****

**FPT UNIVERSITY**

**Taxi Caller Application on Windows Phone**

**F\_Taxi**

**Report #3 – Software Requirements Specification**

|  |  |
| --- | --- |
| F\_Taxi | |
| Group Member | SE02705 - Phạm Gia Hữu |
| SE02900 - Tạ Thiên Hưởng |
| SE02268 - Phạm Ngọc Hoàn |
| SE02314 - Nguyễn Văn Lập |
| Supervisor | Nguyễn Văn Sang |
| Project Code | F\_Taxi |

HaNoi, 11/2015

# Record of change

\*A - Added　M – Modified　D – Deleted

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Effective Date** | **Changed Item** | **A\* M, D** | **Change Description** | **Revision Number** |
| 2015/11/27 | Create new | A | Create document | 0.1 |
| 2015/11/28 | Update section 1.2 and 1.3 | M | Update contents of section 1.2 and 1.3 | 0.2 |
| 2015/11/29 | Update section 1.4, 1.5, 2.1, 2.2 | M | Update contents of section 1.4, 1.5, 2.1, 2.2, | 0.3 |
| 2015/11/30 | Update section 2.1, 2.2 | M | Update contend of section 2.1, 2.2 | 0.4 |
| 2014/12/01 | Modify section structure | M | Modify section structure | 0.5 |
| 2015/12/02 | Update section 3.1, 3.2 | M | Update contend of section 3.1, 3.2 | 0.6 |
| 2015/12/03 | Update section 3.3, 3.4, 3.5 | M | Update contend of section 3.3, 3.4, 3.5 | 0.7 |
| 2015/12/04 | Update section 3.3 and 3.4 | A | Update content of section 3.3 and 3.4 | 0.8 |
| 2015/12/05 | Update section 3.3 and 3.4 | M | Update content of section 3.3 and 3.4 | 0.9 |
| 2015/12/06 | Update section 4.3, 5.1 and 5.2 | M | Update content of section 4.3, 5.1 and 5.2 | 1.0 |
| 2015/12/07 | Update section 4.1 an d 4.2 | M | Update content of section 4.1 and 4.2 | 1.1 |
| 2015/12/08 | Update comment of leader review | M | Check and modify all sections | 1.2 |

# Signature page

**Author:** Phạm Gia Hữu Date: 2015/11/27

*Member*

**Reviewers:** Nguyễn Văn Lập Date: 2015/12/08

*Project manager*

**Approval:** Nguyễn Văn Sang Date:

*Supervisor*

# Table of contents

Record of change 1

Signature page 2

Table of contents 3

Table of figures 5

1 Introduction 6

1.1 Purpose 6

1.2 Audience 6

1.3 Overview 6

1.4 Definition and Acronyms 6

1.5 References 7

2 Test Overview 8

2.1 Test Approach 8

2.1.1 Unit testing 9

2.1.2 Integration testing 9

2.1.3 System testing 9

2.1.4 Acceptance testing 9

2.2 Test Strategy 9

2.2.1 Functional Test 9

2.2.2 User Interface Test 10

3 Test Plan 11

3.1 Feature to be tested 11

3.2 Feature not to be tested 12

3.3 Create test cases plan 13

3.4 Execute test plan 13

3.5 Testing Environments 14

3.5.1 Software 14

3.5.2 Hardware 14

3.6 Resource 14

3.7 Test Milestones 15

4 Test Cases 16

4.1 Unit Test Cases 16

4.2 GUI 21

4.3 Function Test 21

4.3.1 Homepage Screen 21

4.3.2 Transaction Screen & Post information Screen 25

4.4 Check list 38

5 Test Report 40

5.1 Defect Log 40

5.1.1 Homepage Screen 40

5.2 Test Report 45

# Table of figures

[Figure 2‑1 V-model 8](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532671)

[Table 3‑1 Feature to be tested 11](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532676)

[Table 3‑2 Create test cases plan 13](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532677)

[Table 3‑3 Execute test plan 13](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532678)

[Table 3‑4 Software needed for testing 14](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532679)

[Table 3‑5 Resource and Responsibility in Test process 15](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532680)

[Table 3‑6 Test milestones 15](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532681)

[Table 4‑1 Test Cases 16](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532682)

[Table 4‑2 Log in 21](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532683)

[Table 4‑3 Registration 25](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532684)

[Table 4‑4 Check list 38](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532685)

[Table 5‑1 Template 40](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532686)

[Table 5‑2 Test report 45](file:///C:\Users\huuph_000\Downloads\F_Taxi_Report_5_STD.docx#_Toc385532687)

# Introduction

## Purpose

The purpose of this document is to outline the test strategy and overall test approach for the Taxi Caller project. This includes test methodologies, traceability, resources required and estimated schedule.

## Audience

The audience of this document is the project team and the project manager. This document is also written for the extended test team. The test lead, testers, and any outsourced testers should be able to utilize this document to understand the scope of work that must be accomplished by the test team. The document is intended to accomplish its purpose only for the intended audiences.

## Overview

This document describes in detail test approach, test plan, test case, defect log, Test Milestones, Checklist, etc. Which will user to cover all situation arising bugs and validate the quality of our product prior to release? It does not describe implementation details of test cases or technical details of how the product features should work. The goal is to provide a framework that can be used by managers and testers to plan and execute the necessary tests in a timely and cost-effective manner.

## Definition and Acronyms

|  |  |  |
| --- | --- | --- |
| Acronym | Definition | Note |
| STD | Software Test Documentation |  |
| UT | Unit Test |  |
| IT | Integration Test |  |
| ST | System Test |  |
| AT | Acceptance Test |  |
| QA | Quality Assurance |  |

## References

**Documents:**

1. IK group, IK\_Final\_Report, (2013/12/09), FPT University, Hanoi, Vietnam.

# Test Overview

## Test Approach

We use V-model to implement testing process.

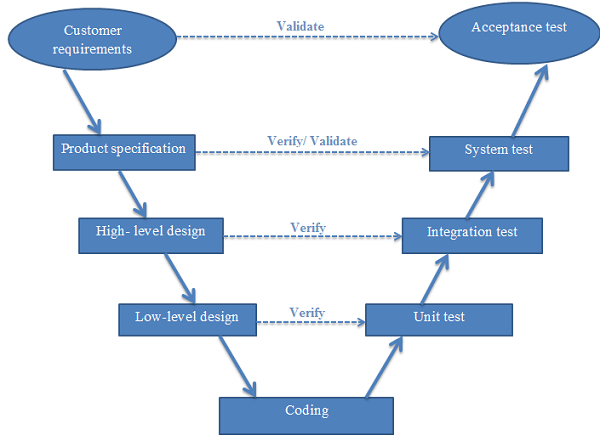


Figure 2‑1 V-model

Due to requirement of project must be always suitable with user, we choose V-model to implement testing process. With V-Model, software development is separated into two appropriate phase’s groups: development and testing. In this model, the verification and validation will be done side by side. It emphasizes the strict process flow to develop a quality product. The errors occurred in any phase will be corrected in that phase. Proactive defect tracking defects which are found at early stages even may be in the development phase before application being tested (Unit test).

Testing progress is divided to 4 phases include: Unit test, Integration test, System test and Acceptance test.

### Unit testing

* The purpose is to verify the internal logic code by testing every possible branch within the function, also known as test coverage.
* Unit test will be done by the developers and will be approved by the implementing team leader to ensure that the building blocks of the system uses work independently of each other and the specific function is working as expected. One function might have multiple tests to catch corner cases in the code

### Integration testing

* The separate modules will be tested together to expose faults in the interfaces and in the interaction between integrated components.
* Integration testing will be done by tester.

### System testing

* Compare the system specifications against the actual system.
* System testing checks if the integrated product meets the specified requirements.

### Acceptance testing

* Acceptance testing will be performed by the test manager and development team leader.
* The acceptance test will be done for a period of 1 week after completion of the System/Integration test process.
* Programs will enter into Acceptance test after all critical and major defects have been corrected.
* Prior to final completion of acceptance testing all open critical and major defects must be corrected.

## Test Strategy

### Functional Test

Functional testing bases its test cases on the specifications of the component under test. Functions are tested by feeding them input and examining the output.  
Functional testing typically involves four steps:

* The creation of input data based on the function's specifications
* The determination of output based on the function's specification
* The execution of the test case
* The comparison of actual and expected outputs

### User Interface Test

During testing, focus is shifted towards application specific aspects of user interaction as they are manifested by interface syntax and semantics. The goal of interface testing is to:

* Uncover errors related to specific interface mechanism.
* Uncover errors in the way interface implements semantics of navigation, web application functionality or content display.

Objectives that should be achieved are:

* Interface features are tested to ensure that design rules, aesthetics, and related visual content are available for user without error.
* Individual interface mechanisms are tested in a manner that is analogous to unit testing.
* Each interface mechanism is tested within the context of a use case or navigation semantic units for a specific user category.
* The complete interface is tested against selected use cases and navigation semantic units to uncover errors in the semantics of the interface.

# Test Plan

## Feature to be tested

Developers will be performed unit test to ensure that all functions are working as expected.

The test team will be undertaking the responsibility of testing and validating the operation of the part that developed. The functions developed require completing design and execution of function tests to ensure proper coverage.

Following are items that will need to be tested by testers:

**Table 3‑1** Feature to be tested

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Component | | Feature Name | | Feature to be tested |
|  | Interface | Rider screen | Screen elements | | Check the display of elements. |
| Driver screen |
|  | Function | Main application |  | Login | Login |
|  |  | Logout | Logout |
|  | Account management |  | Register account |
|  | View profile |
|  | Update profile |
|  | Change password |
|  | Forget password |
|  | Rider actions | Work with trips | Create trip |
|  | Cancel trip |
|  | View completed trips |
|  | View bill detail |
|  | Rate driver |
|  | Favorites management | View favorite driver |
|  | Add favorite driver |
|  | Delete favorite driver |
|  | Call favorite driver |
|  | Find lost assets | Find lost assets |
|  | Send Application Feedback | Send Application Feedback |
|  | Call Taxi Center | Call Taxi Center |
|  | Driver actions | Work with trips | Accept trip |
|  | Reject trip |
|  | Cancel trip |
|  | Complete trip |
|  | Start trip |
|  | View bill detail |
|  | V.I.P customer management | View V.I.P customer |
|  | Add V.I.P customer |
|  | Delete V.I.P customer |
|  | Notify lost assets | Notify lost assets |
|  | Change status | Change status |
|  | Conflict |  | Conflict | | |

## Feature not to be tested

* Maintainability.
* Security testing

## Create test cases plan

**Table 3‑2** Create test cases plan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Date | December | | | | | | Total |
| 1 | 2 | 3 | 4 | 5 | 6 |
| Create TC |  |  |  |  |  |  |  |
| HuuPG | GUI | Account | Rider actions | Log in-Log out | Driver action | Conflict |  |

## Execute test plan

**Table 3‑3** Execute test plan

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Date | Round 1 | | | | | | Total |
| December | | | | | |
| 11 | | | 12 | | |
| Execute Test (TC) |  |  |  |  |  |  |  |
| LapNV |  | x |  | x |  | x |  |
| HuuPG | x |  | x |  | x |  |  |
| Bug |  |  |  |  |  |  |  |
| Method | GUI | Log in-Log out | Account | Driver action | Rider actions | Conflict |  |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Date | Round 2 | | | | | | | | Total |
| December | | | | | | | |
| 13 | | | | 14 | | | |
| Execute Test (TC) |  |  |  |  |  |  |  |  | 648 |
| LapNV |  |  |  |  |  |  |  |  | x |
| HuuPG |  |  |  |  |  |  |  |  |  |
| Bug |  |  |  |  |  |  |  |  |  |
| Method | All | | | | | | | |  |

## Testing Environments

### Software

**Table 3‑4** Software needed for testing

|  |  |  |
| --- | --- | --- |
| Category | Software name | Version |
| Operating System | Windows Phone |  |
| Development framework | Microsoft .NET | 4.0 |
| Development tools | Microsoft Visual Studio | 2013 |
| Notepad++ | Newest |
| Data managing tools | TortoiseGIT | Newest |
| Support tools | Chrome, Internet Explorer, Photoshop | Newest |
| Assign task and tracking bug | Microsoft Excel | Newest |
| Testing tools | - | - |

### Hardware

At server: T-NET

Smart phone 1: LUMINA 630

* OS: Windows phone 8.1
* Install memory (RAM) : 1.00GB

Smart phone 2: LUMINA 720

* OS: Windows phone 8.1
* Install memory (RAM) : 1.00GB

## Resource

Developer, tester and project manager has to join in testing process.

**Table 3‑5** Resource and Responsibility in Test process

|  |  |
| --- | --- |
| Resource | Responsibility |
| Developers | Unit test, fix all bugs after testing |
| Tester | Create test cases, test plan, test report, bug list report |
| Prepare data to test |
| Log bugs |
| Test and retest to limit maximize bugs |
| Project manager | Responsible for project schedules |

## Test Milestones

**Table 3‑6** Test milestones

|  |  |  |  |
| --- | --- | --- | --- |
| Milestone Task | Resource | Start Date | Finish date |
| Create Test plan | Test leader | 2015/11/27 | 2015/11/27 |
| Create Unit Test cases | Developer | 2015/11/28 | 2015/11/29 |
| Create Integration Test cases | Tester | 2015/11/28 | 2015/11/29 |
| Create System Test cases | Tester | 2015/11/30 | 2015/12/01 |
| Review Unit Test cases | Project Team | 2015/11/30 | 2015/11/30 |
| Review Integration Test cases | Project Team | 2015/12/01 | 2015/12/01 |
| Review System Test cases | Project Team | 2015/12/02 | 2015/12/02 |
| Update Test cases | Tester | 2015/12/03 | 2015/12/03 |
| Create data test | Tester | 2015/12/03 | 2015/12/03 |
| Execute Unit Test | Developer |  |  |
| Execute Integration Test | Tester |  |  |
| Execute System Test | Tester |  |  |

# Test Cases

## GUI

Refer to [Test Case](file:///E:\Documents\Study\Software%20engineering\16.Capstone%20Project%20JS\CP-Documents\Reports\Report%205%20Test\Testcases\Test%20Case_RTF_vr0.5.xls) to know all GUI test cases.

## Function Test

Refer to [Test Case](file:///D:\do%20an\Ftaxi\rtf-all-resource\RTF_Reports\Report_5_Test\Testcases\Test%20Case_RTF_vr0.5.xls) to see all test case execution.

### Log in

Following is **Log in** test cases.

**Table 4‑1** Log in test cases



### Register

**Table 4‑2** Register test cases



### Call taxi



### Manage favorite taxi



## Non-functional

### Test conflict



## Check list

**Table 4‑4** Check list

|  |  |  |  |
| --- | --- | --- | --- |
| No. | | Check Point | Yes/No |
| 1 | FUNCTIONALITY | |  |
| 1.1 | LINKS | |  |
| 1.1.1 | Check that the link takes you to the page it said it would. | |  |
| 1.1.2 | Ensure to have no orphan pages (a page that has no links to it) | |  |
| 1.2 | FORMS | |  |
| 1.2.1 | Acceptance of invalid input | |  |
| 1.2.2 | Optional versus mandatory fields | |  |
| 1.2.3 | Input longer than field allows | |  |
| 2 | APPLICATION SPECIFIC FUNCTIONAL REQUIREMENTS | |  |
| 2.1 | DATA INTEGRATION | |  |
| 2.1.1 | Check the maximum field lengths to ensure that there are no truncated characters. | |  |
| 2.1.2 | If a particular, set of data is saved to the database check that each value is saved fully to the database. (i.e.) Beware of truncation (of strings) and rounding of numeric values. | |  |
| 2.2 | ALPHANUMERIC FIELD CHECKS | |  |
| 2.2.1 | Use blank and non-blank data. | |  |
| 2.2.2 | Include lowest and highest values. | |  |
| 2.2.3 | Include invalid characters & symbols. | |  |
| 2.2.4 | Include valid characters. | |  |
| 2.2.5 | Include data items with first position blank. | | Yes |
| 2.2.6 | Include data items with last position blank. | | Yes |
| 3 | COMPATIBILITY | |  |
| 3.1 | BROWSERS | |  |
| 3.1.1 | Is the HTML version being used compatible with appropriate browser versions? | | Yes |
| 3.1.2 | Do images display correctly with browsers under test? | | Yes |
| 3.1.3 | Verify the fonts are usable on any of the browsers | | Yes |
| 3.1.4 | Have you tested Animated GIFs across browsers? | | Yes |
| 3.2 | CONNECTION SPEED | |  |
| 3.2.1 | Does the site load quickly enough in the viewer's browser within 8 Seconds? | | Yes |

# Test Report

## Defect Log

Refer to [Defect Log](file:///D:\do%20an\Ftaxi\rtf-all-resource\RTF_Reports\Report_5_Test\DefectLogs\RTF_Defect_Log.xlsx) to see all Bug List.

**Table 5‑1** Template bug list

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Defect ID | Module | Description | Type | Severity | Priority | Status | Created Date | Close Date |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

## Test Report

**Table 5‑2** Test report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module test | Pass | Fail | Untested | N/A | Number of Test Cases |
| Total of Test Case |  |  |  |  |  |
| GUI |  |  |  |  |  |
| Log in |  |  |  |  |  |
| Log out |  |  |  |  |  |
| Registration |  |  |  |  |  |
| Rider action |  |  |  |  |  |
| Driver action |  |  |  |  |  |
| Conflict |  |  |  |  |  |