

天气应用需求与设计分析报告

软件工程 1102 班 靳杰 U201112375

目录

- 1 天气应用需求分析 1
 - 1.1 需求描述 1
 - 1.2 需求分析 1
 - 1.2.1 应用范围 1
 - 1.2.2 应用功能点定义 1
- 2 系统设计 1
 - 2.1 类图设计 1
 - 2.2 部分算法设计 2
 - 2.2.1 城市定位 2
 - 2.2.2 获取天气信息 3
 - 2.2.3 数据存储与查询 4
- 3 总结 6
- 4 系统截图展示 6

1 天气应用需求分析

1.1 需求描述

基于 iOS 手机操作系统，制作一款天气应用，实现以下基本功能点：

- 从中国天气网，webservice 获取指定的城市天气信息，并可在系统中显示当前天气情况。
- 用 sqlite 数据库保存获取的历史天气信息，查阅存储的历史天气。
- 对以往的天气情况进行统计，各类天气的数量，百分比。
- 绘制温度的变化曲线图。

1.2 需求分析

1.2.1 应用范围

系统运行在 iOS 移动操作系统平台上，要求 iOS 7.0 及以上。适用机型：iPhone4、iPhone4s、iPhone5、iPhone5s。

1.2.2 应用功能点定义

- 定位当前城市，显示当前天气情况。
- 更改城市，显示指定城市天气情况。
- 绘制温度变化曲线图。
- 查询存储的天气情况。

2 系统设计

2.1 类图设计

采用 MVC 模式设计，得到 View 层，Control 层与 Model 层类图：

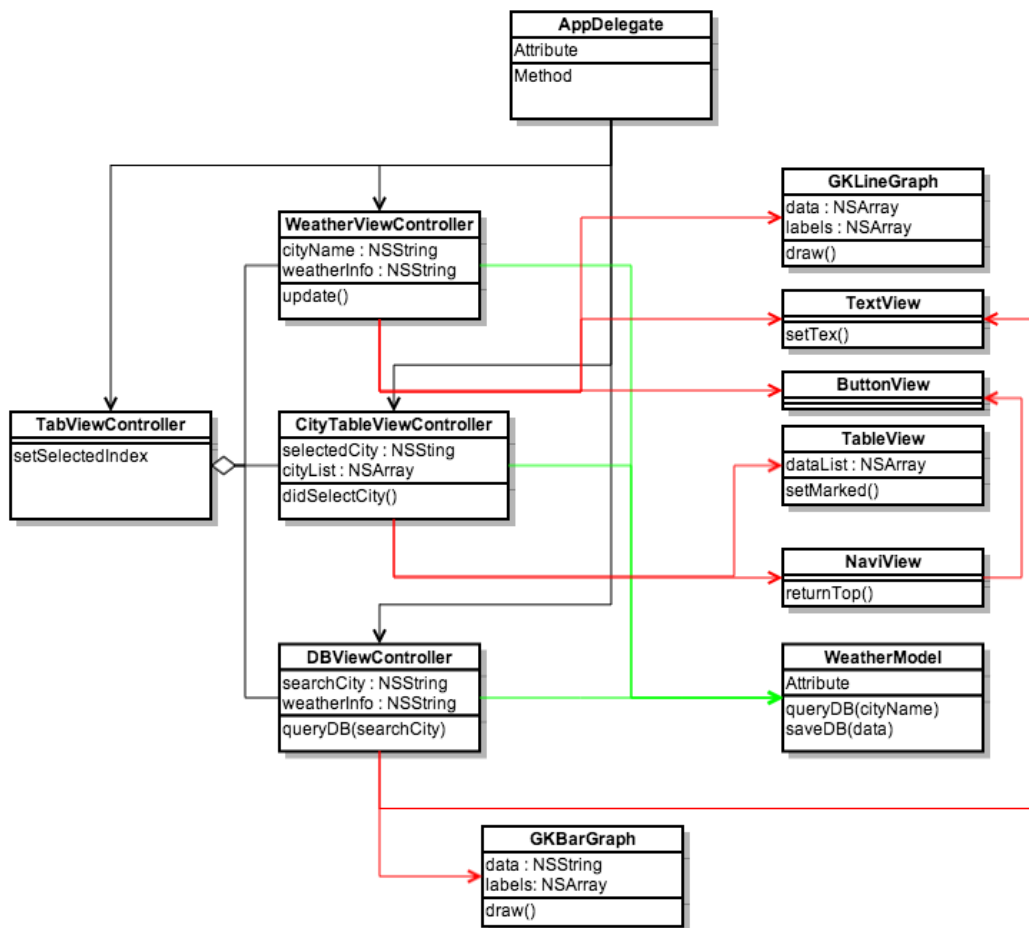


图 1: 系统类图

2.2 部分算法设计

2.2.1 城市定位

未采用 iOS 系统自带的 GPS 定位信息，使用的是通过网络 ip 进行定位。网络地址为 <http://61.4.185.48:81/g/>，通过这个网络地址可以得到一个 json 格式字符串，解析字符串可以得到当前网络 ip 对应的城市代号。

关键代码:

```
1 // 通过ip定位当前城市 获取城市天气代码
2 NSURL *url = [NSURL URLWithString:@"http://61.4.185.48:81/g/"];
3
4 // 定义一个NSError对象, 用于捕获错误信息
5 NSError *error;
6 NSString *jsonString = [NSString stringWithContentsOfURL:url encoding:NSUTF8StringEncoding
7     error:&error];
8 // NSLog(@"-----%@", jsonString);
9 if (jsonString != NULL) {
10     NSString *intString;
11     NSString *Str;
12     for (int i = 0; i <= [jsonString length]; i++) {
13         for (int j = i+1; j <= [jsonString length]; j++) {
14             Str = [jsonString substringWithRange:NSMakeRange(i, j-i)];
15             if ([Str isEqualToString:@"id"]) {
16                 if (![jsonString substringWithRange:NSMakeRange(i+3, 1)] isEqualToString:@"c"
17                     ]) {
18                     intString = [jsonString substringWithRange:NSMakeRange(i+3, 9)];
19                 }
20             }
21         }
22     }
23     self.wm = [WeatherModel getInstance];
24     [self.wm startWithNum:intString];
25     [self.message setText:@"定位成功! "];
26     // 设置cityTableViewController 的城市名为当前城市
27 }
```

2.2.2 获取天气信息

使用中国天气网 api: <http://m.weather.com.cn/atad/城市代号.html>。国内每一个城市都对应有一个城市代号, 通过访问相应的网络地址, 可以获得 json 格式数据, 解析数据得到城市天气信息。

关键代码:

```
1 // 通过城市编号开始获取数据
2 - (void)startWithNum:(NSString *)thecityNum{
3     // 向开源的地址发送连接请求
4     // 这里使用的是异步的请求
5     NSString *urlString = [NSString stringWithFormat:@"http://m.weather.com.cn/data/%@.html",
6         thecityNum];
7     NSURL *url = [NSURL URLWithString:urlString];
8     NSURLRequest *urlRequest = [NSURLRequest requestWithURL:url cachePolicy:
9         NSURLRequestReturnCacheDataElseLoad timeoutInterval:30];
```

```

8     NSURLConnection *urlConnection = [NSURLConnection connectionWithRequest:urlRequest delegate
9         :self];
10    [urlConnection start];
11 }
12 #pragma mark - NSURLConnectionDataDelegate methods
13 // 获取数据失败后显示网络连接失败
14 - (void)connection:(NSURLConnection *)connection didFailWithError:(NSError *)error{
15     UIAlertView * alertV = [[UIAlertView alloc] initWithTitle:@"网络连接失败" message:[NSString
16         stringWithFormat:@"%@",error] delegate:self cancelButtonTitle:nil otherButtonTitles:nil,
17         nil];
18     [alertV show];
19 }
20 // 获取数据后存入数据库
21 - (void)connection:(NSURLConnection *)connection didReceiveData:(NSData *)data{
22     // 网络返回的 JSON 数据 data
23     self.m_jsonData = data;
24     NSError *error = nil;
25     NSDictionary *dict = [NSJSONSerialization JSONObjectWithData:self.m_jsonData options:
26         NSJSONReadingMutableLeaves error:&error];
27     id weatherInfo = [dict objectForKey:@"weatherinfo"];
28
29     self.cityName = [weatherInfo objectForKey:@"city"];
30     // 解决拿到空数据的情况
31     if ( self.cityName == nil) {
32         // 显示警告
33         [[FirstViewController getInstance] showWarning];
34         return;
35     }
36     // 存储数据
37     // 略
38 }

```

2.2.3 数据存储与查询

采用 sqlite 数据库，在 iOS 应用开发中，需要导入 libsqlite3.dylib 动态库。
数据存储与更新关键代码：

```

1 // 数据存储与更新
2 - (void) sqliteOpen{
3     sqlite3 *database;
4     if (sqlite3_open([self dataFilePath] UTF8String], &database) != SQLITE_OK) {
5         // 数据库创建失败！
6         sqlite3_close(database);
7         NSAssert(0, @"open database failed!");
8     }

```

```

9     else{
10         // 创建数据库语句
11         NSString *createSQL = @"CREATE TABLE IF NOT EXISTS weather(cityName TEXT primary key,
cityNum TEXT,cityInfo TEXT, date TEXT, week TEXT, temp1 TEXT, temp2 TEXT, temp3 TEXT,
temp4 TEXT, temp5 TEXT, temp6 TEXT, weather1 TEXT, weather2 TEXT, weather3 TEXT, weather4
TEXT, weather5 TEXT, weather6 TEXT, img1 TEXT, img2 TEXT, img3 TEXT, img4 TEXT, img5 TEXT,
img6 TEXT)";
12
13         char *ERROR;
14         if (sqlite3_exec(database, [createSQL UTF8String], NULL, NULL, &ERROR) != SQLITE_OK){
15             sqlite3_close(database);
16             // 表创建失败
17             NSAssert(0, @"create table failed!");
18         }
19         else {
20             char *saveSQL = "INSERT OR REPLACE INTO weather(cityName ,cityNum ,cityInfo , date
, week , temp1 , temp2 , temp3 , temp4 , temp5 , temp6 , weather1 , weather2 , weather3 ,
weather4 , weather5 , weather6 , img1, img2, img3, img4, img5, img6)""VALUES
(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)";
21             char *errorMsg = NULL;
22             sqlite3_stmt *stmt;
23             if (sqlite3_prepare_v2(database, saveSQL, -1, &stmt, nil) == SQLITE_OK) {
24
25                 sqlite3_bind_text(stmt, 1, [self.cityName UTF8String], -1, NULL);
26                 ... // 略
27             }
28             // 数据更新失败
29             if (sqlite3_step(stmt) != SQLITE_DONE) {
30                 NSAssert(0, @"error updating :%s",errorMsg);
31             }
32             sqlite3_finalize(stmt);
33             sqlite3_close(database);
34         }
35     }
36 }

```

数据查询关键代码:

```

1 // 获取app内文件路径函数
2 -(NSString *) dataFilePath{
3     NSArray *path = NSSearchPathForDirectoriesInDomains(NSDocumentDirectory, NSUserDomainMask,
YES);
4     NSString *document = [path objectAtIndex:0];
5     return [document stringByAppendingPathComponent:@"weather.sqlite"];
6 }
7 // 查找数据库
8 -(BOOL) sqliteCheck{
9     sqlite3 *database;

```

```

10     BOOL returnValue = YES;
11     // 数据库查找失败！
12     if (sqlite3_open([[self dataFilePath] UTF8String], &database) != SQLITE_OK) {
13         sqlite3_close(database);
14         NSAssert(0, @"open database failed!");
15         return false;
16     }
17     else{
18         // 查询语句
19         NSString *query= [NSString stringWithFormat:@"select * from weather where cityName = '%@'", [CityTableViewCell getCurrentCity]];
20         sqlite3_stmt *stmt;
21         if (sqlite3_prepare_v2(database, [query UTF8String], -1, &stmt, nil) == SQLITE_OK) {
22             returnValue = false;
23             while (sqlite3_step(stmt) == SQLITE_ROW) {
24                 self.cityName = [[NSString alloc] initWithUTF8String:(char*)sqlite3_column_text(stmt, 1-1)];
25                 ... // 略
26                 returnValue = true;
27             }
28         }
29         else{
30             // 查询失败
31             NSLog(@"%s",sqlite3_errmsg(database));
32             returnValue = false;
33         }
34         sqlite3_finalize(stmt);
35     }
36     //用完了一定记得关闭，释放内存
37     sqlite3_close(database);
38     return returnValue;
39 }

```

3 总结

项目网络托管地址 <https://github.com/jinjaysnow/Snow>。

此次实验大作业编写 iOS 平台的天气应用程序，是我自己第一次较全面 (网络编程、数据库、UI、多视图控制) 的制作 iOS 平台应用程序。万事开头难，通过这次实验，了解了 iOS 应用程序开发的基本流程和应用程序整个生命周期的运作方式，获得了基本的 *Objective-C* 编程技巧。收获较大。

4 系统截图展示

A Jin Jay Application

Carrier

8:38 PM



武汉



温度: 13°C~5°C



定位成功!

网络更新时间: 2014年3月4日

今日指数: 建议着厚外套加毛衣等服装。年老体弱者宜着大衣、呢外套加羊毛衫。

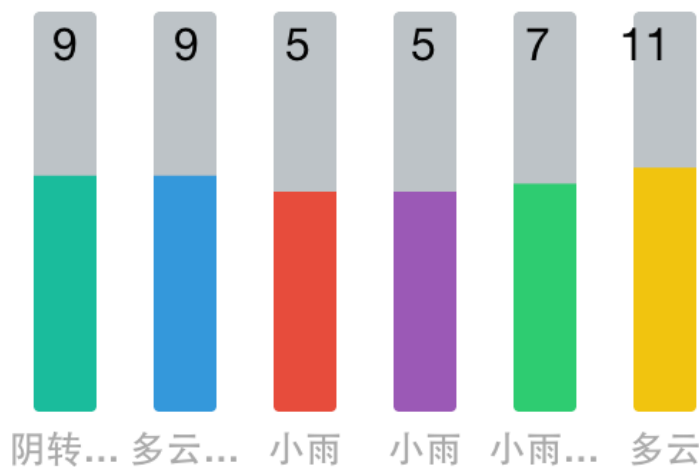


图 3: 系统启动界面



图 4: 城市选择界面

Carrier

8:38 PM



通州



温度: 8°C~-2°C



网络更新时间: 2014年3月4日

今日指数: 天气寒冷, 建议着厚羽绒服、毛皮大衣加厚毛衣等隆冬服装。年老体弱者尤其要注意保暖防冻。

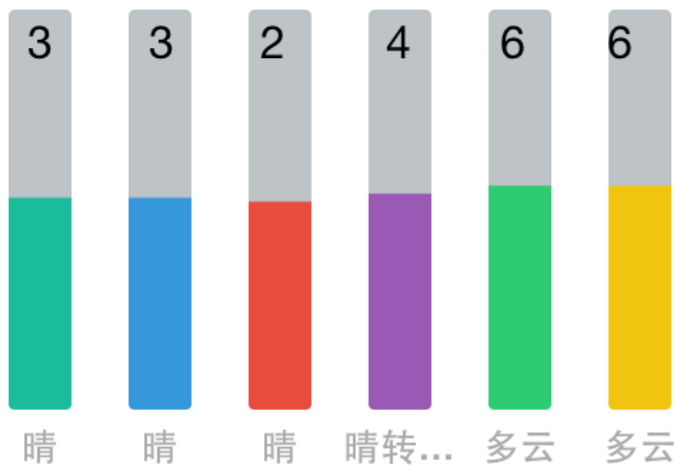


图 5: 选择城市后更新界面



图 6: 城市搜索界面

Carrier

8:39 PM



随州



温度：

14°C~3°C



刷新

网络更新时间：2014年3月4日

今日指数：天气冷，建议着棉服、羽绒服、皮夹克加羊毛衫等冬季服装。年老体弱者宜着厚棉衣、冬大衣或厚羽绒服。

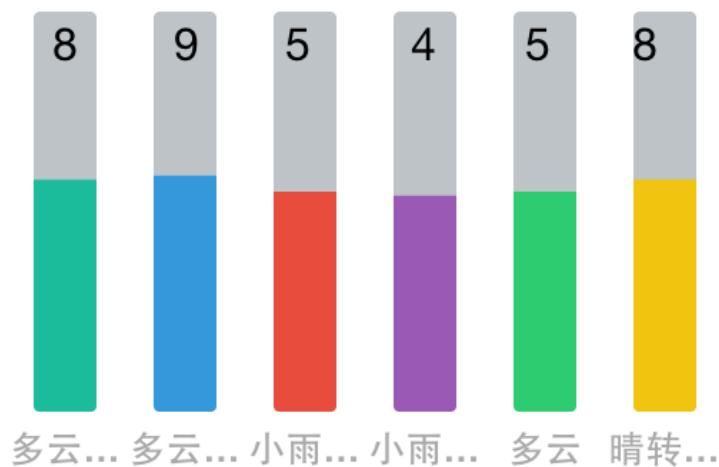


图 7: 搜索后更新界面

请输入城市名

数据库结果

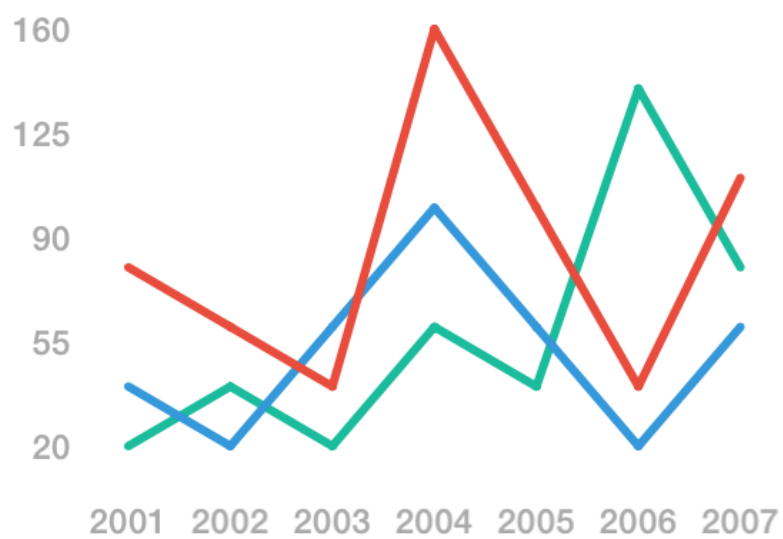


图 8: 数据库界面

请输入城市名

数据库结果



图 9: 数据库搜索界面



请输入城市名

随州

数据库结果

随州

天气冷，建议着棉服、羽绒服、皮夹克加羊毛衫等冬季服装。年老体弱者宜着厚棉衣、冬大衣或厚羽绒服。

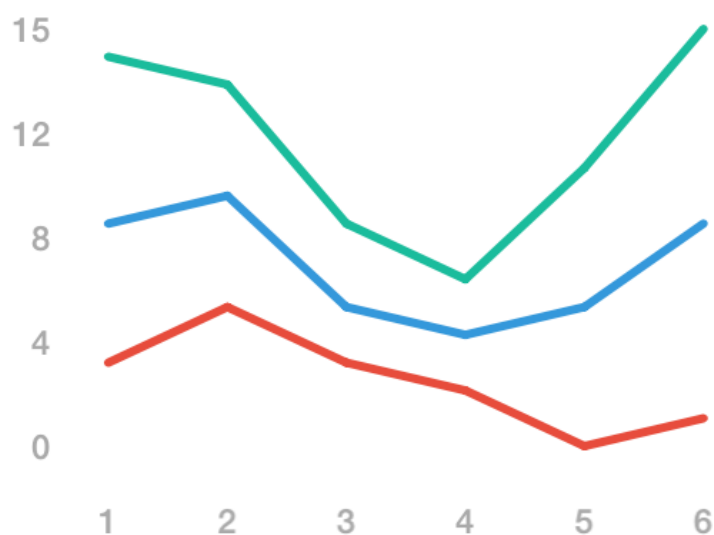


图 10: 数据库搜索结果展示