International Journal of u- and e- Service, Science and Technology
Vol. 7, No. 6 (2014), pp. 21-32
http://dx.doi.org/10.14257/ijunesst.2014.7.6.03

Design and Implementation of iOS-based Mobile Application about Awakening by CNR

Canlin Li[1], Baohua Jin[1] and Wenjie Cao[2]

[1]*School of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou 450000, China*
[2]*Software Engineering College, Zhengzhou University of Light Industry, Zhengzhou 450000, China*
lcl_zju@aliyun.com

Abstract

With the growing popularity of the iPhone, there is a constantly increasing demand of users for mobile applications of iPhone, in which the alarm clock application is contained. But existing alarm clock application is not favored by many users because of functional singleness. This paper presents an iOS-based mobile application on alarm clock with online radio support, and designs and implements its framework by virtue of Objective-C and SQLite in the Xcode 4.5.2 development environment, based on the popular three-tier MVC software design structure. After the application is installed and deployed on iPhone, it is available to users by providing some functions such as auto-playing radio, setting radio alarm clock, turning off the alarm by shaking iPhone, sleeping timer by radio, reserving radio program, binding and sharing microblog. This proposed iOS-based mobile application provides users with more choices and allows the user to enjoy a more colorful leisure time. The practical test from some people shows that the proposed application is very popular with users, which also illustrates its practicability and effectiveness.

Keywords: *iOS platform; CNR; mobile application; alarm clock*

1. Introduction

According to Canalys data, since 2011, the iOS system has been among the best in the global smart phone market share. In November 2011, iOS devices accounted for more than 30% of the global smart phone system, and market share rate in the U.S. even reached 43%. With the growing popularity of the iPhone, and the ongoing enrichment on the functions of iPhone applications, there is a constantly increasing demand of users for mobile applications [1-4], in which the alarm clock application is contained. But existing alarm clock application is not favored by many users because of functional singleness. So it is necessary to design an alarm clock which can provide users with more features as well as more choices.

This paper proposes an iOS-based mobile application on alarm clock with online radio support, which aims to provide users with more choices for alarm clock. In terms of implementation details, we utilize CNR (China National Radio) as sample radio. The multiple functions of the proposed alarm clock application about awakening by CNR are all very close to users' lives. When the user gets up in the morning, if he is tired of hearing the official ringtones, he may listen to the auto-played radio programs to get up by virtue of setting the radio alarm clock. Maybe it is hard to get him up only by the alarm clock; the function of turning off the alarm only by shaking iPhone will clear away his sleepiness. When the user suffers from insomnia and is hard to fall asleep in the night, the function of sleeping timer by

ISSN: 2005-4246 IJUNESST

radio will allow him to fall asleep along with listening to his favorite radio programs. Moreover, the function of reserving radio program will timely remind the user of listening to his favorite radio program when it is just started, and make him not miss it.

The remainder of this paper is organized as follows. Section 2 focuses on the functional and structural analysis and database design of the proposed alarm clock application. We describe schematic design of the proposed application in Section 3, following it up with detailed design of this application in Section 4. Section 5 illustrates the running results of the achieved application. Finally, the paper is concluded in Section 6.

2. Analysis of Application

2.1. Application Functional Analysis

After an analysis of functional requirement on the iOS-based mobile application about awakening by CNR, it is supposed to provide the following functions. (1) Homepage window; (2)Listen to radio: play or pause CNR, show the name of current radio program and clock, play radio in the background; (3)Set radio alarm clock: add alarm, delete alarm, set alarm, show the program about current alarm, save the current list of alarm clocks, show all of the lists of alarm clock; (4)Set bedtime timer: set bedtime, save the bedtime settings, countdown; (5) Reserve radio program: show the list of programs and the current date, program reservation; (6)Set application: bind microblog account, set whether to use only in WIFI network mode, about us etc.; (7) Share through microblog: select sharing platform, input sharing information, count the words, release microblogging. According to the design process of software engineering [5], we design the use case for every above demanded function. Table 1 shows a typical use case about alarm clock settings.

Table 1. Use Case about Alarm Clock Settings

Case Name	Set alarm clock	
Case ID	UC007	
Description	Users can set the name and time of alarm clock, whether to repeat and whether to turn off the alarm only by shaking iPhone	
Priority	high	
Precondition	Already enter alarm setting sub-module	
Post-condition	Set the alarm details	
Operational process	user	If necessary, the user can click on items about alarm name, time, whether to repeat and whether to turn off the alarm by shaking iPhone to set the appropriate circumstances
	system	Save the alarm settings
included cases	UC006: Add alarm clock	

2.2. Application Structure Analysis

The iOS-based mobile application about awakening by CNR is mainly composed of the following five modules: wake-up alarm clock, bedtime timer, radio program reservation, application settings and microblog sharing. For the module of wake-up alarm clock, the user can add, delete and modify the alarm clock, and can customize the name, time, repetitions, duration, notification ringtone and whether to turn off the alarm only by shaking iPhone. For the module of bedtime timer, the users can customize the duration of listening to the radio during bedtime and achieve to countdown the listening, also can adjust the listening volume. For the module of radio program reservation, the users can make an appointment to listen to

his favorite programs according to the program list. For the module of application settings, the user can choose to bind one of the microblogging platforms, and set whether to use only in WIFI network mode. For the module of microblog sharing, the user can call Sina or NetEase or Tencent or microblg open platform interface and share currently listening to radio programs and feelings. After analyzing the actual situation of requirement and application, we get the overall structure of the application function modules shown in Figure 1.

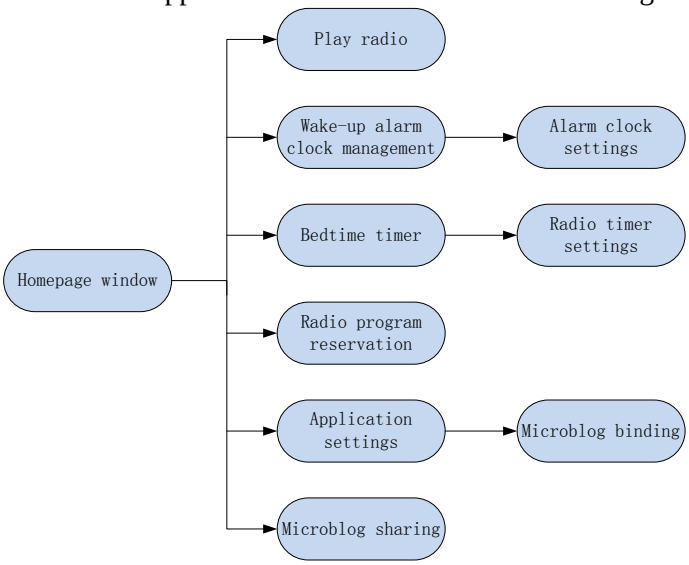

Figure 1. Application Functional Module Diagram

2.3. SQLite Database Design

Database occupies a very important position in an application, and a reasonable database structure will also help to achieve application. The iSO-based CNR wake-up mobile application utilizes SQLite [6] as the system database. The iSO platform contains a SQLite database, which is a simple, lightweight database, and declared as the world's most widely deployed database engine. The SQLite database for the iSO-based CNR wake-up mobile application mainly contains the following tables: the table "programs" of programs about CNR, which is used to record the basic information of programs about CNR; the table "clocks" of alarm clocks, which is used to store the detailed information about custom alarm clocks; the table "appSet" of application settings, which is used to store the basic information about application settings; the table "appShare" of application sharing, which is used to store the basic information about application sharing. Table 2 shows the structure of the table "clocks" of alarm clocks.

3. Schematic Design of Application

The schematic design of application is based on the functional requirements and functional architecture of application to build a system of functional modules, which make precise descriptions and uniform regulations for the application. It means that the "blueprint" for CNR wake-up mobile applications is built. This design is based on the popular three-layer MVC (Model-View-Controller) framework [5, 7].

Table 2. The Structure of the Table "Clocks"

Field Name	Data Types	Field Description
clockId	integer	Alarm number
clockName	varchar	Alarm name
startTime	datetime	Time the alarm rings
DurationTime	varchar	Duration
repeated	varchar	Repetitions
isShaked	integer	Whether to turn off the alarm only by shaking iPhone
isOpened	integer	Whether to open the alarm
clockMusic	varchar	Ringtone of alarm

Model layer is for the SQLite database logic layer, involving the operating table and some entity classes of application. The main models and their functions in the model layer are as follows:

(1) RadioProgDataController: the operating model for the table "programs" of programs about CNR, which is mainly used to add, delete and modify programs;
(2) RadioProg: the entity class of radio programs, which is used to descript the properties and methods of radio program;
(3) ClockDataController: the operating model for the table "clocks" of alarm clocks, which is mainly used to add, delete and modify radio alarm clocks;
(4) Clock: the entity class of radio alarm clocks, which is used to descript the properties and methods of radio alarm clock;
(5) AppDataController: the operating model for the table "appSet" of application settings, which is mainly used to add, delete and modify application settings;
(6) AppSet: the entity class of application settings, which is used to descript the properties and methods of application setting.

Controller layer is for implementing the business logic, involving designing the controller classes for required functions. The main models and their functions in the controller layer are as follows:

(1) MainViewController: the controller class for homepage window of application, which is used to get data from the interface and call other models in the controller layer to achieve main interface functions;
(2) ClockSetViewController: the controller class for setting radio alarm clocks, which is used to get data from the interface and call other models in the model layer to control and manage radio alarm clocks;
(3) BedtimingViewController: the controller class for bedtime timers, which is used to get data from the interface and call other models in the controller layer to control and manage bedtime timers;
(4) RadioProgOrderViewController: the controller class for radio program reservation, which is used to get data from the interface and call other models in the model layer to control and manage radio program reservation;

(5) AppViewController: the controller class for application settings, which is used to get data from the interface and call other models in the model layer to achieve application settings.

View layer is for designing human-computer interface, aiming to enhance application availability and user-friendliness. The models in the view layer are based on and basically correspond one-to-one to the models in the controller layer, only to achieve the controller layer interaction with user. It is unnecessary to go into details here.

We name each model partially according to its belonged logic software layer, and all models in the same layer have the same extension, and the models to achieve the same function but be located at different layers have the same prefix. Every model for iSO-based CNR wake-up mobile application in three layers of MVC is designed in detail, and given the exact description of the interface and attribute definitions, as well as the design of algorithms and data structures. Tables 3 show the interface and attribute definitions of the MainViewController model.

Table 3. Design Table about Homepage Window

Model Name	MainViewController
Function description	The controller class for homepage window of application, which is used to get data from the interface and call other models in the controller layer to achieve main interface functions
interface and attribute	@property (nonatomic, retain) IBOutlet UIButton *playButton; @property (retain, nonatomic) IBOutlet UIImageView *radioBg; @property (retain, nonatomic) IBOutlet UILabel *progNameLab; @property (assign,nonatomic) NSInteger clockId; @property (retain, nonatomic) IBOutlet UIImageView *bg; - (IBAction)playBtnClick:(id)sender; - (IBAction)displayBtnClick:(id)sender; - (IBAction)setBtnClick:(id)sender; - (IBAction)shareBtnClick:(id)sender;

4. Detailed Design of Application

4.1. Application Development Platform

Xcode [8] is a non-open-source integrated development environment of Apple Corporation which is used to develop applications based on Mac OS X and iOS platform. Objective-C [9] is an object-oriented programming language based on an expansion of C which is used to develop such applications. The iSO-based CNR wake-up mobile application is developed using Objective-C and SQLite database in Xcode 4.5.2 integrated development environment. We illustrated the development tools in Figure 2.

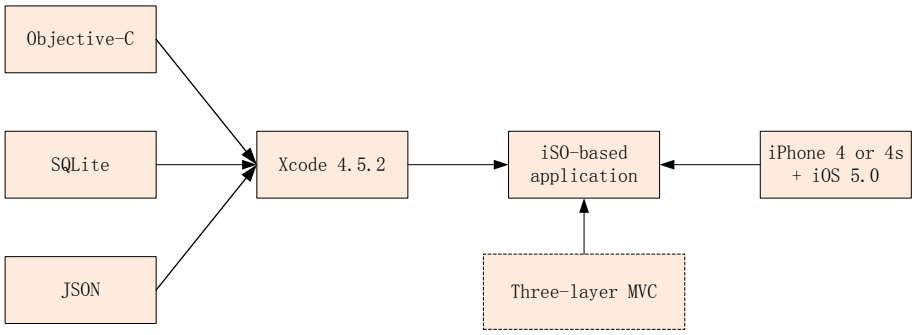

Figure 2. Development Tools

4.2. Key Design Ideas

Detailed design ideas about the major modules of the iSO-based CNR wake-up mobile application are as follows.

(1) The module of homepage window: we parse mms protocol website using RadioTunes, get real-time broadcast from the server and sent to the user. The main code of this module is as follows.

```
- (IBAction)playBtnClick:(id)sender
{
    if(reachability==nil) { reachability = [[Reachability sharedReachability] retain]; }
    if(appDataController==nil) { appDataController = [[[AppDataController alloc] init] retain]; }
    NetworkStatus connectionStatus = [reachability internetConnectionStatus];
    if(connectionStatus == NotReachable)
    {   if(alert= =nil)
        {   alert = [[UIAlertView alloc] initWithTitle:@"无法连接网络" message:@"若要收听
            收音机需要连接网络" delegate:self cancelButtonTitle:@"确认" otherButtonTitles:nil];
            [alert retain];  }
        [alert show];
        return;   }
    else if ([[[[appDataController getAppSet] objectAtIndex:0] objectForKey:@"isWifi"] intValue])
    {   if(connectionStatus!=ReachableViaWiFiNetwork)
        {   if(alert!=nil)
            {alert.message=@"app 被设置为只在 wifi 下使用，若收听收音机需连接 wifi 网络";}
            else
            { alert = [[UIAlertView alloc] initWithTitle:@"无法连接网络" message:@"app 被
              设置为只在 wifi 下使用，若收听收音机需连接 wifi 网络"
              delegate:self cancelButtonTitle:@"确认" otherButtonTitles:nil];
              [alert retain]; }
            [alert show];
            return;  }     }
    if(_radio == nil) {    return;   }
    if([_radio isPlaying]) {   [_radio pause];   }
    else {   [_radio play];   }
}
```

(2) The module of radio alarm clock: we add animation effects to radio alarm control and management, and apply group's tableview to make own editing effects display outside the group. Moreover, by customizing the cell to animate the display of alarm clock in the edit and complete states, to achieve custom edit of alarm. The main code of this module is as follows.

```
-(void)editBtnClick:(id)sender
{
    UIBarButtonItem *buttonItem = (UIBarButtonItem *)sender;
    [UIView beginAnimations:nil context:nil];
    [UIView setAnimationDuration:0.2];
    NSArray *indexPaths = self.myTableView.indexPathsForVisibleRows;
    if([buttonItem.title isEqualToString:@"编辑"]&&[_clockList count]>0)
   {   buttonItem.title = @"完成";
       for(NSIndexPath *indexPath in indexPaths)
      {   UITableViewCell *cell = [self.myTableView cellForRowAtIndexPath:indexPath];
          NSInteger row = [indexPath row];
          if(indexPath.section == 0)
         {   if(row!=0)
             {   UISwitch *mySwitch = (UISwitch *)[cell viewWithTag:row];
                 mySwitch.hidden = YES;
```



```
                UIView *view = (UIView *)[cell viewWithTag:row+1];
                view.frame = CGRectMake(45, 5, 70, 40);
                UIImageView *editView = (UIImageView *)[cell viewWithTag:row+2];
                editView.frame = CGRectMake(10, 5, 30, 28);  }
            else
            {   cell.accessoryType = UITableViewCellAccessoryNone;
                _myTableView.delegate = nil; }        }        }
        }
        //Some codes about editing  alarm  clock is omitted here due to space limitation.
        [UIView commitAnimations];
    }
```

(3) The module of bedtime timer: we achieve stopwatch counting by timer, and call the methods about radio play and stop in the master controller to reckon by time to listen to radio. The main code of this module is as follows.

```
    -(void)timingBtnClick:(id)sender
    {
        if(mainViewController==nil){mainViewController = [[[MainViewController alloc] init] retain];}
        UIButton *btn = (UIButton *)sender;
        //start to set timer
        if(btn.currentTitle==@"开始")
        {  [btn setTitle:@"结束" forState:UIControlStateNormal];
           //Some codes about set background image and text color of button omitted here
           NSTimer *timer = [NSTimer scheduledTimerWithTimeInterval:1
                                target:self selector:@selector(timeInterval:) userInfo:nil repeats:YES];
           currenTimer = timer;
           [currenTimer retain];
           //play radio
           if(_radio == nil) { return; }
           if(![_radio isPlaying]) {[_radio play];}   }
        else
        {  [currenTimer invalidate];
           durationTime = _time;
           [btn setTitle:@"开始" forState:UIControlStateNormal];
           //Some codes about set background image and text color of button omitted here
           NSString *hour = [[_time componentsSeparatedByString:@":"] objectAtIndex:0];
           NSString *minute = [[_time componentsSeparatedByString:@":"] objectAtIndex:1];
           label.text = [NSString stringWithFormat:@"剩余时间：
                        %@小时%@分钟 00 秒",hour,minute];
           //stop playing radio
           if(_radio == nil) { return; }
           if([_radio isPlaying]) {[_radio pause];}       }
    }
```

(4) The module of radio program reservation: we get the reservation settings from the radio reserving user interface, and add a local notification to the device notification center or cancel the notification from the device notification center, where the notifications are distinguished by the userInfo field. The main code of this module is as follows.

```
    -(void)orderRadioPro:(UITapGestureRecognizer *)sender
    {
        UIImageView *clockView = (UIImageView *)sender.view;
        NSArray *arr = [_radioProDataController getRadioProListObject:clockView.tag];
        NSInteger isOrdered = [[[arr objectAtIndex:0] objectForKey:@"isOrdered"] intValue];
        NSString *progName = [[arr objectAtIndex:0] objectForKey:@"progName"];
        NSInteger progId = [[[arr objectAtIndex:0] objectForKey:@"progId"] intValue];
```

```
NSString *beginTime = [[arr objectAtIndex:0] objectForKey:@"beginTime"];
NSString *endTime = [[arr objectAtIndex:0] objectForKey:@"endTime"];
RadioPro *radioPro = [[RadioPro alloc] initWithName:progName proId:progId
                        begTime:beginTime endTime:endTime isOrdered:isOrdered];
UIApplication *application = [UIApplication sharedApplication];
NSArray *scheduledLocalNotifications = [application scheduledLocalNotifications];
if(isOrdered)
{   for( UILocalNotification *myLocalNotification in scheduledLocalNotifications)
    {   if([[myLocalNotification.userInfo objectForKey:@"预约通知"] intValue] == progId)
        {   [application cancelLocalNotification:myLocalNotification];   }
    }
    clockView.image = [UIImage imageNamed:@"alarm_unselect.png"];
    [_radioProDataController updateRadioProListObject:clockView.tag IsOrdered:NO];     }
else
{   clockView.image = [UIImage imageNamed:@"alarm_select.png"];
    [_radioProDataController updateRadioProListObject:clockView.tag IsOrdered:YES];
    [[UIApplication sharedApplication]
     scheduleLocalNotification:[CommonHelper createNotificationWithRadioProg:radioPro]];   }
self.radioProList = [_radioProDataController getRadioProList];
}
```

(5) The module of application settings: we achieve binding microblogging according to kAppKey and kAppSecret in CNR wake-up mobile application by virtue of the third-party technology ShareSDK, and calling the agent about logIn to bind microblog in the first login. When posting microblogging, the application will directly call the distribution method publishWeibo to share microblogging.

```
-(void)boundWeiBo:(id)sender
{
    switchBound = (UISwitch *)sender;
    if(switchBound.tag= =1)
    {   if(switchBound.on)
        {   [_appDataController updateAppShareListObject:switchBound.tag IsBounded:YES];
            SinaWeibo *sinaweibo = [self sinaweibo];
            //binding sinaweibo
            [sinaweibo logIn];        }
        else
        {   [_appDataController updateAppShareListObject:switchBound.tag IsBounded:NO];
            SinaWeibo *sinaweibo = [self sinaweibo];
            //cancelling binding
            [sinaweibo logOut];       }      }
    else if(switchBound.tag= =2)
    {   NSLog(@"********网易微博*********");
        if(switchBound.on)
        {   [ShareSDK authWithType:ShareType163Weibo
            result:^(SSAuthState state, id<ICMErrorInfo> error) {
            if (state == SSAuthStateSuccess)
            {   [_appDataController updateAppShareListObject:switchBound.tag IsBounded:YES];
                self.appShareList = [_appDataController getShareList];
                [_myTableView reloadData];
                NSLog(@"成功");      }
            else if (state == SSAuthStateFail)  {   NSLog(@"失败"); }   } ];  }
        else
        {   [_appDataController updateAppShareListObject:switchBound.tag IsBounded:NO];
            [ShareSDK cancelAuthWithType:ShareType163Weibo];      }
```



```
    }
    else { //Some codes about binding  tencent weibo is omitted here due to space limitation }
}
```

5. Running Results of the Application

We show some of the running results of the achieved iSO-based CNR wake-up mobile application as follows.

(1) Homepage window: As shown in Figure 3, the homepage window provides users to listen to radio, displays the current time in the form of a dynamic calendar, and is used as the entry to main function modules.

Figure 3. Generated Homepage Window

Figure 4. Generated Window about Alarm Management

(2) Alarm clock management: As shown in Figure 4, the window of radio alarm clock management consists of a form for adding radio alarm and a navigation bar for editing alarm. Users can add, delete, modify radio alarm, as well as start and turn off the alarm. By adding radio alarm clock, the user can listen to the radio to get up.

(3) Bedtime timer by radio: As shown in Figure 5, the window of bedtime timer is composed by the form of setting properties and the form about countdown. Users can listen to their favorite programs to fall asleep by bedtime settings.

Figure 5. Generated Window about Bedtime Settings

Figure 6. Generated Window about Customized Programs

(4) Radio program reservation: As shown in Figure 6, the window of reserving programs consists of the navigation bar, the time bar and the program list. Users can reserve and listen to the favorite radio programs according to program list.

(5) Application settings: As shown in Figure 7, the window of application settings is composed by the form for microblog binding and sharing and the form for network settings. Users may bind sharing platform account, and can share microblog without logging on sharing platform the next time, and set whether to use only in WIFI network mode.

Figure 7. Generated Window about Application Settings

6. Conclusion

In this paper we proposed an iOS-based mobile application on alarm clock with online radio support, and we presented its functional and structural analysis and database design, and elaborated its schematic design and detailed design, and illustrated its running results. Existing alarm clock application is not favored by many users because of functional singleness. Compared with existing alarm clock application, the proposed iSO-based CNR wake-up mobile application provides more functionality such as auto-playing radio, setting radio alarm clock, turning off the alarm by shaking iPhone, sleeping timer by radio, reserving radio program, binding and sharing microblog. Thus it provides users with more choices for alarm clock and allows the user's life more comfortable, pleasant and colorful. The practical test from some people shows that the proposed application is very popular with users, which also illustrates its practicability and effectiveness.

Acknowledgements

The work is supported by the National Nature Science Foundation of China (No.61303093), the Natural Science Foundation of Henan Educational Committee (No.12B520068) and the PHD Foundation of Zhengzhou University of Light Industry (No.2011BSJJ002)

References

[1] C. Liu, K. Holroyd, Q. Zhu, K. Shen and W. Zhou, "Design and implementation of a behavioral migraine management iPhone app for adolescents with migraine", The 11th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, **(2010)** June 14-17; Montreal, Canada.
[2] E. Carillo, R. Kadobayashi, M. Chikama, S. Shimojo, K. Rodriguez-Echavarria and D. Arnold, "Making history alive and interactive: designing an iPhone app to present the summer war of Osaka Byobu", The 16th International Conference on Virtual Systems and Multimedia, **(2010)** October 20-23; Seoul, Korea.
[3] J. Liu, Y. Qin and J. Xu, "The Design and Implementation of B2C Web site-Online Mobile Phone Shop", International Journal of u- and e- Service, Science and Technology, vol. 6, no. 2, **(2013)** , pp. 107-114.
[4] N. H. Liu, C. Y. Chiang and Y. H. Wu, "User Interface of Assistant Navigation System in Smart Phone for the Blind", International Journal of u- and e- Service, Science and Technology, vol. 6, no. 4, **(2013)**, pp. 1-12.
[5] H. F. Zhang and Y. M. Mou, "Introduction to Software Engineering, 6th edition", Tsinghua University Press, Beijing **(2013).**
[6] G. Allen and M. Owens, "The definitive guide to SQLite, 2nd Edition", Apress, Berkeley **(2010)**.
[7] L. Hao, "Application of MVC Platform in Bank E-CRM", International Journal of u- and e- Service, Science and Technology, vol. 6, no. 2, **(2013)**, pp. 33-42.
[8] J. Conway, A. Hillegass, "iOS Programming: The Big Nerd Ranch Guide, 3rd Edition", Addison-Wesley Professional, Indianapolis **(2012).**
[9] S.G. Kochan, "Programming in Objective-C, 4th Edition", Addison-Wesley Professional, Indianapolis **(2012).**

Authors

Li Canlin, he received PhD degree in computer science from Shanghai Jiaotong University in 2010. Since 2010 he has been in the School of Computer and Communication Engineering at Zhengzhou University of Light Industry. His research interests include multimedia, digital entertainment, software engineering and wireless network. Dr. Li is a member of IEEE as well as ACM.

Jin Baohua, he is a professor of School of Computer and Communication Engineering at Zhengzhou University of Light Industry. His research interests include artificial intelligence, computer decision support system, computer software and theory and emergency management.

Cao Wenjie, he received the BS degree from Zhengzhou University of Light Industry in 2013. His research interests include software engineering and wireless network.