

Mobile Museum App Testing Plan

Invalid Size of Size 8

Introduction

In this part of the document, the testing plan of this project will be presented. There are two major parts of the testing plan: the general methodologies planned to be used for the project testing as well as the specific arrangement and implementation of how the testing will be conducted.

Methodology

There are two stages of project testing, the first stage is Development Testing and the second stage is User Testing.

In the first stage, to make the system testing more efficient and thorough, as well as increase the coverage of the testing, the entire system will be divided into different major components according to the project requirement document and different user cases. Each component will then be divided into different units according to the user story document and functions.

In each of the sprint, different functional blocks (components) and their units will be implemented and tested in thorough. Before conducting any testing, the code developed by a single developer or a pair of developers will be inspected by other developer(s) working on different aspects of the system. The reason is that peer inspection could improve the coverage of logics and use cases, as well as increase the efficiency of testing as bugs which are difficult to be spotted during the testing stage may be much easier to be located by other developers in the early stage.

After that, Unit Test will be performed for each class and all its methods implemented, for each method in a class, JUnit will be used as an automated testing tools as it is able to conduct testing in tester-specific ways efficiently and automatically. Both Validation testing and Defect testing will be performed for each unit as in this stage any defect of the unit is expected to be exposed. The methodology used for this stage of testing is basically White Box testing, as different paths of execution are expected to be covered and test cases are designed specifically based on possible behaviors of each method.

When all the units successfully passed the test cases, they will be integrated as a component and the testing of a functional block will be conducted. In the Component Testing, the testing will focus on the behaviors of each component's interface and the test cases will be design according to the requirement document and different possible user stories. Assuming that each unit of a component is well-tested, the testing in this stage is basically Black Box testing. The test cases will be designed for both validation testing and defect testing. Instead of using

any automated testing tool, the testing will be performed on a “play-around” basis, where the testers will carry out all possible usages of the component’s interface, try to get valid results for valid operations and expose defect for invalid operations. Also, at the end of each sprint, the components developed will be integrated and tested.

In the end of the last sprint, all the components developed will be integrated and the whole system will be tested. In System Testing, the main focus will be the interactions and compatibilities of different components (both to validate the functions and to expose any defect): if they can successfully perform a complete task and if there any unexpected output for invalid operations.

After the Development Testing discussed above, the next stage is User Testing. Note that the Release Testing has been skipped as it could be difficult to find another testing team to conduct isolated all-level testing. Instead, after the development testing is properly performed, the mobile museum app will be considered released. In this case, peers and friends (who are not related to the project) of the developers would be asked to use the app as “testers”. It is like a Beta Test where the users will be expected to use the app in any way they want and report any unexpected behavior or error encountered. As the last stage of testing, the goal of this testing is to fix and improve any defect of the system before the final delivery.

In addition, apart from the testing focusing on the functionalities, the non-functional aspects of the system will also be tested. Two main focuses of the non-functional testing will be the responding speed as well as security. The responding speed testing will be included in the testing of each component as well as the system testing, and the testing of security will be performed mainly during the testing of user account related functions components such as registration, login, favorite and ticketing.

Testing Plan

According to the project requirements, the mobile museum app could be divided into five major different parts: **detailed items information presentation, user account (registration, login, logout and account information, including ‘favorite’ the item), main page (including items recommendation), search for items, ticket purchasing.**

In the following contents, the how the tests of these parts will be performed, possible test cases, expected results, member(s) in charge as well as planned due date will be presented in format of tables.

Detailed Item Information Presentation

This functional block includes any activity and operation related to item information presentation.

Test Description	Category	Possible Test Cases	Expected Results	Owner	Due Date
Present returned data from a function to the UI	Unit Testing	Set particular output from a function and check the presented value displayed on the UI view.	The displayed data is the returned value from the function.	Qian Fang, Yan Su	03/15/2018
Click an image button and change the UI to another view with a particular data send along, display the ID on the new view.	Unit Testing	Set different image buttons, each with a unique ID, click on each of them and check the displayed ID.	The ID will be properly displayed and will be the same as the image clicked.	Qian Fang, Yan Su	03/15/2018
Sending data via HTTP request to the server and get the response	Unit Testing	Send some particular data (input parameters) to the server like "hello" and receive the response (return values) returns from the server;	Responded data is "hello" on success and the front-end will return "hello" or the server will respond nothing and the front-end will return "failed".	You Lyu	02/20/2018
Sending item name via HTTP request to the server and get particular response	Unit Testing	Send the name of an item to the server and receive the response, i.e. the detailed information (return values) of the item, from the server.	Responded data is the item's information if the combination is valid, if the combination is invalid, the response will be text: "item does not exist".	You Lyu, Zhongyu Li	03/19/2018
Present item's detailed information	Component Testing	Clicking the image of any item on the main view, check the displayed information on the new view.	The view will change to a new one and the detailed information will be displayed.	Qian Fang, Yan Su	03/22/2018

User Account

This functional part includes any activity related the user account management (i.e. registration, login, logout and account information).

Test Description	Category	Possible Test Cases	Expected Results	Owner	Due Date
After submitting, get user's input to the specific element of UI and display input	Unit Testing	Input particular texts to the input box (from null to very long) and submit by clicking a button.	None if nothing is input, and the exact content of input is input is valid (if it is too long, it may go out of the display box's scope, and in this case, it can be ignored as the actually displayed text data will have limited length).	Qian Fang, Yan Su	03/15/2018
Sending data via HTTP request to the server and get the response	Unit Testing	Send some particular data (input parameters) to the server like "hello" and receive the response (return values) returns from the server;	Responded data is "hello" on success and the front-end will return "hello" or the server will respond nothing and the front-end will return "failed".	You Lyu	02/20/2018
Sending user information via HTTP request to the server and the server stores data into the database	Unit Testing	Send a combination of username and password (both valid username and password or invalid username or password) to the server and Manually access to the database, check if it can be successfully stored in the database;	If the both username and password have legal formats, validation of username and password on the server side will pass and the information will be stored in the database, if any of them has illegal format, or the username has been registered before, the data will not be stored.	You Lyu	03/24/2018
Sending user information via HTTP request to the server and get particular response	Unit Testing	Send a combination (input parameters) of username and password to the server and receive the response from the server which is the information (return values) of the user.	Responded data is the user's information if the combination is valid, if the combination is invalid, the response will be text: "wrong username or password".	Zhongyu Li	03/26/2018
User's account information is not accessible if user is not logged in (i.e. update user's status)	Unit Testing	1. Login first, click on the user account tab on the main page, try to access user's information. 2. Click on the user account tab on the main page without logging in.	1. Able to access user's information; 2. Unable to access user's information;	You Lyu, Zhongyu Li	03/27/2018
Add information to the user's favorite item row in the database	Unit Testing	Login first, set a button, by clicking the button the front-end will send an HTTP request with particular data to the server and asking the server to	The particular data will be added to the particular row.	Rongsheng Zhang	03/28/2018

		store the data to the a particular row of a data table.			
User Registration and logging in	Component Testing	Register an account and then use it to login in.	The user account section will be accessible, and user's information will be presented on the page.	You Lyu, Zhongyu Li	03/28/2018
User logging out	Component Testing	Login to a valid account and log out, check the difference on status.	The user account section will not be accessible and when clicked on the corresponding tab, a prompting window will be popped out to alert that the user should login first and the view will jump to the login view.	You Lyu	03/29/2018
'Favorite' is not operable if user is not logged in (i.e. update user's status)	Unit Testing	1. Login first, click on the 'favorite' button, try to add favorite item to the user's information. 2. Click on the 'favorite' button, try to add favorite item to the user's information without logging in.	1. Able to view user's favorite items in the user account page; 2. Unable to add item, and a prompt window will pop up and alerting the user to login first;	Zhongyu Li, Rongsheng Zhang	03/30/2018
'Favorite' items	Component Testing	Clicking the 'favorite' button to add the item to the favorite items list on the user account information.	If logged in, the item will be added and displayed on the user account information; else, 'favorite' is not operable.	Rongsheng Zhang	04/01/2018

Main Page

This functional part includes any activity related the main page management (i.e. recommendation, layout of each interface).

Test Description	Category	Possible Test Cases	Expected Results	Owner	Due Date
Checking if each button is connected to each corresponding page correctly	Unit Testing	Click on each button on the main page to see if each button will lead the user to the correct page.	The page presented will be replaced with the corresponding page connected with the button.	Qian Fang, Yan Su	04/02/2018
Sending data via HTTP request to the server and get the response	Unit Testing	Send some particular data (input parameters) to the server like "hello" and receive the response (return values) returns from the server;	Responded data is "hello" on success and the front-end will return "hello" or the server will respond nothing and the front-end will return "failed".	You Lyu, Zhongyu Li	04/03/2018
Checking if each page could return to the main page correctly.	Unit Testing	Click on the 'return' button on each component page to see if it will lead the user back to the main page.	The page presented will be replaced with the main page after clicking the return button.	Rongsheng Zhang, Qian Fang	04/03/2018
Checking if the rolling part of the main page works correctly	Unit Testing	Roll up and down on the rolling part of the main page to check if it will perform well.	The rolling part will present different contents when user rolls up and down.	Zhongyu Li, Yan Su	04/04/2018
Checking if the recommendation part will present daily items correctly.	Component Testing	Check if the 'item recommendation' will perform correctly. Click on the recommended item to see if it will jump to the correct item presentation page. Change the system time to different dates to see if the recommended item will change.	The item recommendation part should present the daily recommended item. When clicked, the main page should jump to the corresponding page presenting the item. The recommended item should be different on different dates.	You Lyu, Ronghseng Zhang	04/04/2018

Search for Items

This functional part includes any activity related the searching management.

Test Description	Category	Possible Test Cases	Expected Results	Owner	Due Date
Sending data via HTTP request to the server and get the response	Unit Testing	Send some particular data (input parameters) to the server like "hello" and receive the response (return values) returns from the server;	Responded data is "hello" on success and the front-end will return "hello" or the server will respond nothing and the front-end will return "failed".	You Lyu, Rongsheng Zhang	04/05/2018
Clicking on the 'search' button to see if it will change to the item presentation page	Unit Testing	Type in some specific text in the search bar and click the search button.	If the text matches some specific items in the database, the page should change to a list showing the searching result. If there are no matching results, the page should print 'Item not found'.	Qian Fang, Yan Su	04/05/2018
Checking if the searching result is relevant to the input text.	Unit Testing	Type in some general text related to artworks like paint, vase and see if the searching result is related.	After clicking on the searching button, the page should change to a list containing all the relevant items in the database	Zhongyu Li, You Lyu	04/05/2018
Present returned data from a function to the UI	Unit Testing	Set particular output from a function and check the presented value displayed on the UI view.	The displayed data is the returned value from the function.	Rongsheng Zhang, You Lyu	04/05/2018
Checking if the searching page could return to the main page correctly.	Unit Testing	Click on the 'return' button on searching page to see if it will lead the user back to the main page.	The page presented will be replaced with the main page after clicking the return button.	Zhongyu Li, Rongsheng Zhang	04/05/2018
Checking if the rolling part of the searching page works correctly	Unit Testing	Roll up and down on the rolling part of the searching page to check if it will perform well.	The page will roll up and down for the users to scan all the relevant items listed.	You Lyu, Yan Su	04/06/2018
Searching for items	Component Testing	1. Type in some text that are not exactly the name of the item. 2. Type in some valid text that are exactly the name of the item.	1. The searching page should list some of the related searching result and sort the result in the order of relevance. 2. The searching page should show the resulting item's name.	Qian Fang, Zhongyu Li	04/06/2018

Ticket Purchasing

This functional part includes any activity related the ticket purchasing management.

Test Description	Category	Possible Test Cases	Expected Results	Owner	Due Date
Sending data via HTTP request to the server and get the response	Unit Testing	Send some particular data (input parameters) to the server like “hello” and receive the response (return values) returns from the server;	Responded data is “hello” on success and the front-end will return “hello” or the server will respond nothing and the front-end will return “failed”.	You Lyu, Rongsheng Zhang	04/06/2018
Clicking on the ‘purchase’ button to see if it will change to the ticket purchasing page	Unit Testing	Click on the purchase button on the main page or in the user account page.	The page should be changed to the ticket purchase page. If the user has not logged in yet, the system should remind the user to login first.	Qian Fang, Yan Su	04/07/2018
Checking if the system can get the correct user bank account information.	Unit Testing	Send some specific users’ account the server to acquire the corresponding bank account information.	If the user has already registered his bank account in the database, the server should send back the information in the database. If not, the server should send a message to tell the front-end that there is no matching information.	Zhongyu Li, You Lyu	04/08/2018
Present returned data from a function to the UI	Unit Testing	Set particular outputs from a function and check the presented value displayed on the UI view.	The displayed data is the returned value from the function.	Qian Fang, Yan Su	03/15/2018
Checking if the system would present the remaining ticket condition correctly.	Unit Testing	Set the remaining tickets in the database to a small number and click on purchase button many times in the app.	The purchasing process should be fine for the first several times and when the remaining ticket is zero, the app should not let user to buy tickets.	You Lyu, Rongsheng Zhang	04/09/2018
Checking if the rolling part of the ticket purchasing page works correctly	Unit Testing	Roll up and down on the rolling part of the ticket purchasing page to check if it will perform well.	The page will roll up and down for the users to scan all the available tickets and corresponding exhibitions.	Zhongyu Li, Rongsheng Zhang	04/09/2018
Ticket purchasing	Component Testing	Log in as a user and purchase the tickets of some exhibitions.	The user’s information should be sent to the server and be stored in the database. The user’s account page should include information of purchased ticket.	You Lyu, Yan Su	04/09/2018

System Testing and User Testing

After the unit and component testing for each component is finished, the system testing will be performed. The system testing will be conducted based on the requirement and user stories. After the Development Testing is done, there will also be User Testing as the last testing stage. The users are expected to use the app in any way they want in any scenario, hence the testing cases in System Testing may also be covered in User Testing.

Test Description	Category	Possible Test Cases	Expected Results	Owner	Due Date
User register, login and 'favorite' items	System Testing and User Testing	<ol style="list-style-type: none"> 1. Register a new account, login in and 'favorite' certain items, check the user account information and the favorite items list. 2. Do not login, try to 'favorite' item and check favorite items list. 	<ol style="list-style-type: none"> 1. Registration and logging in are successful, 'favorited' items are added to the list and displayed in the user account information menu. 2. Unable to 'favorite' items or to check account information. 	Qian Fang, Yan Su	04/10/2018
View recommended items, search for specific items and view their detailed information	System Testing and User Testing	<ol style="list-style-type: none"> 1. Open app for several times, check if the recommended items are the same. 2. Search for certain items using key partial word and exact key word and check the result. 3. Click the name of the item and check the detailed information. 	<ol style="list-style-type: none"> 1. The recommended items will be updated each time. 2. If using partial key word, items containing the keyword will be displayed, if using exact keyword, the exact item will be displayed as a result. 3. After clicking, the app will jump to another page which displays the detailed information of the item. 	Qian Fang, Yan Su, You Lyu	04/10/2018
User login, purchase tickets and view ticket information	System Testing and User Testing	<ol style="list-style-type: none"> 1. User login, purchase a ticket and view ticket information. 2. Do not login, try to purchase a ticket and view ticket information. 	<ol style="list-style-type: none"> 1. Ticket purchasing will be successfully performed, and the ticket information will be displayed in the user account information page. 2. Ticket purchasing will not be successfully performed, and a prompt window will be displayed indicating that the user should login first. User is not able to view user account information. 	Yan Su, Rongsheng Zhang, Zhongyu Li	04/10/2018