**Project Construction - Stakeholder Product Review #2**

**Taxiping**

*“An affordable dispatching solution for small to mid-size taxi companies”*

**Group: SegFault**

**Alex Kelleher**

**Juan Poma**

**Kevin Stoner**

**Juntao Zhuang**

**Software Design and Documentation Sec. 1**

**November 23, 2015**

**Beta Release**

This beta release of our project covers most of the functionality outlined. In particular, the code covers the test cases with driver to dispatcher communication, dispatcher to driver communication, and administrative functionality outside of the analytics page. A few pieces of functionality are missing such as the driver being able to lookup an address, and the analytics page for administrators. Despite these drawbacks, the core functionality of the program brings it close to a state that is releasable. Major improvements over stakeholder one review include the “forgot password” page, the location tracking of drivers, the “take a break” and “picked up passenger” cases for drivers, and the “send notification” as well as responses between dispatchers and drivers. Overall we our proud of the program and are happy to present to you our beta release.

**Reviews**

**What MOAP said about our project:**

MOAP suggested first of all that we break up our User class into smaller classes. This was good advice, and we already were planning on breaking up the class into smaller classes. MOAP also advised that we be careful to hash our passwords. At the time, one of our developers was absent, but this turned out not to be a concern, as our passwords are indeed hashed and salted. As far as the send notifications, MOAP suggested that we use web sockets to give a live stream. We decided to use that advice and send the messages between drivers and dispatchers using websockets and a chat server. Lastly, MOAP critiqued us on keeping our general file sizes down and adding more comments to the code so that the the project is easier to read and keep track of.

**What we said to Runnr:**

Overall we thought that Runnr was doing a good job on their project and that in general their code looked pretty clean and organized. We also noted that their code matched their UML diagram which was good. One thing we suggested was that they alter the program to make less calls through the database in order to find a user. By doing this their program will be quicker and will not be bogged down by many users trying to access their information at the same time. As far as databases go, we also suggested that they consider having a database class for both the phone and the server, since for now they only have a database class for the server. We also suggested that they look into consolidating their Node.JS calls, since a lot of the code looked to be repetitive. We thought that perhaps there might be a more elegant way to prevent writing similar pieces of code over and over. FInally, we questioned the use case of moving to a new phone or switching between phones. We were not sure if the new phone would be able to pull data from the server, and also if it would lose data stored on the old phone.

**Source Code**

The source code is included in a zip file called “SefFault\_TaxiPing.zip”. Please unzip the zipped file and you will find the following directory structure:

* index.php/ -- this file contains code that will help on the installation of the

application

* install.php/ -- install the application
* sample\_usage\_localhost.html/ -- test file for driver and dispatcher
* config.php/ -- configures and helps in the installation of the application
* front\_end/ -- this file contains the front end
  + loginpage/ -- this page contains files for the login and logout page
    - login.css
    - login.php
    - validate\_login.php
    - register.php
    - change\_password.php
    - forgot\_password.php
    - reset\_password.php
    - new\_password.php
    - images/ -- contains the images that will be used in the front end design
  + logout.php
  + Admin.php
  + header.php
  + map.js
  + dispatcher.js
  + Dispatcher.php
  + Driver.php
  + map.css
* PHP/ -- this file contains the back end of the application
  + check\_logged\_in.php
  + DB.php
  + relations.php
  + taxiCompany.php/ --contains taxi company class
  + User.php/ -- contains Driver, Dispatcher, Admin, and User classes
* android\_code/ -- this file contain the android code that might be use in future implementations
  + backend\_cpp\_server
  + taxiPing
* cpp\_msg\_backend/ --this file contains the backend. C++ server handling interaction

between dispatcher and driver

* SQL/ -- this file contains the code for the database setup
  + init.sql
  + populate.sql

**Testing Strategy**

Testing web sites manually is a task prone to errors. Automating tests reduces the chance to get errors and makes tests more agile. For our testing strategy, a portable software testing framework for web applications called **Selenium** was used.

During our project development, the integration testing phase involves a good amount of execution of system test cases, which are organized and run in a specific sequence to match system requirements. When errors were found during execution of test cases, testing was halted until a new system version was available for testing again. The sequence of tests was restarted on new system version and tests continued to run until next system error appear, causing interruption of tests again, or until all tests were completed, thus ending the testing phase. The testing phase of this project aims to ensure that the system is ready to be available to the user. Testing web systems or websites manually is a task prone to errors. Automate testing with **Selenium** makes this task error free and does not demand tester to play the final user inputting data to the system.

**Testing Plan**

SegFault testing plan will consist of iterations. The first iteration will consist on preliminary results. There is a total of 19 test cases and each test iteration will be described step by step. Also, and outcome of each test iteration will be included. Our testing schedule is as follows:

Iteration 1: 11/22/2015

Iteration 2: 11/25/2015

Iteration 3: 11/27/2015

**Test Cases**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TEST CASE | NAME | STEPS | PASSED OR FAILED TEST | TEST RESULT |
| TC1 | Login | 1. Open the login page 2. Enter username: “[ta@test.com](mailto:ta@test.com)” 3. Enter password: “testtesttest” 4. Press enter 5. Repeat 1-3 then click the submit button | Passed | The application logs in successfully. |
| TC2 | Logout | 1. Open the login page 2. Enter username: “[Admin@test.com](mailto:Admin@test.com)” 3. Enter password: “admin” 4. Press enter 5. log in as an admin 6. Click the logout button 7. Request is sent to server | Passed | The application logs out successfully |
| TC3 | Register | 1. Open the login page 2. Enter username: “[Admin@test.com](mailto:Admin@test.com)” 3. Enter password: “testtesttest” 4. Press enter 5. Logged in as an admin 6. Look for Add password box 7. Enter email: “[pomajuniors@gmail.com](mailto:pomajuniors@gmail.com)” 8. Enter First Name: “Juan” 9. Enter Last Name: “Poma” 10. Enter Password: “password01” 11. Check Driver Option 12. Click the Add User button | Passed | The application adds a new user to the database successfully |
| TC4 | Forget Password | 1. Open the login page 2. Click on Forgot Password 3. Enter email: “[pomajuniors@gmail.com](mailto:pomajuniors@gmail.com)” 4. Check email inbox 5. Open TaxiPing reset password email 6. Click on link to reset password 7. Enter new password: “password01” 8. Re-enter new password: “password01” 9. Click the submit button | Passed | The applications sents an email to user’s email address and resets the password successfully |
| TC5 | Map Clicking Search Address | 1. Open the login page 2. Enter username: “[Dispatcher@test.com](mailto:Dispatcher@test.com)” 3. Enter password: “testtesttest” 4. Press enter 5. Logged in as a dispatcher 6. Click on any location of the map 7. Wait until icon (red person) appear | Passed | The application successfully sends user location to the server and this get updated every 2 secs. |
| TC6 | Search Bar Address | 1. Open the login page 2. Enter username: “[Dispatcher@test.com](mailto:Dispatcher@test.com)” 3. Enter password: “testtesttest” 4. Press enter 5. Logged in as a dispatcher 6. Click on search bar address 7. type 41 Penn St. worcester, ma | Passed | The applications successfully search for 41 Penn St. Worcester, MA and outputs it on the map |
| TC7 | Send Notifications | 1. Open the login page 2. Enter username: “[Dispatcher@test.com](mailto:Dispatcher@test.com)” 3. Enter password: “testtesttest” 4. Press enter 5. Logged in as a dispatcher 6. Click on TaxiPing logo to send notification 7. A new box appears 8. Enter message: “Please go to the following address” 9. Enter address: “41 Penn Ave. Worcester, MA” 10. click send button | Passed | The application send notifications successfully to driver |
| TC8 | Reassign a Rejected Notification | 1. Open the login page 2. Enter username: “[Dispatcher@test.com](mailto:Dispatcher@test.com)” 3. Enter password: “testtesttest” 4. Press enter 5. Logged in as a dispatcher 6. Click on TaxiPing logo to send notification 7. A new box appears 8. Enter message: “Please go to the following address” 9. Enter address: “41 Penn Ave. Worcester, MA” 10. click send button 11. On the upper left corner, a white circle turns red 12. click on red circle button 13. click on reassign address option 14. click on new driver icon 15. click on send notifications button | Passed | the applications successfully reassign the rejected notification from the previous driver |
| TC9 | Map Navigate Move | 1. Access the login page 2. Enter username “[Dispatcher@test.com](mailto:Dispatcher@test.com)” 3. Enter password “testtesttest” 4. Press enter 5. On the map that appears, click and drag the map 6. Repeat step 5 using “[driver@test.com](mailto:driver@test.com)” and password “driver” | Passed | The map should move around as it is dragged |
| TC10 | Navigate Zoom | 1. Access the login page 2. Enter username “[Dispatcher@test.com](mailto:Dispatcher@test.com)” 3. Enter password “testtesttest” 4. Press enter 5. Use the mouse scroll wheel to move in and out | Passed | The map zooms in and out with the movement of the scroll wheel |
| TC11 | Accept Notification | 1. After TC7 2. On the driver page, select “Accept Notification” 3. On the dispatcher page, click the white circle in the top left of the page | Passed | A notification should be in the notification bar that specifies the driver name, the address, and that it was accepted |
| TC12 | Decline Notification | 1. After TC7 2. On the driver page, select “Accept Notification” 3. On the dispatcher page, note the change in color of the white notification circle to red 4. On the dispatcher page, click the red circle in the top left of the page | Passed | A notification should be in the notification bar that specifies the driver name, the address, and that it was declined |
| TC13 | Pick Up Passenger | 1. Open the login page on a phone 2. Enter username “[driver@test.com](mailto:driver@test.com)” 3. Enter password “driver” 4. Press enter 5. Select pick up from the driver screen | Passed | On the dispatcher map, the driver marker turns red to denote that the driver is busy, upon clicking the map near the marker, the marker will remain red, and unselectable |
| TC14 | Drop Off Passenger | 1. After TC13 2. Select drop off from the driver screen | Passed | On the dispatcher map, the marker should turn back to yellow. |
| TC15 | Take a Break | 1. Open login page 2. Enter username “[driver@test.com](mailto:driver@test.com)” 3. Enter password “driver” 4. Press enter - logs in as driver 5. Click “Take a break” | Passed | On the dispatcher map, driver marker disappears. |
| TC16A | Add User | 1. Open login page 2. Enter username “admin@test.com” 3. Enter password “admin” 4. Press enter - logs in as admin 5. Enter email “[newuser@test.com](mailto:newuser@test.com)” 6. Enter first name “new” 7. Enter last name “user” 8. Check Admin 9. Click Add User | Passed | User is added to database |
| TC16B | Add User | Repeat TC16A, but instead select Dispatcher | Passed | User is added to database |
| TC16C | Add User | Repeat TC16A, but instead select Driver | Passed | User is added to database |
| TC17 | Delete User | 1. Open login page 2. Enter username “admin@test.com” 3. Enter password “admin” 4. Press enter - logs in as admin 5. Select user “New User” from drop-down menu 6. Click “Remove User” | Passed | User is removed from database |
| TC18 | Edit User | 1. Open login page 2. Enter username “admin@test.com” 3. Enter password “admin” 4. Press enter - logs in as admin 5. Click user “Admin Admin” 6. Click in Last Name field 7. Change last name to “User” 8. Click “save” button | Failed | Admin user’s last name is changed to “User” |
| TC19 | Check Driver/Dispatcher Status | 1. Open login page 2. Enter username “admin@test.com” 3. Enter password “admin” 4. Press enter - logs in as admin 5. Click user “Driver Driver” | Failed | Individual User Information page is displayed |

**Test Iteration 1**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | NAME | Pass/Fail | Comments |
| TC1 | Login | Fail | Logo covered the login page |
| TC2 | Logout | Pass |  |
| TC3 | Register | Pass |  |
| TC4 | Forget Password | Pass |  |
| TC5 | Map Clicking Search Address | Fail | The marker was added however, if there are drivers on break or busy that come online within distance of the marker, the markers are not clickable |
| TC6 | Search Bar Address | Fail | See TC5 comment |
| TC7 | Send Notifications | Pass |  |
| TC8 | Reassign a Rejected Notification | Pass |  |
| TC9 | Map Navigate Move | Pass |  |
| TC10 | Navigate Zoom | Pass |  |
| TC11 | Accept Notification | Pass |  |
| TC12 | Decline Notification | Pass |  |
| TC13 | Pick Up Passenger | Pass |  |
| TC14 | Drop Off Passenger | Pass |  |
| TC15 | Take a Break | Pass |  |
| TC16 | Add User | Pass |  |
| TC17 | Delete User | Pass |  |
| TC18 | Edit User | Pass |  |
| TC19 | Check Driver/Dispatcher Status | Fail | This functionality has not been added. |

**Documentation for Future Developers**

The site is organized into three general components: PHP, SQL, and Front End. The Front\_End contains the HTML, PHP, JavaScript, and images for the login page, home page, driver and dispatcher profile pages, and admin pages. Check\_logged\_in.php inject javascript into the admin.php, Driver.php, and Dispatcher.php files. The admin.php file allows an admin to have add new Driver and Dispatchers to the database. Login.php logs the account on or redirects to the forgot\_password.php page if the user forgot his password. If the user’s account is registered as a Driver the page redirects to Driver.php, and if the user’s account is registered as a Dispatcher, the page redirects to the Dispatcher.php page. Logout.php logs the user’s account out and redirects to the Login.php page.

The database contains two main databases, the User.php and taxiCompany.php files, which exist within the PHP folder and are connected by relational tables within relations.php. Relations.php contains various functions that connect the users’ information to a table of Dispatcher and Drivers. The user information in User.php is connected to a relational table of passwords to ensure that they are secure.

A PHP back end was built so that when using information on the front end, we don’t need to directly query the database. The back end contains scripts to process the user data and act as an in-between for AJAX requests and the database. This area also properly queries the database for the data and parses and returns the data as JSON files.

**Installation and Configuration**

1. Add (and insert appropriate dir path) the following to your httpd.conf file:

<Directory “path\_to\_root\_of\_SD&D”>

AllowOverride All

Options None

Require all granted

</Directory>

2. Add (and insert appropriate dir path) the following to your httpd-vhosts.conf file:

<VirtualHost \*>

DocumentRoot “path\_to\_root\_of\_SD&D”

ServerName TaxiPing

</VirtualHost>

3. Add the following to your windows hosts file:

TaxiPing 127.0.0.1

4. Start Apache & MySQL

5. Open taxi-app/install.php in a browser and enter your MySQL root/admin username and

password.

6. Type the domain name TaxiPing to visit the website

7. Login with the following username/password combinations:

- Driver@rpi.edu / mahonk3

- Dispatcher@test.com / testtesttest

- admin@test.com / admin

## Contribution Summary

Alex Kelleher:

* Test cases
* Status Report

Juan Poma:

* Test cases
* Test plan

Kevin Stoner:

* Test cases
* Code review

Juntao Zhuang:

* Test cases
* Documentation for future developers

## 

## **Status Report**

Since the beginning of the project, the following items have been figured out:

* Project started on 9/13/2015
* Meeting schedule was agreed upon - team meets twice a week (Wednesdays & Sundays)
* Team metrics - consists of four parts: timeliness of task completion, involvement in team education, breadth of experience, team communication
* Created team folder and discussion group on google drive and Facebook
* GitHub repository was set up
* Server with static IP was set up
* Established coding standards for Java, JavaScript, HTML5, PHP
* Initial research on Google API, Android API, HTML interface as well as some backend work were done
* Inception Deliverables were finished
* Diagrams about user interface were drawn, covering all the features of our application
* Login, logout pages were implemented (front end & back end were finished)
* Front end of forgot password page were implemented
* Project elaboration was finished
* Completed GPS proof-of-concept
* MySQL database was researched
* Register page was implemented
* Successfully moved our PHP code from development environment to a Linux server
* Embedded Google API into our front end and was able to mark locations on map
* Search by clicking and search by inputting address has been implemented
* Drivers were able to update their latest location and dispatcher was able to access these data from backend
* Was able to access GPS module of android devices from Javascript call
* Driver can send notifications to Dispatcher and vice versa
* Dispatcher can assign pickup to Driver
* Driver can accept or reject pickup