International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 7 4607 - 4616

Design and Implementation of Running Support System by Providing Common Routes of Runners

Zhuoyi Cui
Graduate School of Science and Engineering
Doshisha University, 1-3, Miyakotani, Tatara
Kyotanabe, Japan
e-mail:zcui@ishss10.doshisha.ac.jp

Hirohide Haga
Graduate School of Science and Engineering
Doshisha University, 1-3, Miyakotani, Tatara
Kyotanabe, Japan
e-mail:hhaga@mail.doshisha.ac.jp

***Abstract*—**This paper proposes the design and implementation of running support system using mobile devices. As technology continues to develop extremely quickly, new technologies provide more convenience, but their evolution simultaneously includes many problems. One of these problems, about which people are becoming more concerned, is the lack of exercise to maintain his/her health. Cellphones and smartphones have already become more intelligent and indispensable, and mobile applications (APPs) are no longer novel. Based on the integrated, multi-functional, personal customization, and other unique advantages of applications, modern society requires an exercise application that includes interaction, competition, and quality communication to encourage people to do more exercising. Such applications must motivate people to communicate with each other easily and discuss the daily process of their exercising. Such mobile applications might resemble a new kind of fitness application software that assists people become healthier. First, this paper performs survey research on the demand for health in daily life and what kind of exercise applications people might use to become healthier by smartphones. Our survey includes both people in Japan and from other countries. Through survey management, considering some specific samples, and analyzing their characteristics and flaws, we easily summarized and improved our application software. Next, based on the research results, it was not hard to find identical parts about the concepts of other exercise application software; we also explain our new design concept APP. Our explanation includes an automatic algorithm that generates a common path and a determination algorithm, both of which generate new fitness applications. In general, through these two algorithms, a common path was generated. In other words, the common path is the main concept that concerns new sport applications built on mobile phones. Finally, based on our design concept and the above algorithms, we implement fitness applications on an iOS that we call Run-Map-APP.

Keywords*-Running support; Mobile Device; Healthcare; Running; iOS; SQLite; Augmented Reality*

*****

I. INTRODUCTION

A running map application on mobile devices [1] focuses on fitness tracking in combination with the Internet. This mobile application can track exercise data with a GPS-enabled phone like other social sport applications. The tracking data are kept on websites to monitor performances over time. Another function is that users can get support from Internet communities to inspire them to stay motivated. If Facebook and Instagram friends are less active, new group members might continue to encourage users to exercise. In this way, users can relax and focus on their progress and share tips with others. By tracking and analyzing performances, sharing workout data and photos with friends, and most critically, people will exercise more and become healthier. Run-Map-APP shares such similar functions with other exercise APPs but also through professional researches, we produced a new creative function to find common paths of several users. When people walk or run, Run-Map-APP measures their route, their speed, and heart rate information and follows their progress. It records this information and matches people who like walking or running on the same street. Run-Map-APP can send messages or displays to the screens of cellphones and notices about people near users. Since Run-Map-APP's main approach is to encourage and facilitate people who like exercising to get together, it also needs an incentive mechanism. Based on the results of our investigation, user incentive mechanisms are determined by running or jogging miles. Due to the technical progress of the running support applications, improvement systems are included in role-playing games (RPGs). On the other hand, social community elements are introduced on websites that enable RPGs' players to interact with other players. The users eventually become willing to exercise more, and the applications encourage and attract more users. Another advantage is making healthier bodies.

In recent years, social progress and the improvement of living conditions have made millions of people's lives more convenient and comfortable. Unfortunately, people are spending less time exercising. Many people have overly sedentary office jobs and the lack of physical activity has become a huge public health issue. The incidence of obesity, diabetes, and cardiovascular disease is increasing. Based on disease statistics, health problems and loss of life are also increasing. Steven Blair, an American sports epidemiologist, warns that less exercise and inadequate physical activity are going to become the most serious public health problems in the 21st century [2]. Underlying such a bleak premise, more people are focusing on the topic of sports and exercise; regular exercise can improve skills and fitness and provide a variety of health benefits to the human body.

Running and jogging are simple exercises that do not require any equipment or specific places; both activities can be

 ISSN: 2321-8169
4607 - 4616

enjoyed anytime and anywhere. Daily running, which only requires 30 minutes, is a very popular activity that not only provides a good healthy effect but it can also ease work-related stress. Even though running and jogging are simple and effective forms of exercise, compared to such competitive sports as football and basketball, they fail to garner much attraction from participants. Unlike other sports that promote communication during games, running does not trigger heated discussions after it is finished. The sense of achievement is not very strong. However, even though running as a healthy sport is easily accepted by people, it is also difficult to ask people to persevere, especially those who do not want to exercise daily. In this article, based on the lack of attraction by participants as a starting point of running and the existing mobile applications, running map applications add interactive communication and share sport experiences in communities. Also, according to the sport application's new design, our implement fitness application, Run-Map-APP, is more creative and attractive.

This article is structured as follows: Chapter 2 describes previous research and instructions about exercise APPs. Chapter 3 shows our design idea and the system structure of incentives. Chapter 4 shows our iOS application module implementation. Chapter 5 explains the system test results. Chapter 6 summarizes this paper and presents future works.

II. Related Works

The research direction of this article is the design and implementation of sport support software that runs on an iOS platform. We conducted a detailed investigation of relevant and similar software and analyzed its advantages and disadvantages.

A. *Edomondo*

Endomondo [3] is an application that is suitable for running, cycling, skiing, etc. Friends of its users may share motion data and routes on Facebook. Their friends may encourage other users. Endomondo has a professional sports data track and can scientifically calculate the expended calories, hydration rate, and several statistical data. All Endomondo data are based on people's body weight, gender, age, and running direction. Sport exercises require route and speed records during such processes as running, walking, bike riding, horse riding, skiing, and skating. Other sports do not require such route and speed records, including fencing, dancing, diving, swimming, volleyball, and sailing. They are shown in Fig. 1. Endomondo is a kind of comprehensive assistant sport application.

Figure 1. Edomondo.

One of Endomondo's shortcomings is that it can only be shared on Facebook. The other is that maps in Endomondo are not always accurate.

B. *Nike+Running*

Nike + Running [4] is another sports support software developed by a company with a high reputation in the field of sport. Like other sports support software that records information on runners, this application can also be combined with a unique running chip to record more accurate record information, such as the number of runs, the accumulated mileage, running medals, and the feelings of runners; all of the above advantages are traits of the Nike + Running application. Nike + Running constantly inspires users to challenge themselves. In addition, in recent years, the software is paying more attention to social communication around runners and offers its own friend list in the software. Its users can share running records in SNS communities. The friends of the users in the Nike + Running community who notice a runner who is exercising can immediately receive cheers with others. Such encouragement highlights the accomplishments of users (Fig. 2).

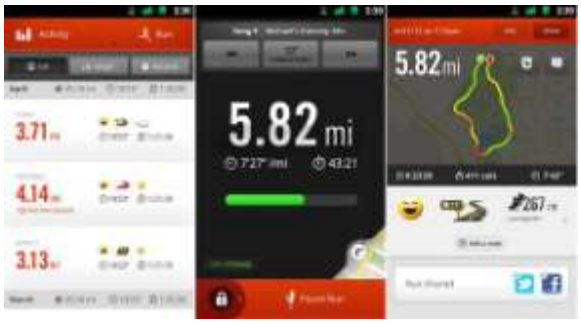

Figure 2. Nike+Running

Nike + Running is very good sports support software, but unfortunately it suffers from many drawbacks. For example, it cannot supply accurate positioning or generate running routes, and the routes are not always accurate when users go around buildings, over hills, and along rivers. Its design concept focuses on individuals, and the social interaction appears slightly thin.

C. *Codoon*

Codoon [5] is Chinese sports motivation software. Like Nike + Running, it also uses a combination with special hardware and software and exploits people who wear rings to increase the records of its users' sport data. Users also can upload and process the data on applications. In addition to calculating basic running data, Codoon can also track precise movement routes, distances, speeds, and monitor altitudes through GPS global positioning technology. Users check locations and charts and can accurately calculate the amount of expended calories. Furthermore, a variety of customizable exercise patterns, goals, and challenges can be chosen on Codoon. Codoon automatically and synchronously uploads the routes of running and data to its website and shares sporting procedures on popular social network websites. On the other hand, more people inspire users to make great improvements. Such functions strengthen the application's interactivity. This point completely caters to the needs of modern runners (Fig. 3).

 ISSN: 2321-8169
4607 - 4616

Figure 3. Codoon

As a personal and professional running application, Codoon's achievement is impressive. Codoon has more useful functions than others sport applications. For instance, it caters to people who want to run and provides good interaction and a mechanism that rewards running as well as is suitable for individual workouts. However, Codoon is less effective for two or more people who do sports outdoors.

D. Incentive for sports

Incentives for sports were mentioned in the beginning of this article. Running is a simple and effective method of exercise, but compared to such competitive sports as football and basketball, it fails to get enough attraction from participants. Based on this inability to attract attention, all three above APPs have difficulty properly raising running's appeal to users and the corresponding processing. For instance, Nike + Running advocates that users effectively spend time on applications and uses a level system that often appears in RPG games. After people finish exercising, they might have a greater sense of achievement. Codoon also uses the same technique by utilizing online social communities and running incentives, and after users are done exercising, they might also feel a greater sense of achievement. But most of these incentive settings are merely self-discipline. People who stress self-discipline can be encouraged to keep exercising through incentive mechanisms. This above content offers a good promotion effect in the early stages, but when people adapt such a mechanism, the focus on interactions on Facebook and blogs will decrease and the effect on running will gradually be reduced.

Are there any more effective incentives for running/jogging except reward mechanism? Football, basketball, and other sporting projects emphasize specific skills and competition is widely embraced. Electronic sports (E-Sports) simply exploit this point as a specific skill and competition has risen in recent years [6]. More people are being attracted to E-sports.

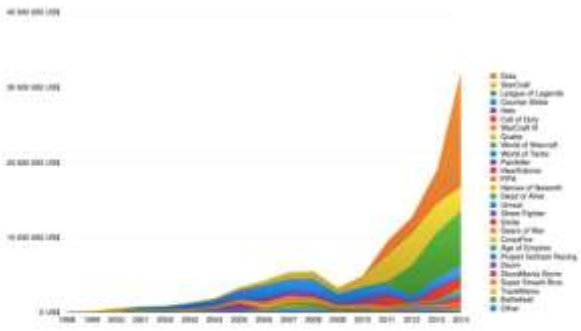

Figure 4. Esports tournament prize amout, 1998-2014 (from [7])

E-sports bonuses are increasing, suggesting that e-sport are becoming more popular. One factor cannot be ruled out; the graphics have become more magnificent and realistic. The skill and feeling of the competition supply enough satisfaction. This feeling is more than a simple achievement mechanism. Dota [8], StarCraft [9], and League of Legends [10], which are the top three e-sport games, are highly competitive and popular Jogging, unlike other closed running movements, does not require a site with competitors and lacks competition that brings people much pleasure. In reality, even though people want to improve this issue, it is almost impossible. This is one basic reason for the lack of attraction to jogging. Based on this reason, this research is different from other similar jogging software because it presents a specific perspective about more effective incentive mechanisms that will be shown in a later chapter.

III. DESIGN CONCEPT OF RUN-MAP-APP

Even though the concepts of running/jogging and racing are very similar, there is one main difference; racing is more competitive, which is what this research wants to achieve in Run-Map-APP. This chapter analyzes the basic concept of racing by the concept of common paths. This article will explain racing's design concept, which has already been input into Run-Map-APP.

A. Necessary conditions for competition

A race is a running event in which more than two people simultaneously start at the same place and advance on the same path to reach identical destinations. The duration of the interval determines the competition's outcome. On the other hand, no competition exists during running/jogging because it is an individual form of exercise. Running / jogging is not realistic if people want to engage in competition. Hence, users require an APP on mobile phones which virtually implement the situation of competition.

The following three problems must be solved to realize competition:

1. **Judgment**: Judging the winner among all of the participants,
2. **Time**: APP's design focus promotes daily exercise, so people cannot be required to exercise at the same time.
3. **Path**: The purpose is the same as the time.

Although the application encourages people to develop a habit of running, it cannot require all people to arrive at a fixed location or to exercise by a fixed path. Especially for people who often run/jog, they already have a fixed route,

 ISSN: 2321-8169
4607 - 4616

and if the application program generates a competition path, it may exert the opposite effect and discourage participation.

B. Common Paths

In the real world, it is impossible to make one track that satisfies everyone, but with applications, users can create their own virtual routes. The application automatically chooses a running path that is near the user homes and can also collect running route data around them. The application can identify the common points from nearby users. These common points are the basis of a game's virtual path called a common path.

1) Generating a common path

If the application treats users as a center, the existing path directly displays any path within a 1 km area. If no path exists, it generates a common path that must satisfy the following points:

- When runners are placed at the corner, there are more than two paths within 1 km
- The path data are recorded by different participants
- Paths must be entered within a week
- There are more than 300 meters of similar coordinates.

Figure 5. Generation of common paths

For example, Fig. 5 shows four-color lines that represent the running/jogging paths of different people. The light blue area represents the four participants' common running paths that are automatically generated by the application. The specific generation methods are explained in Chapter 4.

2) Display of common paths

The common path in the user display interface only expresses itself and the user running routes (Fig. 6). The user interfaces show only the generated blue part of the common and blue paths of the user running routes.

Figure 6. Display of common paths

3) Calculating time through common paths

Figure 7 shows green and red ovals as the beginning and the end of a common path. The timing application begins when runners enter the green part and leave the red part or switch out of each part. After the user completes the above process, the application records and saves the time in a database. This entire procedure is deemed the only method for calculating that users have gone through the common path. Other situations cannot be recorded.

Figure 7. Calculating time through common paths

4) Cancellation of common paths

Within two weeks, if fewer than five people are using a common path, the application will cancel it.

5) Ranking common paths

When the users go through a common path, the application displays a list of those who take the shortest time.

This application establishes a virtual racing track and lets users challenge each other while running/jogging and encourage them to keep running. If the list of rank data is lost, the ranking list is refreshed every week.

 ISSN: 2321-8169
4607 - 4616

In Case one, some users treat ranking lists as if they are Olympic records. They make lists with no change for a long time so that the runners may be discouraged.

Figure 8. Screenshot of "Temple run" (from [11])

In Case two (Fig. 8), screenshots are seen of iOS's Game Center, which is a very popular game on iOS. Breaking into the top 10 scores is very difficult. The application also prevents users from driving or operating vehicles on the common path to exhaust remarkable achievements.

6) Statistics using time through common paths

Every time a person goes through a common path, the application counts the time and collects statistics for the convenience of other users who demand information about the use of common paths. Users can utilize the time distribution to appropriately adjust their own running times (Fig. 9).

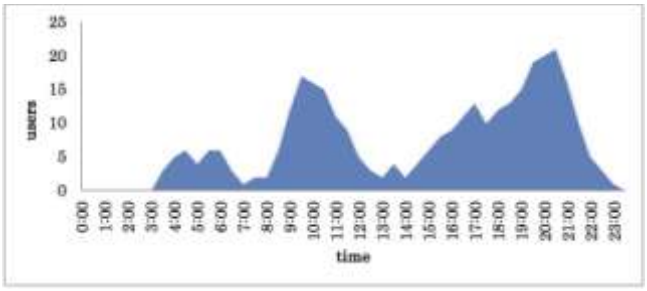

Figure 9. Statistics using time through common paths

Figure 9 shows the example of statistics using time through common paths. Users might adjust their running time to 10 o'clock in the morning or to 8 o'clock in the evening. Most people prefer to exercise during these two times. In this way, they don't need to change their own running routes, which encourages them to meet more people rather than merely relying on network interaction. The people who enjoy running by themselves can also reduce the chance of meeting others on the street.

C. Other settings

In addition to common paths, this article calculated the running distance, the running time, and the average running time. People can use this running software without using the common paths.

D. Summary

By building the common path proposed in this chapter, our application puts into the virtual world a running race that has the most time and path requests. As Fig. 10 shows, the application combines people in the same time axis through lists and common paths in different times and different running paths [12]. By allowing people to communicate with others without improving their original running routes and time arrangements, users can also challenge themselves or the running records of others with this application. Its purpose is to inspire people to run faster.

Figure 10. Common path images (from [12])

IV. IMPLEMENTATION OF IOS APPLICATION

A. Introduction to iOS development

iOS is an operating system that runs on Apple's mobile devices such as iPad, iPhone, and iPod. Operating systems control the device hardware and provide the technologies that are required to implement native applications. They are also equipped with several system applications, such as Phone, Mail, and Safari that provide standard system services to users.

In addition, the iOS Software Development Kit (SDK) includes the tools and interfaces needed to develop, install, run, and test native applications that appear on an iOS device's home screen. Native applications are built using the iOS system frameworks and Objective-C language and run directly on iOS. Unlike web applications, native applications are installed physically on a device and are therefore always available to the user, even when the device is in Airplane mode. They reside next to other system applications, and both the application and any user data are synced to the user's computer through iTunes on Macintosh personal computer.

The iOS architecture is layered, and at the highest level, iOS function as a medium between the fundamental hardware and the created applications. Applications do not directly communicate with the fundamental hardware.

 ISSN: 2321-8169
4607 - 4616

Instead they intercommunicate with the hardware through a set of well-defined system interfaces that simplify writing applications that work systematically on devices with dissimilar hardware capabilities.

The effectuation of iOS applied sciences can be viewed as The Pyramids, which owns a set of layers. For example, Fig. 11 shows that lower layers include rudimentary services and technologies. Higher-level layers, which are built upon the lower layers, provide more advanced services and technologies.

The **Cocoa Touch layer** includes the main frameworks for building iOS applications. These frameworks delimitate the visual aspects of applications and supply the basic application substructure and support such main technologies as multitasking, touch-based input, push notifications, and many high-level system services. When contriving applications, people ought to investigate whether the technologies in this layer satisfy their requirements.

The **Media layer** includes graphics, audio, and video technologies. People implement multimedia experiences in their applications. The technologies in this layer make it easy to establish applications that look and sound great.

The **Core Services layer** includes the basic system services for applications. One key among these services is the Core Foundation and Foundation frameworks that define the basic types used by all applications. This layer also includes specific technologies to support features, such as location, iCloud, social media, and networking.

The **Core OS layer** includes the low-level features that most other technologies are based on. Even if people do not use them directly in their applications, they will want to use other frameworks most. In situations where a person needs to expressly deal with security or communicate with an external hardware accessory, she uses the frameworks in this layer.

Figure 11. Layers of iOS

B. SQLite

SQLite [13], which is a small database that follows the relational database management system of Atomicity, Consistency, Isolation, Durability (ACID), is contained in a relatively small C programming library. The SQLite program, built by D. Richard Hipp, is in the public domain. Its design is embedded and has been used in many embedded products. SQLite costs fewer resources in embedded devices and only consumes a few hundred K Bytes of memory. Most mainstream operating systems can be supported by SQLite, such as Windows/Linux/Unix. SQLite can also be combined with many other programming languages, such as Tcl, C #, PHP, Java, etc. Also for comparison with MySQL [14] and PostgreSQL [15], both are famous database management systems all over the world that people can easily find online. Its processing speed is also faster than Mysql and PostgreSQL.

Therefore, based on these SQLite advantages, Run Map's relational database management system should use SQLite.

SQLite is a small relational database. iOS SDK supports the SQLite form in the very early stage. When people use SQLite underneath iOS SDK, they only need to join `libsqlite3 dylib`, which depends and introduces header file `Sqlite3.h`. However, the original SQLite API is quite unfriendly and very inconvenient when people are using it. In the open source community, FMDB (Flying Meat Data Base) [16] encapsulates the SQLite API in a series of libraries and is deemed outstanding by the open source community. Fig. 12 shows its simple use cases.

```
NSString* do csdir = [NSSearchPathForDirectoriesInDomains(NSDocmentDirectory,
NSUserDomainMask, YES) lastObject];
NSString* dbpath=[do csdir stringByAppendingPathComponent:@" user.sqlite"];
FMDatabase* db=[FMDatabase databaseWithPath:dbpath];
[db opne];
FMResultSet *rs=[db executeQuery:"@ select * from people"];
while ([rs next]) {
  NSLog(@"%@%@",
    [rs stringForColumn:@" firstname"],
    [rs stringForColumn:@" lastname"]);
}
[db close];
```

Figure 12. Simple use cases

C. APP structure

The core functional groups of Run-Map-APP must be divided into two parts: a logic control class and a calculation analysis function class. The control class is in the Class group, and the functional class is in the Common Tool group. The App structure is shown in Fig. 13.

Figure 13. APP structure

In addition to the main function of the above groups, APP also includes the Obj, View, and Kernel Algorithm

 ISSN: 2321-8169
4607 - 4616

groups. The SDK groups mainly use the SQLiteFMDB database engine. Obj only includes a temporary data object for the moment. The View groups contain some custom UIs. The Kernel Algorithm stores some related auxiliary algorithms.

D. APP framework structure

Run Map APP uses the Model View Controller (MVC) to design patterns, which are driven by a logic control UI to connect the data. The APP UI structure belongs to the stack structure of hash. On the other hand, based on the pattern, `ViewController` is a root container, and `NaviationController` is a classification stack container. The application can implement different interfaces that jump to each other. The application framework structure is shown in Fig. 14.

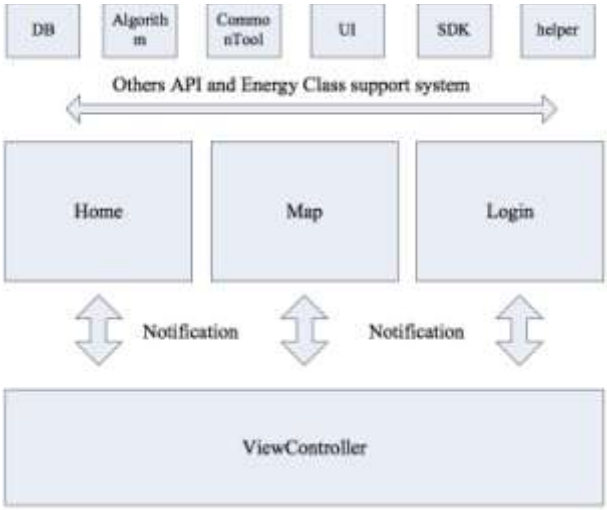

Figure 14. Framework sketch

1) Main Control classes

The main control classes in the class groups are divided into two parts: the logic control class among all the interfaces, and the business logic control class called the `MMapController` class that can directly manipulate the UI of maps.

a) Logic control class among interfaces at all levels

The logic control class is among the interfaces at all the levels. The logic control class at all the level interfaces includes `HomeViewController`, `LoginViewController`, and `MapViewController`. They are mainly responsible for dealing with navigation relations among the interfaces at all levels and the UI inside the interface displays the logic of interface. For receiving a notice from others, Run-Map-APP uses the `NSNotificationCenter` to implement.

b) MMapController

In the APP, the `MMapController` Class is the logic of the control class that only directly operates the UI map. This class involves the `MMapController` class and directly operates the UI maps, but `MMapController` also operates the data interactions with `DBAction`. First, to introduce the main functions of the MMapController, an interaction bridge connects the maps and APP and exploits MapKit, which supplies API to realize the basic operation on maps. MMapController is shown in Fig. 15.

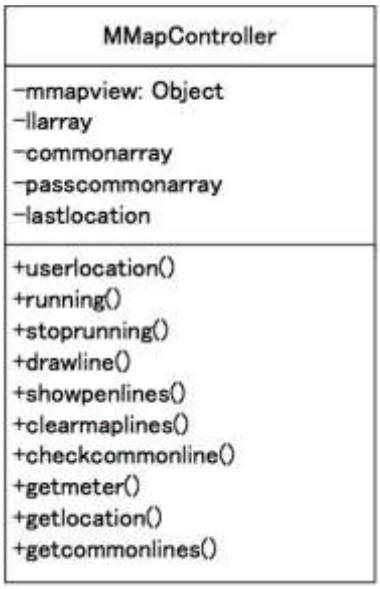

Figure 15. MMapControoller

The following are `MMapController`'s main properties:

- `mmapview`: inherits `MKMapView` to display system maps.
- `llarray`: records the coordinate arrays about the number of users who go through the path while running.
- `commonarray`: saves the coordinate arrays based on the calculation of the common path.
- `passcommonarray`: saves data about users through the common path.
- `lastlocation`: users locate the last position to confirm their mobile node.

The following are `MMapController`'s main methods:

- `userlocation()`: This method uses `MKMapView`, which has its own method, to get the user's current location. The `MKMapView` benefits only bring a positioning method when the user coordinates change, triggering a callback method to get the appearance of the coordinate noise that is decreasing.
- `running()` and `stoprunning()`: By sending notices, the application on cellphones achieves user starts to run and stop in normal life. During the running's start, it deletes the initial running data.
- `drawline()`: The `MKMapKit` method changes the two coordinate points into a straight line.
- `showopenlines()`: The `showopenlines()` method realizes two functions. The first, the common path data, is removed from the database and drawn on the map, and the second is `showopenlines()`, which uses the algorithm to get the intersection parts of all the frequently-used paths, namely, the common path. Following is an algorism of finding common paths. (1) Get two paths and coordinate arrays *A*: $\{A_1 \ldots A_n\}$,

 ISSN: 2321-8169
4607 - 4616

$\{B_1...B_n\}$, P' is the latitude and longitude offset, and $C_n = \{\}$ is a common path array.

- (2) The common path algorithm, $C = A \cup B$, is the offset from point to point and determines whether points A_x and B_x are identical. If there is a point identical to coordinate point A_x, the key point is A_x. Then it builds a collection of similarities $\{B_x...B_y\}$. The realized algorithm is shown in Fig. 16.

```
map Amap{ }
for (A) {
  array Axary{ };
  for (B) {
    if (like(Ax, Bx)) {
      Axary.add(Bx);
    }
  }
  if (Axary != NULL) {
    Amap.set(Axary);
  }
}
```

Figure 16. Common path algorithm

- `checkcommonlines()`:This method judges whether the user goes through the common path. The algorithm uses two time nodes (Fig. 7 shows green and red parts) *intime* and *outtime* to determine whether users go through the common path. When the judgments to the common path go through the first time point, it empties the uncompleted original data information and resets the *intime*. When `checkcommonlines()` detects a user of the node at the end on the same common path, it may set the current time to the *outtime*, and the process above expresses a complete common path calculation and records the relevant data through the common path.

2) Main helper classes

In this application, the main objective of the function is to simplify the code. At the same time, the main helper classes reduce the similarity between business and logic. The second objective, the main helper classes, can strip the mode layer from the control layers. In `CommonTool`, a function class mainly includes `CommonTool`, `DBAction`, and `LocationHelper`. In addition, the `CommonTool` class is a global public library function; all public methods are implemented with abstract class methods in this class.

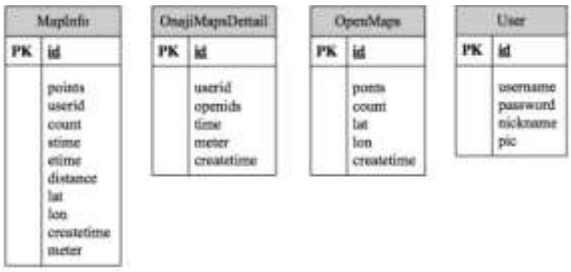

Figure 17. Database table structure diagram

There are two main kinds of methods in `DBAction`. The first is reading and writing methods that are associated with the user information; the other is reading and writing methods that are related to map data. `DBAction` mainly provides a way to register, check, and access user information. The map data class method mainly goes through the associated user ID and records in a corresponding table or queries the relevant user data under specific maps. In the map data table, the data of the path coordinates are converted into a JSON text format to record the new text format data.

V. FUNCTION TEST

A. Basic functions of system tests

This article tests the basic functions of systems, and the test results are shown in Figs. 18 and 19. The user and map interfaces are displayed well. Since there is no user information, the user and map interfaces cannot display anything related.

Figure 18. User Interface

Figure 19. Map Interface

B. Login function test

We tested the system's login function, and the results are shown in Figs. 20 and 21. The login function was achieved very well, and the generation of user information on the user interface showed out.

ISSN: 2321-8169
4607 - 4616

Figure 20. Login interface

Figure 21. Login successful interface

Our investigation tested the automatic generation function of common paths, and two paths were recorded at the end (Fig. 22).

path1：	path2：
longitude ,latitude	longitude ,latitude
34.799566,135.774041	34.797348,135.769414
34.799191,135.773282	34.797289,135.769442
34.798897,135.772831	34.796643,135.770107
34.798527,135.772209	34.796476,135.770239
34.798247,135.771538	34.796530,135.770344
34.798152,135.771307	34.796354,135.770536
34.797912,135.771043	34.796241,135.770894
34.797759,135.771296	34.796299,135.771246
34.797293,135.771620	34.796218,135.771582
34.798265,135.773381	34.796254,135.771736
34.798721,135.774547	34.796769,135.771708
34.799372,135.774222	34.797275,135.771631
34.799566,135.774041	34.797741,135.771296
	34.797908,135.771048
	34.798030,135.770899
	34.797605,135.770173
	34.797352,135.769414

Figure 22. Input path

The common path generations of the two path results are shown in Fig. 23.

Figure 23. Common path generation

As shown, the orange part is the entry path, and the blue shaded part is the automatically generated common path. The upper left corner is the user running timer, the orange button is the stop button, the user presses the stop button, the App ends the sport, and all the related items are recorded in the database. After testing all the functions, the APP operated well.

VI. CONCLUSION

This article designs and implements auxiliary sport software based on an iOS platform on smartphones. Based on the main idea of common paths, we designed and implemented a Run Map application, which is health assistant software that recommends that people exercise more. The following is a summary of the main innovations of this article:

- Developed multi-sensor applications on iOS.
- Proposed an incentive mechanism for becoming healthy using competition with other exercisers
- Designed and implemented health assistant software, mainly based on the idea of common paths.

On the other hand, although our original implementation purpose was just a prototype for our experiment, notable issues remain:

- Users get a better experience by linking to social network systems
- Designing a better algorithm generates more accurate common paths
- Adding music appreciation auxiliary functions improves the running experience of users.

Future work will investigate the influence of the heat of the hardware and the short battery life that concerns the software.

REFERENCES

[1] http://www.tomsguide.com/us/best-running-apps,review-2285.html

[2] Blair S. N. Physical inactivity: the biggest public health problem of the 21st century. *Br J Sports Med*, vol.43, No.1, pp.1~2, 2009

[3] Endomondo. https://www.endomondo.com/

[4] Nike + Running. http://www.nike.com/cn/zh_cn/c/running/nikeplus/gps-app

[5] Codoon. http://www.codoon.com

[6] Yuri Seo, Electronic sports: A new marketing landscape of the experience economy, *Journal of Marketing Managemen*t, Vol.29, Nos.13-14, pp.1542-1560, 2013

[7] E-sports Earnings. http://esportsearnings.com/

[8] http://blog.dota2.com

[9] http://us.blizzard.com/en-us/games/sc2/

[10] http://na.leagueoflegends.com

[11] Wikipedia.Templerun http://en.wikipedia.org/wiki/Temple_Run

[12] U-car. Jaguar virtual windshield display technology. http://news.u-car.com.cn/31334.html

[13] Michael Owens, *The Definitive Guide to SQLite*, Apress, 2006

[14] Charles Bell, Mats Kindahi, and Lars Thalmann, *MySQL High Availability: Tools for Building Robust Data Centers* (2nd edition), O'Reilly Media, 2014

[15] Richard Stones, *Beginning Databases with PostgreSQL: From Novice to Professional*, (2nd edition), Apress, 2007

[16] http://www.theappguruz.com/tutorial/sqlite-database-in-ios/