**《数据库实训》 课程报告**

**(2020-2021 学年第2 学期)**

**题目 投资组合评比器**

**提交日期： 2021 年 6月 18日**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学 院** | **软件学院** | | **专业班级** | **软件工程19级1班** | |
| **课程名称** | **数据库实训** | | **任课教师** | **曾兵** | |
| **人数** | **3** | |  |  | |
| **学 号** | | **学生姓名** | | | **分 数** |
| **201936380086** | | **陈涵** | | |  |
| **201930381317** | | **梁永豪** | | |  |
| **201930381355** | | **廖宇延** | | |  |
|  | |  | | |  |
|  | |  | | |  |
|  | |  | | |  |
|  | |  | | |  |
| **教师评语：**  **教师签名：** | | | | | |

目录

[正文 2](#_Toc74948162)

[一、需求分析 2](#_Toc74948163)

[功能性需求 2](#_Toc74948164)

[非功能性需求 3](#_Toc74948165)

[二、概要设计 3](#_Toc74948166)

[1. 系统逻辑架构 3](#_Toc74948167)

[2. 系统物理架构 4](#_Toc74948168)

[3. 系统数据模型 4](#_Toc74948169)

[4. 系统开发架构 4](#_Toc74948170)

[三、详细设计 4](#_Toc74948171)

[1. 概述 4](#_Toc74948172)

[2. 术语表 5](#_Toc74948173)

[3. 设计概述 5](#_Toc74948174)

[四、软件测试 9](#_Toc74948175)

[测试用例1 9](#_Toc74948176)

[测试用例2 10](#_Toc74948177)

[测试用例3 11](#_Toc74948178)

[测试用例4 11](#_Toc74948179)

[五、软件演示 12](#_Toc74948180)

[用例1 12](#_Toc74948181)

[用例2 15](#_Toc74948182)

[用例3 16](#_Toc74948183)

[用例4 17](#_Toc74948184)

[六、软件使用教程 18](#_Toc74948185)

[安装要求 18](#_Toc74948186)

[程序使用 18](#_Toc74948187)

[小组讨论纪要 19](#_Toc74948188)

[个人总结 20](#_Toc74948189)

[项目进度安排等项目管理内容 20](#_Toc74948190)

# 正文

## 需求分析

### 功能性需求

1、实现从蛋卷基金和且慢基金网站上爬取相关基金数据。

2、将爬取的相关基金数据存储至本地数据库中。

3、实现图形化界面，可对以下内容进行展示。

a.可以把投资组合放在同一时间段下进行比较，时间段可由用户指定，例如, 20180101-20210401。

b.可以对比不同的指标，如：涨幅/年化收益率, 最大回撤, 夏普比, 年化波动率。

c.可以根据上述指标进行排序。

4、可扩展数据源，可以添加且慢和蛋卷平台新的数据源。

### 非功能性需求

1、给予用户准确及时的信息。

2、爬取时间不可过长，防止用户等待。

3、客户输入不正确的数值，系统能给予反馈。

4、易于用户更新数据。

5、易于用户操作。

6、易于用户下载安装。

## 二、概要设计

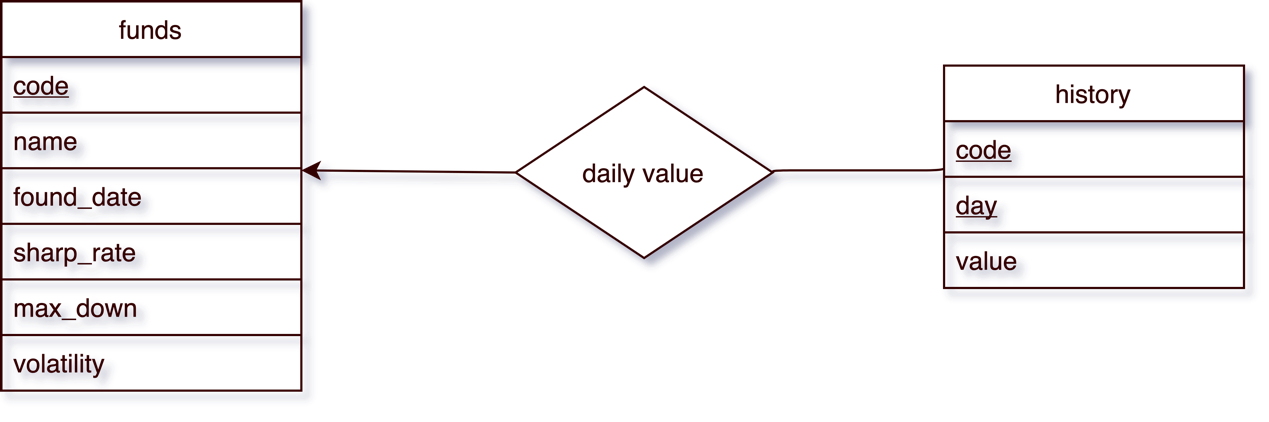
### 1. 系统逻辑架构



### 2. 系统物理架构

|  |  |
| --- | --- |
| 建议CPU | Intel® Core™ i7 Gen8 or higher  AMD Ryzen R5 Gen2 or higher |
| 建议内存 | 4GB DDR4 Memory or more |
| 系统 | Windows 10 20H2 or later/MacOS Big Sur 11.4 or later |

### 3. 系统数据模型

E-R图

### 4. 系统开发架构

图形化界面-python3.8 with tkinter, matplotlib

数据库-mysql8.0

爬虫-python3.8 with requests, json

## 三、详细设计

### 1. 概述

本系统可用于爬取蛋卷基金、且慢基金网站上的投资组合信息，存储至本地数据库中。可以不断更新新获取的数据。同时能用图形化界面向用户展示基金的各方面信息，便于用户将不同的投资组合放在一起进行比较。

本系统基于python3.8，配合mysql8.0用作数据存储，实现本地爬取，存储，展示数据。

### 2. 术语表

|  |  |
| --- | --- |
| 历史净值 | 该基金某日的净值。 |
| 最大回撤 | 在选定周期内任一历史时点往后推，产品净值走到最低点时的收益率回撤幅度的最大值。 |
| 年化收益率 | 某一时间段内收益率，除以时间长度乘以360天，用来衡量区间内收益。 |
| 无风险化利率 | 指没有风险的理财方式所获得的资金收益比例。专业点说是对无信用风险和市场风险的资产的投资，指到期日期等于投资期的国债的利率。 |
| 年化波动率 | 一年里基金收益的波动程度。年化波动率=一年里日收益的标准差\* |
| 夏普率 | 它代表基民每多承担一分风险，可获得高出无风险回报率几分的回报。夏普率=(年化收益率-无风险利率)/年化波动率。 |

### 3. 设计概述

#### 系统接口设计

##### 用户与系统间交互

用户通过图形化界面与系统进行交互

电脑萤幕的截图

描述已自动生成

##### 系统内的交互

1. 主程序(main.py)直接调用爬虫模块(creeper.py)从网站上爬取数据。
2. 主程序(main.py) 通过数据库模块(databaseOP.py)中定义的函数初始化数据库。
3. 主程序(main.py)通过数据库模块(databaseOP.py)中定义的连接函数与数据库连接，将爬取的数据存储至数据库。
4. 图形界面(chart1.py)通过数据库模块(databaseOP.py)从数据库中获取数据后，将数据存储在内存中，根据用户指令对数据进行操作后再进行展示。
5. 图形界面(chart1.py)根据用户输入通过数据库模块(databaseOP.py)和爬虫模块(creeper.py)向数据库中添加新的投资组合。
6. 图形界面(chart1.py)通过数据库模块(databaseOP.py)和爬虫模块(creeper.py)更新数据库中的信息。

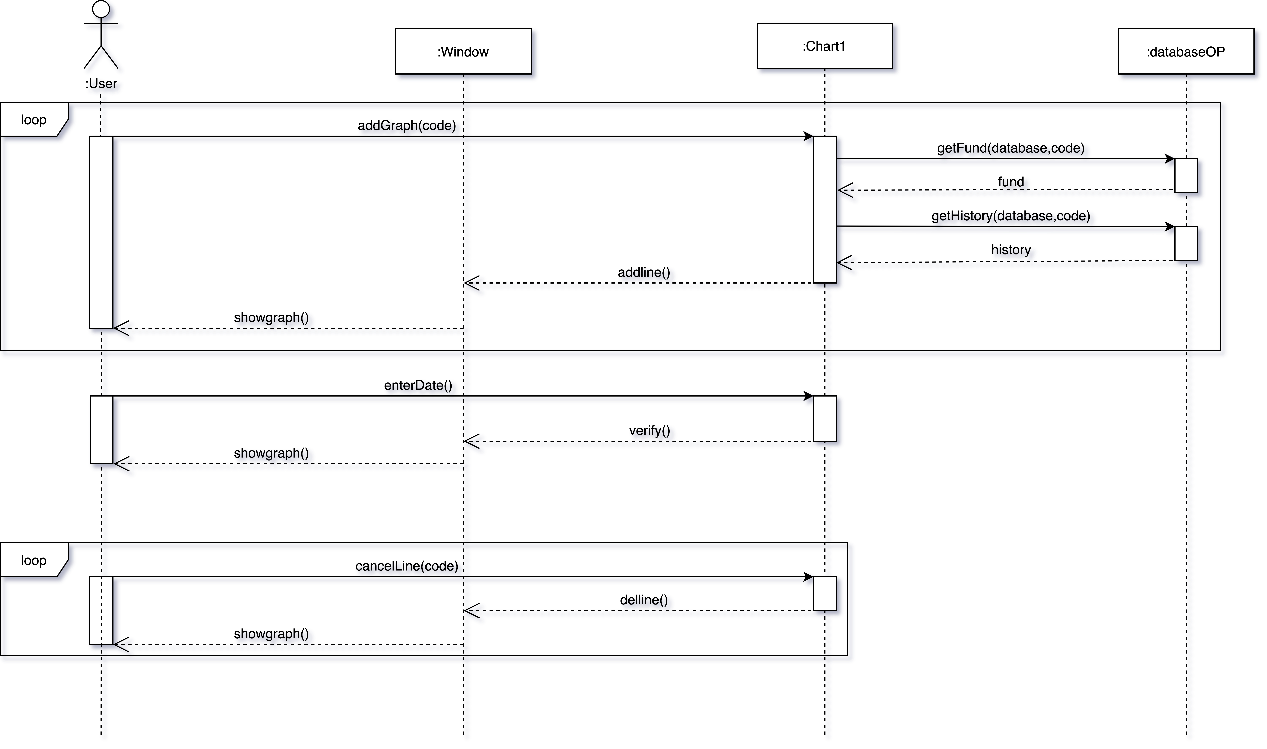
#### 对象模型设计

屏幕上有字

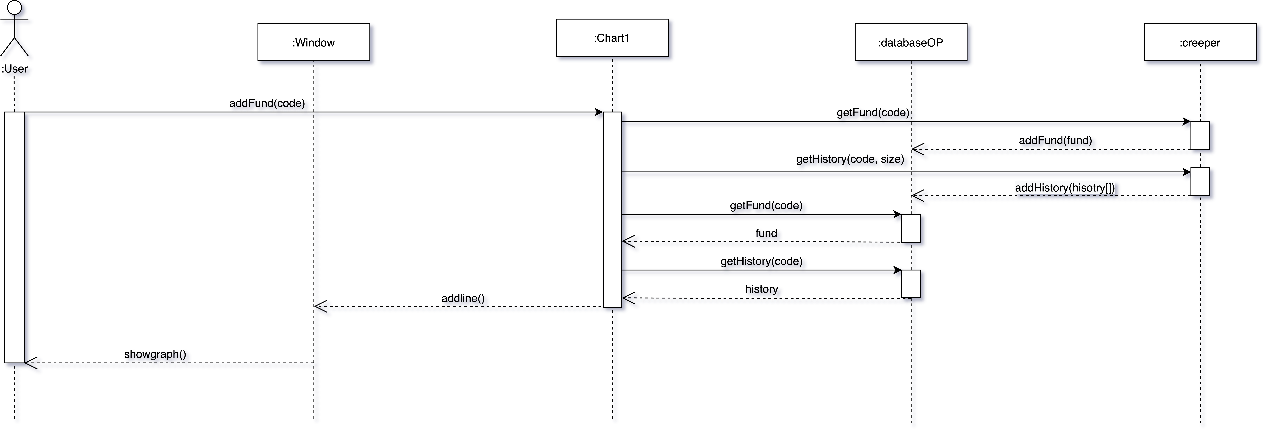
描述已自动生成

#### 系统用例实现详细设计

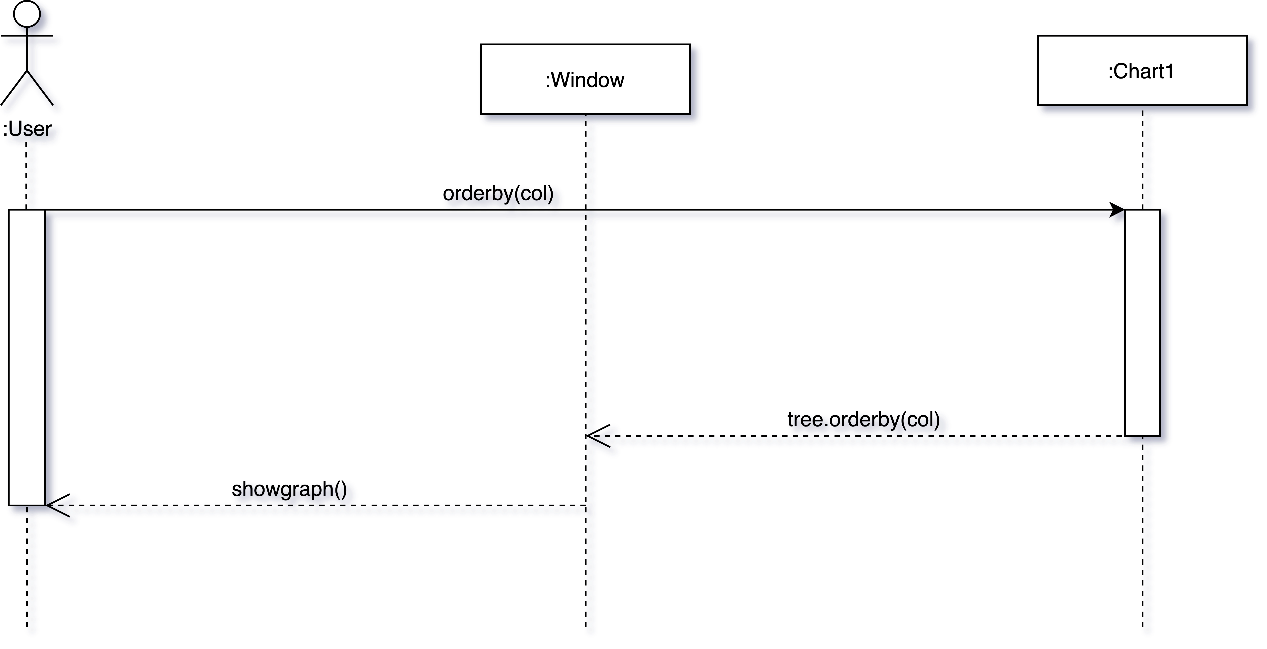
用例1 用户往主视图中添加或删除数据库已有的基金数据



用例2 用户往主视图中添加数据库中没有的基金数据



用例3 用户点击某一字段按该字段排序



用例4 用户对当前数据进行更新

图形用户界面

描述已自动生成

#### d)系统非功能设计

1、为了加快爬取速度，增加了多线程爬取。

2、为了加快数据的获取，在数据库中建立了索引。

3、针对数据库操作增加了异常处理。

## 四、软件测试

### 测试用例1

随机输入基金编号，点击确定

图形用户界面

描述已自动生成

跳出弹窗“找不到该基金”

图形用户界面, Teams

描述已自动生成

### 测试用例2

随机输入更新的x-sign值

图形用户界面, 应用程序, Teams

描述已自动生成

弹出x-sign有误

图形用户界面, 应用程序, Teams

描述已自动生成

### 测试用例3

输入错误的日期

电脑萤幕的截图

描述已自动生成

系统无应答

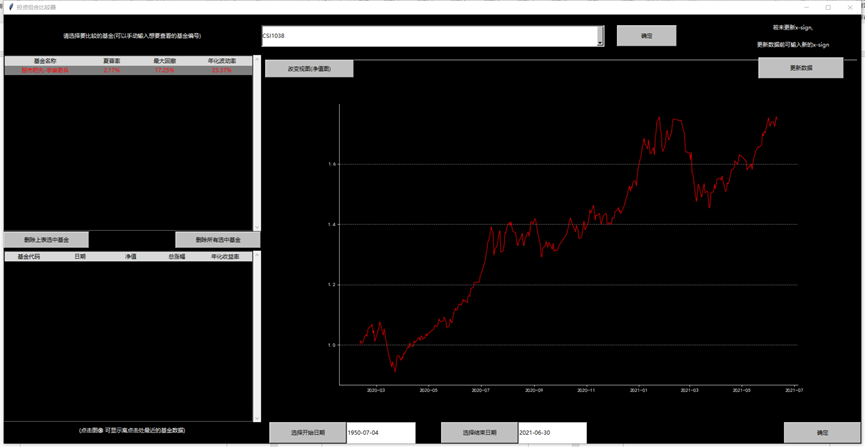
### 测试用例4

输入年代久远的日期

图形用户界面

描述已自动生成

系统正常判断但依旧显示当前基金最远记录日期



## 五、软件演示

### 用例1

选择基金

图形用户界面

描述已自动生成

点击确定添加

图形用户界面

描述已自动生成

按照上述步骤添加多条记录

电脑萤幕的截图

描述已自动生成

修改日期

电脑萤幕的截图

描述已自动生成

删除其中一条记录

图形用户界面

描述已自动生成

点击某一时间点展示信息

电脑萤幕的截图

描述已自动生成

### 用例2

在方框中输入想要查询的基金编号

图形用户界面, 应用程序

描述已自动生成

点击确定将其添加

图形用户界面

描述已自动生成

### 用例3

点击想要排序的字段

图形用户界面, 应用程序

描述已自动生成

将按该字段数值进行排序

电脑萤幕的截图

描述已自动生成

再次点击同一字段，改变排序方向

图形用户界面, 应用程序

描述已自动生成

### 用例4

点击更新数据按钮

图形用户界面, 应用程序

描述已自动生成

## 六、软件使用教程

### 安装要求

Python3.8 或更高版本。

Mysql8.0 或更高版本。

### 程序使用

1. 安装python3.8或更新版本，安装mysql8.0或更新版本。
2. 安装好reqirements.txt里的所有库并配置selenium库的webdriver至默认路径。
3. 将所有python文件放在同一文件夹下。
4. 运行main.py文件。
5. 选择是否更新数据，选择“是”后，等待数据更新完毕，弹出“数据更新完毕”之后，关闭此弹窗，出现用户界面，或者选择否，直接进入用户界面。

# 小组讨论纪要

|  |  |  |  |
| --- | --- | --- | --- |
| 第一次小组讨论 | | 时间 2021年5月20日 | |
| 参会人员 | 陈涵 | 梁永豪 | 廖宇延 |
| 1. 在Gitee上初始化仓库，填写项目基本信息。 2. 讨论爬虫的实现，且慢基金因为header的反爬校验暂时不能爬取。 3. 讨论前端实现运用的语言。 4. 确认后期工作任务进度。 5. 确认要实现功能的相应模块。 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 第二次小组讨论 | | 时间 2021年5月30日 | |
| 参会人员 | 陈涵 | 梁永豪 | 廖宇延 |
| 1. 讨论实现前端界面所用到的库。 2. 对前端样式进行初步设定。 3. 讨论前端所要展示的内容以及呈现样式。 4. 对爬取过程进行优化，能够只爬取数据库中没有的新数据。 | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| 第三次小组讨论 | | 时间 2021年6月15日 | |
| 参会人员 | 陈涵 | 梁永豪 | 廖宇延 |
| 1. 检查所有已实现的工作。 2. 确定所有必须功能已全部实现。 3. 对软件进行测试。 4. 整理报告以及相应文档。 | | | |

# 个人总结

|  |
| --- |
| 陈涵 |
| 通过本次数据库实训，我实践了从爬虫到运用python操作数据库等各方面内容，收获了小组开发项目的经验，提高了对软件工程的认识，加深了自己对关系型数据库的理解，增强了运用python语言和MySQL数据库的实践能力。 |

|  |
| --- |
| 梁永豪 |
| 这次实训用Python实现GUI界面，几乎整个过程都是边学边做，决定用tkinter库完成窗口，用matplotlib库实现基金可视化，treeview实现基金详细信息的表格显示，最终也都实现了所需功能，我学到了很多。 |

|  |
| --- |
| 廖宇延 |
| 通过本次实训，在两位组员的帮助下，我对python有了基本的认识，学会了如何用python去操作数据库，对于数据库在软件开发中的重要地位有了深刻的认识。也增强了自己实践的能力，虽然对整个项目的贡献不是很多，但还是收获了不少。 |

# 项目进度安排等项目管理内容

|  |  |
| --- | --- |
| 12周 | 完成爬取 |
| 13周 | 设计数据库和对象，完成数据的存储 |
| 14周 | 完成数据的计算和比较 |
| 15周 | 设计图形化展示界面 |
| 16周 | 优化设计 |

[项目仓库(https://gitee.com/concyclics/db\_big\_homework)](https://gitee.com/concyclics/db_big_homework)