On Demand High Capacity Taxi Visualization Framework – TVF

Master Test Plan

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 04/05/2019 | 1.0 | Master testing plan with UI and unit testing | V.G.M.I. Karunathilaka  160289X |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Evaluation Mission and Test Motivation 3

2. Target Test Items 3

3. Test Approach 4

3.1 Testing Techniques and Types 4

3.1.1 Data and Database Integrity Testing 4

3.1.2 Function Testing 5

3.1.3 User Interface Testing 5

3.1.4 Performance Profiling 6

3.1.5 Load Testing 7

3.1.6 Security and Access Control Testing 7

3.1.7 Failover and Recovery Testing 8

3.1.8 Configuration Testing 9

4. Deliverables 9

4.1 Test Evaluation Summaries 9

4.2 Reporting on Test Coverage 10

5. Risks, Dependencies, Assumptions, and Constraints 10

6. References

Master Test Plan

# Evaluation Mission and Test Motivation

# This document is focused on the master testing of the system with developed components until this reporting date. Goal of this test plan in to ensure the developed system has required specifications & design criteria. Finally, this testing plan will enable to release a stable and bug free version of the TVF system.

# This system works on the MVC architecture. Therefore, TVF has Models and Controllers. They will provide the main functionalities of the system. In this report used several testing methodologies to test the functionalities of each Model, Controller and View.

The main testing strategies of TVF system includes,

1. Data & database integrity testing
2. Function testing
3. User interface testing
4. Performance profiling
5. Load testing
6. Security and access control testing
7. Failover and recovery testing
8. Configuration testing

At this point most of the main use cases of the system are completed and few of them will complete in near future. In this stage project transferring from construction phase to transition phase.

The main objective of the testing process are as follows,

* Find and fix as many bugs as possible
* Verify the system specification
* Find out special problems and risks
* Confirm and evaluate the use case realization

# Target Test Items

There several main targets that needs to be tested in TVF system. TVF requires high performances with user friendly web interfaces to interact with the system.

When consider about the TVF system, following test targets identified to process the test cases and make sure those are working with good performance level.

* User interfaces

User interfaces of the TVF system will tested for check the responsiveness of the components.Qunit testing tool will use to test the user interfaces of the system. Otherwise Selenium web driver and Katalon automation recorder can be used for the testing.[1],[2]

* Data model (MySQL Database )

This is system implementation which is related to all the system data storing and management. This is one of the core implementations of the system and this must be work with highly reliability with the system.

* Objects and Functions:

This is contained all the functions written in JavaScript which are handle the behavior of the system.

# Test Approach

## Testing Techniques and Types

### Data and Database Integrity Testing

### TVF will use one database table to gain the performance of the system. In other words, instead of normalization TVF uses denormalization to increase the system performances.

|  |  |
| --- | --- |
| Technique Objective: | Check database access methods without using directly with the integration of the interface. This may help to log incorrect functional behaviors. |
| Technique: | When the data is querying and inserting into the database, data will log in the console to check the correctness of the data. |
| Oracles: | Verify that, all the data saved in to database completely saved, and transactions are complete with ACID properties  Logging the data that is loaded and checking them with the data in the database is a way of observing the accuracy. |
| Required Tools: | * Php Myadmin to view tables and their contents * Tool like VScode or browser for see the console logged data |
| Success Criteria: | The technique supports the testing of all key database access methods and processes |
| Special Considerations: | * Database needs to tested manually also, to check all the system data entered in database. |

### Function Testing

|  |  |
| --- | --- |
| Technique Objective: | Validation of the correct functionalities of the functional requirements and all the functional and non-functional requirements are verify in this section. |
| Technique: | TVF uses JavaScript functions in the system. Therefore, Qunit testing is used to implement the functional testing. TVF can verify that the all the functions are functioning correctly without bugs. |
| Oracles: | Verify there is no dead pages or invalid redirects in the web application.  Verify the data integrity  Verify the workflow of the system.  Verify that user inputted correct file type |
| Required Tools: | * QUnit testing framework |
| Success Criteria: | All the essential use cases are tested.  All the test cases can automate and reliable with used testing approaches. |
| Special Considerations: | When we testing system using automated tool in localhost response time of the system get lower that in hosting space. Therefore, we have to test the system with higher performance level in localhost. |

### User Interface Testing

|  |  |
| --- | --- |
| Technique Objective: | Following to observe and log standard conformance and target behavior   * Window objects and characteristics can be exercised. * Responsiveness of the UI elements * Check the navigation of longer sequences |
| Technique: | * Google chrome developer console also used to find out bugs and test some UI components. * QUnit testing framework for write javascript automated testing codes |
| Oracles: | * QUnit testing framework can be used easily for the automated UI testing. |
| Required Tools: | * QUnit testing framework * Googel Chrome * VScode |
| Success Criteria: | QUnit testing is one of the industry accepted framework for test UI using JavaScript. Therefore, this technology can used to make the TVF system industry acceptable one |
| Special Considerations: | This testing technique must test the all possible scenarios to the system while user is using this |

### Performance Profiling

|  |  |
| --- | --- |
| Technique Objective: | Observe the system performance under the different operating systems. Response time, transaction rate, functional throughput will test to ensure the system runs with good performance level. |
| Technique: | * Use the test procedures developed to test the functions in the system. * Modify the data files to increase the no of transactions or the scripts to increase the no of iterations that occur in each transaction. * Repeat all the tasks performed by single user using multiple user instances simulated through a script and measure the performance of the system with each of the instances. |
| Oracles: | It tests results of the performance profiling is within the acceptable region, then the performance of the system is accepted to deploy the system. |
| Required Tools: | Katalon Automation recorder |
| Success Criteria: | * Single transaction or single user: successful emulation of the transaction scripts without any failures due to test implementation problems. * Multiple transaction or multiple users : Successfully tested with the workload without any failures due to test implementation problems |
| Special Considerations: | * The database used for performance testing should be either actual size or scaled quality. * All the test cases need to run with server multi-threaded (multi users) workload.. |

### Load Testing

|  |  |
| --- | --- |
| Technique Objective: | To execute designed transactions or business cases under varying workload situations to observe and log target behavior |
| Technique: | Use Katalon automation recorder tool to automate some of the user testing while some database transaction happening with other user instance |
| Oracles: | The system will be subjected to run with varying workloads. Under any workload instance system response time and other non-functional characteristics are compared with the currently available system. |
| Required Tools: | * Katalon automation recorder |
| Success Criteria: | * If the system can with stand with expected load, then we can ensure system is in the deployable workload managing level. |
| Special Considerations: | * Load testing should be performed on a dedicated device or at a dedicated time period. This permits the all measurements get more accurate. * The database used for load testing should be either actual size or scaled equally. |

### Security and Access Control Testing

|  |  |
| --- | --- |
| Technique Objective: | * To ensure that the system provides necessary application and system level security |
| Technique: | * Application level security Verify that users cannot access beyond their scope. |
| Oracles: | * The system should give fail by throwing errors with warnings if the needed criteria for the security did not meet. |
| Required Tools: | * Node security platform [3] * Katalon automation recorder [1] * Google chrome |
| Success Criteria: | All the unauthorized access should be prevented by the system to achieve the required security and access control. |
| Special Considerations: | * We need to consider all possible accessing paths to the system and needs to ensure they worked properly with high reliable level. |

### Failover and Recovery Testing

|  |  |
| --- | --- |
| Technique Objective: | * The following types of situation can happen while system running and those can be the events that trigger the system failures.   + Communication interruption in network   + Incomplete data transfers   + Incomplete map loading |
| Technique: | Communication interruption in network: simulate the data transfer through the network and measure the data losses. |
| Oracles: | * All above mentioned system failure methods are done while system processing some functionalities * In this section also needs to test about the database consistence as well. |
| Required Tools: | * Php Myadmin |
| Success Criteria: | * One or more tested failure situations involving one or more combinations of the application, database and system. * One or more simulated recoveries involving one or more combinations of the application, database and system. |
| Special Considerations: | * Resources from the system, database and networking are required. |

### 

### Configuration Testing

|  |  |
| --- | --- |
| Technique Objective: | * To execute the system in different web browsers |
| Technique: | * Use function test scripts to ensure the database connection and verify the availability of permissions required by the system. * Execute the selected transactions to simulate the actor’s interaction of the system. |
| Oracles: | Unit tests that used in functional testing are used in configuration testing as well. |
| Required Tools: | * Different web browsers * Google chrome * Edge * Mozilla Firefox * Mobile browsers |
| Success Criteria: | The system should work fine on any device with any operating system. |
| Special Considerations: | * The end users are not needed to configure anything while they are using the system. They only need to have an internet connected device to use this system. * Deployment of the system only happens once, and all the deployment configurations are done once. |

# 

# Deliverables

In this section, the various artifacts that created with the testing are listed.

## Test Evaluation Summaries

## Test evaluation contain all the analysis data related to the tests of the system. In addition to that, test evaluation summaries contain general statements of quality and suggested recommendations to the future testing attempts

The main output of the test evaluation summary report,

1. Test results
2. Change in build
3. Assess quality
4. Improve the effort

All the analyzed data related to,

* Unit testing
* integrated testing
* Security and Access Control Testing
* Load testing
* Performance profiling
* User Interface Testing
* Functional testing
* Data and Database Integrity Testing

Are status inside this report.

## Reporting on Test Coverage

## This test plan mainly discussed under evaluation mission and test motivation, the targeted test items, test approaches and the deliverables of the testing phase. Several techniques types such as data and database integrity testing, function testing, user interface testing, performance testing, load testing, security and access control testing, failover and recovery testing and configuration testing will also be discussed under different testing processes with other details such as Technique objective, testing techniques, oracles, required tools and success criteria’s and special considerations.

# Risks, Dependencies, Assumptions, and Constraints

| Risk | Mitigation Strategy | Contingency (Risk is realized) |
| --- | --- | --- |
| Load testing difficulty | Tester will define the prerequisites that must be met before Load Testing can start.    Customer will endeavour to meet prerequisites indicated by Tester. | * Meet outstanding prerequisites * Consider Load Test Failure |
| Test data proves to be inadequate. | User will ensure a full set of suitable and protected test data is available.  Tester will indicate what is required and will verify the suitability of test data. | * Redefine test data * Review Test Plan and modify * components (that is, scripts) * Consider Load Test Failure |
| Database requires refresh. | Tester will endeavour to ensure the Database is regularly refreshed as required by tester. | * Restore data and restart * Clear Database |

# References

[1] Katalon Studio. (2019). *Katalon Automation Recorder - Powerful Selenium IDE to record, debug, play tests in any browser*. [online] Available at: https://www.katalon.com/resources-center/blog/katalon-automation-recorder/ [Accessed 4 Apr. 2019].

**[2]** js.foundation, J. (2019). *QUnit*. [online] Qunitjs.com. Available at: https://qunitjs.com/ [Accessed 4 Apr. 2019].

[3] The npm Blog. (2019). The Node Security Platform service is shutting down 9/30. [online] Available at: https://blog.npmjs.org/post/175511531085/the-node-security-platform-service-is-shutting [Accessed 4 Apr. 2019].