**Chapter 5: Software Testing Documentation**

**5.1 Introduction**

**5.1.1 Purpose**

This chapter’s primary goal is to create testing plans and execute the defects detection and prevention procedures, which may cause software malfunctioning. Another objective of this chapter is to provide details about the software quality and to ensure that the end result meets the business and user requirements.

**5.1.2 Scope of testing**

* **Stages of testing:**

There are 4 phases in the Testing Process: Unit testing, Integration testing, System testing and Acceptance testing.

|  |  |  |
| --- | --- | --- |
| **ID** | **Test Stages** | **Description** |
| 1 | Unit test | Unit Tests are performed by developers to ensure that individual units of source code function as intended. Unit tests are primarily used to test complicated algorithms and automatically test important functions. |
| 2 | Integration testing | Integration Tests are performed by testers to test whether combined unit function as intended. We also know defects between the modules/functions. After that, developer will fix the system suitably. |
| 3 | System testing | System Tests are performed by testers. In this test, tester will be a customer, to test the system as a whole and determine whether or not the system meets the requirements. |
| 4 | Acceptance testing | Testers create check list include some rules between team members, customers and the supervisor. If customers don’t have any requirements, we will build acceptance test base on the purpose of project |

* **Type of testing**

The test team has to test the following type on Google Chrome

* Integration testing
* System testing
* User interface testing
* **Range of testing**

Team performs all functions defined in the SRS based on the approved version.

**5.2 Test plan**

**5.2.1 Testing tools and environment**

**5.2.1.1 Testing tools**

* **­Chrome Developer Tools: To view logs, inspect elements.**



*Figure 5-1: Chrome Dev Tools*

* **Trello and Backlog: bug control service to log, manage and resolve bugs.**
* **Microsoft Excel: To manage test cases**



*Figure 5-2: Microsoft Excel*

* **Postman: A tool to test API endpoints and returned results.**



*Figure 5-3: Postman*

**5.2.1.2 Testing environment**

|  |  |  |
| --- | --- | --- |
| **Type of testing** | **Software** | **Hardware** |
| System test | Google chrome | Personal computer for developing with the minimum configuration:  - Windows 10 Pro 64-bit.  - Intel® Core™ i5 5200 CPU.  - Installed memory (RAM): 8.00GB |
| Integration Test | Google chrome | Personal computer for developing with the minimum configuration:  - Windows 10 Pro 64-bit.  - Intel® Core™ i5 5200 CPU.  - Installed memory (RAM): 8.00GB |
| Unit test | Visual Studio | Personal computer for developing with the minimum configuration:  - Windows 10 Pro 64-bit.  - Intel® Core™ i5 5200 CPU.  - Installed memory (RAM): 8.00GB  - Visual Studio 2017 or 2019 |
| API testing | Postman | Personal computer for developing with the minimum configuration:  - Windows 10 Pro 64-bit.  - Intel® Core™ i5 5200 CPU.  - Installed memory (RAM): 8.00GB |

**5.2.2 Resources and responsibilities**

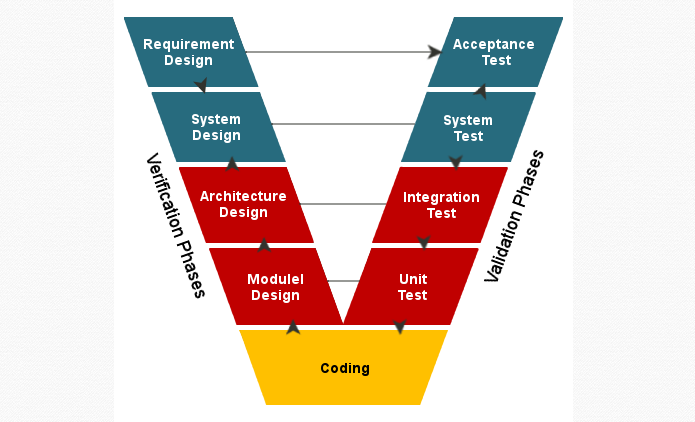
This table shows the staffing assumptions for the project.

|  |  |  |
| --- | --- | --- |
| **ID** | **Resources** | **Responsibilities** |
| 1 | Project Manager | * Responsible for Project Schedules and overall success of the project. * Review Test-case and report. |
| 2 | Tester | * Preforming the actual system testing. * Manage test resource and assign test tasks. * Create Test Plan. * Create Test Cases. * Create Test Report. * Execute Test. * Test Log report. |
| 3 | Developer | * Create unit test and integration test scripts. * Fix bugs. |

**5.2.3 Test strategy**

**5.2.3.1 Test model**

* Trip-Sharing deploys a contemporary of traditional software development models is "V-Model":
* At V model, corresponding to a test phase is a software development phase, testing in the V-model is done in parallel with the software development cycle.
* Each level of the development lifecycle is verified before moving on to the next level.
* This helps in identifying errors very early in the lifecycle and minimizes potential future defects appearing in the code later in the lifecycle.



*Figure 5-4: Architecture of V Model*

*Trip-Sharing API has 2 levels of test:*

* Unit testing: Developer will write unit test to cover logic of Models and Functions
* API testing: Use Postman tool tests that involve testing APIs directly to determine whether APIs return the correct response for a broad range of feasible requests, react properly to cases such as failures and unexpected/extreme inputs.
  + - 1. **Test types**
* **Integration testing**
* Integration Test combines individual components and units together and checks their communication.
* Main goals of Integration Test:
* Detecting communication errors between units.
* Integrating individual units into subsystems (Subsystems) and finally a complete system (System) to prepare for system-level testing (System Test).
* Integration Test should only be performed on previously tested units with the Unit Test
* **System testing**
* System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.
* The purpose of System Test is to test whether the design and the whole system (after integration) meet the requirement.
* System Test tests both the software's functional behavior and quality requirements such as reliability, usability, performance and security.
* **User Interface testing**
* User Interface testing verifies a user’s interaction with the software. The goal of GUI testing is to ensure that the GUI provides the user with an appropriate access and navigation through the functions of the target-of-test
* GUI test will be performed fully on all screens.
* This test targets to cover the verification of the overall look and feel of the Trip-Sharing system including initial position, font, text size, color, focus, initial button, tab order, label, screen sizes and sentences width.
* Images have high resolution and properly adjusted for different screen sizes.

**5.2.3.3 Test schedule**

Table below is the Test Schedule for Trip-Sharing Project

|  |  |  |
| --- | --- | --- |
| **Test Schedule** | **Start Date** | **End Date** |
| *Phase 1:* | 06/06/2019 | 20/07/2019 |
| Unit Testing and API Testing | 10/06/2019 | 30/06/2019 |
| Integration Testing | 15/06/2019 | 10/07/2019 |
| System Testing | 17/06/2019 | 20/07/2019 |
| *Phase 2:* | 20/07/2019 | 25/08/2019 |
| Unit Testing and API Testing | 23/07/2019 | 10/08/2019 |
| Integration Testing | 26/07/2019 | 12/08/2019 |
| System Testing | 28/07/2019 | 15/08/2019 |
| Acceptance Testing | 10/08/2019 | 20/08/2019 |

*Table 5-1: Test schedule*

**5.2.4 Features to be tested**

All features are listed in the use case list.

**5.2.5 Features not to be tested**

All features are listed in 1.5.1.3 above will not to be tested.

**5.3 Test Case**

**5.3.1 Unit testing and API testing**

Unit testing and API testing will be done by the developers and approved by the development team leader.

The Trip-Sharing development team embrace these features to gain the following advantages:

* Reduce the number of bugs in production code.
* Ensure the functions of the software are in accordance with customer requirements
* Identify errors very early in the development process and minimize potential future errors that may appear

**5.3.1.1 Unit testing framework**

* We use NUnit is a unit testing framework for .NET. It is the most used framework for writing unit test cases and JetBrains dotCover is a .NET unit test runner and code coverage tool.
* Mock function are simulated functions that mimic the behavior of real functions in controlled ways, most often as part of a software testing initiative.
* Unit testing scripts are created manually and saved to ProjectName.Tests directory of Trip-Sharing API services



Figure 5-4: Test directory structure

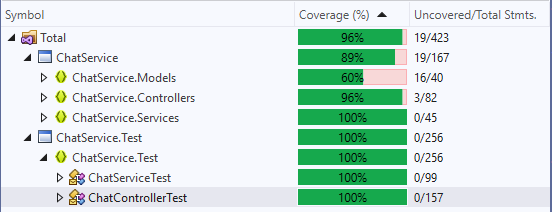
* Unit tests focus on individual functions in a class and are created as in the picture.



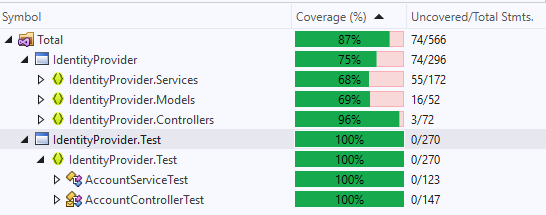
*Figure 5-5: Unit test case sample*

This factory function used to get recommend articles

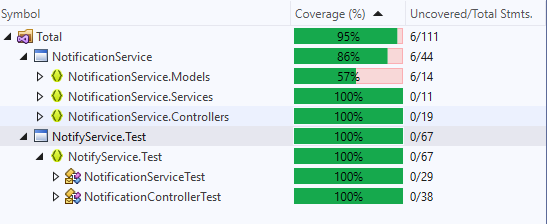
* Coverage report:



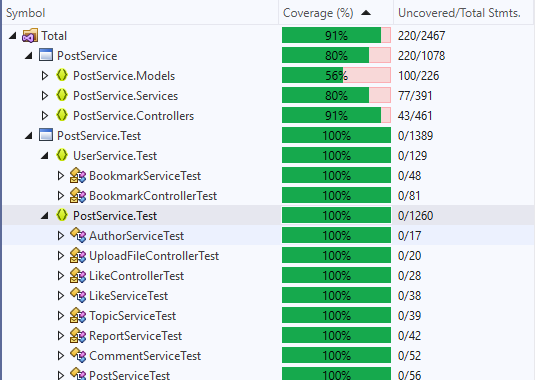
*Figure 5-6: Unit test chat service coverage*



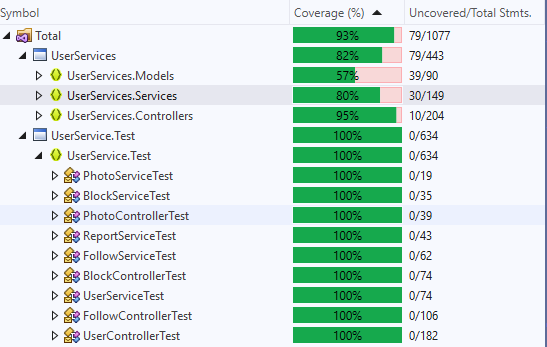
*Figure 5-7: Unit test identity provider service coverage*

**

*Figure 5-8: Unit test notification service coverage*

**

*Figure 5-9: Unit test post service coverage*

**

*Figure 5-10: Unit test user service coverage*

**5.3.2 Integration testing and System test**

Detailed Test cases will be described in TripSharing\_IntegartionTest\_Phase1.xlsx file and TripSharing\_IntegartionTest\_Phase2.xlsx file.

* Integrated test cases must ensure that the feature between different module must works well.

As a standard definition, Trip-Sharing Project defines that a test case is:

* A set of test data, expected results and actual results. A test case validates one or more system requirements and generates a pass or fail
* Trip-Sharing Project System testing will not focus on common logic of system like length of text but focus on behavior of website and aims to validate that all software module dependencies are functionally correct
* Data integrity is maintained between separate modules for the entire solution.

**5.3.4 Acceptance Test**

Acceptance testing is a level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system’s compliance with the business requirements and assess whether it is acceptable for delivery. But our project will use the checklist as a substitute for acceptance testing.

The content of the checklist is shown in the table below:

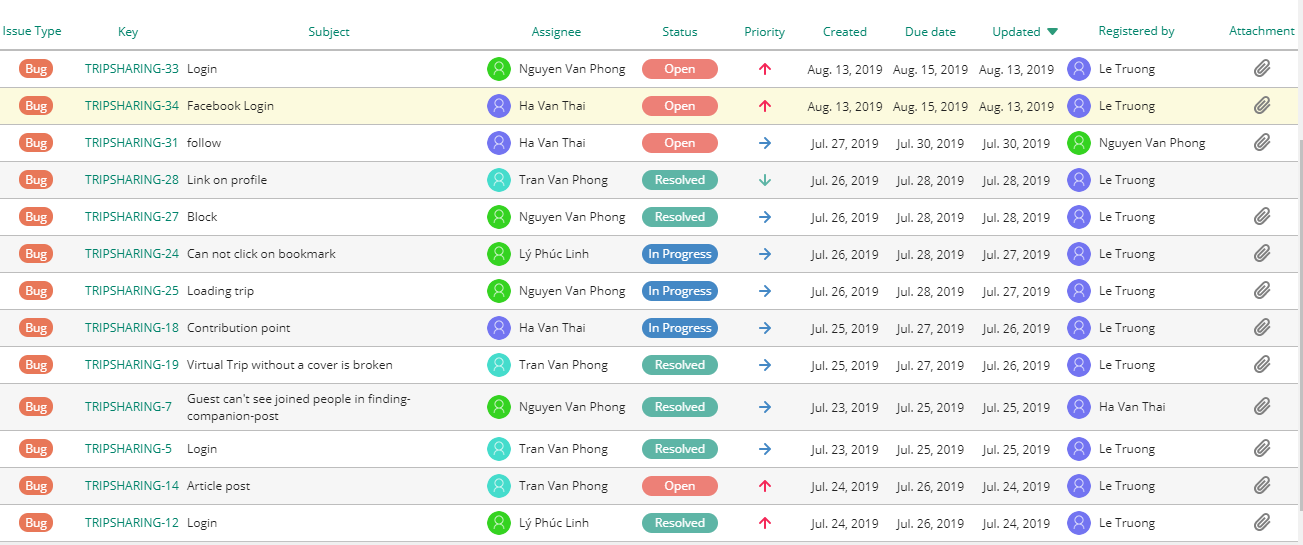
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Checklists** | **Yes** | **No** |
| **Functional** | | | |
| CL-001 | User can sign up a new account | ✓ |  |
| CL-002 | User can sign in to trip-sharing.net by facebook account | ✓ |  |
| CL-003 | User can sign in to trip-sharing.net by google account | ✓ |  |
| CL-004 | User can sign in to trip-sharing.net by normal account | ✓ |  |
| CL-005 | User can change the new password for account | ✓ |  |
| CL-006 | User can reset the password | ✓ |  |
| CL-007 | User can update profile and update information | ✓ |  |
| CL-008 | User can choose more interested topics | ✓ |  |
| CL-009 | User can create a new article | ✓ |  |
| CL-010 | User can create a new virtual trip | ✓ |  |
| CL-011 | User can create a new finding companion post | ✓ |  |
| CL-012 | Users are allowed to delete their own posts | ✓ |  |
| CL-013 | Users are allowed to edit their own posts | ✓ |  |
| CL-014 | User can join to a companion group that other member created | ✓ |  |
| CL-015 | User can leave to a companion group that other member created | ✓ |  |
| CL-016 | User can share one or more posts from trip-sharing.net to facebook application | ✓ |  |
| CL-017 | If a article violates the website's rules, user can report it to the administrator. | ✓ |  |
| CL-018 | If a comment violates the website's rules, users can report it to the administrator. | ✓ |  |
| CL-019 | If a member violates the website's rules, users can report them to the administrator. | ✓ |  |
| CL-020 | User can bookmark one or more posts and add them to the bookmarks list. | ✓ |  |
| CL-021 | User can view bookmarks list | ✓ |  |
| CL-022 | User can remove one or more posts from bookmarks list. | ✓ |  |
| CL-023 | User can add a new comment below the content of any post | ✓ |  |
| CL-024 | User can edit their own comments | ✓ |  |
| CL-025 | User can delete their own comments | ✓ |  |
| CL-026 | User can like any comment | ✓ |  |
| CL-027 | User can unlike any comment | ✓ |  |
| CL-028 | User can follow other member | ✓ |  |
| CL-029 | User can unfollow other member | ✓ |  |
| CL-030 | User can block one or more other members | ✓ |  |
| CL-031 | User can unblock one or more other members | ✓ |  |
| CL-032 | User can like one or more posts | ✓ |  |
| CL-033 | User can unlike one or more posts | ✓ |  |
| CL-034 | User can create a group chat | ✓ |  |
| CL-035 | User can remove a member from group chat |  |  |
| CL-036 | User can send or receive messages and system will notify for user when they have new messages | ✓ |  |
| CL-037 | Administrator can add a new travel topic to topics list | ✓ |  |
| CL-038 | Administrator can remove one or more travel topic to topics list | ✓ |  |
| CL-039 | Administrator can edit a travel topic | ✓ |  |
| CL-040 | Users can search 3 posts type: article, virtual trip and finding companion | ✓ |  |
| CL-041 | User can search a member from members list of system | ✓ |  |
| CL-042 | Users can view recommended articles on home page | ✓ |  |
| CL-043 | Users can view popular articles on home page | ✓ |  |
| CL-044 | Users can view recently articles on home page | ✓ |  |
| CL-045 | Users can view a post detail with category: article, virtual trip and finding companion | ✓ |  |
| CL-046 | Users can view a other member’s profile | ✓ |  |
| CL-047 | Users can view a other member's followings/followers number | ✓ |  |
| CL-048 | Users can view a other member's created posts | ✓ |  |
| CL-049 | Administrator can view all user’s account | ✓ |  |
| CL-050 | Administrator can view all accounts that users reported | ✓ |  |
| CL-051 | Administrator can ban an account | ✓ |  |
| CL-052 | Administrator can unban an account |  |  |
| CL-053 | Administrator can view all posts reported by users | ✓ |  |
| CL-054 | Administrator can remove a post reported by users | ✓ |  |
| CL-055 | Administrator can restore a post that previously removed | ✓ |  |
| CL-056 | Administrator can view all comments that is reported by other members | ✓ |  |
| CL-057 | Administrator can remove a comment that is reported by other members | ✓ |  |
| CL-058 | Administrator can restore a comment that previously removed | ✓ |  |
| CL-059 | Administrator can view statistic of users | ✓ |  |
| CL-060 | Administrator can view statistic of posts | ✓ |  |
| CL-061 | Check application logout functionality. | ✓ |  |
| **GUI and Usability** | | | |
| CL-062 | All mandatory fields are validated. | ✓ |  |
| CL-063 | Validation error messages are displayed properly below the field. | ✓ |  |
| CL-064 | All error messages are displayed in red color. | ✓ |  |
| CL-065 | Delete functionality for any record on page are asked for confirmation. | ✓ |  |
| CL-066 | All like, comment number values are formatted properly. | ✓ |  |
| CL-067 | Application crash or unavailable pages are redirected to error page. | ✓ |  |
| CL-068 | The screen is well organized and easy to interact with. | ✓ |  |
| CL-069 | All fields on page (e.g. text box, radio options, dropdown lists) should be aligned properly. | ✓ |  |
| CL-070 | System display notification message when meet trouble, error. | ✓ |  |
| CL-071 | All images displayed on the website must be of good quality | ✓ |  |
| **Database** | | | |
| CL-072 | All data must saved in database when submit data successful. | ✓ |  |
| CL-073 | The id field in the data base is required, cannot be left blank. | ✓ |  |
| CL-074 | Field length shown to user on page and in database schema should be same. | ✓ |  |
| CL-075 | Database fields are designed with correct data type and data length. | ✓ |  |
| CL-076 | The relationship between the tables must be clear and true to the customer's requirements | ✓ |  |

**5.3.5 Defect Log**

* Trip Sharing project used http://www.trello.com in phase 1 and http://www.backlog.com in phase 2 to manager tasks and defects.
* Every member of Trip-Sharing project creates an account backlog and trello to take part in activities: control bugs, fix bugs, re-test bugs and close bug. Bug will be log by tester or developer in develop progress.



*Figure 5-11: Control task and bug with Trello*



*Figure 5-12: Control task and bug with Backlog*

**5.4 Test Report**

**5.4.1 Unit test report**

The contents of the Unit Test Case Report are shown in the table below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Service Name** | **Controller/Service Name** | **Pass** | **Fail** | **Not available** | **Number of Test Case** |
| Chat Service | Chat Controller | 15 | 0 | 0 | 15 |
| Chat Service | 10 | 0 | 0 | 10 |
| **Total** | **25** | **0** | **0** | **25** |
| Identity Provider | Account Controller | 16 | 0 | 0 | 16 |
| Account Service | 15 | 0 | 0 | 15 |
| **Total** | **31** | **0** | **0** | **31** |
| Notification Service | Notification Controller | 2 | 0 | 0 | 2 |
| Notification Service | 2 | 0 | 0 | 2 |
| **Total** | **4** | **0** | **0** | **4** |
| Post Service | Article Controller | 11 | 0 | 0 | 11 |
| Article Service | 8 | 0 | 0 | 8 |
| Author Service | 2 | 0 | 0 | 2 |
| Bookmark Controller | 6 | 0 | 0 | 6 |
| Bookmark Service | 5 | 0 | 0 | 5 |
| Comment Controller | 5 | 0 | 0 | 5 |
| Commnet Service | 5 | 0 | 0 | 5 |
| Companion Controller | 13 | 0 | 0 | 13 |
| Companion Post Service | 15 | 0 | 0 | 15 |
| Like Controller | 2 | 0 | 0 | 2 |
| Like Service | 4 | 0 | 0 | 4 |
| Post Controller | 6 | 0 | 0 | 6 |
| Post Service | 6 | 0 | 0 | 6 |
| Report Controller | 6 | 0 | 0 | 6 |
| Report Service | 5 | 0 | 0 | 5 |
| Topic Controller | 8 | 0 | 0 | 8 |
| Topic Service | 5 | 0 | 0 | 5 |
| Upload File Controller | 2 | 0 | 0 | 2 |
| Upload File Service | 1 | 0 | 0 | 1 |
| Virtual Trip Controller | 7 | 0 | 0 | 7 |
| Virtual Trip Service | 7 | 0 | 0 | 7 |
| **Total** | **133** | 0 | 0 | **133** |
| User Service | Block Controller | 6 | 0 | 0 | 6 |
| Block Service | 4 | 0 | 0 | 4 |
| Follow Controller | 10 | 0 | 0 | 10 |
| Follow Service | 7 | 0 | 0 | 7 |
| Photo Controller | 3 | 0 | 0 | 3 |
| Photo Service | 2 | 0 | 0 | 2 |
| Report Service | 5 | 0 | 0 | 5 |
| User Controller | 19 | 0 | 0 | 19 |
| User Service | 10 | 0 | 0 | 10 |
| **Total** | **66** | 0 | 0 | **66** |
| **Total of Test Case** | | **259** | **0** | **0** | **259** |

*Table 5-2: Unit test case report*

|  |  |  |
| --- | --- | --- |
| **Service Name** | **Coverage** | |
| **Phase 1** | **Phase 2** |
| Chat Service | 96% | 96% |
| Identity Provider | 87% | 87% |
| Notify Service | 95% | 95% |
| Post Service | 88% | 91% |
| User Service | 90% | 93% |

*Table 5-3: Unit test coverage report*

**5.4.2 Integration test report**

|  |  |  |  |
| --- | --- | --- | --- |
| **Module code** | **Pass** | **Fail** | **Number of test case** |
| Sign Up | 17 | 0 | 17 |
| Sign In | 14 | 0 | 14 |
| Forgot password | 6 | 0 | 6 |
| Sign Out | 2 | 0 | 2 |
| Change Password | 16 | 0 | 16 |
| Update Profile | 14 | 0 | 14 |
| Choose Interested Topic | 14 | 0 | 14 |
| View user's profile | 2 | 0 | 2 |
| Create a post | 60 | 0 | 60 |
| Edit a post | 59 | 0 | 59 |
| Like Unlike a post | 7 | 0 | 7 |
| Delete a post | 11 | 0 | 11 |
| Comment to a post | 6 | 0 | 6 |
| Edit a comment | 5 | 0 | 5 |
| Delete a comment | 5 | 0 | 5 |
| Like Unlike a comment | 8 | 0 | 8 |
| View a post detail | 14 | 0 | 14 |
| Bookmark a post | 6 | 0 | 6 |
| View all bookmarks | 2 | 0 | 2 |
| Remove a bookmark | 7 | 0 | 7 |
| Follow Unfollow an user | 13 | 0 | 13 |
| ViewPost | 6 | 0 | 6 |
| View user's follower following | 3 | 0 | 3 |
| View user's created posts | 6 | 0 | 6 |
| TravelTopic | 14 | 0 | 14 |
| Search | 11 | 0 | 11 |
| Block Unblock an user | 8 | 0 | 8 |
| Join Leave to a companion group | 5 | 0 | 5 |
| Share a post | 4 | 0 | 4 |
| Send receive messages | 9 | 0 | 9 |
| View all users' account | 3 | 0 | 3 |
| View Statistic | 5 | 0 | 5 |
| View Reported | 3 | 0 | 3 |
| Reported | 17 | 0 | 17 |
| Ban an user | 4 | 0 | 4 |
| Unban an user | 4 | 0 | 4 |
| Remove a reported post | 4 | 0 | 4 |
| Restore a removed post | 4 | 0 | 4 |
| Remove a reported comment | 4 | 0 | 4 |
| Restore a removed comment | 4 | 0 | 4 |
| **Total of Test Case** | **406** | **0** | **406** |

**5.4.3 System test report**