A Real Time and Interactive Dashboard in Tourism Industry

**Test Record**

By

Junyu Zhou 592115508

Yawei Li 592115518

Department of Software Engineering,

College of Arts, Media and Technology,

ChiangMai University

Project Advisor



Table of Contents

[1. Document History 6](#_Toc27051779)

[2. Introduction 8](#_Toc27051780)

[2.1 Purpose 8](#_Toc27051781)

[2.2 Responsibility 8](#_Toc27051782)

[2.3 Acronyms 8](#_Toc27051783)

[3. Unit Testing 9](#_Toc27051784)

[UTC-01: Front-end: Method name: onUsernameChanged (e) 9](#_Toc27051785)

[UTC-02: Front-end: Method name: onPasswordChanged (e) 10](#_Toc27051786)

[UTC-03: Front-end: Method name: submitLogin (e) 11](#_Toc27051787)

[UTC-04: Front-end: Method name: logout () 12](#_Toc27051788)

[UTC-05: Front-end: Method name: showSummary () 13](#_Toc27051789)

[UTC-06: Front-end: Method name: componentDidMount () 17](#_Toc27051790)

[UTC-07: Front-end: Method name: getAllHotel () 17](#_Toc27051791)

[UTC-08: Front-end: Method name: getHotel () 18](#_Toc27051792)

[UTC-09: Front-end: Method name: submit(e) 19](#_Toc27051793)

[UTC-10: Front-end: Method name: getAll () 20](#_Toc27051794)

[UTC-11: Back-end: Method name: login () 22](#_Toc27051795)

[UTC-12: Back-end: Method name: admin\_login () 22](#_Toc27051796)

[UTC-13: Back-end: Method name: delete\_account (username) 24](#_Toc27051797)

[UTC-14: Back-end: Method name: register () 25](#_Toc27051798)

[UTC-15: Back-end: Method name: get\_all\_account () 26](#_Toc27051799)

[UTC-16: Back-end: Method name: edit\_username () 27](#_Toc27051800)

[UTC-17: Back-end: Method name: edit\_password () 28](#_Toc27051801)

[UTC-18: Back-end: Method name: approve\_user () 29](#_Toc27051802)

[UTC-19: Back-end: Method name: reject\_user () 30](#_Toc27051803)

[UTC-20: Back-end: Method name: get\_pending () 31](#_Toc27051804)

[UTC-21: Back-end: Method name: add\_user () 32](#_Toc27051805)

[UTC-22: Back-end: Method name: get\_account\_by\_username () 33](#_Toc27051806)

[UTC-23: Back-end: Method name: get\_all () 34](#_Toc27051807)

[UTC-24: Back-end: Method name: get\_count\_by\_date () 35](#_Toc27051808)

[UTC-25: Back-end: Method name: get\_count\_by\_date\_positive () 36](#_Toc27051809)

[UTC-26: Back-end: Method name: get\_count\_by\_date\_negative () 37](#_Toc27051810)

[UTC-27: Back-end: Method name: get\_count\_by\_date\_neutral () 38](#_Toc27051811)

[UTC-28: Back-end: Method name: get\_hotel () 39](#_Toc27051812)

[UTC-29: Back-end: Method name: get\_hotel\_by\_name () 40](#_Toc27051813)

[UTC-30: Back-end: Method name: add\_comment () 41](#_Toc27051814)

[UTC-31: Back-end: Method name: record\_log () 42](#_Toc27051815)

[UTC-32: Front-end: Method name: Heatmap () 45](#_Toc27051816)

[UTC-33: Front-end: Method name: WordCloud () 46](#_Toc27051817)

[UTC-34: Back-end: Method name: get\_words () 47](#_Toc27051818)

[UTC-35: Back-end: Method name: get\_Top10 () 48](#_Toc27051819)

[UTC-36: Back-end: Method name: get\_positive\_line () 49](#_Toc27051820)

[4. System Testing 50](#_Toc27051821)

[STC-01: Admin can login to the system and manage accounts. 50](#_Toc27051822)

[STC-02: Admin can view system log 54](#_Toc27051823)

[STC-03: Staff can login or register account. 55](#_Toc27051824)

[STC-04: Staff can edit information 56](#_Toc27051825)

[STC-05: Staff can logout from system. 57](#_Toc27051826)

[STC-06: View the summary of data visualization result 58](#_Toc27051827)

[STC-07: View the comments 60](#_Toc27051828)

[STC-08: Write the comments 61](#_Toc27051829)

[STC-09: View the heatmap of data 62](#_Toc27051830)

[STC-10: View the Wordcloud of data 64](#_Toc27051831)

# 1. Document History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| History | Status | Date | Viewable | Editable | Responsible |
| Project-Test Record\_v1.docx  - Create:  Document History  Introduction  Unit Testing  System Testing | Draft | 29 July, 2019 | ZJY, LYW,  AJP | ZJY, LYW | ZJY, LYW |
| Project-Test Record\_v2.docx  - Modify:  Unit Testing  System Testing | Draft | 23 Aug 2019 | ZJY, LYW,  AJP | ZJY, LYW | ZJY, LYW |
| Project-Test Record\_v3.docx  - Modify:  Unit Testing  System Testing | Draft | 8 Oct, 2019 | ZJY, LYW,  AJP | ZJY, LYW | ZJY, LYW |
| Project-Test Record\_v4.docx  - Modify:  Unit Testing  System Testing | Final | 12 Dec, 2019 | ZJY, LYW,  AJP | ZJY, LYW | ZJY, LYW |

ZJY = Junyu Zhou

LYW = Yawei Li

AJP = Dr. Pree Thiengburanathum

# 2. Introduction

### 2.1 Purpose

The purpose of test record for A Real Time and Interactive Dashboard in Tourism Industry is for recording the actual result and pass/fail criteria of each test case that have designed in the test plan and executed.

### 2.2 Responsibility

|  |  |
| --- | --- |
| Item | Responsibility |
| Unit test of web application | ZJY, LYW |
| Record unit test of web application | ZJY, LYW |
| System test of web application | ZJY, LYW |
| Record system test of web application | ZJY, LYW |

### 2.3 Acronyms

URS = User Requirement Specification

UTC = Unit Test Case

STC = System Test Case

# 3. Unit Testing

### UTC-01: ****Front-end: Method name: onUsernameChanged (e)****

**Test Date:** 23 Aug 2019

**Description:** This method will validate format of username.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 1 | - | - | Pass |
| 2 | console.log(“Username cannot contain special characters!”) | console.log(“Username cannot contain special characters!”) | Pass |
| 3 | console.log (“Format correct”) | console.log (“Format correct”) | Pass |

### UTC-02: ****Front-end: Method name: onPasswordChanged (e)****

**Test Date:** 23 Aug 2019

**Description:** This method will validate format of password.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 4 | console.log (“Password must be more than 6 digits!”) | console.log (“Password must be more than 6 digits!”) | Pass |
| 5 | Password= admin001 | Password= admin001 | Pass |
| 6 | - | - | Pass |

### UTC-03: ****Front-end: Method name: submitLogin (e)****

**Test Date:** 23Aug 2019

**Description:** This method will login to the dashboard.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 7 | console.log (“login successful”) | console.log (“login successful”) | Pass |
| 8 | console.log (“login failed”) | console.log (“login failed”) | Pass |

### UTC-04: ****Front-end: Method name: logout ()****

**Test Date:** 23 Aug 2019

**Description:** This method will log out from dashboard.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 9 | console.log (“Log out successfully”) | console.log (“Log out successfully”) | Pass |

### UTC-05: ****Front-end: Method name: showSummary ()****

**Test Date:** 23 Aug 2019

**Description:** This method will display summary on the dashboard page.

**Test By:** ZJY, LYW

Test Data:

sparklineData = [47, 45, 54, 38, 56, 24, 65, 31, 37, 39, 62, 51, 35, 41, 35, 27, 93, 53, 61, 27, 54, 43, 19, 46]

expectedData2 = {

seriesSpark1: [{data: [25, 66, 41, 89, 63, 25, 44, 12, 36, 9, 54]}],

seriesSpark2: [{data: [47, 45, 74, 14, 56, 74, 14, 11, 7, 39, 82]}],

seriesSpark3: [{data: [12, 14, 2, 47, 42, 15, 47, 75, 65, 19, 14]}],

seriesSpark4: [{data: [15, 75, 47, 65, 14, 2, 41, 54, 4, 27, 15]}],

}

expectedData3 = {

total\_comments: '135,965',

positive\_comments: '99,821',

negative\_comments: '10,212',

neutral\_comments: '25,932',

percent\_total: '100%',

percent\_positive: '73.4%',

percent\_negative: '7.5%',

percent\_neutral: '19.1%',

}

expectedData4 = [

[ 1483203600000, 5], [ 1483808400000, 22],

[ 1484413200000, 52], [ 1485018000000, 87],

[ 1485622800000, 57], [ 1486227600000, 84],

[ 1486832400000, 87], [ 1487437200000, 43],

[ 1488042000000, 29], [ 1488646800000, 69],

[ 1489251600000, 81], [ 1489856400000, 46],

[ 1490461200000, 14], [ 1491066000000, 79],

[ 1491670800000, 9], [ 1492275600000, 51],

[ 1492880400000, 76], [ 1493485200000, 9],

[ 1494090000000, 51], [ 1494694800000, 77],

[ 1495299600000, 22], [ 1495904400000, 96],

[ 1496509200000, 18], [ 1497114000000, 67],

[ 1497718800000, 24], [ 1498323600000, 51],

[ 1498928400000, 41], [ 1499533200000, 83],

[ 1500138000000, 56], [ 1500742800000, 42],

[ 1501347600000, 76], [ 1501952400000, 20],

[ 1502557200000, 73], [ 1503162000000, 14],

[ 1503766800000, 67], [ 1504371600000, 8],

[ 1504976400000, 22], [ 1505581200000, 64],

[ 1506186000000, 44], [ 1506790800000, 1],

[ 1507395600000, 34], [ 1508000400000, 34],

[ 1508605200000, 70], [ 1509210000000, 16],

[ 1509814800000, 20], [ 1510419600000, 17],

[ 1511024400000, 86], [ 1511629200000, 78]

]

expectedData5 = [400, 430, 448, 470, 540, 580, 690, 1100, 1200, 1380]

expectedData6 = [{name: 'Neutral',

data: [44, 55, 57, 56, 61, 58, 63, 60, 66]}, {

name: 'Positive',

data: [76, 85, 101, 98, 87, 105, 91, 114, 94]}, {

name: 'Negative',

data: [35, 41, 36, 26, 45, 48, 52, 53, 41]}];

expectedData7 = [{name: 'Neutral',

data: [44, 55, 41, 67, 22, 43, 21, 49, 39]}, {

name: 'Positive',

data: [13, 23, 20, 8, 13, 27, 33, 12, 14]}, {

name: 'Negative',

data: [11, 17, 15, 15, 21, 14, 15, 13, 9]}];

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 10 | console.log (“Summary is showing!”) | console.log (“Summary is showing!”) | Pass |
| 11 | {sparklineData}  {expectedData2}  {expectedData3} | {sparklineData}  {expectedData2}  {expectedData3} | Pass |
| 12 | {expectedData4} | {expectedData4} | Pass |
| 13 | {expectedData5} | {expectedData5} | Pass |
| 14 | {expectedData6} | {expectedData6} | Pass |
| 15 | {expectedData7} | {expectedData7} | Pass |

### UTC-06: ****Front-end: Method name: componentDidMount ()****

**Test Date:** 27 Sep 2019

**Description**: This method is initial method when component render.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 16 | Console displays (“View component render!”) | Console displays (“View component render!”) | Pass |

### UTC-07: ****Front-end: Method name: getAllHotel ()****

**Test Date:** 27 Sep 2019

**Description**: This method is used to get all hotels.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 17 | expectedData8 | expectedData8 | Pass |
| 18 | [] | [] | Pass |
| 19 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-08: ****Front-end: Method name: getHotel ()****

**Test Date:** 27 Sep 2019

**Description**: This method is used to get data of a hotel.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 20 | expectedData9 | expectedData9 | Pass |
| 21 | [] | [] | Pass |
| 22 | [] | [] | Pass |

### UTC-09: ****Front-end: Method name: submit(e)****

**Test Date:** 27 Sep 2019

**Description**: This method is used to submit a comment.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 23 | {“message”: “comment added”} | {“message”: “comment added”} | Pass |
| 24 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-10: ****Front-end: Method name: getAll ()****

**Test Date:** 8 Oct 2019

**Description**: This method is used to get data of log information.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 20 | expectedData10 | expectedData10 | Pass |
| 21 | [] | [] | Pass |
| 22 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-11: ****Back-end: Method name: login ()****

**Test Date:** 8 Oct 2019

**Description**: This method will receive http POST request “/users/login”

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 17 | {“message": "Login successfully", "token": "eyJ…g34GY", "user": "admin001”} | {“message": "Login successfully", "token": "eyJ…g34GY", "user": "admin001”} | Pass |
| 18 | {“error": "Invalid username and password"} | {“error": "Invalid username and password"} | Pass |

### UTC-12: ****Back-end: Method name: admin\_login ()****

**Test Date:** 23 Aug 2019

**Description**: This method will receive http POST request “/admin/login”

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 19 | {“message": "Super admin login successfully", "token": "eyJ0…TEjCzJln\_wXR79yR-s”} | {“message": "Super admin login successfully", "token": "eyJ0…TEjCzJln\_wXR79yR-s”} | Pass |
| 20 | {“error": "Invalid username and password"} | {“error": "Invalid username and password"} | Pass |

### UTC-13: ****Back-end: Method name: delete\_account (username)****

**Test Date: 23 Aug 2019**

**Description**: This method will receive http DELETE request “/admin/delete/<username>”

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 21 | {“result": {“message": "User deleted”}} | {“result": {“message": "User deleted”}} | Pass |
| 22 | {“result”: {  "message": "No user found"}  } | {“result”: {  "message": "No user found"}  } | Pass |

### UTC-14: ****Back-end: Method name: register ()****

**Test Date: 23 Aug 2019**

**Description**: This method will receive http POST request “users/register”

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 23 | {"result": {  "message": “admin001 registered”}  } | {"result": {  "message": “admin001 registered”}  } | Pass |

### UTC-15: ****Back-end: Method name: get\_all\_account ()****

**Test Date: 23 Aug 2019**

**Description**: This method will receive http POST request “users/get-all”.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 24 | [{“status”: "super admin added",  "trial\_time": "Fri, 06 Sep 2019 07:05:07 GMT",  "username": "test"},  {"status": "super admin added",  "trial\_time": "Fri, 06 Sep 2019 07:05:46 GMT",  "username": "test02"}] | [{“status”: "super admin added",  "trial\_time": "Fri, 06 Sep 2019 07:05:07 GMT",  "username": "test"},  {"status": "super admin added",  "trial\_time": "Fri, 06 Sep 2019 07:05:46 GMT",  "username": "test02"}] | Pass |
| 24 | [] | [] | Pass |

### UTC-16: ****Back-end: Method name: edit\_username ()****

**Test Date:** 23 Aug 2019

**Description**: This method will receive http POST request “users/edit/username”.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 25 | {"result": {  "message": "test is changed to feiji@test.com"}} | {"result": {  "message": "test is changed to feiji@test.com"}} | Pass |
| 26 | {  "result": {  "message": "No user found"  }  } | {  "result": {  "message": "No user found"  }  } | Pass |

### UTC-17: ****Back-end: Method name: edit\_password ()****

**Test Date:** 23 Aug 2019

**Description**: Description: This method will receive http POST request “users/edit/password”.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 27 | {  "message": "Password changed"  } | {  "message": "Password changed"  }  } | Pass |
| 28 | {"error": "Invalid username and password"} | {"error": "Invalid username and password"} | Pass |

### UTC-18: ****Back-end: Method name: approve\_user ()****

**Test Date:** 23 Aug 2019

**Description**: This method will receive http POST request “admin/approve”.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 29 | {"result": {"message": "testagain's request is approved"}} | {"result": {"message": "testagain's request is approved"}} | Pass |
| 30 | {"result": {"message": "No user found"}} | {"result": {"message": "No user found"}} | Pass |

### UTC-19: ****Back-end: Method name: reject\_user ()****

**Test Date:** 23 Aug 2019

**Description**: This method will receive http POST request “admin/reject”.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 31 | {"result": {"message": "testagain's request is rejected"}} | {"result": {"message": "testagain's request is rejected"}} | Pass |
| 32 | {"result": {"message": "No user found"}} | {"result": {"message": "No user found"}} | Pass |

### UTC-20: ****Back-end:**** ****Method name: get\_pending ()****

**Test Date:** 23 Aug 2019

**Description**: This method will receive http GET request “admin/get-pending”.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 33 | {"results": [{"status": "pending",  "trial\_time": "pending",  "username": "hello"}]} | {"results": [{"status": "pending",  "trial\_time": "pending",  "username": "hello"}]} | Pass |
| 34 | {  "results": []  } | {  "results": []  } | Pass |

### UTC-21: ****Back-end: Method name: add\_user ()****

**Test Date:** 23 Aug 2019

**Description**: This method will receive http POST request “admin/add”.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 35 | {"result": {"message": "test\_admin\_add@test.com added"}} | {"result": {"message": "test\_admin\_add@test.com added"}} | Pass |

### UTC-22: ****Back-end: Method name: get\_account\_by\_username ()****

**Test Date:** 23 Aug 2019

**Description**: This method will receive http GET request “/users/get-by-username”.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 36 | {"result": {"message": "User found",  "status": "approved",  "trial\_time ": "Fri, 06 Sep 2019 07:34:06 GMT",  "username": "test\_admin\_add@test.com"}} | {"result": {"message": "User found",  "status": "approved",  "trial\_time ": "Fri, 06 Sep 2019 07:34:06 GMT",  "username": "test\_admin\_add@test.com"}} | Pass |
| 37 | {  "result": {  "message": "No user found”}  } | {  "result": {  "message": "No user found"}  } | Pass |

### UTC-23: ****Back-end: Method name: get\_all ()****

**Test Date:** 27 Sep 2019

**Description**: This method will receive http GET request and get comment data from database.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 50 | expectedData11 | expectedData11 | Pass |
| 51 | [] | [] | Pass |
| 52 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-24: ****Back-end: Method name: get\_count\_by\_date ()****

**Test Date:** 27 Sep 2019

**Description**: This method will receive http GET request and get the comment count by date from database.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 53 | expectedData12 | expectedData12 | Pass |
| 54 | [] | [] | Pass |
| 55 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-25: ****Back-end: Method name: get\_count\_by\_date\_positive ()****

**Test Date:** 27 Sep 2019

**Description**: This method will receive http GET request and get the positive comment count by date from database.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 56 | expectedData13 | expectedData13 | Pass |
| 57 | [] | [] | Pass |
| 58 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-26: ****Back-end: Method name: get\_count\_by\_date\_negative ()****

**Test Date:** 27 Sep 2019

**Description**: This method will receive http GET request and get the negative comment count by date from database.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 59 | expectedData14 | expectedData14 | Pass |
| 60 | [] | [] | Pass |
| 61 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-27: ****Back-end: Method name: get\_count\_by\_date\_neutral ()****

**Test Date:** 27 Sep 2019

**Description**: This method will receive http GET request and get the neutral comment count by date from database.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 62 | expectedData15 | expectedData15 | Pass |
| 63 | [] | [] | Pass |
| 64 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-28: ****Back-end: Method name: get\_hotel ()****

**Test Date:** 27 Sep 2019

**Description**: This method will receive http GET request and get the number of comments of each hotel from database.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 65 | expectedData16 | expectedData16 | Pass |
| 66 | [] | [] | Pass |
| 67 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-29: ****Back-end: Method name: get\_hotel\_by\_name ()****

**Test Date:** 27 Sep 2019

**Description**: This method will receive http GET request and get the comment of inputting hotel from database.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 68 | expectedData9 | expectedData9 | Pass |
| 69 | [] | [] | Pass |
| 70 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |
| 71 | [] | [] | Pass |

### UTC-30: ****Back-end: Method name: add\_comment ()****

**Test Date:** 27 Sep 2019

**Description**: This method will receive http POST request and store the comment to database.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 72 | expectedData17 | expectedData17 | Pass |
| 73 | expectedData18 | expectedData18 | Pass |
| 74 | expectedData19 | expectedData19 | Pass |
| 75 | expectedData19 | expectedData19 | Pass |
| 76 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-31: ****Back-end: Method name: record\_log ()****

**Test Date:** 27 Sep 2019

**Description**: This method will receive http GET request and get comment data from database.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 77 | {"content": "super admin logged in",  "creator": "super admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Login" }, | {"content": "super admin logged in",  "creator": "super admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Login" }, | Pass |
| 78 | {"content": "admin logged in",  "creator": "admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Login" }, | {"content": "admin logged in",  "creator": "admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Login" }, | Pass |
| 79 | {"content": "admin wrote a comment", "creator": "admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Comment" }, | {"content": "admin wrote a comment", "creator": "admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Comment" }, | Pass |
| 80 | {"content": "admin edit username", "creator": "admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Edit" }, | {"content": "admin edit username", "creator": "admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Edit" }, | Pass |
| 81 | {"content": "admin edit password", "creator": "admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Edit" }, | {"content": "admin edit password", "creator": "admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Edit" }, | Pass |
| 82 | {"content": "admin login failed", "creator": "admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Login" }, | {"content": "admin login failed", "creator": "admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Login" }, | Pass |
| 83 | {"content": "super admin delete an account", "creator": "super admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Delete" }, | {"content": "super admin delete an account", "creator": "super admin", "log\_id": 1,"time": "2019-10-08 15:40:37", "type": "Delete" }, | Pass |
| 84 | {"content": "all user approved.", "creator": "super admin", "log\_id": 1,"time": "201910-08 15:40:37", "type": "Approve" }, | {"content": "all user approved.", "creator": "super admin", "log\_id": 1,"time": "201910-08 15:40:37", "type": "Approve" }, | Pass |
| 85 | {"content": "all user rejected.", "creator": "super admin", "log\_id": 1,"time": "201910-08 15:40:37", "type": "Reject" }, | {"content": "all user rejected.", "creator": "super admin", "log\_id": 1,"time": "201910-08 15:40:37", "type": "Reject" }, | Pass |

### UTC-32: ****Front-end: Method name: Heatmap ()****

**Test Date:** 28 November 2019

**Description**: This method is used to get data from backend and call the chart API.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 86 |  |  | Pass |
| 87 | console.log(“Heatmap is showing!”) | console.log(“Heatmap is showing!”) | Pass |
| 88 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-33: ****Front-end: Method name: WordCloud ()****

**Test Date:** 28 November 2019

Description: This method is used to get data from backend and call the chart API.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 89 |  |  | Pass |
| 90 | console.log(“WordCloud is showing!”) | console.log(“WordCloud is showing!”) | Pass |
| 91 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |

### UTC-34: ****Back-end: Method name: get\_words ()****

**Test Date:** 28 November 2019

Description: This method will get the frequency of each word in comments from database.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 92 | expectedData20 | expectedData20 | Pass |
| 93 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |
| 94 | alert: “Database is empty!” | alert: “Database is empty!” | Pass |

### UTC-35: ****Back-end: Method name: get\_Top10 ()****

**Test Date:** 28 November 2019

Description: This method will get the comments details of top 10 hotels with the most comments.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 95 | expectedData21 | expectedData21 | Pass |
| 96 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |
| 97 | alert: “Database is empty!” | alert: “Database is empty!” | Pass |

### UTC-36: ****Back-end: Method name: get\_positive\_line ()****

**Test Date:** 28 November 2019

Description: This method will get the positive comments of each year.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 98 | expectedData22 | expectedData22 | Pass |
| 99 | alert: “Cannot connect to database, please try again” | alert: “Cannot connect to database, please try again” | Pass |
| 100 | alert: “Database is empty!” | alert: “Database is empty!” | Pass |

# 4. System Testing

### STC-01: Admin can login to the system and manage accounts.

**Test Date:** 23 Aug 2019

**Description:** The system testing for URS-01 – URS07. The system should provide an interface. The interface includes “All account”, “Waiting List”, “Dashboard”, and “Logout” buttons. Admin can log in to system, view all accounts or waiting list page, edit account information, delete account, add staff account, accept or reject staff account register request.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 1 | The system redirects to super admin page. | The system redirects to super admin page. | Pass |
| 2 | Show an error message “Login failed” | Show an error message “Login failed” | Pass |
| 3 | Show all accounts information | Show all accounts information | Pass |
| 4 | Show waiting list information | Show waiting list information | Pass |
| 5 | Show a message “No record to display” | Show a message “No record to display” | Pass |
| 6 | Show a message “Cannot connect to database” then provide refresh and cancel button | Show a message “Cannot connect to database” then provide refresh and cancel button | Pass |
| 7 | Show a message “No record to display” | Show a message “No record to display” | Pass |
| 8 | Show a message “Cannot connect to database” and provide refresh and cancel button | Show a message “Cannot connect to database” and provide refresh and cancel button | Pass |
| 9 | User’s trial time increased 30 days and system saves the new information to database. | User’s trial time increased 30 days and system saves the new information to database. | Pass |
| 10 | System cancel the edit action. | System cancel the edit action. | Pass |
| 11 | User’s trial time decreased 30 days and system saves the new information to database. | User’s trial time decreased 30 days and system saves the new information to database. | Pass |
| 12 | Remove user account from database. | Remove user account from database. | Pass |
| 13 | System cancel the delete action. | System cancel the delete action. | Pass |
| 14 | System records account information to database. | System records account information to database. | Pass |
| 15 | System cancel the add admin account action. | System cancel the add admin account action. | Pass |
| 16 | System records account information to database. | System records account information to database. | Pass |
| 17 | System cancel the accept admin account register request action. | System cancel the accept admin account register request action. | Pass |
| 18 | System records account information to database. | System records account information to database. | Pass |
| 19 | System cancel the reject admin account register request action. | System cancel the reject admin account register request action. | Pass |

### STC-02: Admin can view system log

**Test Date:** 27 Sep 2019

**Description:** The system testing for URS-21 – URS-23. The system should provide an interface. The interface includes a table which shows “ID”, “User”, “Time”, “Type” and “Content” of each log. Super admin view log information and account information.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 20 | The system displays the table which shows “ID”, “User”, “Time”, “Type” and “Content” of logs. | The system displays the table which shows “ID”, “User”, “Time”, “Type” and “Content” of logs. | Pass |

### STC-03: Staff can login or register account.

**Test Date:** 23 Aug 2019

**Description:** The system testing for URS-08 –URS09. The system should provide an interface. The interface includes “Login”, “Register” buttons. Staff can log in and register.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 21 | The system redirects to admin page. | The system redirects to admin page. | Pass |
| 22 | Show an error message “Login failed” | Show an error message “Login failed” | Pass |
| 23 | The system records account information to database | The system records account information to database | Pass |

### STC-04: Staff can edit information

**Test Date:** 23 Aug 2019

**Description:** The system testing for URS-10. The system should provide an interface. The interface includes “Edit” buttons. Staff can edit their username and password.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 24 | The system stores information to database. | The system stores information to database. | Pass |
| 25 | System cancel the edit action. | System cancel the edit action. | Pass |
| 26 | The system stores information to database. | The system stores information to database. | Pass |
| 27 | System cancel the edit action. | System cancel the edit action. | Pass |

### STC-05: Staff can logout from system.

**Test Date:** 23 Aug 2019

**Description:** The system testing for URS-11. The system should provide an interface. The interface includes “Logout” buttons. Staff can logout system.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 28 | System remove user token from local storage. | System remove user token from local storage. | Pass |

### STC-06: View the summary of data visualization result

**Test Date:** 23 Aug 2019

**Description:** The system testing for URS-12 – URS-16. The system should provide an interface. The interface includes all the summary sparkline chart, positive comments rate line chart, word frequency bar chart, number of comments bar chart, types of comments stacked column chart.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 29 | System displays the data in Appendix expectedData3, expectedData4 | System displays the data in Appendix expectedData3, expectedData4 | Pass |
| 30 | System displays the data in Appendix expectedData4 | System displays the data in Appendix expectedData4 | Pass |
| 31 | System displays the data in Appendix expectedData5 | System displays the data in Appendix expectedData5 | Pass |
| 32 | System displays the data in Appendix expectedData6 | System displays the data in Appendix expectedData6 | Pass |
| 33 | System displays the data in Appendix expectedData7 | System displays the data in Appendix expectedData7 | Pass |
| 34 | Show a message “Cannot connect to database” then provide refresh and cancel button | Show a message “Cannot connect to database” then provide refresh and cancel button | Pass |

### STC-07: View the comments

**Test Date:** 27 Sep 2019

**Description:** The system testing for URS-22 – URS-24. The system should provide an interface. The interface includes a list of all hotels and the user could click to access each hotel page to view comments.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 35 | System displays the data in Appendix exceptedData11. | System displays the data in Appendix exceptedData11. | Pass |

### STC-08: Write the comments

**Test Date:** 27 Sep 2019

**Description:** The system testing for URS-25. The system should provide an interface. The interface includes an input area for inputting comment.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 36 | User could input a comment and system displays new comment at hotel page, the comment includes content, date, ID and rating. | User could input a comment and system displays new comment at hotel page, the comment includes content, date, ID and rating. | Pass |

### STC-09: View the heatmap of data

**Test Date:** 28 Nov 2019

**Description:** The system testing for URS-18 The system should provide an interface. The interface includes heatmap and legend.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 37 | System displays the data in Appendix expectedData21 | System displays the data in Appendix expectedData21 | Pass |
| 38 | System displays the chart with positive comments only. | System displays the chart with positive comments only. | Pass |
| 39 | System displays the chart with negative comments only. | System displays the chart with negative comments only. | Pass |
| 40 | System displays the chart with neutral comments only. | System displays the chart with neutral comments only. | Pass |
| 41 | Show a message “Cannot connect to database” then provide refresh and cancel button | Show a message “Cannot connect to database” then provide refresh and cancel button | Pass |

### STC-10: View the Wordcloud of data

**Test Date:** 28 Nov 2019

**Description:** The system testing for URS17. The system should provide an interface. The interface includes heatmap and legend.

**Test By:** ZJY, LYW

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Expect Result | Actual Result | Pass/Fail |
| 42 | System displays the data in Appendix expectedData20 | System displays the data in Appendix expectedData20 | Pass |
| 43 | System displays the frequency of word. | System displays the frequency of word. | Pass |
| 44 | Show a message “Cannot connect to database” then provide refresh and cancel button | Show a message “Cannot connect to database” then provide refresh and cancel button | Pass |