功能：信息查询

总类：用户接口

<template T>

T={Date,Student,Total,College}

interface user(T)

{

String getInformation(T)

{

ConnectServer()

SendRequestToServer(T)

ReceiveInformation()

InformationHandler()------type1:data package

}

Present(type1):find specific component

OffConnect()

}

分类1：学生

数据内容：A，日期；B，开始时间；C，结束时间；D，跑步时长；E，平均速度；F，最快速度；G，跑步距离；H，获得分数；A1，是否达标(0,1)；

数据内容扩展：I，跑步天数；J，总跑步时长; K,白天跑步总时长；L，晚间跑步总时长，M，跑步总距离；N，获得总分数；

class student(Date) implements user(Date)

{

String getInfomation(T)

{

//通过touch日历中某一天------获得日期（date）

ConnectServer()

sendRequestToServer(A)

/\*服务器内容

IdentifyTypeofMessage()

DataBaseOperation(A)

DataBase---Information(type2)---Server

ReturnMessage(type2)

\*/

ReceiveInformation()---String

InformationHandler()---type2:data package

}

Present(type2):find specific component in student model //将获得的信息以预期方式组合并呈现

offConnect()

}

分类2：体育教师

数据内容：班级代码，A；班级人数，B；班级跑操次数总和，C；班级跑操时间总和，D;班级完成/未完成人数，E；全体完成/未完成/存在未达标风险/不可能完成人数

//以上数据基本由数据库操作来完成

Class TotalInformation(void) implements user(void)

{

//extra type T:get all of the information from database

Private sendingType;

TotalInformation(T)---sengdingType==T

TotalInformation(A)---sendingType==A

//与学生类的函数操作基本相同，只不过需要传递的参数有所不同，服务器接受相应参数后所要调用的数据库数据也有所不同，基本函数沿用上述。

Present():在具体界面的对应component装入不同的数据(文字，图像) //总体数据

ClassSearching(sendingType=A) //通过输入具体班级代码，查询具体班级的跑操情况

{

具体实现仍然同上所述  
}

}

数据内容：A，学生学号；B，学生姓名；C，学生性别；D，学生身体素质

//预先处理类：获得学生清单(学号，姓名，性别)

Define sendingType=GL

Class getStudentList(void):total

{

//同理，基本函数操作相同

Problem：列表的呈现

}

Class SingleInformation(A) implements user()

{

//由预处理类获得的列表可以了解到具体某个学生的学号信息

//要查询具体学生的跑操信息，需要键入学号（由上一步提供），执行搜索

SingleSearching(A)

{

服务器交流操作，具体实现同上

}

Present()

}

分类3：辅导员/班主任

数据内容：A，院系代码；

Class TotalManagement(院系代码/班级代码) implements user() //辅导员/班主任界面

{

Initialization() //对初始信息进行初始化，将辅导员与院系/班主任与班级进行绑定

\\可能的解决方案：在初始化时，由辅导员输入具体的院系代码，服务器将之与该手机的IP地址绑定，每次需要查询时，服务器会根据传来指令的IP地址，自动的返回相应的信息。

\\普通的解决方案还是通过传输特定的信息类型，服务器通过判断这种类型，返回对应的信息。

Solution1()

{

Private String;

getCollegeCode():I/O(String)

getIPAddress():?(String)

sendBothMessage()

ConnectBoth():?

}

String getInformation()

{

具体服务器交流操作基本相同

1. sendRequestToServer(IP(或者其他能够关联IP的信息))
2. IdentifyTypeOfMessage() \\通过IP地址，由ConnectBoth所提供的服务，寻找到对应的学院代码，然后利用数据库操作返回信息。

}

Solution2()

{

传输类型是学院/班级代码，其余操作基本不变。

}

Present()

}

Class GetStudentList():college/class

Class SingleManagement() implements user()

\\该类的实现也是通过获取学生列表，然后执行查询，获得具体学生运动信息来完成

所以具体操作与体育老师的SingleInformation以及其预先处理类基本相同。