UDC 004.421

*Kadyrova A., Pradeep A.

Kazakh-British Technical University Department of Information SystemsManagement,
Tole bi, 59, Almaty, Kazakhstan
*e-mail:altynaykadyrovakbtu@gmail.com

Piu-an application to hire reliable and affordable driver for ios platform

Abstract. In this graduation diploma project I present a new mobile application for IOS Operating System in Xcode IDE (Integrated Development Environment). It is employed in order to overcome the problem of traffic flow in cities like Almaty and Astana. Moreover, I designed a software which enables us to easily find nearest drivers for passengers, and vice versa, by using Objective C programming language. Objective C can be very efficient there, since it is a general-purpose, object-oriented programming language and uses model-view-controller design pattern.My mobile application makes it possible for users, i.e. both drivers and passengers, to select trustworthy drivers via simplerequests.
Key words: Mobile application, IOS, iPhone, Objective C, MVC, Xcode IDE, Singleton pattern, MySQL, Storyboard, Laravel PHP Framework, JSON format.

Introduction

Nowadays, taxi applications play vital role in the whole globe, and Kazakhstan is not an exception.

In this regard, in the graduation diploma project is considered a mobile application which allows hiringdependable and affordable taxi driver. And name of that application is Piu.

The graduation project contains set of technologies which were used during the development phase of the application. For instance, I used the following key technologies such as Xcode, Objective C, Singleton pattern, Storyboard, and MVC (model-view-controller) pattern for the client side of the application. Server side of the application is done using LARAVEL 4 PHP Framework [1].Laravel is a free, open source PHP web application framework, designed for the development of MVC web applications. JSON (JavaScript Object Notation) format was used in order to communicate with server, i.e. the application sends POST query to a web page that has whole API behind it, and this page returns needed information in JSON format [2].So the application sends data that is stored in keys and values. Server returns data in the same way: keys and values. My application uses NSDictionaries and NSArrays for that purpose.

NSDictionary class represents an unordered group of objects; however, they associated each value with a key, which acts like a label for the value [3]. This is useful for modeling relationships between pairs of objects. Therefore, when server returns information to database, the application parses JSON to NSDictionaries and NSArrays.

Moreover, registration of users is done fully in the server side, because, it is important to draw your attention that we will not collect any extra information, i.e. credit card number, etc. from our users in IOS Operating System because we will obey the IOS policy.

Every user, including both drivers and passengers, can earn their own ratings and here the most challenging part of the algorithm of the project comes. Therefore, I am going briefly describe the calculation of the rating.

Evaluation and votes are roughly equal in importance of factors. To adjust these two values - you must bring them to the same scale. The maximum number of votes is always different, but a rating from 1 to N. For simplicity we take N = 10. Therefore, the problem is reduced to bring the number of votes by a percentage of the maximum possible among all users. For calculating the percent of the vote, we need the

 Printed in Kazakhstan

max function and logarithm. If we have already made equivalent to the number of votes and an assessment and decided lead to voice dial a scale of 1 to 10, it is sufficient to use a logarithmic scale.

So the code will look like this:

```
$rating = DB::table('users')
->select(DB::raw('((LOG(POW(max(count_votes), 1/10),count_votes))+raiting)/2 as actual_raiting'))
->orderBy("actual_raiting")
->where('id','=', $user)
->lists('actual_raiting');
```

Figure 1 – part of the code of calculating of rating

We get the root of the 10th degree of the maximum number of votes for the subsequent calculation of the logarithm. So we get a share of the number of votes a particular user of the maximum, reduced to 10. Fold with an average rating and divide by 2 - so we do correlate(see Figure 1). The algorithm of searching the nearest driver is that I take values of latitude and longitude of the users, and average radius of the Earth as 6371 (see Figure 2).

```
$user = Auth::user();
$lat = $user->current_lat;
$lon = $user->current_lon;
$user = User::select(
        DB::raw("*,
                        ( 6371 * acos( cos( radians(?) ) *
                          cos( radians( current_lat ) )
                          * cos( radians( current_lon ) - radians(?)
                          ) + sin( radians(?) ) *
                          sin( radians( current_lat ) ) )
                        ) AS users"))
        ->having("users", "<", "5")
        ->orderBy("users")
        ->setBindings([$lat, $lon, $lat])
        ->where('rd', '=', 1)
        ->get();
```

Figure 2 – part of the code of finding nearest driver

Conclusion

In conclusion, Piu application gives a benefit for drivers to enrich their knowledge about the city by GPS navigator on their cars.

For the first time, application will be fully in Russian language. In near future, we are going to make it in Kazakh and English languages.

In terms of CRM (Customer Relationship Management), we will collect comments from customers (driver has opened door, provided cold water or smoked), and will analyze feedbacks in order to make our clients happy.

The relevance of graduation diploma project and its deliverable, Piu application, is that both users – Rider (i.e. client) and Driver can make a bargain about price. We are confident that Piu will work in Kazakhstan i.e. people in Kazakhstan have a mentality to become a driver even though they have their own job positions. In other words, the project offers free market where clients determine their own price for the trip. That is a big advantage in our city, namely in Almaty.

References

1. Raphael Saunier. Getting Started with Laravel 4. 20th January 2014

2. JitendraKotamraju. Java API for JSON Processing: An Introduction to JSON. Published July 2013

3. Lime Jelly. NSDictionary and NS Mutable Dictionary. 23rd of November 2011

4. Stephen G. Kochan. Programming in Objective-C, 3rd edition. Pearson Education 2011

5. Bert Altenberg, Alex Clarke and Philippe Mougin. Become an Xcoder. 2008

6. Jeff Kelley. Advanced Objective. – P. 2014

7. Paris Buttfield-Addison, Jonathon Manning, Tim Nugent. Learning Cocoa with Objective – C, 4th edition. O'Reilly Media, 2014

8. Mattew Campbell. Objective – C Quick Syntax Reference. Appress, 2013

9. Carlos Oliveira. Objective – C Programmer's Reference. Appress, 2013

10. Matt Neuburg. IOS 7 Programming Fundamentals. O'Reilly Media, 2013