1．引言 2

1.1编写目的 2

1.2项目背景 2

1.3定义 2

1.4参考资料 2

2．总体设计 2

2.1需求概述 2

2.2软件结构 2

3．程序描述 3

3.1功能 3

3.2性能 3

3.3输入项目 3

3.4输出项目 3

3.5算法 3

3.6程序逻辑 3

3.7接口 4

3.8存储分配 4

3.9限制条件 4

3.10测试要点 4

# 1．引言

## 1.1编写目的

在当今信息社会中，移动技术使我们走到哪里都可以获取最新的信息，都可以和自己的朋友联系。尼尔森2010年发布了《中国手机市场洞察报告》，该报告对全国范围内19个城市中4946名15岁及以上的消费者进行研究，该调查采用与消费者进行面对面访问的形式，实地调查时间为2010年3月。根据这份报告，中国手机用户已经达到了7.55亿人次，超过中国总人口数的一半。

在大学里，更可谓人人一部手机，通过技术，使同学们在任何时间、任何地点都能获取最新的信息，意义非凡。

移动平台上的校园信息化建设，目前在国内还鲜有高校在做。并且在校园信息化建设中部署移动技术的应用，是举足轻重的。

## 1.2项目背景

软件名称：学生信息门户。

产品用户：所有在校学生。

读者对象：产品的使用者。

## 1.3定义

USTB: 为北京科技大学之英文缩写。

InfoPortal: 软件英文名，为信息门户之意。

IP: 在程序中用到，为InfoPortal之缩写，通常作为前缀使用。

db: 为数据库Database之缩写，常用于文件扩展名。

软件：指代此学生信息门户软件。

## 1.4参考资料

1. Dave Mark, Jack Nutting, Jeff LaMarche著

iPhone 4与iPad开发基础教程

ISBN: 9787115255525

2011年7月

人民邮电出版社

1. Mark Dalrymple, Scott Knaster著

Objective-C基础教程

ISBN: 9787115208774

2009年8月

人民邮电出版社

1. Carlo Chung著

Objective-C编程之道 iOS设计模式解析

ISBN: 9787115265869

2011年11月

人民邮电出版社

# 2．总体设计

## 2.1需求概述

学生，“学”则为学习之意，而“生”则为古代对于有学问的男子的称呼，因此称之为学生，必定要不断的学习。除了上课的时间以外，其他学习的时候就是上自习了。如何有效的找到可以供自己上自习的教室就是第一个需求。原来寻找一个无课的教室，需要自己到逸夫楼或者教学楼一个一个的去找，费时费力；或者根据学生会整理出来，并且做成展板放在楼门口的无课表来查询，但是风吹日晒，不是找不到了，就是因为改课程计划等种种原因出现错误。相对于传统的由学生会学习部整理出的无课表，手机平台上的无课表有众多优势，主要体现在：

1. 随时随地的获取 – 打开手机，就可看到
2. 特殊需求查询（比如上午连续两节课都可供自习的一个教室）-满足各种不同的需求
3. 当前无课教室-根据当前教学周和星期几和当前时间，智能判断出哪个教室可以供上自习

对于学生而言，借阅的图书可以看三十天。但因在繁杂的校园中，上课、实验、社团和论文等种种因素，总是会搞得自己团团转。也许过了大概20天之后就忘了自己准确借阅的时间了，更可怕的是忘记了自己曾经借过一本书。

通过个人借阅信息，可以通过在软件中轻松查询到自己图书馆借阅信息，如所借图书，到期日期等。并可通过软件，在手机中的闹钟和行事历中进行定时设置，通过这种智慧化的方式可以在到期前一天提醒你“哦，是该还书的时间了”。

图书馆在藏数目查询功能可以通过输入图书的部分信息来查询图书在图书馆中的收藏情况，以及是否可借。

## 2.2软件结构

【如给出软件系统的结构图。】

# 3．程序描述

## 3.1功能

1.ClassDatabase类

用以对无课表Controller提供所需数据。

1.1getBuild函数

提供教学楼的列表

1.2getBuildName函数

通过id返回教学楼名称

1.3getFloor函数

通过教学楼id返回楼层数

1.4getTimes函数

获取节次数组  
 1.5getTermStart函数

获取开学时间

1.6getTermEnd函数

获取放假时间

1.7getWeekNumber函数

获取教学周数值

1.8getUsage函数

获取某个教室连续几天的使用情况

1.9getResult函数

获取符合查询条件的教室列表

2.IPDatabase类

对SQLite方法进行封装

2.1open函数

创建sqlite3实例

2.2close函数

关闭sqlite3实例

2.3query函数

执行一个select查询

2.4moveToNext函数

获取下一条结果

2.5insert函数

执行一个insert查询

2.6remove函数

执行一个delete查询

2.7update函数

执行一个update查询

3.IPDate类

提供不同格式时间之间的转换的简易方法

3.1timeToStamp函数

将不同字符串格式的时间转换为Unix时间戳。

3.2getWeekName函数

获取星期的三个英文字符缩写

4.Initialize类

提供程序第一次启动时的初始化工作

4.1checkAndMove函数

检测数据库文件是否存在，不存在的话将数据库文件复制到Document目录

5.StructBuild类

提供教学楼查询结果的返回数据结构

6.StructUsage类

提供教室使用情况的查询结果的返回数据的结构

7.StructResult类

提供条件查询结果的返回数据的结构

8.HttpClient类

提供异步的Http请求与下载

8.1initWithURLString函数

初始化URL参数

8.2start函数

重载NSOperation的start方法，使其在系统队列中执行时自动连接到远程服务器。

8.3connect函数

异步处理NSURLConnection的方法，用于对成功收到数据与连接失败的情况处理。

9.HttpParser类

提供了对Html文件的解析方法

9.1decodeUnicode函数

将输出为UTF8编码字符的编码转换为字符。

9.2searchAllReplace函数

查找并替换所有符合条件的字符串。

9.3hexToInt函数

将String类型的十六进制转换为int。

10.ClassViewController

控制无课表主界面

10.1initWithNibName

初始化无课表主界面

10.2resetBuildingButton

重置楼名按钮，根据用户选择的教学楼情况设置教学楼按钮的背景

10.3ChooseBuilding

用户点击楼名按钮时调用，更改保存的用户选择数据，并重置楼名按钮和楼层弹出列表

10.4NextButton

用户点击开始查询时调用，检查用户所选数据是否正确，不正确则弹出警告，正确则跳转查询结果界面

10.5getdate

获取TimePickerView子窗体中的DatePicker控件的date数据

10.6getnumber

获取ChooseView子窗体中的TableView控件的Index数据

10.7viewDidload

载入窗体时调用，进行初始化和子窗体对象的生成

10.8initData

初始化需要用到的数据

10.9initBuilding

初始化楼名模块

10.10initMap

初始化地图模块

10.11initTime

初始化时间选择模块

10.12initFloor

初始化楼层选择模块

10.13initClass

初始化节次选择模块

10.14viewDidUnload

窗体关闭时调用

10.15getFloor

获取目前用户选择的教学楼的最大层数

10.16resetSearchButton

使searchButton可交互

10.17checkFloor

查看当前是否选择了楼名，以判断能否弹出楼层选择窗体

10.18checkTime

监测用户所选的时间是否符合规则

10.19shake

晃动窗口

10.20ClickButton

点击主界面中除了楼名按钮和开始查询按钮以外按钮时调用，主要用于弹出对应的ChooseView和TimePickerView实例

10.21resetBuildID

将以楼层多少为顺序排列的楼层选择结果转换为以id为顺序排列

11.ClassResultViewController

控制无课表查询结果界面

11.1initWithNibName

初始化无课表查询结果界面，调用数据库函数得到查询结果

11.2viewDidLoad

无课表查询结果界面载入时调用，绘制查询结果

11.3ClickButton

每个教室按钮点击时调用，弹出ClassTableView

11.4ViewDidUnload

界面关闭时调用

12.TimePickerView

日期选择窗体的所有方法属性

12.1initWithSuperView

初始化日期选择窗体

12.2close

关闭日期选择窗体，播放关闭动画

12.3animationFinished

动画结束时调用，关闭窗体

12.4ButtonClick

按钮按下时调用，根据按下的按钮不同选择设置日期或返回

12.5loadView

载入界面，播放弹出动画

12.6dateChanged

所选日期改变时调用，同步更新label显示

13.ClassTableView

无课表窗体的所有方法属性

13.1initWithSuperView

初始化无课表窗体

13.2drawTable

绘制无课表的表格元素

13.3close

关闭无课表窗体，播放关闭动画

13.4animationFinished

动画结束时调用，关闭窗体

13.5loadView

载入界面，播放弹出动画

13.6touchesBegan

在手势开始时调用获得手指位置

13.7touchesMoved

在手指移动时调用，时事获取手指位置与开始位置比较进行手势判断，若符合要求则调用close

14.ChooseView

选择窗体的所有方法属性

14.1initWithSuperView

初始化选择窗体

14.2close

关闭选择窗体，播放关闭动画

14.3animationFinished

动画结束时调用，关闭窗体

14.4ButtonClick

叉形按钮按下时调用，调用close

14.5loadView

载入界面，播放弹出动画

14.6tableViewnumberOfRowsInSection

获取tableview项数

14.7tableViewcellForRowAtIndexPath

绘制tableview的每个cell

14.8tableViewdidSelectrowAtIndexPath

点击tableview后调用，掉去主界面get number方法

15.MapView

地图窗体的所有方法属性

15.1initWithSuperView

初始化地图窗体

15.2close

关闭地图窗体，播放关闭动画

15.3animationFinished

动画结束时调用，关闭窗体

15.4ButtonClick

叉形按钮按下时调用，调用close

15.5loadView

载入界面，播放弹出动画

15.6initMap

初始化Mapview设置mapview的显示范围和其他属性

15.7initMapKit

初始化Mapkit，根据Datalist绘制MapKit

15.8LocationManagerdidUpdatetoLocationfromLocation

当用户所在位置更新后调用，初始化Mapview

15.9setCurrentLocation

当用户选择指定地点后调用，

16.MapAnnotations

MapKit的所有方法和属性

16.1initWithCoordinate

初始化MapKit

17.UserChoosedData

用户选择数据类

17.1resetChoosedData

## 3.2性能

查询时间，响应时间等均在人类可察觉范围之外。

## 3.3输入项目

1.ClassDatabase类

无

1.1getBuild函数

无

1.2getBuildName函数

buildId: 教学楼id

1.3getFloor函数

buildId: 教学楼id

1.4getTimes函数

无  
 1.5getTermStart函数

无

1.6getTermEnd函数

无

1.7getWeekNumber函数

curTimeStamp: Unix时间戳

1.8getUsage函数

buildId: 教学楼id，roomNum: 房间号，startTimeStamp: 开始时间Unix时间戳，days: 持续天数

1.9getResult函数

buildSelect: 教学楼选择值（生成方法：值的位数与教学楼的数量相等，从小到大分别为第一座，第二座。。。置1为选中，置0为未选中），floorFrom: 楼层起始值，floorTo: 楼层结束值，timeFrom: 节次起始值，timeTo: 节次结束值，timeStamp: 查询时间Unix时间戳。

2.IPDatabase类

无

2.1open函数

无

2.2close函数

无

2.3query函数

table: 表格名，columns: 列名数组，selection: where语句，orderBy: 排序方法，limit: 获取前几项。

2.4moveToNext函数

无

2.5insert函数

table: 表格名，values: 需要插入的键值对，selection: where语句

2.6remove函数

table: 表名，selection: where语句

2.7update函数

table: 表名，values: 需要修改的键值对，selection: where语句

3.IPDate类

无

3.1timeToStamp函数

date: 字符串格式的时间，format: 字符串时间格式

3.2getWeekName函数

timestamp: unix时间戳

4.Initialize类

无

4.1checkAndMove函数

无

5.StructBuild类

无

6.StructUsage类

无

7.StructResult类

无

8.HttpClient类

无

8.1initWithURLString函数

url: url字符串

8.2start函数

无

8.3connect函数

无

9.HttpParser类

无

9.1decodeUnicode函数

dataStr: Unicode十六进制值字符串

9.2searchAllReplace函数

inString: 输入字符串，src: 被替换字符串，dest: 替换字符串

9.3hexToInt函数

hexNumber: Unicode十六进制值字符串

10.ClassViewController

10.1initWithNibName

(NSString \*)nibNameOrNil bundle:(NSBundle \*)nibBundleOrNil 所用nib文件名

10.3ChooseBuilding

(id)sender 调用他的按钮的指针

10.4NextButton

(id)sender 调用他的按钮的指针

10.5getdate

(NSDate \*)date TimePickerView子窗体中的DatePicker控件的date数据

10.6getnumber

(NSString \*)data:(int)number 获取ChooseView子窗体中的TableView控件的Index数据和与之对应的名字

10.19shake

(UIView \*)shakeView 要晃动的窗口名

10.20ClickButton

(id)sender 调用他的按钮的指针

10.21resetBuildID

(long)ID 以楼层多少为顺序排列的楼层选择结果

11.ClassResultViewController

11.1initWithNibName

(NSString \*)nibNameOrNil bundle:(NSBundle \*)nibBundleOrNil:(UserChoosedData \*)data:(ClassDatabase \*)db 所用nib文件名，主界面的用户选择数据，主界面的数据库实例

11.3ClickButton

(id)sender 调用他的按钮的指针

12.TimePickerView

12.1initWithSuperView

(UIView \*)superView:(ClassDatabase \*)db 父窗体和父窗体的数据库实例

12.4ButtonClick

(id)sender 调用他的按钮的指针

12.5loadView

(UIView \*)superView 父窗体

12.6dateChanged

(id)sender 改变日期的Datepickerview指针

13.ClassTableView

13.1initWithSuperView

(UIView \*)superView:(ClassDatabase \*)db:(long)time:(UIButton \*)Sender

父窗体，父窗体数据库实例，用户选择的时间，调用他的按钮的指针

13.5loadView

(UIView \*)superView 父窗体

14.ChooseView

14.1initWithSuperView

(UIView \*)superView 父窗体

14.4ButtonClick

(id)sender 调用他的按钮的指针

14.5loadView

(UIView \*)superView 父窗体

15.MapView

15.1initWithSuperView

(UIView \*)superView:(NSArray \*)Building 父窗体和所有的教学楼数组

15.4ButtonClick

(id)sender 调用他的按钮的指针

15.5loadView

(UIView \*)superView 父窗体

15.7initMapKit

(double)x:(double)y:(NSString \*)title 经度，纬度，和教学楼名字

16.MapAnnotations

MapKit的所有方法和属性

16.1initWithCoordinate

(CLLocationCoordinate2D)c 新建的MapKit

## 3.4输出项目

1.ClassDatabase类

无

1.1getBuild函数

NSArray: StructBuild类的数组

1.2getBuildName函数

NSString: 教学楼名

1.3getFloor函数

int: 楼层数

1.4getTimes函数

NSArray: 获得节次列表  
 1.5getTermStart函数

long: 获取开学时间的Unix时间戳

1.6getTermEnd函数

long: 获取放假时间的Unix时间戳

1.7getWeekNumber函数

int: 第几周的数值

1.8getUsage函数

NSArray: 教学楼id，roomNum: 房间号，startTimeStamp: 开始时间Unix时间戳，days: 持续天数

1.9getResult函数

NSArray: StructResult数组，包括了返回的结果数值。

2.IPDatabase类

无

2.1open函数

int: 返回数据库打开成功与否状态。成功为1，不成功为0。

2.2close函数

无

2.3query函数

int: 返回数据库查询成功与否，成功为1，不成功为0。

2.4moveToNext函数

NSArray: 返回每条结果，为一个字符串数组。

2.5insert函数

int: 返回查询状态，成功为1，不成功为0。

2.6remove函数

int: 返回查询状态，成功为1，不成功为0。

2.7update函数

int: 返回查询状态，成功为1，不成功为0。

3.IPDate类

无

3.1timeToStamp函数

long: 对应时间的Unix时间戳数值。

3.2getWeekName函数

NSString: 对应的周名称。

4.Initialize类

无

4.1checkAndMove函数

返回布尔值。

5.StructBuild类

无

6.StructUsage类

无

7.StructResult类

无

8.HttpClient类

无

8.1initWithURLString函数

id: 返回实例。

8.2start函数

无

8.3connect函数

无

9.HttpParser类

无

9.1decodeUnicode函数

NSString: 对应的字符串。

9.2searchAllReplace函数

NSString: 对应的替换后的字符串。

9.3hexToInt函数

int: 十六进制字符串对应的int值。

## 3.5算法

对教室无课情况的表示，使用了二进制数字，每一位表示一个教学周。从低到高位分别为第一周、第二周……第16周。每一周如果有课的话，此位置零，没有课的话，此位置1。获取情况时，若为第n周，只需要将1向左移动n-1位，与存储数值进行与操作，结果为1，则此周教室空闲，为0，则此周教室占用。

对教学楼的选择情况也使用了二进制数字，每个楼用一个二进制数表示，选择置1，不选置0。这样就不需要新建一个数组，节省了空间，计算也简便。

为了防止重复的数据库访问，在类中添加了类可访问的全局变量，减少了重复查询的次数。

## 3.6程序逻辑

|  |
| --- |
| ClassDatabase |
| classNumber: int  termStartStamp: long |
| getBuild(): NSArray  getBuildName(int buildId): NSString  getFloor(int buildId): int  getTimes: NSArray  getTermStart: long  getTermEnd: long  getWeekNum(long curTimeStamp): int  getUsage(int buildId, int roomNum, long startTimeStamp, int days): NSArray  getResult(long buildSelect, int floorFrom, int floorTo, int timeFrom, int timeTo, long timestamp): NSArray |

|  |
| --- |
| IPDatabase |
| database: sqlite3  statement: sqlite3\_stmt  columnSize: int |
| open(): int  close(): void  query(NSString table, NSArray columns, NSString selection, NSString orderBy, NSString limit): int  moveToNext(): NSArray  insert(NSString table, NSDictionary values, NSString selection): int  remove(NSString table, NSString selection): int  update(NSString table, NSDictionary values, NSString selection): int |

|  |
| --- |
| IPDate |
|  |
| timeToStamp(NSString date, NSString format): long  getWeekName(long timestamp): NSString |

|  |
| --- |
| Initialize |
|  |
| checkAndCopy(): BOOL |

|  |
| --- |
| StructBuild |
| buildId: int  buildName: NSString  location: NSString |
|  |

|  |
| --- |
| StructResult |
| buildId: int  roomName: NSMutableArray |
|  |

|  |
| --- |
| StructUsage |
| timestamp: long  usageStatus: NSArray |
|  |

|  |
| --- |
| HttpClient |
|  |
| initWithURLString(NSString url): id  start(): void  connection(NSURLConnection connection, NSData data): void  connection(NSURLConnection connection, NSError error): void |

|  |
| --- |
| HttpParser |
|  |
| decodeUnicode(NSString dataStr): NSString  searchAllReplace(NSString inString, NSString src, NSString dest): NSString  hexToInt(NSString hexNumber): int |

|  |
| --- |
| AppDelegate |
|  |
|  |

|  |
| --- |
| ChooseView |
| delegate: NSObject<UIViewPassValueDelegate>  background: UIView  ListData: NSArray  alertShowView: UIView |
| ButtonClick(ind sender): IBAction  loadView(UIView superView): void  initWithSuperView(UIView superView): id |

|  |
| --- |
| ClassResultViewController |
| myResultData: NSMutableArray  myResult: StructResult  myDatabase: ClassDatabase  myUserData: UserChoosedData  myClassTableView: ClassTableView |
| initWithNibName(NSString nibNameOrNil, NSBundle nibBundleOrNil, UserChoosedData data, ClassDatabase db): id |

|  |
| --- |
| ClassTableView |
| alertShowView: UIView  background: UIView  myData: NSArray  myTime: long  myDatabase: ClassDatabase  mySender: UIButton  myUsage: StructUsage  gest: CGPoint |
| initWithSuperView(UIView superView, ClassDatabase db, long time, UIButton sender): id  loadView(UIView superView): void |

|  |
| --- |
| ClassViewController |
| myTimePickerView: TimePickerView  myChooseView\_1: ChooseView  myChooseView\_2: ChooseView  myChooseView\_3: ChooseView  myChooseView\_4: ChooseView  myMapView: MapView  myUserData: UserChoosedData  choosedate: NSDate  ListData: NSArray  Building: NSArray  TimeNow: NSDataFormatter  myBuild: StructBuild  mychoosedBuildingID: long  floors: NSArray  Button: NSMutableArray  ButtonSender: IBOutlet UIButton  myDatabase: ClassDatabase  myIPDate: IPDate  myAlertView: UIAlertView  myResultViewController: ClassResultViewController |
| ClickButton(id sender): IBAction  ChooseBuilding(id sender): IBAction  NextButton(id sender): IBAction |

|  |
| --- |
| LibraryViewController |
| myAlertView: UIAlertView  gest: CGPoint |
| backgroundTap(id sender): IBAction |

|  |
| --- |
| MapAnnotations |
| corrdinate: CLLocationCoordinate2D  subtitle: NSString  title: NSString |
| initWithCoordinate(CLLocationCoordinate2D coordinate): id |

|  |
| --- |
| MapView |
| delegate: NSObject<UIViewPassValueDelegate>  background: UIView  ListData: NSArray  myBuild: StructBuild  alertShowView: UIView  myMap: MKMapView  myLocationManager: CLLocationManager  loc: CLLocationCoordinate2D |
| ButtonClick(id sender): IBAction  loadView(UIView superView): void  initWithSuperView(UIView superView, NSArray Building): id  setCurrentLocation(CLLocation location): void |

|  |
| --- |
| TimePickerView |
| delegate: NSObject<UIViewPassValueDelegate>  alertShowView: UIView  background: UIView |
| dateChanged(id sender): IBAction  ButtonClick(id sender): IBAction  loadView(UIView superView): void  initWithSuperView(UIView superView, ClassDatabase db): id |

|  |
| --- |
| UIViewPassValueDelegate |
|  |
| getdate(NSDate date): void  getnumber(NSString data, int number): void  resetSearchButton(): void |

|  |
| --- |
| UserChoosedData |
| choosedBuildingID: long  choosedFloorFromID, choosedFloorToID: int  choosedClassFromID, choosedClassToID: int  choosedDate: long |
| resetChoosedData(): void |

## 3.7接口

UIViewPassValueDelegate

-(void)getdate:(NSDate \*)date;

-(void)getnumber:(NSString \*)data:(int)number;

-(void)resetSearchButton;

## 3.8存储分配

按照类的大小动态分配。

## 3.9限制条件

无

## 3.10测试要点

测试类中方法的返回值，直接输出Log，或者在程序中嵌入Log函数来实时观测程序的运行情况。

对SQL语句输出，用以直接查看语句是否符合要求。

主界面注意每个按钮都要点击，尝试不选择楼名点击楼层，尝试选择周六日点击开始查询，尝试在不同时间登陆程序观察默认选择的节次和日期变化，点击Map里的MapKit

查询结果界面注意点击各个教室，查看其近7天内的课表。在无课表弹出窗口使用向上轻划手势退出，选择楼层多一些使空闲教室多一些观察ScrollView情况