

Universiti Teknologi Malaysia Car Ride or Rental

Mohd Zafri Zulkipli¹ and Noraniah Md Yassin²

Department of Software Engineering, Faculty of Computing, Universiti Teknologi
Malaysia, 81310 UTM SKUDAI Johor, Malaysia

¹zafrizulkipli@gmail.com, ²noraniah@utm.my

Abstract. The purpose of this project is to investigate the problems existing in Universiti Teknologi Malaysia (UTM) students' method of finding a rental car or booking a ride. The analysis of these problems was needed as requirements to develop the UTM Car Ride or Rental (UTM CARRR) mobile application. With the current non-systemised operations car ride or rental in UTM students are restricted to seek out a car either by calling or messaging the student with a car henceforth called Owner. Students are finding it hard to find a reliable Owner that will lend the car for rental purposes. In addition, students have a distrust in the taxi system in UTM which compels them to seek out Owners but faced a problem when a car owner may be late in picking-up the passenger. Price range differs from Owner to Owner. This makes ride or rental service among UTM students hard to rely on since there is no specific price range. UTM CARRR was developed to assist both students who require this service and car owners in UTM in dealing with such problems. This system is designed to facilitate both ride and rental service between UTM students in one application, which makes it unique. UTM CARRR helps students place request for car ride or rental anytime anywhere this system offers and standardized the price using suitable metric.

Keywords: Extreme Programming, Agile, Student, Owner

1 Introduction

Nowadays, there are many UTM students located in Skudai, Johor, bring their own car to campus (herein referred as Owner). Some of these students give services such as renting their cars or taking ride request from one place to another. It is good for UTM students because they have a lot of choice in choosing the service offered by fellow UTM students themselves. However, sometimes it consumes time to find the right service.

For those who do not have cars but have a driving license, they most probably need a car for transportation. Students must make telephone calls or send messages to the driver to request for his/her service specifically. Sometimes this can pose problems such as the driver is not available at the requested time or the driver has discontinued his/her service. In view of such problems, a mobile-based car ride or rental system is suggested. With a mobile-based car ride or rental system, students have the power to make requests for rental or ride to all drivers available in UTM that register to this system. It is just as easy as just tapping their fingers on their smartphone.

This online ride and rental system is available for use 24 hours a day and seven days a week. UTM Students can make their request anytime. With the introduction of a mobile-based ride and rental system, drivers will also be able to increase their income by accepting customer request anytime.

2 Methodology

Extreme Programming (XP), a software development model which uses agile is chosen because circumstances for this project will require a model that is flexible all stages of development. It is a type of agile development

method. It is used to improve software project in a few important ways which is simplicity, feedback, respect and courage. XP implements simple phases which are planning, managing, designing, coding and testing (Extreme Programming Rules, 2016).

2.1 Planning Phase

Requirements for this project are gathered based from UTM students that has experienced rides on both taxi and ride services. Elicitation of requirements are also collected from those that have use car rental service in UTM.

2.2 Management Phase

For this project since there is only one developer developed this project, management is not something of concern.

2.3 Designing Phase

For this system, most of the code modules are revisited to improve their readability as well as remove codes that are duplicates. This makes for more efficient code and will ensure future developers that want to improve this system will have easier experience in understanding the structure of code for this system.

2.4 Coding

The system was developed after all requirements are gathered and analyzed. This is still within the realm of XP since as the coding begin, requirements will be revised again and again in designing phase. Figure 1 shows snippet of registering a new user.

```

1. public function register(Request $request)
2.     {
3.         $validator = Validator::make($request->all(), [
4.             'name' => 'required',
5.             'password' => 'required',
6.             'matric_no' => 'required|unique:users'
7.         ]);
8.         if ($validator->fails()) {
9.             return response()->json([
10.                'code' => 500,
11.                'message' => 'Failed to register you',
12.                'errors' => $validator->errors()
13.            ]);
14.        }
15.        if ($request->car_name != null) {
16.            $request['role_id'] = 3;
17.        }
18.
19.        User::create($request->all());
20.
21.        return response()->json([
22.            'code' => 200,
23.            'message' => 'Successfully registered to this system.'
24.        ]);
25.    }

```

Figure 1 Register Code Snippet

Register uses Validation static class to validate all the input from the user to ensure needed information are being filled. In line 3, the validator will check all the necessary inputs, if one fails then it will return error on line 9. Line 19 will create a new user using the User object and finally return JSON response on line 21.

2.5 Testing

Testing was done after most of the functions are completed. Upon testing if a found an error or not the desired output, then the function responsible for the testing will be refactored.

Black Box Testing was done on this project however the testing procedure is done by developer due to limited time and resources. Table 1 shows the Black Box Testing for registering new user.

Table 1(a) Register Test Case

Test Case ID	TCR1	TCR2	TCR3
Input/Action/Condition			
Name	Zafri	""	Zafri
Phone	0174403225	0174403225	0174403225
Matric No	A13CS0059	A13CS0059	A13CS0059
Email Address	zafri@gmail.com	zafri@gmail.com	""
Password	123	123	123
Re-Password	123	123	123
Car Name	Perodua Axia	""	""
Tap "Toggle" button	X		
Tap "Register" button	X	X	X
Expected Result			
"Register" button disabled		X	X
Navigate to Login View	X		
Actual Result			
"Register" button disabled		X	X
Navigate to Login View	X		
Result	PASS	PASS	PASS

Table 1(b) Register Test Case

Test Case ID	TCR4	TCR5	TCR6
Input/Action/Condition			
Name	Zafri	Zafri	Zafri
Phone	0174403225	0174403225	0174403225
Matric No	A13CS0059	""	A13CS0059
Email Address	zafri@gmail.com	zafri@gmail.com	zafri@gmail.com
Password	123	123	123
Re-Password	1243	123	""
Car Name	Perodua Axia	Perodua Axia	Perodua Axia
Tap "Toggle" button	X	X	X
Tap "Register" button	X	X	X
Expected Result			
"Register" button disabled	X	X	X
Navigate to Login View			
Actual Result			
"Register" button disabled	X	X	X
Navigate to Login View			
Result	PASS	PASS	PASS

3 Result

This application is developed using Ionic Framework 2 which is a hybrid mobile software development kit (SDK) for developing native mobile application (Ionic Documentation, 2017) and Laravel Framework 5.4 which is a framework for the language Pre-Hypertext Process (PHP) (Otwell, 2017). The source code for used for developing the mobile application was built on top of Ionic Framework.

Figure 2 shows the register interface, Figure 3 shows the login interface and Figure 4 shows the accept request interface.

The image displays three distinct screens from a mobile application. The first screen, labeled 'Figure 2 Register Interface', contains a registration form with fields for Name, Phone, Matric No, Email Address, Password, and Re-Password, along with a toggle for car rental services and a 'REGISTER' button. The second screen, 'Figure 3 Login Interface', features a car icon, input fields for Matric No and Password, and 'LOGIN' and 'SIGNUP' buttons. The third screen, 'Figure 4 Accept Request Interface', shows a request card with a car icon, destination, price, pickup time, and location, and a summary of the rental duration and price.

Figure 2 Register Interface

Figure 3 Login Interface

Figure 4 Accept Request Interface

By using this mobile application, user can request for generic transportation service requested by student in UTM which is car rental and car ride. User can easily accept the requests mentioned and keep track of how much expense they have spent using this application.

5 Discussion & Future Work

This system has very basic user interface design which is not appealing to many users. Besides that, it lacks feedback to user. For example, it does not have push notification service to notify a user if their request has been accepted.

Aside from that, since the user interface is not intuitive, user have mild difficult experience using this application. They need to first familiarize with the application before they can use it with confidence.

Some ideas have been suggested for future enhancement of UTM Car Ride Or Rental. For functionality enhancement, a better user interface design would increase the usability of the system. A notification service would also improve user experience.

References

- Extreme Programming Rules*. (2016, April 18). Retrieved from Extreme Programming : A gentle introduction: <http://www.extremeprogramming.org/rules.html>
- Ionic Documentation*. (2017, May 6). Retrieved from Ionic Framework: <https://ionicframework.com/docs/> Otwell, T. (2017, May 6). *Laravel - The PHP Framework for Web Artisans*. Retrieved from Laravel - The PHP Framework for Web Artisans: <https://laravel.com/>
- Rahimian, V., & Ramsin, R. (2008). Designing an Agile Methodology for Mobile. 1-6.
- Rouse, Margaret;. (2016, April 14). *systemsdevelopment life cycle (SDLC)*. Retrieved from techtarget: <http://searchsoftwarequality.techtarget.com/definition/systems-development-life-cycle>
- Wells, D. (2013, October 8). *Extreme Programming: A Gentle Introduction*. Retrieved May 19, 2017, from Extreme Programming: <http://www.extremeprogramming.org/>