完成情况进行管理

项目：大作业

描述：

截止日期：2025-07-17

收起子任务

删除项目

☐ 1

描述：

截止日期：2025-06-07

☐ 2

描述：

截止日期：无

功能一：对大项目以及内部的子任务进行分别管理，设置其名称、详细描述等。

功能二：可以实现对每个子任务选择性地添加 ddl，并将其同时同步到 ddl 上，便于 ddl 的统一管理。

功能三：可以对每个子任务设置并查看其完成状态

二、类实现细节：

1、main 函数：创建一个 QApplication 以及一个主窗口，程序开始运行。

2、mainwindow 类：作为主窗口，在创建对象的时候分为四大部分，即分别创建 The calendar/ddlreminder/dailyplan/longtermproject 四个新对象，进行初始化操作并且在主窗口上按照一定预先设定好的比率展示。

3、Thecalendar 类：

① Ui 界面在 Qt 的设计栏内搭建完成

② 按钮交互利用信号槽函数实现

③ 创建了一个计时器对象，用于根据时间更新“星期几”按钮的状态以及每日一句的随机种子。利用一个 QStringList 存储每日一句。

④ 创建了一个子类 calendardialog，对 Qt 本身的 Calendar 插件进行了功能扩展，通过信号槽函数实现了光标悬停和点击对应的响应功能。

4、ddlReminder 类

① 创建了最小的对象——ddlitem 类，作为最小单位 ddl 单项的记录，并且在类内实现了把对象转存入 json 文件与从 json 文件中恢复的方法。

② 利用 ddlwidget 类实现从已有的 ddlitem 容器中提取 ddl 对象并且加载到屏幕相应区域。在类内实现了 ddl 显示随日期更新，以及利用信号槽函数实现鼠标点击功能。

③ ddlReminder 类内主要进行数据的存储与加载，以及提供了 ddlitem 添加的接口，实现对于数据的整体管理。

5、longtermproject 类

①设置了 projectitem 以及 goalitem 两个类，其中每个 projectitem 由多个 goalitem 组成，对其使用数组形式存放，分别代表每个大项目以及其含有的各个子项目，并在类内部分别实现了对大项目以及子项目的信息的存取，并同样通过 json 文件的方式实现对信息的存储和读取。

②采用 projectwidget 类，完成对于已有大项目及其对应子项目的显示，并可以对每个子项目设置其完成状态

③采用 longtermproject 类，实现整体的添加大项目，删除项目，加载存储数据等功能

④在用于添加新的大项目的函数中，对于其中设置了截止日期的项目，将其同步添加到 ddlreminder 中进行显示，方便统一查看

6、pomodorowidget 类

用于实现番茄钟这一功能，可以设置工作时长，休息时长。

7、dailysentencedialog 类

用于实现每日一句的显示

8、dailyplan 类

①使用 dailydata 类实现对于每日信息的存储，通过 json 进行数据储存。对于每天的日程，通过以日期为索引，创建不同的 json 进行储存。对于习惯项目，全部归结到一个 json 文件中。在使用的时候呢进行读取。

②交互上，为每一个项目创建一个按钮，按钮会统一释放一个信号，引起细节界面更新相应的内容。创建日程和创建习惯则通过弹出一个窗口进行事件的创建。新创建的事件先录入后台数据，再更新前端的日程表。日程表和习惯表按钮通过释放信号实现切换。每过一秒钟更新一次界面，通过 Qt 的布局构建来调整不同组件之间的位置关系

三、小组成员分工

张艺洋：初期方案的整体框架设计、代码四大模块的类框架搭建与 API 设计、界面整体设计、TheCalendar 模块和 deadline 模块的设计与实现、日历部分功能与 DailyPlan、LongTermProject 的联动的实现。

赵俊凯：参与初期方案的设计，实现 longtermproject 模块的设计实现与交互显示、数据存储、longtermproject 与 ddl 模块的联动显示，录制项目运行演示视频，整理项目报告。

龙俊宇：参与初期方案的设计。负责完成 `dailyplan` 界面和相关的交互实现。完成 `dailyplan` 数据的储存和复用。搭建 `dailyplan` 和其他组件沟通的逻辑。

四、总结与反思：

总结：

我们基本实现了预定项目计划的各项功能，同时也在项目的不断进行过程中对项目的具体预期实现进行了修正

出现的问题：平衡各模块的整体性与独立性

在进行框架的搭建时，为了方便小组分工，把整个程序拆成了四个很独立的大板块。这在一开始的工作中为我们提供了诸多便利。然而当我们已经基本上实现完自己的部分，开始进行各模块的联动时，之前给我们带来巨大便利的独立式编程显露其弊端。这弊端主要体现在由于每个成员自己写代码的风格不同，导致当初认为只需提供一个 API 就可以解决的问题也由于对应模块内部耦合性过强而需要对代码进行大改。

反思：

在完成整个项目的时候要关注初期的 API 的整体商定，即 API 应该是大家一起商量出来的，然后再分别完成对应的 API 内部的细节内容，这样也便能更符合“高内聚、低耦合”的要求。

另外，方案的设计需要明确的区分前端和后端，在不同模块的搭建中，小组内需要实时保持沟通，不断地同步彼此的想法，最后的合成才能顺利。