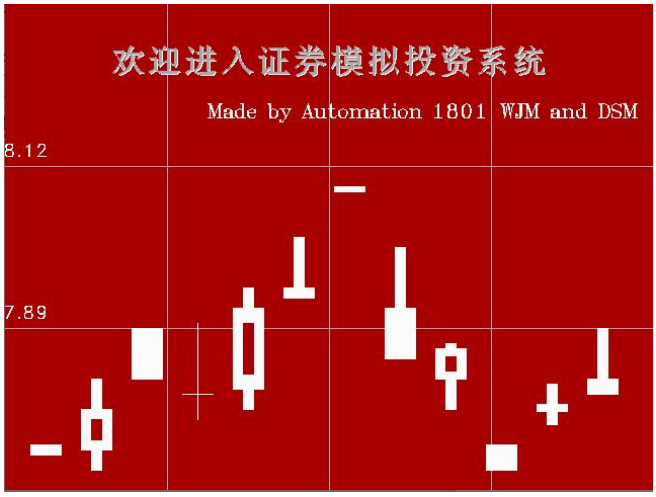
证券模拟投资系统

需求分析报告

C 语言课程设计



专业班级：自动化 1801
小组成员：魏靖旻 U201814224
邓述民 U201814206
指导老师：周纯杰、何顶新、彭刚、周凯波、桑农、左峥嵘、高常鑫、汪国有、陈忠。
上交时间：2019 年 9 月 29 日

目录

1、前言

编写背景：

2019 是新中国成立 70 周年，在这 70 年间，祖国母亲走过了风风雨雨。而国家的发展和民族的振兴，经济的发展无疑占据主要地位。

而在经济发展中，证券行业无疑也充当着重要的角色。相比欧美的成熟市场，中国证券市场确实是一个年轻的行业。从改革开放到现在，证券的发展历经 30 多年的风雨，从年轻慢慢走向成熟。随着中国经济的发展和进一步的对外开放，中国证券市场逐步与国际接轨，A 股相继纳入 MSCI、富时罗素等新兴指数，我国的证券行业的发展仍值得瞩目。对大多数人来说，证券相关知识并不简单，它不仅需要我们学习有关经济金融方面的专业知识，也需要我们有丰富的实战经验，对市场和企业有个人的认识。证券的快速发展正如我们国家的发展，在发展中既有机遇，也有挑战。

因此，编者编写了这个证券模拟投资系统，旨在帮助没有投资经验或者对于投资证券行业有一定想法，对于金融知识有兴趣的人。对于我们的用户而言，可以像真的证券软件一样进行基本面技术面的学习，学习布林线 MACD 等指标，学习专业知识；也可以模拟交易，在其中获得投资的乐趣，也体验从亏损到盈利的过程；甚至可以看到多只股的近日行情，从而对数据进行分析，逐步成长为一个稳健的投资者。

编写目的：

通过“证券模拟投资系统”对证券需求者和使用者的调查和分析，对同花顺，益盟和通信达等证券软件的使用，对一些技术指标的学习和网络资料查询，我们编写出这一份软件需求分析和功能设计报告。

该项报告对于整个“证券模拟投资系统”进行了全面的用户需求和功能分析。包括可行性分析，需求分析，系统功能设计，代码实现，软件亮点和不足，集成测试等等。本报告明确了本软件系统架构设计，软件结构与数据结构设计，各模块之间的接口和调用，系统界面设计，系统功能设计（函数罗列），具体算法设计以及整个软件的源代码。

同时，该项报告也明确了两位开发者的设计与分工，增强了后期测试人员对于软件的调试和验收的可读性与可修改性。

本报告的预期受众为证券从业人员，证券投资者，项目委托方，软件开发与测试人员，工程管理人员。

参考资料：

1. 王士元. C 高级实用程序设计. 北京: 清华大学出版社. 1996 年

2. 周纯杰，何顶新等. 程序设计与应用（用 C/C++编程）. 北京: 机械工业出版社. 2008 年

3. [美] Prata. C Primer Plus（第六版）北京：人民邮电出版社. 2016 年

4. 严蔚敏，吴伟民编著. 数据结构（C 语言版）北京：清华大学出版社. 2018 年

参考软件：

5. 益盟财经软件与同花顺证券软件.

6. 数据来源：通达信.

编者的话：

一开始选择编写这个证券模拟投资系统其实是有原因的。确实我们考虑过做模拟军演或者小区智能停车系统，但是最后我们仍然选择了证券。

目前中美贸易战仍在继续，面对国内外市场竞争日益激烈，迫使许多企业不断进行技术改造和设备更新。我们不仅看到了市场和政府之间密不可分的关系，也看到了经济对于一个国家的未来，一个民族的发展的重要作用。这也让我们对于我国经济的发展和其中的本质有了想要探索的欲望。所以我们选择了证券模拟。

股票是世界上最赚钱的金融投资产品。可是大多数普通投资者为什么赚不到钱？一是投资理念的错误，二是不具备选择优秀公司的能力，三是缺乏耐心，带有赌博性质。市场是位让人难以捉摸的小机灵鬼，我们也想通过编写这个模拟软件更好地了解市场，通过用户对股票的选择更好地了解到人们在做选择方面所考虑的各方因素...

2、任务概述

目标功能：

该证券模拟投资系统可实现证券投资软件（如同花顺等）的部分基本功能。在欢迎界面点击之后，系统用户可以在最初界面选择用户或者游客实现登陆、注册、与忘记密码修改密码的操作。

1、游客能了解股票的相关概念，可以像真的证券软件一样进行基本面技术面的学习，学习布林线 MACD 等指标，学习专业知识。浏览市场今日以及近日的行情指数、大盘数据、通过按键来自行选择股票、浏览不同指标的股票排行、可以根据基本面、技术面选股，以及根据对股票的不同需求来进行快速选股、智能选股，同时可以在观察股票的时候进行登录。

2、登陆的用户能在游客的基础上进行开户、充值金额、查询已选择的股票、修改银行密码、根据需求进行买入卖出、撤单、持仓等操作。

3、管理员可使用该系统进行证券后台数据维护，冻结违规用户与解冻。

该证券模拟投资系统旨在使其实现证券交易功能模拟买卖股票、股票信息查询、股票数据图形化显示、股价预测等功能。来给用户、游客与管理员带来便利。

编写规范：

1. 命名规范：

变量命名，涉及用户以及股票信息的，应该尽量用英文表达其准确定义。其他类型变量名应给出详细注释以说明其主要功能。

函数命名应该用英文表达其确定含义

文件命名都用小写，并且表达出该文件所包含函数的主要功能。

涉及数据结构的命名应参考数据结构（C 语言版），并进行适当修改。比如说沪深界面使用队列命名时可以用 stoqueue。

2. 注释:

函数功能都要在函数原型后注明。

部分令测试者比较难以理解的算法和流程应该给出相应的注释。

3、运行环境和配置

1、硬件接口

处理器：Intel Pentium 166 MX 或以上。

硬盘：空间 500MB 以上。

屏幕适配器：VGA 接口。

系统运行内存：要求 32MB 以上。

2、软件接口

开发软件工具：Borland C 3.0。

文字编辑工具：Notepad++、Visual Studio 2019 Community

数据库：文本存储（记事本）或 MySQL。

操作系统：DOS WINDOWS 9X/ME/2000/XP/WINDOWS 7。

3、控制

该系统通过鼠标与键盘直接进行控制。用户将鼠标移至需要操作的功能区进行点击，或者按 F1~F6 键来显示登陆后的不同界面，同时通过键盘来完成登陆、注册的输入功能。操作完毕后下拉菜单点击相应位置退出系统。通过中断技术来获取鼠标的位置与键盘的输入功能。

4、需求分析与系统设计

根据我们对于证券软件用户的需求分析，以及参考同花顺，益盟炒股，通信达等证券软件，我们设置了如下功能。

根据用户对于股票查看与排序的功能，我们设计了沪深界面。

根据证券软件在行情方面的设计，我们设计了行情界面，罗列 K 线图以及各种技术指标。

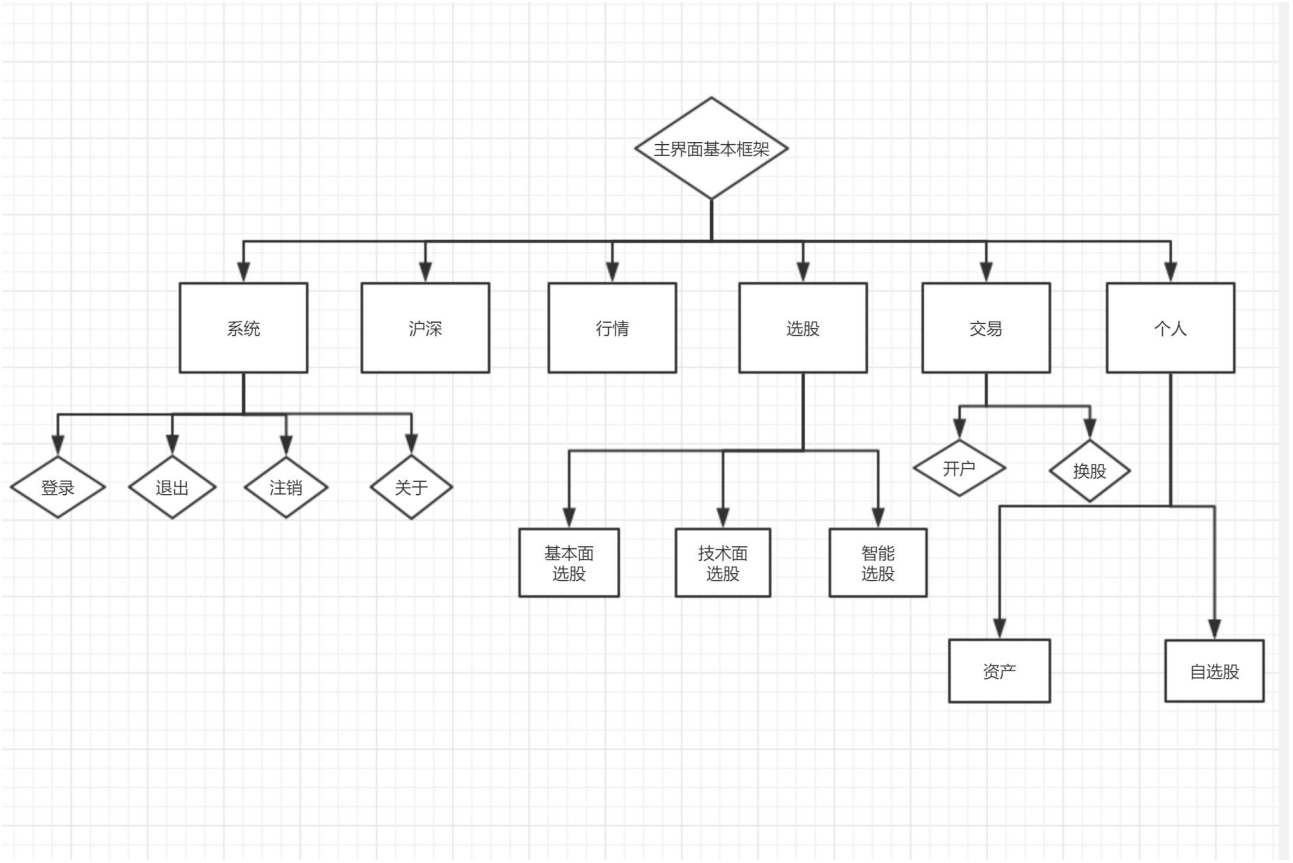
根据用户在选股方面的需求，我们设计了三大选股方式。基本面，技术面和智能选股。

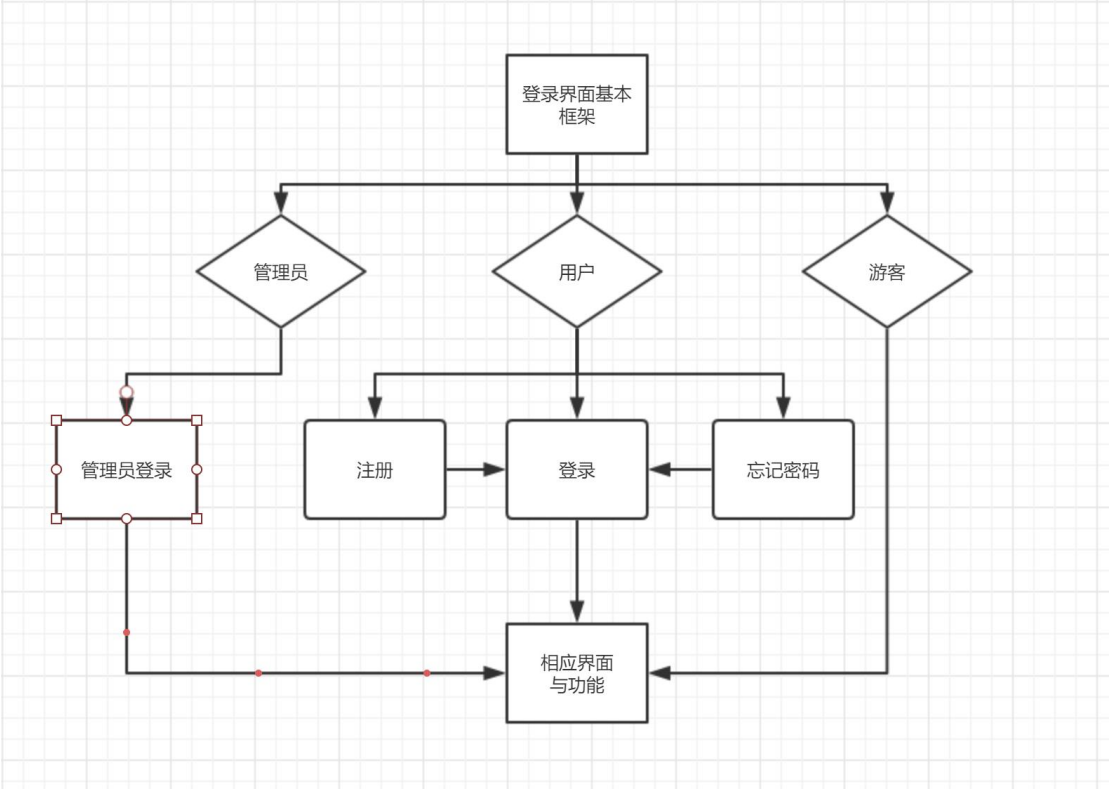
我们也根据股票规则设计了交易界面和资产界面。

为了方便管理与股票浏览，我们也设计了管理员和游客登录界面。

同时为了方便用户使用，以及提高软件安全性和可用性，我们也设计了登入登出，注销，资产限制，随机验证码，智能选股等一些功能。

基本框架：





沪深界面：

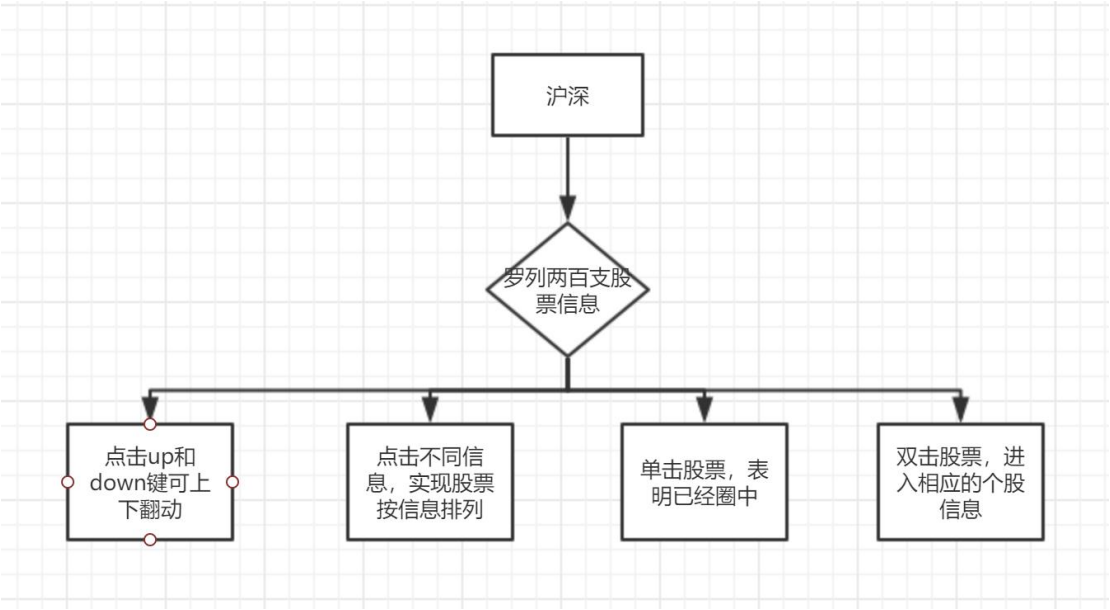
a.队列列表显示所有的股票，涨股标红，跌股标绿，通过按键和鼠标选择来决定选择查看某只股的详情，以方便用户查看相应沪深 A 股数据。

b.列表中包括股票代码、名称、涨幅、现价、涨跌、买价、卖价、总量、现量、涨速、换手、今开等多条股价信息。

c.通过点击列表中的不同信息，即可完成按该信息升序或降序的排列显示。以方便用户对股票进行简单的筛选。

d.单击股票，该行出现绿线，表示正选定的股票。

e.双击股票，进入个股，详细显示 K 线、5 天成交量均线，10 天成交量均线、以及其他股票详细信息方便用户了解选择的股票。



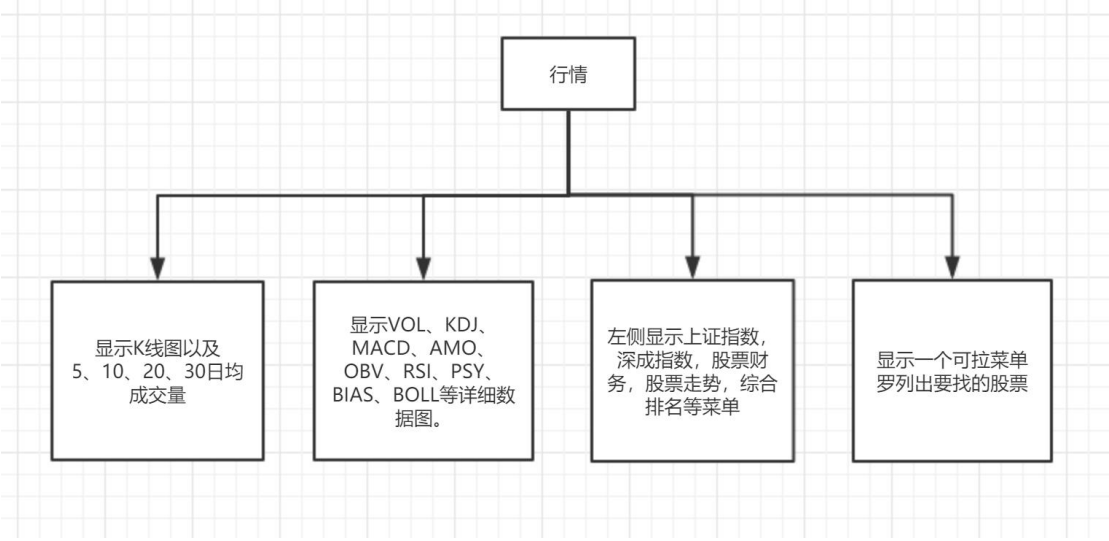
行情界面：

a.显示 K 线图以及 5、10、20、30 日均成交量。

b.根据技术面选股的关键因素以及用户对选股方式的需求，显示 VOL、KDJ、MACD、AMO、OBV、RSI、PSY、BIAS、BOLL 等详细数据图。

c.左侧显示上证指数，深成指数，股票财务，股票走势，综合排名等菜单。

d.左下方显示一个可拉菜单罗列出要找的股票。

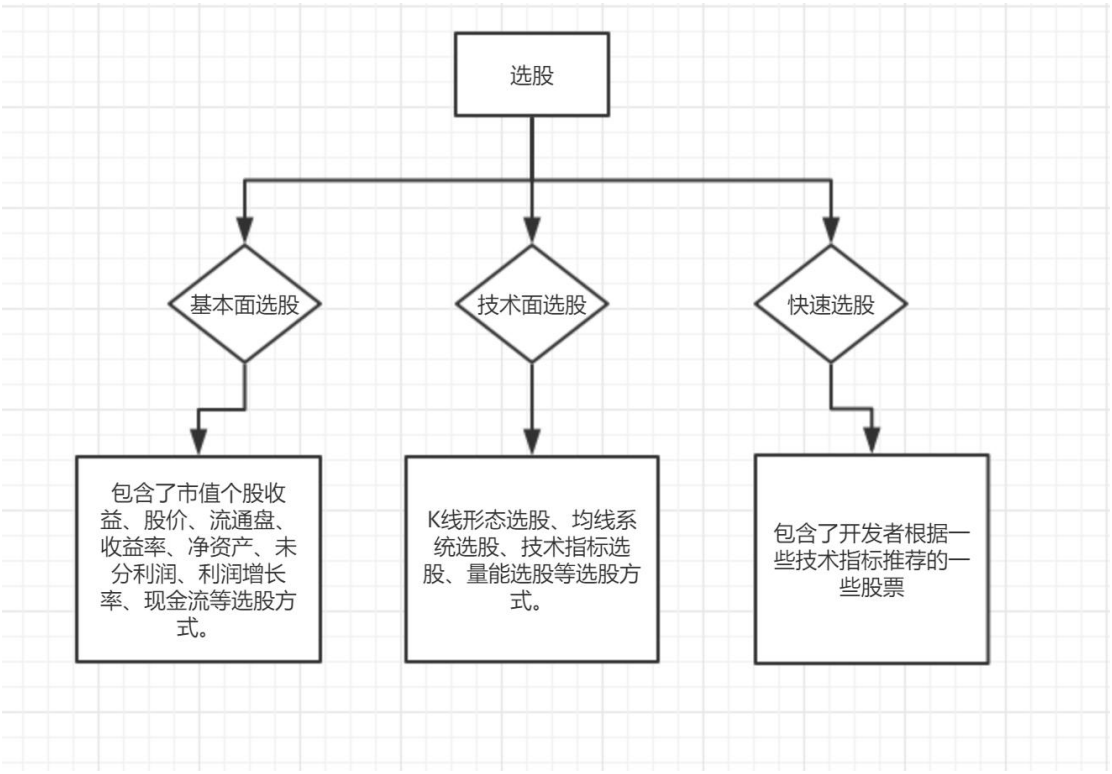


选股：

a.基本面选股包含了市值个股收益、股价、流通盘、收益率、净资产、未分利润、利润增长率、现金流等选股方式。

b.技术面包含了 K 线形态选股、均线系统选股、技术指标选股、量能选股等选股方式。

c.快速选股则包含了开发者根据一些技术指标推荐的一些股票。



交易：

A.开户：

a.输入银行卡号、银行卡密码、银行卡确认密码全部输入完全后才可开户。

b.完成注册则跳转回资产界面，若未填写完全，或相应内容不合规范则会有报错提示，待修改后方可开户成功，若点右下角返回按钮，则直接跳转回资产界面。

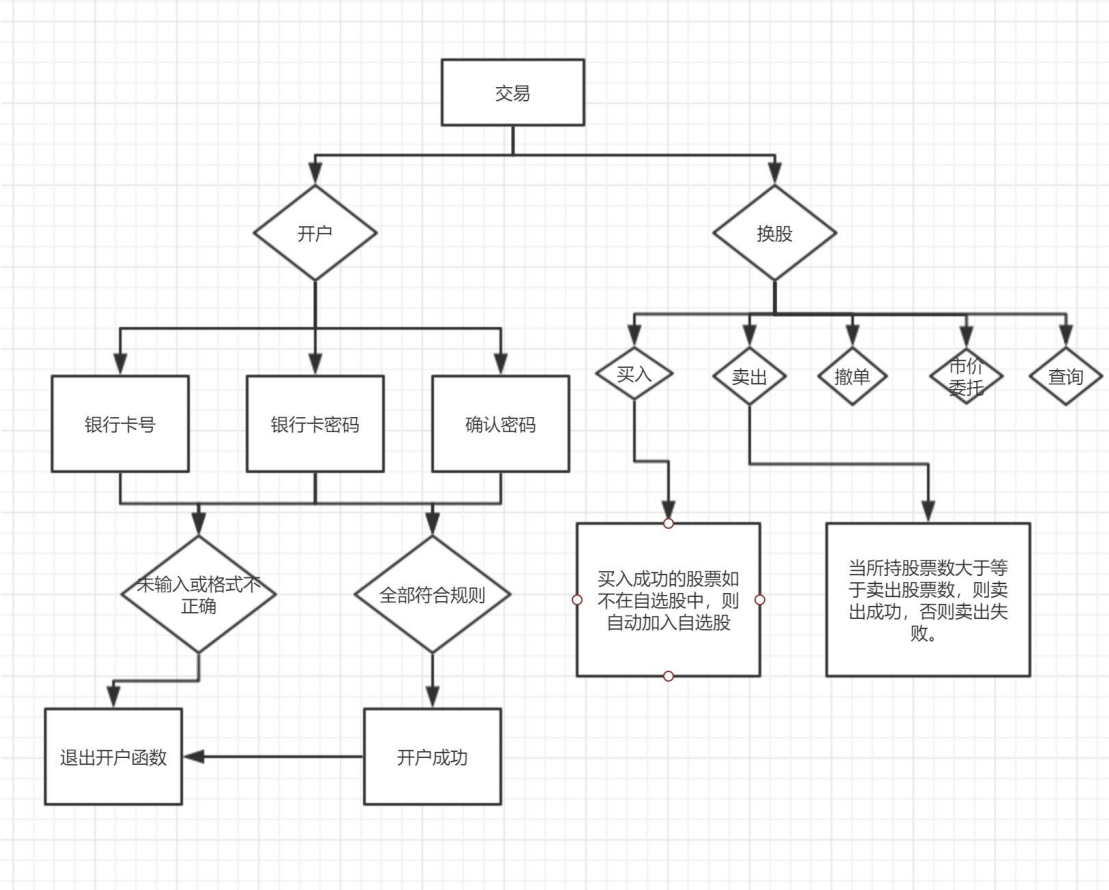
c.点击左下角重置界面即可将所有已输入归零开始重新输入。

B.换股：

a.包含了买入、卖出、撤单、市价委托、查询功能。

b.买入成功的股票如不在自选股中，则自动加入自选股。

c.当所持股票数大于等于卖出股票数，则卖出成功，否则卖出失败。



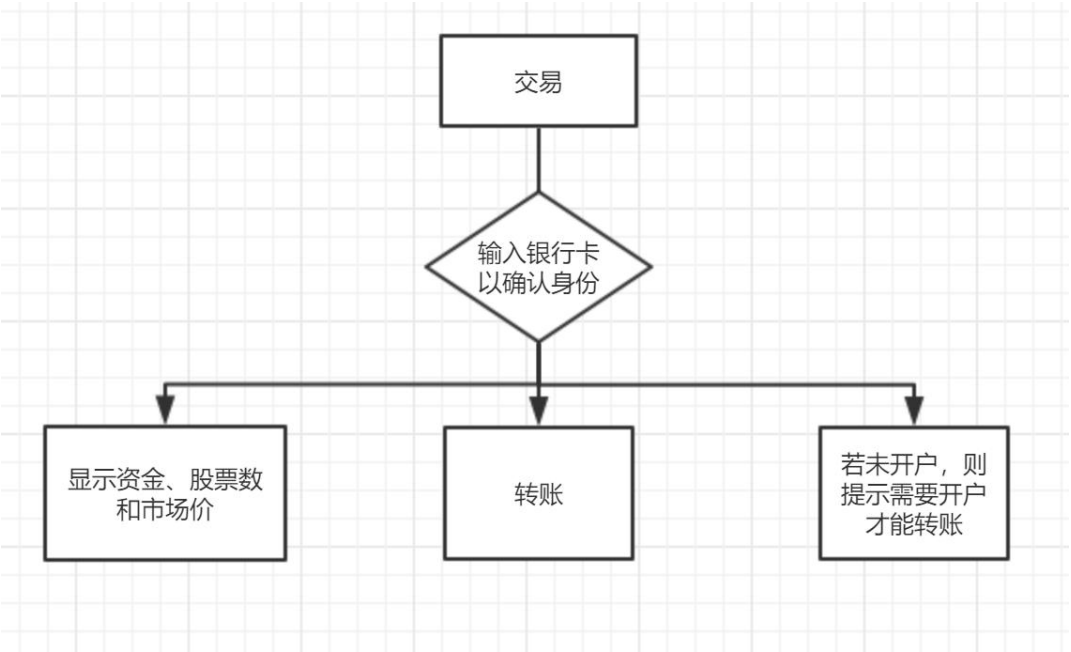
资产：

a.用户点击资产时需要输入银行卡密码确认身份，并且可以点击忘记密码来修改密码。

b.显示目前所拥有的资金及股票相应股票数和市场价。

c.用户可以选择从银行转账进入证券账户。

d.初始用户登录时，并没有开户，则会提示用户开户，待用户开户后，才可进行资产界面的操作。

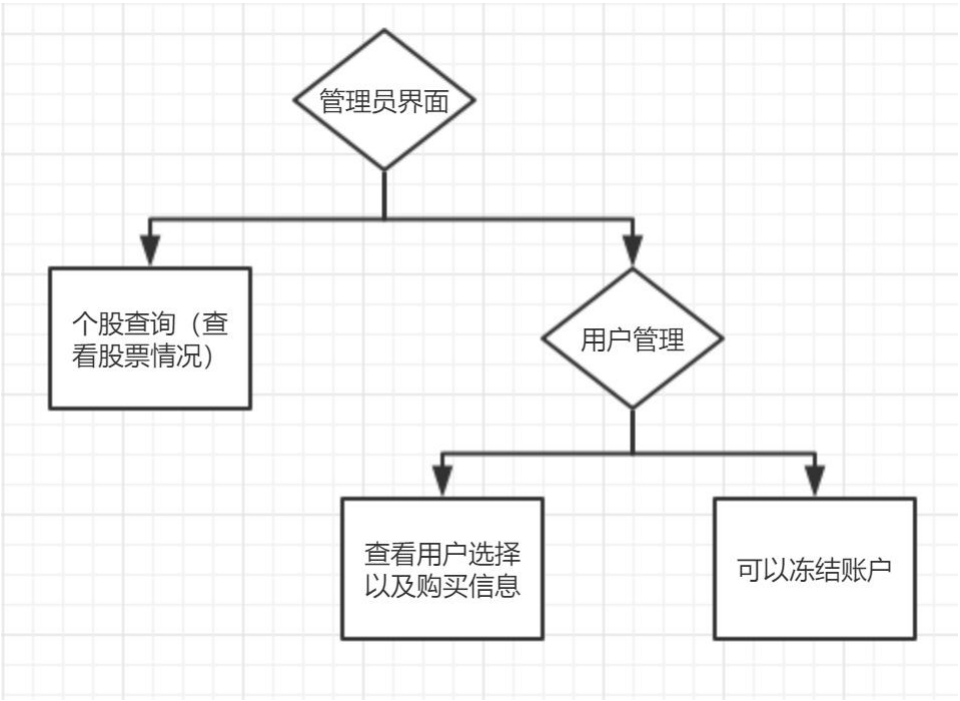


管理员界面：

a.个股查询可观察每支个股的详细情况。

b.用户管理可观察用户的股票购买情况。

c.用户管理可以冻结账户（停户）

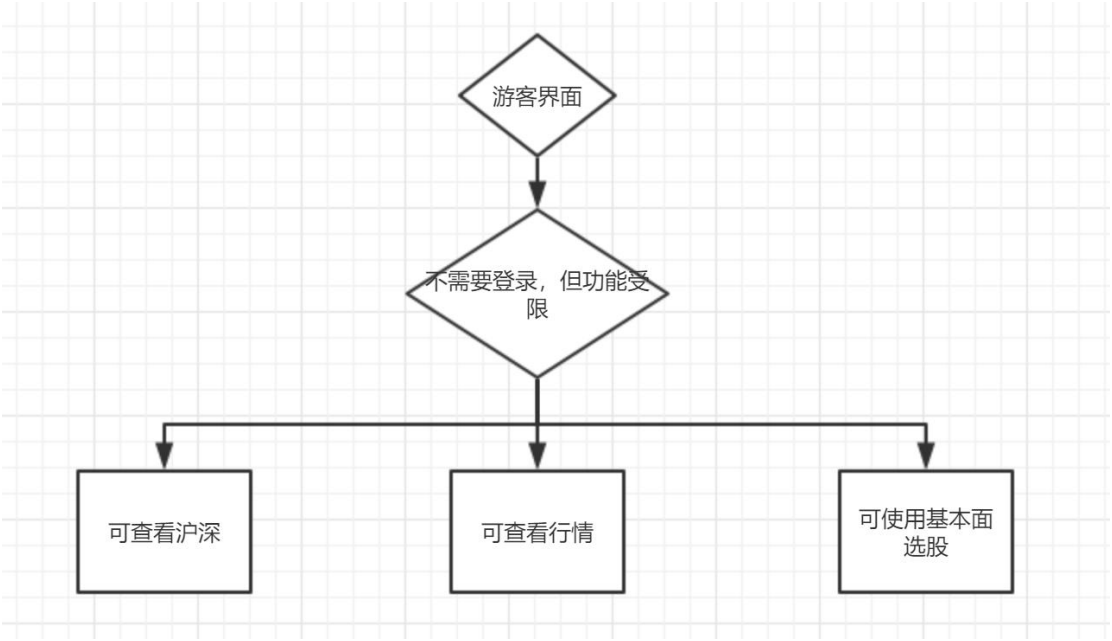


游客界面：

功能限制：

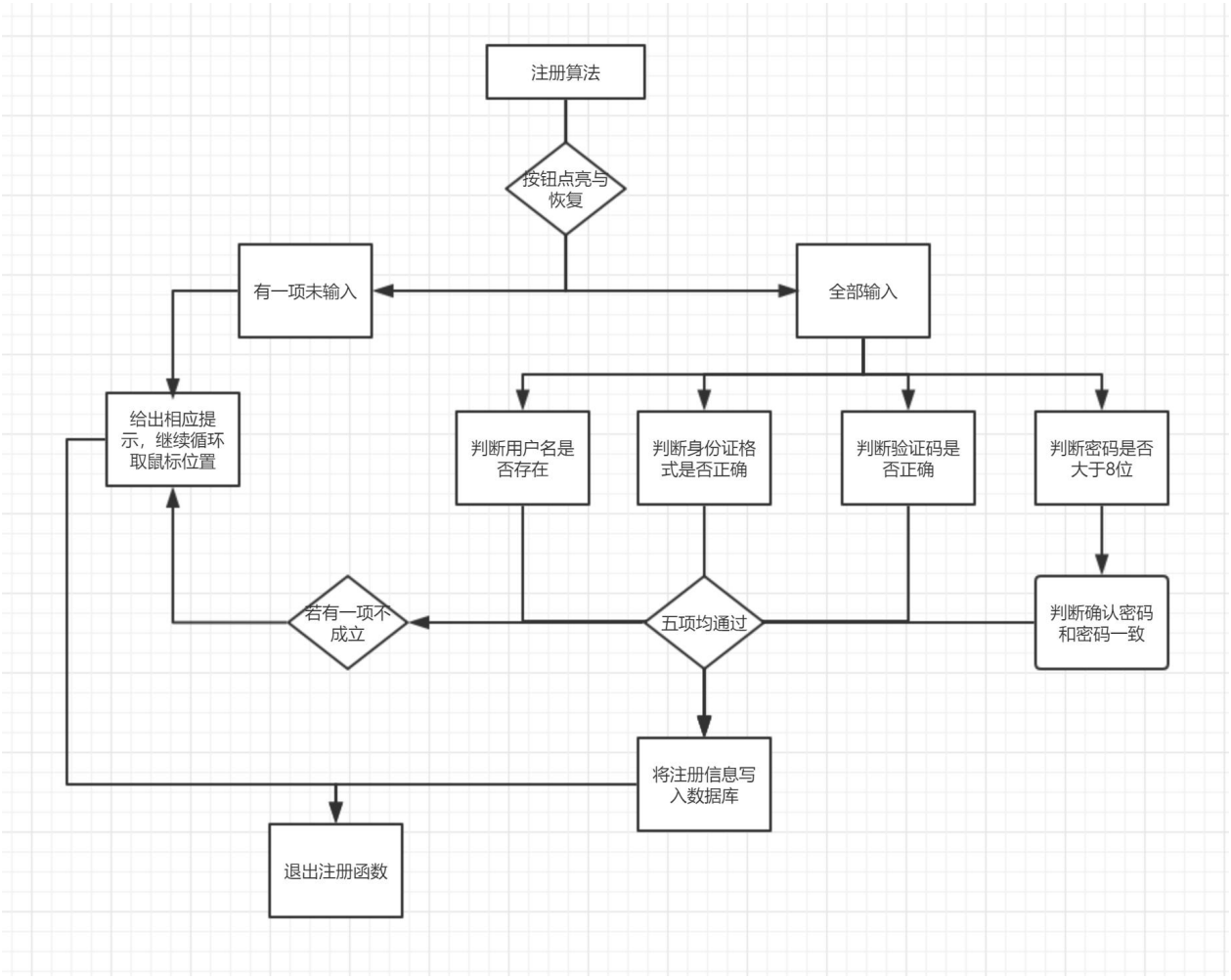
根据部分证券软件的游客功能，游客不需要注册，但是功能受到限制。

游客只能游览沪深，行情和基本面选股的界面以及功能。其他功能必须要登录后才能使用。



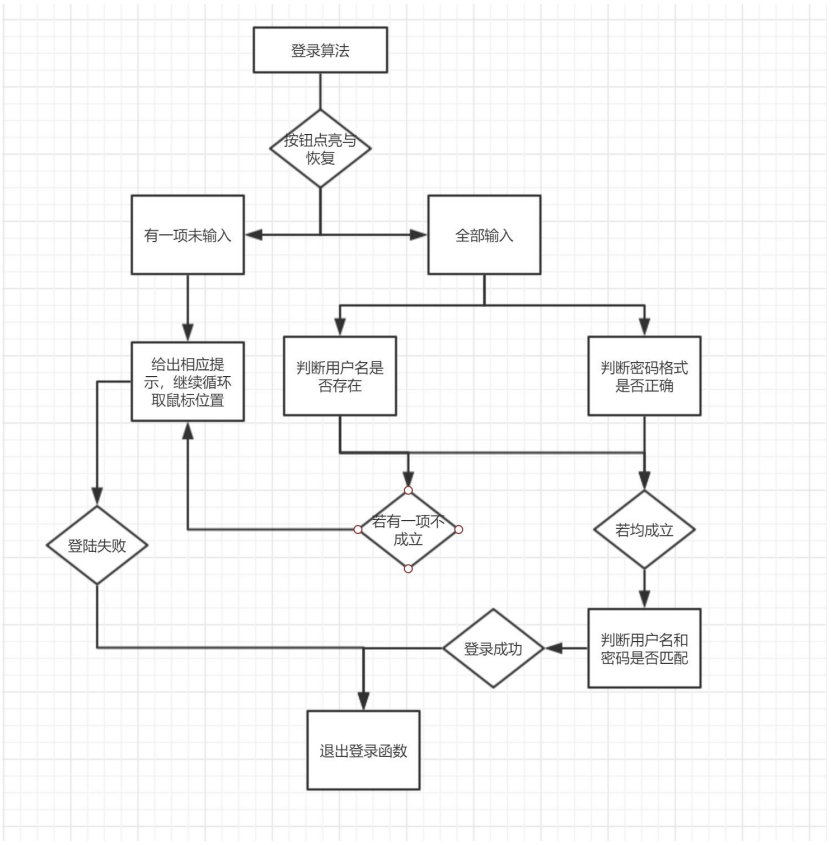
注册:

在注册中，为了保障用户信息的安全性，防止刷机等情况发生，我们设计了随机生成的验证码。同时我们设立了一套完整的判断机制，全部满足才能把注册信息写入数据库。这样也增加了注册的安全性。



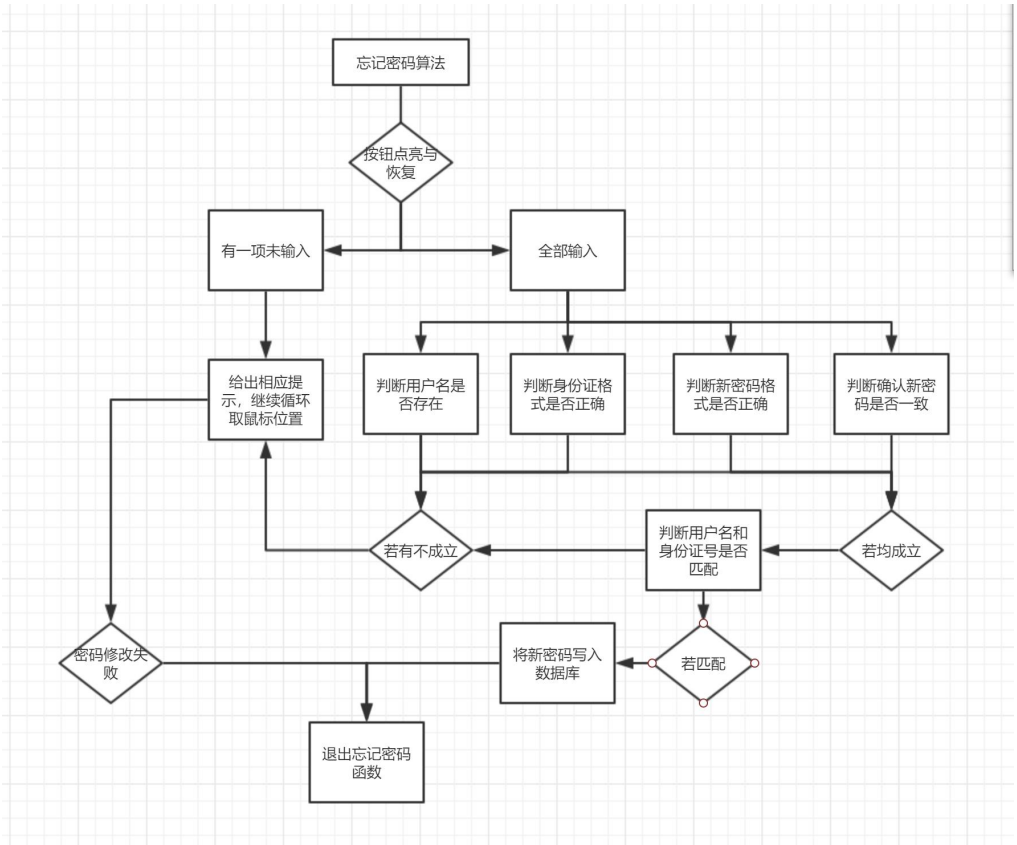
登录：

我们参考了两大证券软件的登录界面绘制了我们的登录界面，也给出了相应的用户提示信息。



忘记密码：

忘记密码与登录类似，只有满足了全部的信息判断，密码才会被成功修改。

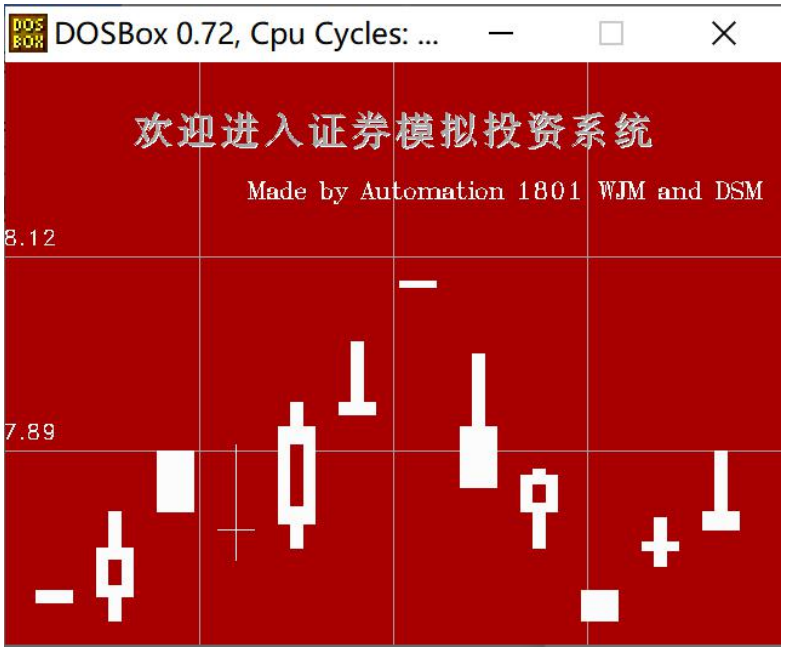


5、界面设计

鼠标设计：

我们设计了 MouseS 这一变量，为 0 时为默认值，为 1 时为手型，为 2 时为光标，以提供给用户最好的使用反馈。

界面设计（目前已经完成部分）：



欢迎界面：

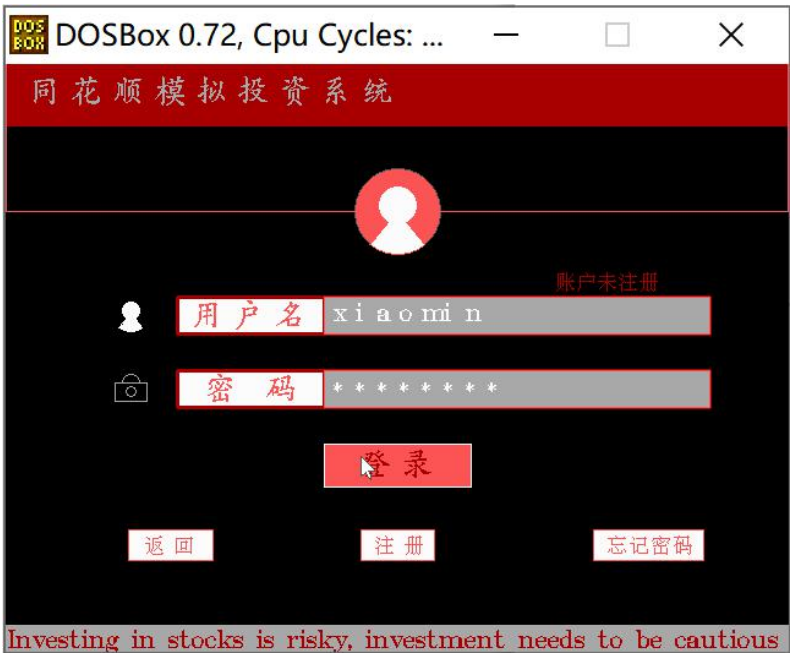
灵感来源于百度证券某图片，采用证券的关键特征 K 线，列出包括锤线、T 线、十字线、一字涨停板等关键的 K 线特征。

选择界面：

进入欢迎界面后就是我们的登录账户选择界面，分为用户、管理员和游客。

我们也设置了按钮的点亮算法，点击右上角 X 按钮可以直接退出程序。

同时在每个界面中为了追求更好的用户体验和反馈，我们优化了鼠标。包括但不仅限于按钮会变成手型，输入会变成 I 型。



登录界面：

除了基本的返回注册和忘记密码等，我们也优化了界面。包括会提示“未输入”，“账号未注册”，“密码错误”，“登录成功”等等，以追求给用户最完美的反馈。

同时我们在登录下方也给出了提示。证券有风险，投资需谨慎。

具体算法会在下面提到，此处不再赘述。

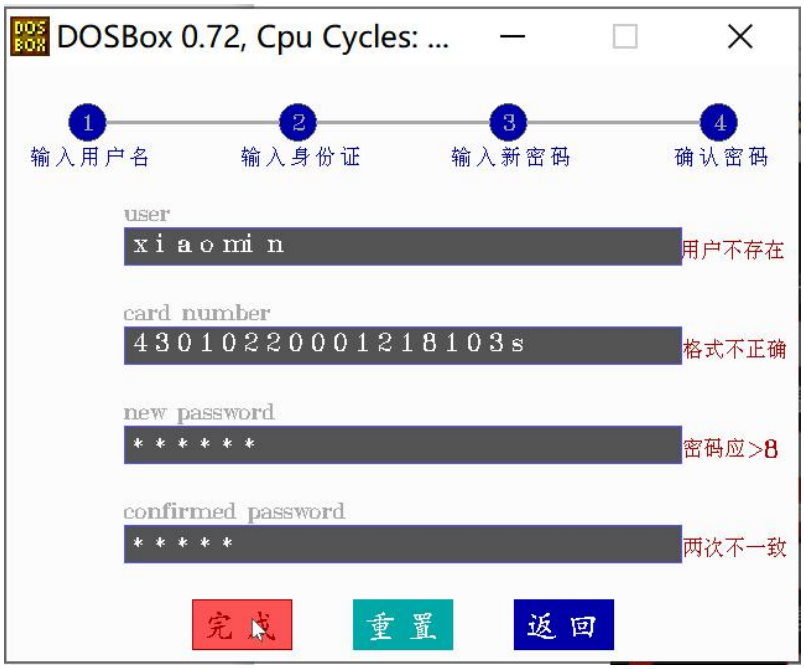
用户注册界面：

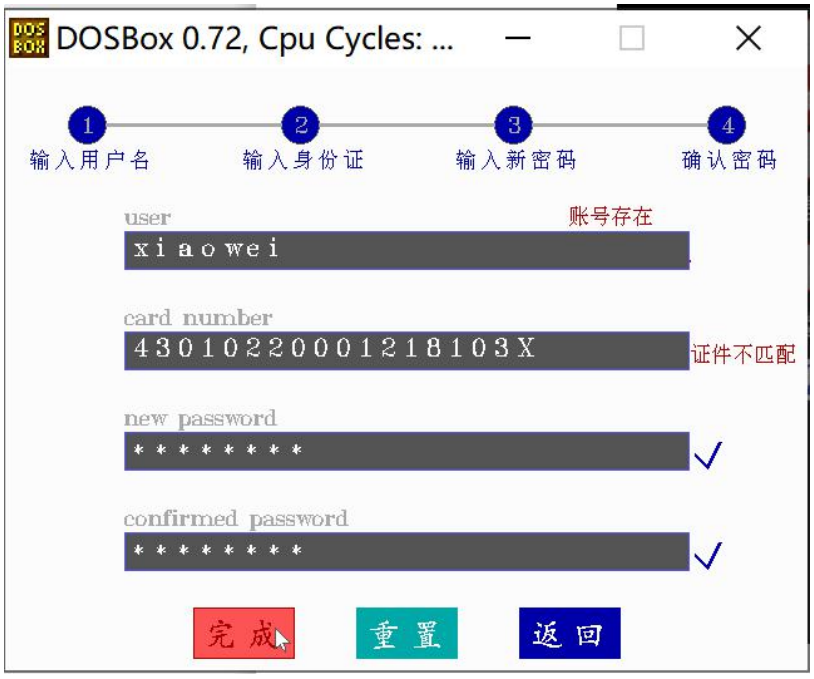
和登录一样，为了追求完美的用户反馈，我们给出了丰富的提示信息，告诉用户哪项必须满足，哪项信息输入有误。

在界面设计方面，我们采用数组和 fillpoly 画出了之后的涨跌箭头。也比较符合我们证券的一些特征。

我们也有很多亮点，比如说设置了重置按钮，可以让用户重新输入所有信息，方便修改。我们也设置了验证码点击切换的功能（这点会在算法部分给出详细信息），以防止刷机等其他问题。

具体算法会在下面提到，此处不再赘述。

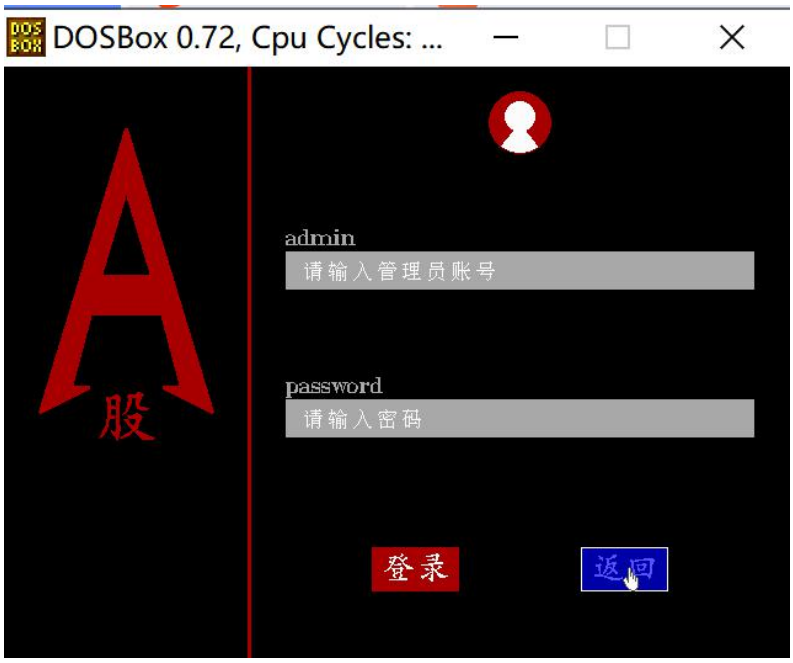


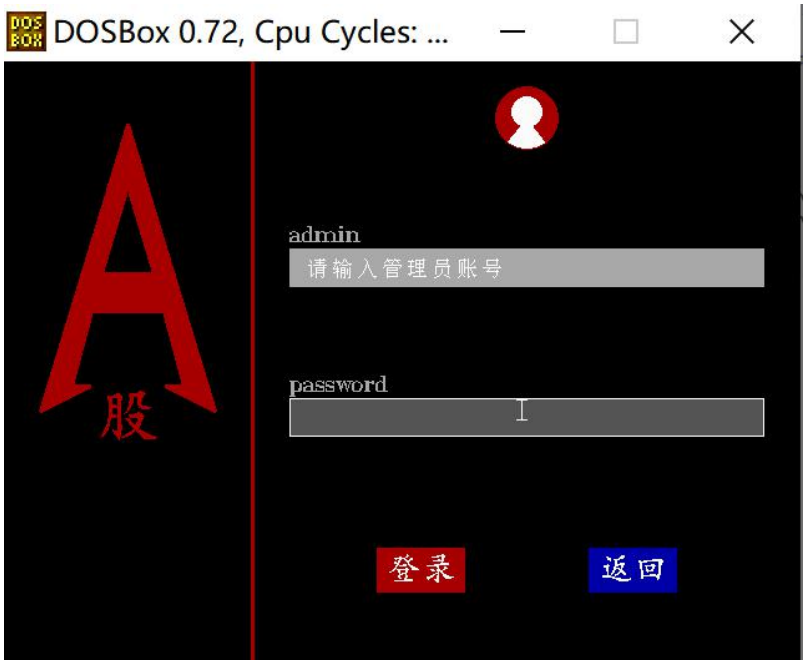


忘记密码界面：

和注册登录一样，我们也设置了丰富的用户反馈。并以流程图的方式让用户明白忘记密码的流程修改，我们也设置了重置按钮方便修改用户信息。

具体算法会在下面提到，此处不再赘述。



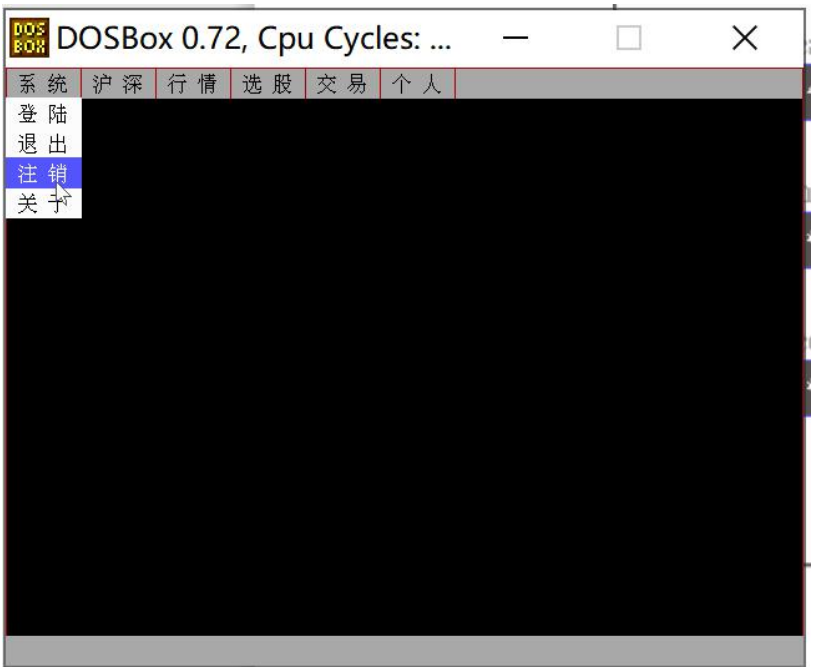


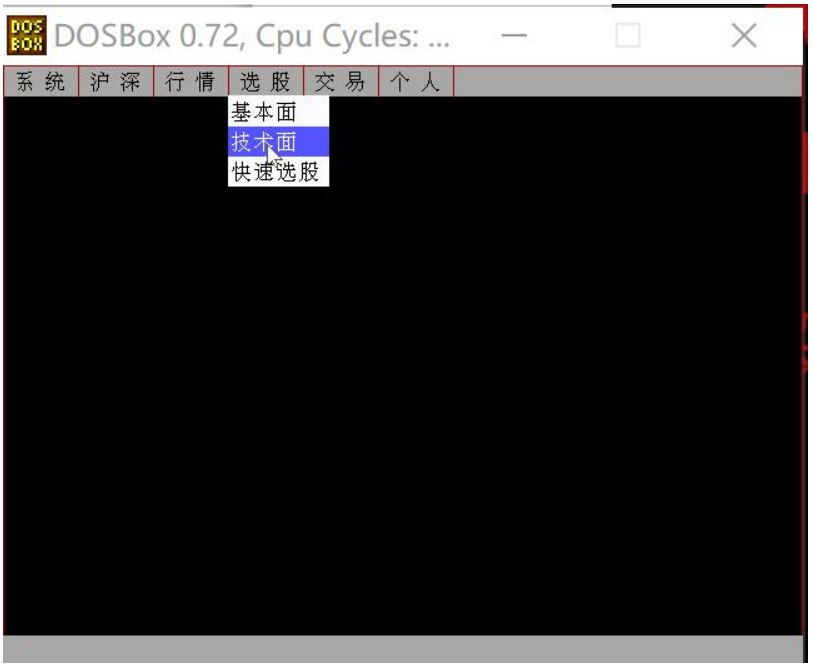
管理员登录界面：

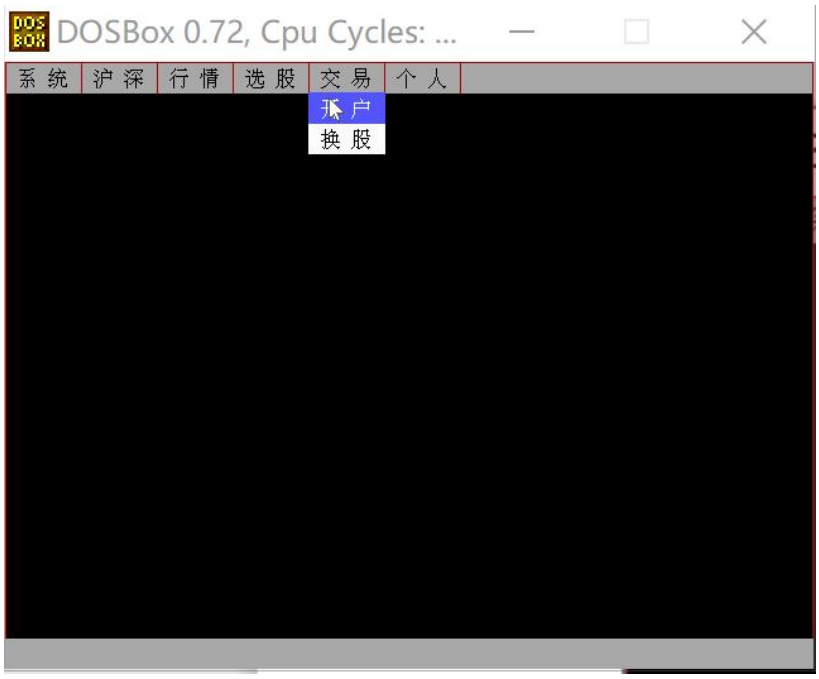
目前我们还没有录入管理员的身份信息，也还没有做出管理员的管理界面。

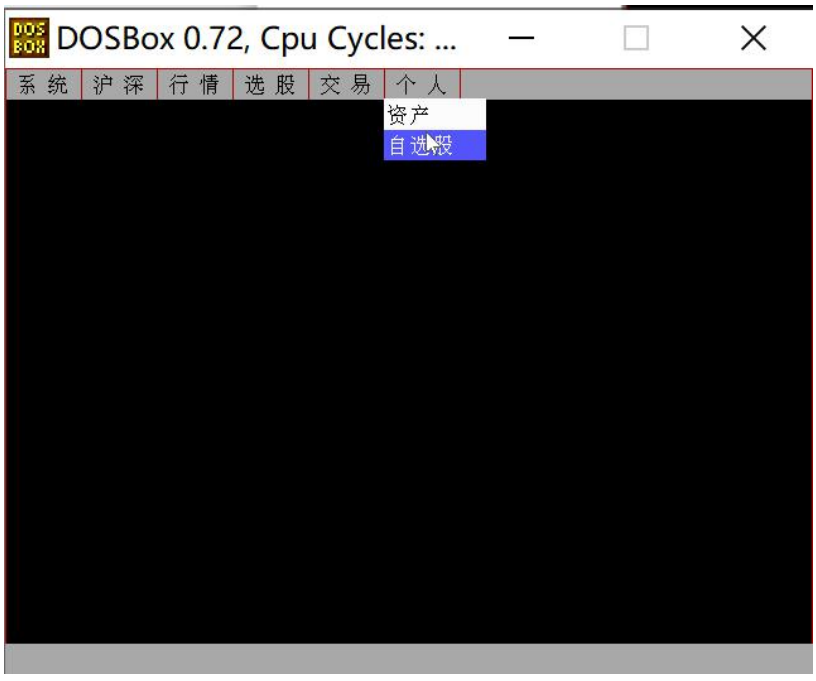
我们优化的界面灵感来自网易云的登录。当鼠标放在输入框上会自动点亮输入框并覆盖提示文字。

具体功能设计之后会提出，此处不再赘述。


DOSBox 0.72, Cpu Cycles: ...
系 统
沪 深
行 情
选 股
交 易
个 人
登 陆
退 出
注 销
关 于


DOSBox 0.72, Cpu Cycles: ...
系 统
沪 深
行 情
选 股
交 易
个 人
基本面
技术面
快速选股


DOSBox 0.72, Cpu Cycles: ...
系 统
沪 深
行 情
选 股
交 易
个 人
开 户
换 股


DOSBox 0.72, Cpu Cycles: ...
系 统
沪 深
行 情
选 股
交 易
个 人
资产
自选股

Interface——主菜单设计：

我们设置了系统，沪深，行情，选股，交易和个人六大主菜单。

其中系统，选股，交易和个人我们采取下拉菜单的方式，以节约空间，方便用户在不同功能和界面之间切换。

1. 系统：我们设计了下拉菜单。登录，退出，注销和关于。点击退出按钮会直接退出程序，点击注销按钮会返回登录界面。

2. 沪深：罗列股票信息，会有简单的搜索和排序。

3. 行情：K 线，技术指标和股票的一些数据分析。

4. 选股：我们设计了下拉菜单。选股分为基本面选股，技术面选股和智能选股。

5. 交易：我们设计了下拉菜单，分为开户和换股。

6. 个人：下拉菜单，分为资产和自选股。

以上菜单中的功能会在报告中的功能与算法设计，程序框图与流程设计中有详细的讲解与分析，此处不再赘述。

沪深板块设计：

代码	名称	涨幅%	现价	涨跌	买价	卖价	总量	现量	涨速%	换手%	今开
000001	平安银行	1.21	15.90	0.19	15.90	15.91	1007706	13071	0.00	0.59	15.85
000002	万科 A	-0.84	26.00	-0.22	26.00	26.01	397930	8578	0.00	0.41	26.21
000004	国农科技	1.51	18.88	0.28	18.88	18.89	6679	179	0.27	0.81	18.68
000005	世纪星源	1.94	3.16	0.06	3.15	3.16	84953	2007	0.00	0.80	3.14
000006	深振业 A	0.00	5.10	0.00	5.10	5.11	57092	1358	0.20	0.42	5.12
000007	全新好	-1.97	8.46	-0.17	8.46	8.47	86867	1637	-0.93	2.81	8.30
000008	神州高铁	0.29	3.48	0.01	3.48	3.49	969799	4027	0.29	3.74	3.47
000009	中国宝安	1.35	4.50	0.06	4.50	4.51	113796	888	0.00	0.45	4.49
000010	*ST美丽	-0.59	3.37	-0.02	3.37	3.38	7031	60	0.60	0.13	3.30
000011	深物业业	0.21	9.75	0.02	9.74	9.75	17551	269	-0.09	0.33	9.80
000012	南玻 A	0.71	4.26	0.03	4.25	4.26	72632	1069	0.24	0.37	4.23
000014	沙河股份	-1.14	9.54	-0.11	9.53	9.54	28877	669	0.21	1.43	9.64
000016	深康佳 A	1.44	4.22	0.06	4.22	4.23	81515	616	-0.23	0.51	4.15
000017	深中华 A	2.74	4.87	0.13	4.86	4.87	95317	898	0.00	3.15	4.76
000018	*ST神城	-1.02	0.97	-0.01	0.97	0.98	253258	2472	-1.01	2.54	0.98
000019	深粮控股	0.82	6.15	0.05	6.15	6.16	22073	734	-0.15	0.53	6.12
000020	深华发 A	3.30	11.27	0.36	11.27	11.28	15952	172	-0.08	0.88	10.91
000021	深科技	4.06	11.28	0.44	11.28	11.29	509269	5408	-0.17	3.47	10.80
000023	深天地 A	2.87	13.25	0.37	13.25	13.26	20573	388	0.23	1.48	12.98
000025	特力 A	-0.15	20.47	-0.03	20.47	20.48	30386	851	-0.04	0.77	20.38

代码	名称	涨幅%	现价	涨跌	买价	卖价	总量	现量	涨速%	换手%	今开
000001	平安银行	1.21	15.90	0.19	15.90	15.91	1007706	13071	0.00	0.59	15.85
000002	万科 A	-0.84	26.00	-0.22	26.00	26.01	397930	8578	0.00	0.41	26.21
000004	国农科技	1.51	18.88	0.28	18.88	18.89	6679	179	0.27	0.81	18.68
000005	世纪星源	1.94	3.16	0.06	3.15	3.16	84953	2007	0.00	0.80	3.14
000006	深振业 A	0.00	5.10	0.00	5.10	5.11	57092	1358	0.20	0.42	5.12
000007	全新好	-1.97	8.46	-0.17	8.46	8.47	86867	1637	-0.93	2.81	8.30
000008	神州高铁	0.29	3.48	0.01	3.48	3.49	969799	4027	0.29	3.74	3.47
000009	中国宝安	1.35	4.50	0.06	4.50	4.51	113796	888	0.00	0.45	4.49
000010	*ST美丽	-0.59	3.37	-0.02	3.37	3.38	7031	60	0.60	0.13	3.30
000011	深物业业	0.21	9.75	0.02	9.74	9.75	17551	269	-0.09	0.33	9.80
000012	南玻 A	0.71	4.26	0.03	4.25	4.26	72632	1069	0.24	0.37	4.23
000014	沙河股份	-1.14	9.54	-0.11	9.53	9.54	28877	669	0.21	1.43	9.64
000016	深康佳 A	1.44	4.22	0.06	4.22	4.23	81515	616	-0.23	0.51	4.15
000017	深中华 A	2.74	4.87	0.13	4.86	4.87	95317	898	0.00	3.15	4.76
000018	*ST神城	-1.02	0.97	-0.01	0.97	0.98	253258	2472	-1.01	2.54	0.98
000019	深粮控股	0.82	6.15	0.05	6.15	6.16	22073	734	-0.15	0.53	6.12
000020	深华发 A	3.30	11.27	0.36	11.27	11.28	15952	172	-0.08	0.88	10.91
000021	深科技	4.06	11.28	0.44	11.28	11.29	509269	5408	-0.17	3.47	10.80
000023	深天地 A	2.87	13.25	0.37	13.25	13.26	20573	388	0.23	1.48	12.98
000025	特力 A	-0.15	20.47	-0.03	20.47	20.48	30386	851	-0.04	0.77	20.38

我们通过循环队列的算法，罗列出沪深 A 股部分股票的有效信息。

包括代码、名称、涨幅%、现价、涨跌、买价、卖价、总量 、现量、 涨速%、换手%和今开。

为了方便用户翻页，我们取消了鼠标点击翻页，而是使用键盘键值键入翻页，同时我们使用二进制刷新界面。这样极大地增加了界面的刷新效率。

6、函数说明及函数原型

借助软件 typora 优化头文件定义：

- **MAIN.H**

```
#ifndef _MAIN_H_
#define _MAIN_H_

#include"welcome.h"
#include"initmenu.h"
#include"login.h"
#include"forgetpass.h"
#include"admin.h"
#include"register.h"
#include"interface.h"
#include"database.h"
#include"opaccount.h"

#endif
```

- **WELCOME.H**

```
#ifndef _welcome_H
#define _welcome_H

void welcome_page(void);    //欢迎界面

#endif
```

- **INITMENU.H**

```
#ifndef _INITMENU_H
#define _INITMENU_H

void initmenu_set(int* func);   //初始化选择界面
void drawmenu(void);//初始化界面绘制
void lightbutton_initmenu(int x, int y, int x1, int y1, int color1, int color2, int
flag);//按钮点亮
void recoverbutton_initmenu(int btnum);//点亮恢复

#endif
```

- **LOGIN.H**

```
#ifndef _LOGIN_H
#define _LOGIN_H
#include"database.h"

void user_login(int *func,USER* u); //用户登录界面
void drawlogin(void);//登录界面绘制
void lightbutton_login(int x, int y, int x1, int y1, int color1, int color2, int flag);
//按钮点亮
void recoverbutton_login(int status);//点亮恢复

#endif
```

- **ADMIN.H**

```
#ifndef _ADMIN_H
#define _ADMIN_H

void admin_login(int* func);    //管理员登录界面
void drawadmin(void);//管理员界面绘制
void lightbutton_admin(int x, int y, int x1, int y1, int color1, int color2, int flag);
//按钮点亮
void recoverbutton_admin(int status);//点亮恢复

#endif
```

- **REGISTER.H**

```
#ifndef _REGISTER_H_
#define _REGISTER_H_


void user_register(int* func);//注册函数
void random_str(char* str);//生成验证码
void draw_signin(char* str);//注册界面绘制
void lightbutton_register(int x, int y, int x1, int y1, int color1, int color2, int
flag); ////按钮点亮，color1为边框色,color2为填充色
void recoverbutton_register(int status);//点亮恢复

#endif
```

- **FORGETPASS.H**

```
#ifndef _FORGETPASS_H
#define _FORGETPASS_H

void user_forgetpass(int *func);    //忘记密码界面
void lightbutton_forget(int x, int y, int x1, int y1, int color1, int color2, int flag);
//按钮点亮
void draw_forgetpass(void);
void recoverbutton_forget(int status);//点亮恢复

#endif
```

- **DATABASE.H**

```
#ifndef _DATA_H_
#define _DATA_H_
typedef struct userinfo
{
    char user[15];//6-12位，用户名
    char ID[20];//18位，身份证
    char password[20];//8-16位，密码
    char bankcard[21];    //19位，银行卡
    char bankpassword[8];    //6位,银行卡密码
}USER;
//封装用户信息
typedef struct admininfo
{
    char user[15];//6-12位，管理员账号
    char password[10];//8位密码
}ADMIN;

#endif
```

- **LGFUN.H**

```
#ifndef _LGFUNC_H_
#define _LGFUNC_H_
#include"database.h"
void inputadmin(char* id, int x1, int y1, int charnum, int color);
//x1,y1分别是账号，验证码等输入信息的左端点和上端点

void inputpassword(char* id, int x1, int y1, int charnum, int color);
//x1,y1分别是密码等隐藏信息输入的左端点和上端点

int login_complete(char* u, char* p);
//用来判断登录信息是否输入完全和正确

int forgetpass_complete(char* u, char* id, char* np, char* cp);
//检查是否可以更改新密码

int judge_rightpassword(char* name, char* pass);
//用户名和密码是否匹配

int name_id(char* name, char* id, char* pass, char* againpass);
//判断账号与身份证是否匹配，并判断两次输入的密码是否相同，若匹配且相同则写入新密码。

int register_complete(char* u, char* p, char* cp, char* id, char* vc, char* rdvc,char*
bank,char* bankcode);
//用来判断注册信息是否输入完全和正确

void judge(char* str, int* state, int x, int y);
//判断注册的状态
```

```
void recoverhz(int x, int y, int color);
//覆盖之前输出的汉字

int judge_sameuser(char* new_user,int flag);
//判断账户是否出现重名,以及找回密码时判断用户名是否存在      1为

int checkright_user(char* str, int x, int y);
//检查用户名是否大于6位

int checkright_password(char* str, int x, int y);
//检查用户名是否大于6位

int checkright_confirmedpassword(char* str1, char* str2, int x, int y);
//检查确认密码是否一致

int checkright_ID(char* str, int x, int y, int flag);
//检查身份证位数与格式

int checkright_verificationcode(char* str1, char* str2, int x, int y);
//检查验证码是否一致

void input_database(char* name, char* id, char* code,char* bank,char* bankcode);
//把用户注册信息写入文件

int input_uinfo(USER* us);
//登陆后把用户信息读入u

#endif
```

- **PUBLIC.H**

```
#ifndef _PUBLIC_H_
#define _PUBLIC_H_

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<graphics.h>
#include<math.h>
#include<bios.h>
#include<conio.h>
#include<dos.h>
#include<time.h>
#include"HZ.H"
#include"MOUSE.H"
#include"DATABASE.H"
```

```
#define NUM0 0x5230
#define NUM1 0x4f31
#define NUM2 0x5032
#define NUM3 0x5133
#define NUM4 0x4134
#define NUM5 0x4c35
#define NUM6 0x4d36
#define NUM7 0x4737
#define NUM8 0x4838
#define NUM9 0x4939
#define F2 0x3c00
#define F3 0x3d00
#define F4 0x3e00
#define F5 0x3f00
#define F6 0x4000
#define F7 0x4100
#define F8 0x4200
#define F9 0x4300
#define F10 0x4400
#define F1 0x3b00
#define ENTER 0x1c0d
#define BACK 0x0e08
#define ESC 0x011b
#define UP 0x4800
#define DOWN 0x5000
#define RIGHT 0x4d00
#define LEFT 0x4b00

#endif
```

- **INTERFACE.H**

```
#ifndef _INTERFACE_H_
#define _INTERFACE_H_
#define F1 0x3b00
#define F2 0x3c00
#define F3 0x3d00
#define F4 0x3e00
#define F5 0x3f00
#define F6 0x4000

void menu_interface(int* func,USER* u);//登陆完成之后进入界面画含有下拉、标亮、恢复常亮的菜单
void mainmenu_draw(void);//画主菜单
void submenu_draw(int itemnum);//根据n的不同来画出属于n号父菜单的子菜单
void lightbutton_interface(int ord);//子菜单点亮
void recoverbutton_interface(int ord_last);//点亮恢复
int control_mainmenu(short int* itemnum, short int* ord_last);//通过鼠标点击或者键盘输入特殊键F1~F6实现父菜单的选择

#endif
```

- **OPACCOUNT.H**

```
#ifndef _OPACCOUNT_H_
#define _OPACCOUNT_H_
#include"database.h"

int opaccount(USER* u);//绘画开户界面之后的功能实现调用
void draw_opaccount(void);//画开户界面
int opaccount_complete(char* b,char* bp,char* bcp,USER* u);
//将功能框标亮，边框色为color1，填充色为color2；将x、y、x1、y1位置的功能框标亮
void lightbutton_opa(int x, int y, int x1, int y1, int color1, int color2, int flag);
//将序号为status的功能框恢复常量
void recoverbutton_opa(int status);//判断开户条件是否满足以及调用写入文件的函数
int checkright_bankcard(char* str);//判断银行卡号是否为19位
int checkright_bankpassword(char* str);//判断银行卡密码是否为6位
int checkright_bankconfirmedpassword(char* str1,char* str2);//判断两次输入是否一致
int judge_samecredit(char* new_credit,int flag);//打开文件判断银行卡号是否已经存在
int input_database_bk(char* bankcard,char* bankcode,USER* us);//开户成功判定之后，将银行卡、银行卡密码写入文件

#endif
```

- **QUEUE.H**

```
#ifndef _QUEUE_H_
#define _QUEUE_H_

#define MAXQSIZE 20

typedef struct                  //股票列表队结构
{
    Stodata* base;
    int front;                  //front端为屏幕上端
    int rear;                   //rear端为屏幕下端
    int flag;
}StoQueue;//股票信息的队列


void initqueue(StoQueue* Q);//构造存放股票信息的队列

int enqueue(StoQueue* q, FILE* fp1, int line, int prelen);//队头或队尾添加元素，向下翻页队头指针赋值，向上翻页队尾指针赋值

void destroyqueue(StoQueue* q);//销毁队列

void inputqueue(StoQueue* Q,int k);//队列赋值，k为文件开始位置

int countqueue(int k);//数每一页有多长

int countline(FILE* fp1, int prelen);//数每排有多长

#endif
```

1 函数名: void welcome_page();

输入参数：无;

函数描述：欢迎界面点击后延时 2000 毫秒进入 initmenu 界面;

返回值：无;

2 函数名:void initmenu_set(int* func);

输入参数：int 型指针变量 func;

函数描述：画初始化选择界面以及标亮恢复功能，点击不同的按钮进入不同的功能函数;

返回值：无;

3 函数名：void lightbutton_initmenu(int x, int y, int x1, int y1, int color1, int color2, int flag);

输入参数：标亮框的 x,y,x1,y1，颜色框 color1，填充颜色 color2;

函数描述：将功能框标亮;

函数描述：将 x、y、x1、y1 位置的功能框标亮;

返回值：无;

4 函数名：void recoverbutton_initmenu(int num);

输入参数：恢复常亮的序号 num;

函数描述：将序号为 num 的功能框恢复常量;

返回值：无;

5 函数名：void drawmenu(void);

输入参数：无;

函数描述：画初始化菜单界面;

返回值：无;

6 函数名：void user_login(int* func,USER* u)；

输入参数：int 型指针变量 func，USER 结构指针变量 u；

函数描述：完成用户的登陆功能、点击可进入注册、忘记密码、返回功能函数；

返回值：无；

7 函数名：int login_complete(char* u, char* p)；

输入参数：从键盘输入的字符串用户的字符型指针，与密码字符型指针；

函数描述：判断是否完成注册；

返回值：0 或 1；

8 函数名：int judge_rightpassword(char* name, char* pass)；

输入参数：字符型指针 name 和 pass；

函数描述：检测输入的用户名和密码是否匹配；

返回值：匹配则返回 1，否则返回 0；

9 函数名：void user_register(int *func)；

输入参数：int 型指针变量 func；

函数描述：绘画注册界面，并完成注册功能，点击不同区域则有标亮功能与恢复常亮功能，点击验证码即可更换验证码；

返回值：无；

10 函数名：int register_complete(char* u, char* p, char* cp, char* id, char* vc,char* rdvc,char* b,char *bp)；

输入参数：用户名、密码、确认密码、身份证、输入的验证码、随机验证码、银行卡号、银行卡密码等字符型指针；

函数描述：判断注册格式是否正确与是否存在同名用户名，以及调用其他函数；

返回值：无；

11 函数名：void draw_signin(char* str)；

输入参数：指向验证码的字符串指针；

函数描述：画含有验证码的注册界面；

返回值：无；

12 函数名：void random_str(char* str)；

输入参数：指向验证码字符串的字符型指针；

函数描述：随机给与含大小写字母和数字的验证码；

返回值：无；

13 函数名：void lightbutton_register(int x, int y, int x1, int y1, int color1, int color2, int flag)；

输入参数：标亮框的 x,y,x1,y1，颜色框 color1，填充颜色 color2，功能号 flag；

函数描述：将 x、y、x1、y1 位置的功能框标亮，如果 flag 为 6、7、8 则在相应位置输出汉字；

返回值：无；

14 函数名：void recoverbutton_register(int status)；

输入参数：恢复常亮的序号 status；

函数描述：将序号为 status 的功能框恢复常量；

返回值：无；

15 函数名：void input_database(char* name, char* id, char* code,char* bank,char* bankcode)；

输入参数：指向用户名、身份证、密码、银行卡号、银行卡密码的字符型指针；

函数描述：将用户信息输入文件中；

返回值：无；

16 函数名 int judge_sameuser(char* new_user,int flag)

输入参数：指向新输入的用户名的字符指针和功能参数 flag；

函数描述：当 flag 为 1 时，则在注册的时候提示该用户已存在、当 flag 为 2 时，在忘记密码完成的时候提示用户存在；

返回值：当重名时返回 0，不重名返回 1；

17 函数名：int checkright_user(char* str, int x, int y)；

输入参数：指向用户名的字符指针和某个位置的 x、y 值；

函数描述：检查用户名是否在 6-12 位之间（符合格式），若符合格式则返回 0，小于 6 位则返回一，并且在 x、y 处给与提示“名称应>6”；

返回值：0 或 1；

18 函数名：int checkright_password(char* str, int x, int y)；

输入参数：指向密码的字符型指针和位置的 x、y 值；

函数描述：检查密码是否在 8 到 16 位之间（符合格式），若符合格式则返回 0，小于 8 位则返回 1 并在 x、y 处给与提示“密码应>8”，否则则返回 1；

返回值：0 或 1；

19 函数名：int checkright_confirmedpassword(char* str1, char* str2, int x, int y)；

输入参数:指向密码的字符型指针 str1 和指向确认密码框中字符串的字符型指针 str2，以及位置的 x、y 值；

函数描述：检查两次输入的密码是否一致，若一致则返回 0，否则返回 1；

返回值：0 或 1；

20 函数名：int checkright_ID(char* str, int x, int y,int flag)；

输入参数：指向身份证字符串的字符指针、位置 x、y 值、功能参数 flah；

函数描述：检查身份证位数与格式；若身份证位数不等于 18 位，则在 x、y 处给与提示，若身份证位数等于 18 位，则开始检查格式的正确性；若 flag 为 1 则会在格式正确时画勾再返回、flag 为 2 时则直接返回；

返回值：格式正确返回 0；格式不正确返回 1；

21 函数名：int checkright_verificationcode(char* str1, char* str2,int x,int y)；

输入参数：指向随机验证码和输入验证码字符串的字符指针，位置参数 x、y；

函数描述：检查验证码是否一致，若不一致则返回 1，一致则返回 0；

返回值：0 或 1；

22 函数名：void judge(char* str, int* p, int x, int y)；

输入参数：指向字符串的字符指针、输入框状态参数 p、位置参数 x、y；

函数描述：判断是否输入，若未输入则提示并将 p 返回为 1，输入则返回为 0；

返回值：无；

23 函数名：void recoverhz(int x, int y,int color)；

输入参数：位置参数 x、y 和颜色参数 color；

函数描述：将 x、y 位置到 x+100，y+20 位置的汉字覆盖；

返回值：无；

24 函数名：void inputadmin(char* id, int x1, int y1, int charnum, int color)；

输入参数：输入的字符串，输入的 xy，输入的字符限制，输入框的颜色；

函数描述：在 x、y 位置开始输入的输入法，将最大输入数限制 charnum，以及使输入框为 color 色

返回值：无；

25 函数名：void inputpassword(char* id, int x1, int y1, int charnum, int color)；

输入参数：输入的密码字符串，位置参数 x、y，字符限制参数 charnum、输入框的颜色；

函数描述：在 x、y 位置开始输入的输入法，将最大输入数限制 charnum，以及使输入框为 color 色，并将每个输入的字符用*号遮盖；

返回值：无；

26 函数名：int input_uinfo(USER* us)；

输入参数：USER 型指针；

函数描述：在开户时，将已登陆账号的银行卡和银行卡密码输入文件中；

返回值：输入成功返回 1，否则返回 0；

27 函数名：void user_forgetpass(int *func)；

输入参数：功能参数 int 型指针；

函数描述：忘记密码界面，以及忘记密码功能的完成，点击重置即清屏并重新输入，点击完成则进入 forgetpass_complete 函数判断并完成修改密码输入文件，点击返回即通过改变*func 的值并返回登陆界面；

返回值：无；

28 函数名：void lightbutton_forget(int x, int y, int x1, int y1, int color1, int color2, int flag)

输入参数：标亮框的 x,y,x1,y1，颜色框 color1，填充颜色 color2；

函数描述：将功能框标亮，边框色为 color1，填充色为 color2；

函数描述：将 x、y、x1、y1 位置的功能框标亮；

返回值：无；

29 函数名：void recoverbutton_forget(int status)；

输入参数：恢复常亮的序号 status；

函数描述：将序号为 status 的功能框恢复常量；

返回值：无；

30 函数名：void draw_forgetpass(void)；

输入参数：无；

函数描述：画忘记密码界面；

返回值：无；

31 函数名：int forgetpass_complete(char* u, char* id, char* np, char* cp)

输入参数：指向用户名、身份证、新密码、确认密码字符串的字符指针；

函数描述：检查是否可以更改新密码并调用写入文件的函数

返回值：若成功写入则返回 1；

32 函数名：int name_id(char* name, char* id, char* pass, char* againpass)；

输入参数：指向输入框中的用户名、身份证、密码、确认密码的字符型指针

函数描述：判断账号与身份证是否匹配，并判断两次输入的密码是否相同，若匹配且相同则写入新密码。

返回值：成功写入则返回 1，否则返回 0；

33 函数名：void menu_interface(int *func,USER* u)；

输入参数：功能参数 int 型指针 func，以及指向用户信息的 USER 型指针 u；

函数描述：登陆完成之后进入界面画含有下拉、标亮、恢复常亮的菜单，，用鼠标完成菜单功能区的选择调用，通过比较 ord 和 ord_last 的相同性来完成子菜单的恢复常亮功能。以及父菜单 itemnum 号展开的情况下点击子菜单的功能实现；

返回值：无；

34 函数名：void mainmenu_draw(void)；

输入参数：无；

函数描述：画主菜单；

返回值：无；

35 函数名：void submenu_draw(int n)；

输入参数：父菜单的按钮号 n;

函数描述：根据 n 的不同来画出属于 n 号父菜单的子菜单；

返回值：无；

36 函数名：void lightbutton_interface(int ord)；

输入参数：子菜单号 ord;

函数描述：将号码为 ord 的子菜单点亮；

返回值：无；

37 函数名：int control_mainmenu(short int *itemnum,short int *ord_last)；

输入参数：父菜单号和上一个子菜单号；

函数描述：通过鼠标点击或者键盘输入特殊键 F1~F6 实现父菜单的选择，清屏后重画主菜单，将父菜单号传给 itemnum，并将 ord_last 归零；

返回值：成功选择父菜单则返回 1，否则返回 0；

38 函数名：void recoverbutton_interface(int ord_last)；

输入参数：需要恢复常亮的上一个菜单号；

函数描述；将 ord_last 菜单号恢复常亮；

返回值：无；

39 函数名：int opaccount(USER* u)；

输入参数：用户信息的 USER 型指针；

函数描述：绘画开户界面之后的功能实现调用；

返回值：开户成功返回 1，否则返回 0；

40 函数名：void draw_opaccount(void)；

输入参数：无；

函数描述：画开户界面；

返回值：无；

41 函数名：void lightbutton_opa(int x, int y, int x1, int y1, int color1, int color2, int flag)；

输入参数：标亮框的 x,y,x1,y1，颜色框 color1，填充颜色 color2；

函数描述：将功能框标亮，边框色为 color1，填充色为 color2；将 x、y、x1、y1 位置的功能框标亮；

返回值：无；

42 函数名：void recoverbutton_opa(int status)；

输入参数：恢复常亮的序号 status；

函数描述：将序号为 status 的功能框恢复常量；

返回值：无；

43 函数名：int opaccount_complete(char* b,char* bp,char* bcp,USER* u)；

输入参数：指向银行卡号、银行卡密码、银行卡确认密码和用户信息 USER 型指针 u；

函数描述：判断开户条件是否满足以及调用写入文件的函数；

返回值：开户成功返回 1，否则返回 0；

44 函数名：int checkright_bankcard(char* str)；

输入参数：指向银行卡号的 str 字符型指针；

函数描述：判断银行卡号是否为 19 位，若不为 19 位则给与提示并返回值 1，如果银行卡号为 19 位则返回 0；

返回值：0 或 1；

45 函数名：int checkright_bankpassword(char* str)；

输入参数：指向银行卡密码的 str 字符型指针；

函数描述：判断银行卡密码是否为 6 位，若不为 6 位则给与提示并返回 1，若为 6 位则画勾并返回 0；

返回值：0 或 1；

46 函数名：int checkright_bankconfirmedpassword(char* str1,char* str2)；

输入参数：银行卡密码和银行卡确认密码的字符指针；

函数描述：判断两次输入是否一致，是则返回 0 并画勾，若不一致则返回 1；

返回值：0 或 1；

47 函数名：int judge_samecredit(char* new_credit,int flag)；

输入参数：新银行卡号字符指针与功能参数 flag；

函数描述：打开文件判断银行卡号是否已经存在，当 flag 为 1 时，且与文件中的银行卡号相同时，则在开户界面给与提示并返回 0，若未存在相同银行卡号则返回 1；

返回值：0 或 1；

48 函数名：int input_database_bk(char* bankcard,char* bankcode,USER* us)；

输入参数：银行卡号、银行卡密码的字符型指针，用户信息 USER 型指针；

函数描述：开户成功判定之后，将银行卡、银行卡密码写入文件，与当前的用户信息 USER 型指针 us 中；成功写入则返回 1，否则返回 0；

返回值：1 或 0；

8、时间安排

周数	任务
第一周	进行需求分析并设计报告
第二周	构思主要界面，绘制登录，注册，欢迎和忘记密码等界面，优化鼠标
第三周	完成注册，登录等数据库构建和搭建，设计主界面框架和下拉菜单
第四周	开户与沪深队列设计，整理中期报告，中期验收
第五周	选股与行情（K线）设计与函数
第六周	交易，管理员与游客设计，开始完成最终报告
第七周	debug与优化算法，做一些创新点
第八周	程序调试，整理报告，准备最后的验收