

# A股金融证券 分析系统

Quantourist

## 迭代一项目总结文档

南京大学ASI

ASI of Nanjing University

2017-03-18

## 目录

一、	冯俊杰_151250037 .....	2
二、	董金玉_151250032 .....	3
三、	高源_151250040 .....	4
四、	龚尘淼_151250043 .....	5

# 一、 冯俊杰\_151250037

## 1. 个人贡献

- 1) 在项目启动时搭建基本框架，将项目搭载在 IDEA 上。
- 2) 在项目启动时定义部分接口。
- 3) 实现数据层 Dao 和 DataHelper，为数据层和 DataHelper 层撰写测试用例，保证每一个方法都在各种情况下被测到。
- 4) 负责处理数据，将数据进行适当变更后方便项目系统的读取。

## 2. 项目总结

- 1) 代码的注释规范应该尽早确定，在合适的需要的地方才写注释，不要到处写一些无用的注释。
- 2) 自己一直在后端，应该适当学一些前端的开发，以便在项目后端开发结束之后能够在前端发光发热。
- 3) 就算定义好了接口，也应该定期交流各自的开发进度，不然还是会出现理解上的偏差，造成无用的浪费。
- 4) 小组所有成员明显在工作日开发进度慢，周末开发进度快，故在下一次迭代中应该加强对周内的利用，并尽力增加周末的效率。
- 5) 多想想代码，学会重构，开发更高质量的代码。

## 二、 董金玉\_151250032

1：定下心来，踏踏实实：我学的是软件工程专业，自己学的只是一个基础，只是有些方面或许比别人走的快一步，所以一切都要靠自己.自己要定得心下来学习.成功需要耐得住寂寞，不求最快，但求最好.

2：课外学习：我们才只是开始阶段，需要不断去学习新的技术，新的领域，不断扩充自己。不要仅仅局限域书本和大作业需要的知识

3：遇到问题的时，自行解决问题的能力变强了，以前常常会想要询问别人，不想自己动手动脑

自身缺点：

1：沟通问题：自己的沟通能力只能算一般，因为对于某些事的阐释还是不怎么好，语言表达能力有点差，希望通过平时的交流和沟通来加强。

2：小组互审代码很必要，互相抽查核心代码对于我们这种在校的小开发团队来说很必要，尤其是没有自动化单元测试的情况下。

3：做事太过心急，心急反而做不了工作，还有可能影响他人的工作，从而影响整个工作的进度，我需要认真的反思

迭代二的计划：

1：向组员学习，学习代码，学习技术，努力开发高质量代码

2：提醒大家进行代码复审工作，提高代码质量

3：调正自己，努力做到做事平稳，按部就班，不心急

### 三、 高源\_151250040

在本次迭代一中，虽然时间很短，但是却感受到了软件的魅力，上学期一学期的内容竟然在短短的两周时间内重新做了一遍，我深切体会到了几点。

在组里我负责的是界面模块，在大腿队友的指导下，软件有条不紊的进行的，同时也让我认识到，我应该弥补自己后端的能力，要向其它方面拓展学习。定下心来，踏踏实实，自己学的只是一个基础，只是有些方面或许比别人走的快一步，所以一切都要靠自己。自己要定得心下来学习，不求最快，但求认真。团队合作方面我们做的很好，自己完成自己的任务，对 git 的使用也慢慢熟悉起来，同时我觉得应该多学习一些同组人的优势方面，多向别人请教问题，多听听他人意见，会犯更少错误，会更长见识，做事才更有效。开发代码需要用全身心，每一个提交到 git 的代码一定要认真核实并加上注释，而我却因为某些事情敷衍和消极开发，结果就是现在越来越难挽回。界面层次写的有点乱，多亏同组人的帮助，帮我重构了界面层的代码。

存在一些，自身缺点，沟通问题：自己的沟通能力只能算一般，因为对于某些事的阐释还是不怎么好，语言表达能力有点差，希望通过平时的交流和沟通来加强。心态问题：自己对于做某些事过于着急，一心想急切完成，确反而误时，希望以后时刻注意。

开发团队的互相协作和沟通是必须的，决不能因为接口的确定就互相不沟通了，结对编程的重要类似于此。

迭代二应该积极完成界面工作，同时完成部分后端任务，力争在软件开发能力和水平上一个新台阶，多吸取组员的优点，集中精力把软件做好。

## 四、 龚尘淼\_151250043

### 1. 个人贡献

- a) 在项目启动的时候负责编写需求规格文档，定义好我负责的模块的街口
- b) 编写负责模块的数据层和逻辑层的代码，一边开发一边编写测试用例
- c) 在写好负责的模块的代码后，检查界面层代码，并进行重构

### 2. 项目总结

- a) 开发代码需要用全身心，每一个提交到 Git 的代码一定要认真核实并加上注释。
- b) 自动化单元测试很重要，可能之前会浪费一些时间，可对于软件开发的成熟来说十分必要，磨刀不误砍柴工。
- c) 小组互审代码很必要，互相抽查核心代码对于我们这种在校的小开发团队来说很必要，尤其是没有自动化单元测试的情况下。
- d) 永远不要怀疑重构。可能短期内会出现问题，但是长远考虑换回的是代码的精简和设计的清晰。
- e) 开发团队的互相协作和沟通是必须的，决不能因为接口的确定就互相不沟通了，结对编程的重要类似于此，旁观者清。