



Computer Science

ICBC Flex Work Test Plan

March 12, 2020
Version 0.1

Team Flex:

Srijon Saha
Yifan Wei (Kevin)
John Zou
Charlie Chen
Ravina Gill
Linh Phan

Introduction

This document is the **Test Plan** for **Flex Work**, the desk-sharing solution by *Team Flex*, for *ICBC*. We are a team of six UBC Computer Science students. This Test Plan will recognize the overall testing approach, functional and non-functional test plans, approach on test scripts, approach on security testing, approach on test data, and regression testing needs.

From the period of Jan 6 - April 26, 2020, we are creating **Flex Work**, the desk-sharing solution, for ICBC. At its core, externally, it will be a web application for employees to make their desks available and book desks made available by others, across all ICBC locations. There will also be an admin interface for updating information of office locations among other functions. Internally, this is implemented with a modern web technology stack comprising a MySQL database, a REST API server in Node.js, and a single page application in React.

Summary

We combined automated testing and manual testing in this test plan. We use Mocha, Chai, and JMeter for automated tests. The manual tester should have internet access to deployed server and database. We plan to have 2 testing cycles, the first cycle is for bug discovering. After all bugs are fixed we can start the second testing cycle. We should have all Unit and SAT testing passed in the second cycle. If a major fix is needed after the second cycle, we will have to run the third cycle.

Testing tool version

Mocha: 7.1.0

Chai 4.2.0

JMeter 5.2.1

Test Environment

Internet Browser

For all manual tests, the tests shall be run on Internet Explorer 11 (Version 11.1098.17763.0), Microsoft Edge (Version 80.0.361.50), and Firefox (Version 73.0).

Active Directory

System Acceptance Tests will use our Test Active Directory hosted on Azure. A link to the login information is in the Appendix. User Acceptance Test will use ICBC's on-premise Active Directory.

Test Database

We will provide a script to populate the Test Database. It will create a fresh MySQL database with 1000 live active bookings and 6000 users, with dates dynamically generated based on the current date so that bookings are not in the past. It should be run before each Script.

Functional Test Plan

The full set of functional components (16 total) and their full scripts are provided in the Appendix. We present three here as part of the main document.

They are representative of the primary requirements of Flex Work: lending a workspace (Component 3), selecting an office to book (Component 5), and facility administrative act of editing workspace information (Component 14).

Component 3 : Employee is able to create an availability (lend workspace)

Testing Technique

Manual black box

Automated white box

Test Script

Happy Path Script

Step	Action	Expected Outcome
1	Access the 'Lend Office'	Show the Lend Office page with staff's information and location shown
2	Pick a date range from the calendar from today onwards	Calendar should have picked date range highlighted
3	Enter some text in the comments	Comments field should respond to user input
4	Click 'Confirm Availability'	Should be redirected to a confirmation page

Negative Path Script

Step	Action	Expected Outcome
1	Access the 'Lend Office'	Show the Lend Office page with staff's information and location shown
2	Pick a date range from the past	Past date ranges should be disabled and not be highlighted

Tools

Web browser

Risks

No risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Automated white box	Manual black box	Manual black box

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: 1

Component 5: Employee is able to filter their bookings search by location, features, and date

Testing Technique

Black-Box Testing

Test Script

Manual test

Happy path script

Step	Action	Expected Outcome
1	On the 'Book Office' page, select a date range	The options of availabilities should update reflecting the specified date range (all availabilities are within the date range)

2	Select a location	The options of availabilities should update reflecting the location chosen
3	Select a floor	The options of availabilities should update reflecting the floor chosen
4	Select some features from the feature checkbox list	The options of availabilities should update reflecting the features chosen

Negative path script

Step	Action	Expected Outcome
1	On the 'Book Office' page, select a date range in the past	Past date ranges should be disabled and the calendar shouldn't respond to clicks on past date ranges

Tools

Web browser

Risks

No risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Automated white box	Black-box testing	Black-box testing

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: 1

Negative Path script

Due to the design of the front-end ,wrong format of booking ID or non -existing booking ID will not be possible

Tools

Browser

Risks

None

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
White-box testing	Black-box testing	Black-box testing

Number of Anticipated Scripts

- Number testing happy-path: 1
- Users can cancel their bookings successfully
- Number testing negative-path: 0

Component 14: Admin can edit workspaces

Testing Technique

White-box testing on backend, manual test on frontend

Test Script

Assuming navigation to admin page and admin authentication is completed

Happy Path script

Step	Action	Expected result
1	Select a location button	Enters the location page
2	Select a floor in table	Floor map shows up with correct map
3	Select edit button on the floor row	Workspace table associated to floor appears, Add Floor buttons become Add Workspace button
4	Select edit button on workspace row	Pop up screen shows up to enter edited workspace info
5	Enter a new workspace name	Editing succeeds, workspace row is updated with new workspace name
6	Select edit button on a workspace row again, enter a new valid ownerId	Editing succeeds, workspace row is updated with the new owner information associated to the workspace

Negative Path script

Step	Action	Expected result
1	Select a location	Enters the location page
2	Select a floor in table	Floor map shows up with correct map
3	Select edit button on the floor row	Workspace table associated to floor appears, Add Floor buttons become Add Workspace button
4	Select edit button on workspace row	Pop up screen shows up to enter edited workspace info
5	Enter another existing workspace name	Error message pops up - no duplicated workspace
6	Enters a new workspace name, with a non-existing employeeID	Error message pops up - office owner's ID does not exist
7	Enters a new workspace name, with non-numeric employeeID	Error message pops up - office owner's ID invalid
8	Enters a new workspace name with 6 or more characters, with a valid employeeID	Error message pops up - workspace name cannot be more than 5 characters

Tools

mocha/chai, web browser

Risks

None

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Manually written white-box testing	Manual testing in different OS/browser environment	Manual testing on different browsers

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: 1

Non-Functional Testing

The non-functional requirements are (1) concurrent user capacity, (2) peak active bookings capacity, and (3) a DBA script to offload non-active bookings into an archive table.

(1) Concurrent usage of up to 20 users at peak times

To test concurrent usage, we will provide a JMeter script that simulates concurrent users accessing the REST API. This is an automated black box testing using JMeter.

(See Appendix for Script)

(2) Handle approximately 1000 peak live active bookings in the system with good performance

This non-functional requirement is not tested individually. Instead, we will have a script, as mentioned in the Test Database section, that has 1000 peak live active bookings, which creates a MySQL database, and this is used for every test mentioned in the functional settings.

(3) A DBA script to offload non-active bookings into an 'archive' table

Black box testing done manually.

This non-functional component will be tested by manually running DBA script on Test Database.

(See Appendix for Script)

(4) Internet Explorer 11, Edge, and Firefox compatibility

This non-functional requirement is not tested individually. Instead, we are testing this as an integral part of our test environment (browser specification). (No Associated Script.)

Security Testing and Test Data Approach

We will only use fake data, so data masking is not applicable.

Authentication

We use MS Active Directory for normal user login, therefore in Security testing, we want to verify all webpages and backend endpoints are only usable after authenticated with MS Active Directory. Unauthorized attempts to web pages should be redirect to Active Directory login. The backend server should also reject unauthorized requests.

Testing approach

We will use mocha to automate our backend test suite. Three different tests will be performed on this section. First, endpoints should be usable for authenticated users. Testing script for this part should login to AD with a fake account to get an ID token. With this token stored in local storage, the testing script should be able to make requests to backend endpoints and get valid response. Second, backend endpoints should reject attempts with expired tokens. The ID token should expire in an hour. To test this case, the testing script should make a request to the backend with an expired ID token. The backend endpoint should be able to reject this request. Third, backend endpoints should reject unauthorized requests. The testing script will attempt to make requests to endpoints without ID token. The backend endpoint should reject this request.

Since we do not have scripts for AD login, we will test frontend user authentication manually. When user type in our URL in browser the first time, user should be redirected to AD login. After login, user should be able to see our homepage. If login failed, user can choose to try again but will not be able to enter homepage. The second scenario is when user type in sub directory URL in browser (e.g. /availabilities), our website should redirect user to login page.

Administrative Functions

Other than Active Directory, Administrator is required a simple password to enter admin page. Therefore, the administrative endpoint and page require both MS Active Directory login and admin password login. The Active Directory authentication part is covered in the previous section. The testing approach below in this section only covers the admin login part.

Testing approach

Similar to previous section, we will use mocha for automatic testing for backend, manual testing for frontend. For backend endpoints, we will first test if all admin endpoints respond normally after admin login. Same idea with MS Active Directory if the admin is unauthorized or expired the backend endpoints will reject the request. If the admin is authorized, the admin endpoints should respond correctly.

In frontend, when admin enters admin page, admin should be provided a text field for password, if admin enters correct password, he/she will be allowed to enter admin control page, otherwise there will be message prompt to admin for wrong password. If user enters a sub URL for admin in browser, the user will be redirect to admin login page.

Password Resets

Since we are using Active Directory to manage user login, user password reset should be handled by Active Directory. For admin password reset, after the admin enters admin control page, admin can click password reset to reset the password. When resetting password, admin have to provide old password to verify identity.

Test Data

As mentioned in the Test Environment section, we will provide automated script to insert fake data to our database. These data are generated randomly just for the testing purpose. We will also provide fake account for Active Directory login.

Regression Testing Needs

Since this is the first version of our application and no further versions are planned, regression test for previous version is not applicable.

Appendix - All Functional Components and their Test Scripts

Component 1: Log into Active Directory

Testing Technique

Black box

Test Script

Happy Path SAT Script

Step	Action	Expected Outcome
1	Ensure user is not authenticated into AD	N/A
2	Access the Flex Work website	Redirected to Active Directory login
3	Login to AD (using 'fakeemail@gmail.com' and password 'fakepassword')	Redirect to Flex Work homepage

Negative Path SAT Script

Step	Action	Expected Outcome
1	Ensure user is not authenticated into AD	N/A
2	Access the Flex Work website	Redirected to Active Directory login
3	Login to AD (using 'fakeemail@gmail.com' and password 'wrongpassword')	Login to fail and not be redirected to Flex Work homepage

UAT Script

Step	Action	Expected Outcome
------	--------	------------------

1	Be logged into ICBC system	N/A
2	Access the Flex Work website	Should go directly to Flex Work website without any login required
3	Login to AD (using 'fakeemail@gmail.com' and password 'fakepassword')	Redirect to Flex Work homepage

Tools

Web browser

Risks

No risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
N/A	Manual black box	Manual black box

Number of Anticipated Scripts

- Number testing happy-path: 2
- Number testing negative-path: 1

Component 2: Functional home page that links to all other pages

Testing Technique

Manual black box

Test Script

Happy path script

Step	Action	Expected Outcome
1	Access the Flex Work homepage	Should see a page with 3 buttons and a link to admin page
2	Click on 'Lend Office' button	Redirected to Lend Office page

3	Go back to homepage and click on 'Book Office' button	Redirected to Office Booking page
4	Go back to homepage and click on Manage Bookings/Lendings	Redirected to manage bookings and lendings page
5	Go back and click on the 'Admin' at the bottom	Redirected to Admin page

Tools

Web browser

Risks

No risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
N/A	Manual black box	Manual black box

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: no possible negative paths

Component 3 : Employee is able to create an availability

Testing Technique

Manual black box

Automated white box

Test Script

Happy Path Script

Step	Action	Expected Outcome
1	Access the 'Lend Office'	Show the Lend Office page with staff's information and location shown

2	Pick a date range from the calendar from today onwards	Calendar should have picked date range highlighted
3	Enter some text in the comments	Comments field should respond to user input
4	Click 'Confirm Availability'	Should be redirected to a confirmation page

Negative Path Script

Step	Action	Expected Outcome
1	Access the 'Lend Office'	Show the Lend Office page with staff's information and location shown
2	Pick a date range from the past	Past date ranges should be disabled and not be highlighted

Tools

Web browser

Risks

No risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Automated white box	Manual black box	Manual black box

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: 1

Component 4: Employee is able to see default options for booking

Testing Technique

Manual black box

Test Script

Manual script

Step	Action	Expected Outcome
1	Access the 'Book Office' page'	Redirected to the Book Office page with some default availability packages being shown

Tools

Web browser

Risks

No risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Automated white box	Manual black box	Manual black box

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: no possible negative paths

Component 5: Employee is able to filter their bookings search by location, features, and date

Testing Technique

Black-Box Testing

Test Script

Manual test

Happy path script

Step	Action	Expected Outcome
1	On the 'Book Office' page, select a date range	The options of availabilities should update reflecting the specified date range (all availabilities are within the date range)

2	Select a location	The options of availabilities should update reflecting the location chosen
3	Select a floor	The options of availabilities should update reflecting the floor chosen
4	Select some features from the feature checkbox list	The options of availabilities should update reflecting the features chosen

Negative path script

Step	Action	Expected Outcome
1	On the 'Book Office' page, select a date range in the past	Past date ranges should be disabled and the calendar shouldn't respond to clicks on past date ranges

Tools

Web browser

Risks

No risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Automated white box	Black-box testing	Black-box testing

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: 1

Component 6: Confirming a booking option (20 mins to confirm)

Testing Technique

Black-Box Testing

Test Script

Happy Path script

Step	Action	Expected result
1	Select “Book Office” button	Enters the Booking page
2	Select a booking	Booking row selected is highlighted
3	Select “Confirm Booking” button	Enters confirmation page, booking is now locked from other’s access for 20 mins
4	Have another user enter Booking page	The locked booking should not be on the list of availabilities to this user

Negative Path script

All information for booking confirmation is requested from AD. Therefore, there is no input error possible. There are also no other exception/errors expected from this component

Tools

React-render for the frond-end

Chai for back-end

Risks

No risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Black-Box Testing	Black-box testing	Black-box testing

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: 0

Component 7: Showing the floor plan associated with availability on click

Testing Technique

Manual Testing, Black-box testing

Test Script

Happy Path script

Step	Action	Expected result
1	Select "Book Office" button	Enters the Booking page, which shows default bookings.
2	Select the first availability in the default bookings. It has an associated floor plan.	Floor map shows up with correct map
3	Click somewhere away from the floor map	Floor map disappears
4	In the calendar, select any two dates within a month of the current date	The list of bookings updates with different bookings
5	Select each availability in the displayed table	Floor map shows up with correct map
6	Click somewhere away from the floor map	Floor map disappears

Negative Path script

If there is no floor plan image for the associated floor of the booking, then it is handled gracefully.

Step	Action	Expected result
1	Select "Book Office" button	Enters the Booking page, which shows default bookings. The second booking has no associated floor plan. However, there is no clickable floor plan icon for it.
2	Mouse over the area where there is a clickable floor plan icon in other bookings, but for the second booking	The mouse icon turns into the forbidden action icon, notifying the user that there is no floor plan image.
3	Click on it	Nothing should happen

Tools

React-render for the frond-end

Chai for back-end

Risks

No risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Black-box testing	Black-box testing	Black-box testing

Number of Anticipated Scripts

- Number testing happy-path: 1
 - Test to check if floor plan associated with availability shows up
- Number testing negative-path: 1
 - Test to check if there is no floor plan associated with availability, then it is handled correctly

Component 8: User can view their bookings

Testing Technique

Manual Testing, Black-Box Testing

Manual Test Script

Happy Path script

Step	Action	Expected result
1	Click "Book Office" button	The Book Office page appears, showing default booking options
2	Select the first option, and click Next	The confirmation screen appears, with 20 minute countdown timer
3	Select the Confirm button	A second confirmation screen confirms the booking along with the Home button
4	Click the Home button	The Home page is shown

5	Select "Manage Bookings" button	Enters the Manage Bookings page, which shows their current bookings and availabilities. The booking made in Step 1-3 is expected to appear here, with correct information (workspace, start date, end date)
---	---------------------------------	---

Negative Path script

Not applicable, as there is no user decision that can lead to a negative path for this component.

Tools

Manual testing

Chai for back-end

Risks

No risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Black-Box Testing	Black-box testing	Black-box testing

Number of Anticipated Scripts

- Number testing happy-path: 1
 - Test to check if user can view their own bookings

Component 9: User can cancel availability

Testing Technique

Manual Black-Box Testing for the front-end

Automated white-box test for the black-end

Test Script

Happy Path script

step	Action	Expected result
1	Click the manage booking button from home page	Navigate to manage booking page and the table of bookings and list of availabilities should be shown

2	Click the delete icon of one of the availabilities row shown	The availability chosen is deleted. Page is updated and that availability should be removed
---	--	--

Negative Path script

Due to the design of the front-end, wrong format of booking ID or non-existing booking ID will not be possible. Therefore, there is no negative path

Tools

Browser

Risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
White-box Testing	Black-Box Testing	Black-Box Testing

Number of Anticipated Scripts

- Number testing happy-path:1
- Availability canceled successfully by the user
- Number testing negative-path:2
- Incorrect format of availability can be handled properly
- Non- existing availability can not be canceled

Component 10: User can cancel booking

Testing Technique

Manual Black-Box Testing for the front-end

Automated white-box test for the black-end

Test Script

Happy Path script

Step	Action	Result
1	Click the manage booking button	Navigate to manage booking page and

	from Home Page	the table of bookings and list of availabilities should be shown
2	Navigate to manage booking and click the Icon to delete one of the existing booking	Canceled successfully, Page refresh and that booking is removed from the page

Negative Path script

Due to the design of the front-end ,wrong format of booking ID or non -existing booking ID will not be possible

Tools

Browser

Risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
White-box testing	Black-box testing	Black-box testing

Number of Anticipated Scripts

- Number testing happy-path: 1
- Users can cancel their bookings successfully
- Number testing negative-path: 0

Component 11: Admin login/authentication

Testing Technique

Manual Black-box testing

Test Script

Manual test

step	Action	Expected result
------	--------	-----------------

	Happy Path	
1	Click the admin button located at the bottom of the home page	A field for entering the password should be shown
2.	1.Try to log in with correct password	Log in successfully,show the main page of Admin
	Negative Path	
3	Redo Step1 Try to log in nothing entered for the password field	Fail to submit
4	Redo Step1 Try to Log in with wrong password(eg.11111)	Fail to Login Show wrong password error message

Tools

Web browser

Risks

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Black-box testing	Black-box Testing	Black-box Testing

Number of Anticipated Scripts

- Number testing happy-path:1
- Admin can log in successfully
- Number testing negative-path:2
- Admin fail to log in due to wrong password
- Exception case of entering with no input can be handled properly

Component 13: Admin can add workspaces

Testing Technique

White-box unit tests on backend, manual test on frontend

Test Script

Assuming navigation to admin page and admin authentication is completed

Happy Path script

Step	Action	Expected result
1	Select a location	Enters the location page
2	Select a floor in table	Floor map shows up with correct map
3	Select edit button on the floor row	Workspace table associated to floor appears, Add Floor buttons become Add Workspace button
4	Select Add Workspace	Pop up screen shows up to enter new workspace info
5	Enter a new workspace name, with an existing employeeID	A new workspace is created, table of workspaces updated with the newly created workspace row

Negative Path script

Step	Action	Expected result
1	Select a location	Enters the location page
2	Select a floor in table	Floor map shows up with correct map
3	Select edit button on the floor row	Workspace table associated to floor appears, Add Floor buttons become Add Workspace button
4	Select Add Workspace	Pop up screen shows up to enter new workspace info
5	Enter an existing workspace name, with a valid employeeID	Error message pops up - no duplicated workspace
6	Enter a new workspace name, with a non-existing employeeID	Error message pops up - office owner's ID does not exist
7	Enter a new workspace name, with non-numeric employeeID	Error message pops up - office owner's ID invalid
8	Enter a new workspace name with 6 or more	Error message pops up - workspace name cannot be more than 5 characters

	characters, with a valid employeeID	
--	-------------------------------------	--

Tools

mocha/chai, web browser

Risks

None

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Manually written white-box testing	Manual testing in different OS/browser environment	Manual testing on different browsers

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: 1

Component 14: Admin can edit workspaces

Testing Technique

White-box testing on backend, manual test on frontend

Test Script

Assuming navigation to admin page and admin authentication is completed

Happy Path script

Step	Action	Expected result
1	Select a location button	Enters the location page
2	Select a floor in table	Floor map shows up with correct map
3	Select edit button on the floor row	Workspace table associated to floor appears, Add Floor buttons become Add Workspace button
4	Select edit button on	Pop up screen shows up to enter edited workspace info

	workspace row	
5	Enter a new workspace name	Editing succeeds, workspace row is updated with new workspace name
6	Select edit button on a workspace row again, enter a new valid ownerID	Editing succeeds, workspace row is updated with the new owner information associated to the workspace

Negative Path script

Step	Action	Expected result
1	Select a location	Enters the location page
2	Select a floor in table	Floor map shows up with correct map
3	Select edit button on the floor row	Workspace table associated to floor appears, Add Floor buttons become Add Workspace button
4	Select edit button on workspace row	Pop up screen shows up to enter edited workspace info
5	Enter another existing workspace name	Error message pops up - no duplicated workspace
6	Enters a new workspace name, with a non-existing employeeID	Error message pops up - office owner's ID does not exist
7	Enters a new workspace name, with non-numeric employeeID	Error message pops up - office owner's ID invalid
8	Enters a new workspace name with 6 or more characters, with a valid employeeID	Error message pops up - workspace name cannot be more than 5 characters

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: 1

Component 15: Admin can upload floor plan

Testing Technique

Black-box testing on backend, manual test on frontend

Test Script

Assuming navigation to admin page and admin authentication is completed

Happy Path script

Step	Action	Expected result
1	Select a location button	Enters the location page
2	Select a floor in table	Floor map shows up with correct map
3	Select 'Upload New Floor Plan' button under floor map	Pop up screen shows up to choose a new document from computer
4	Select a valid floor plan jpg file	File accepted, current floor map is updated with chosen image file
5	Unselect the floor in table	Floor map section went back to blank canvas
6	Reselect the floor in table	Correct floor map that was uploaded shows up again

Negative Path script

Step	Action	Expected result
1	Select a location	Enters the location page
2	Select a floor in table	Floor map shows up with correct map
3	Select 'Upload New Floor Plan' button under floor map	Pop up screen shows up to choose a new document from computer
4	Select a non-jpg file	Error message pops up - file has to be jpg format

Tools

mocha/chai, web browser

Risks

None

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Manually written black-box testing	Manual testing in different OS/browser environment	Manual testing on different browsers

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: 1

Component 16: Admin can change global workspace feature labels

Testing Technique

Black-box testing on backend, manual test on frontend

Test Script

Assuming navigation to admin page and admin authentication is completed

Happy Path script

Step	Action	Expected result
1	Select the "Features" tab	Enters the edit feature list page
2	Select edit button on a feature	Pop up screen appears asking for feature name edit
3	Edit the feature to new feature name	Edit successful, feature list updated with the feature name changed
4	Select delete button on a feature	Delete successful, feature disappear from list
5	Select "Add feature" button	Pop up screen appears asking for new feature name
6	Enter a new string for feature name	Feature added successfully, feature list updated with new feature

Negative Path script

Step	Action	Expected result
1	Select the “Features” tab	Enters the edit feature list page
2	Select “Add feature” button	Pop up screen appears asking for new feature name
3	Enter existing feature name	Error message pops up - cannot add duplicate feature
4	Enter non-string input for new feature name (“ ” or “123”)	Error message pops up - new feature name not valid

Tools

mocha/chai, web browser

Risks

None

Differences in Approach for Unit, SAT, UAT Testing

Unit	SAT	UAT
Manually written black-box testing	Manual testing in different OS/browser environment	Manual testing on different browsers

Number of Anticipated Scripts

- Number testing happy-path: 1
- Number testing negative-path: 1

Test Scripts for Non-Functional Tests

(1) Concurrent usage of up to 20 users at peak times

As stated in the main document, this will be tested using JMeter.

It is an automated test that simulates HTTP requests to the back end from 20 users at a time. It will test a variety of end points and validate the results.

The JMeter file that contains the test information will be provided in the /test/jmeter folder.

Steps:

Input: Our custom JMeter file.

The JMeter GUI (graphical user interface) will be run and our custom JMeter file will be used to run a set of automated tests simulating 20 concurrent users.

Once the test is finished, the results will be summarized.

Output:

The results will be shown in the GUI itself. For each of the automated tests, the result will be individually displayed, as for whether the tests passed or failed.

(3) A DBA script to offload non-active bookings into an 'archive' table

Once a month, the back end server will run a script on the MySQL database in order to archive non-active bookings. This will be tested by execution of this script on a Test Database script that will populate the database with active bookings that are more than 1 month old, and a second script (which we provide) will be run to see if these bookings are archived.

Input:

The custom Test Database script that generates test bookings.

Output:

The result of the archive check script.

Steps:

In the /test folder, run command "node initialize_test_database_archive.js" in a terminal.

It will automatically run a script to initialize the Test Database to have bookings more than 1 month old.

At this point, the tester may optionally inspect the database with a tool such as MySQL Workbench.

Next, run command "node dba_script.js" in a terminal. This is the act of running the script manually, in order to test it.

At this point, the tester may manually inspect the database with a tool such as MySQL Workbench.

Finally, run command “node archive_test.js” in a terminal. This will check for the presence of all the archived bookings in the archive tables, along with their absence from the normal active bookings tables.

The result will be shown in the same terminal, at the end of this test.

(2) and (4) have no associated scripts since they are not individually tested.