< Journey Assistant >
Software Testing Plan
Version 1.0

Group Member

Yiwen Song
Zhihui Xie
Weizhe Wang
Huangfei Jiang
Haoping Chen

Modification History

Date	Version	Description	Author
2019-06-18	1.0	Finish the first version.	Yiwen Song

Contents

1 Intruduction

1.1 Purpose

The software testing plan aims to plan and formalize the testing procedure for our Journey Assistant software. This document will show the testing statics and environment, list the specific testing use cases, and make testing timetables, etc. It is limited to guide the development group and the testing group to do testing jobs.

1.2 Scope

Software the document is applied on: Journey Assistant.

Characteristics, subsystems, models and codes related to the software all fit the contents of this the document.

1.3 Definition

The terms referred to in this document are defined in the project glossary document (Glossary.pdf).

1.4 Bibliography

1. <Object Oriented Software Engineering (Version 3)> (Tsinghua University Press)
2. <Object Oriented Software Engineering Practice Guidelines>

1.5 Sketch

The document is made up of 6 parts, i.e., testing strategies, testing ranges & methods, testing use cases, testing environment, testing timetable, and risk management. Testing strategy arranges the event flow of our testing procedure. Testing range & method plans the specific testing requirements and methods. Testing use cases give the specific content of testing. Testing environment fixes the requirements of software and hardware during the testing procedure. Testing timetable arranges the involving personnel and time. Risk management points out the possible risks. Each part of the document links with one another, and complements each other to show an overview of out testing procedure.

2 Testing Strategy

2.1 Outlook

The table 1 below shows the outlook of our testing strategy.

2.1.1 Entering Criterion

The entering condition of testing, i.e., the entering criterion, is that: the system is submitted, the software is released, the testing use cases have been designed, the testing environment has been set up, and the testing personnel are ready.

Testing Necessity	Involving Personnel	Predicted Result
Functional Test & Compatibility Test	Xie Zhihui	Make sure that under different Android versions, all functions appointed by the users can be executed well.
Performance Test	Wang Weizhe	Test the response time of our program under different works.
Safety Test	Chen Haoping	Test if the access to the data are normal.
Interface Test	Jiang Huangfei	Test if the interface fits the standard or the usage of different access method.
Installation Test	Song Yiwen	Test if the software can be installed and uninstalled properly.
Total Test	Song Yiwen	Test if the software can be used in a general scope.

Table 1: Testing Strategy Outlook

2.2 Pause / Exit Criterion

The pause criterion of testing is that bugs are discovered or the software collapsed during testing.

The exit criterion of testing is that all system functions are tested. The number of problems that have severity level PI is 0. The testing use case are all done, and the testing document is submitted.

3 Testing Range and Method

3.1 Subsystem Objects of Testing

The subsystems that need to be tested are: Login subsystem, Recommendation subsystem, Customization subsystem.

The subsystems that don't need to be tested are: Feedback subsystem.

3.2 Testing Requirement

3.2.1 Fundamental Requirement

The testing requirements of the use cases are listed in the following table.

Use Case	Function	Index	Requirements	Method
Register	Add a new user	TR-01	1. User information should include username, password, e-mail. 2. Non-repeatable username and e-mail. 3. E-mail should be legally-formatted. Password should be 4-10 characters.	BB/MT.
Login	Enter the system if e-mail and password is correct.	TR-02	1. Check if the e-mail and password exists in the database. 2. All inputs should be non-empty.	BB/MT.
Recommend	Recommend an itinerary according to the user's likes.	TR-03	1. The user's like should be evaluated by an integer list ranging from 1 to 4. 2. Recommended itinerary should be correctly shown on map.	BB/AT.
Customize	User customize itinerary	TR-04	User can select the spots they would like to go sequentially.	BB/MT
View Itinerary	Show itinerary on map	TR-05	1. Click on the spot on map and a spot information page will pop out. 2. The itinerary should be displayed properly.	BB/MT

Table 2: Fundamental Requirement(Abbreviations for testing method: BB=Black Box, WB=White Box, MT=Manual Test, AT=Auto Test.)

3.2.2 Other Requirement

Requirements	Need to be tested?	Test demands	Test methods
Usability	Y	System has full function	BB
Reliability	Y	Safety test.	MT
Performance	Y	Performance test.	AT
Stability	Y	Compatibility test.	BB/MT

Table 3: Other Requirement

4 Testing Use Case

Testing use cases are shown in Table [4 - 6].

5 Testing Environment

5.1 Hardware Environment

Server A laptop with Intel Core I5-6550H, 8GB Memory and 1TB Hard Disk.

Client An Android smart phone with Qualcomm 835 Processor, 4GB RAM and 64GB; An Android smart phone with Kirin 960 Processor, 4GB RAM and 64GB ROM.

5.2 Software Environment

Server

OS Microsoft Windows 10.

Server Software Apache Tomcat.

Development Software NetBeans.

Development Language Java Web 8.0.

Client

OS Android 9.0.

5.3 Communication Environment

Server 100 Mbps of WAN connection, WIFI/WLAN or Cellular Network.

Client WLAN or Cellular Network.

5.4 Safety Environment

The server should have specific authority control in case of illegal access.

5.5 Specific Environment Requirement

None.

6 Testing Plan

6.1 Workload Estimation

Workload estimation is shown in Table 7.

6.2 Personnel Requirement

Personnel Requirement is shown in Table 8.

6.3 Progress Plan

Progress Plan is shown in Table 9.

6.4 Other Resource Requirement

None.

6.5 Delivered Document

- Software Testing Plan
- Software Testing summary

6.6 Risk Management

Risk management is shown in Table 10.

Required Item	Index	Use Case Index	Use Case	
UR-01-Register	TR-01-01	TR-01-01-01	Name	Detect if the e-mail address is legal
			Object	Registration Page
			Priority	Middle
			Input	abcd/(EMPTY)/abc@d
			Output	Not a legal address
			Step	If the address is illegal, pop a warning
			Method	MT/WB
		TR-01-01-02	Name	Detect if the username is longer than 6 characters
			Object	Registration Page
			Priority	Middle
			Input	abcd
			Output	Username should be more than 6 characters
			Step	If the input is illegal, pop a warning
			Method	MT
		TR-01-01-03	Name	Check if the username or email exists
			Object	Registration Page
			Priority	High
			Input	abcd@163.com
			Output	E-mail already exists
			Step	If the username or e-mail exists, pop a warning
			Method	MT/WB
		TR-01-02-01	Name	Detect if the password length is more than 6 characters
			Object	Registration Page
			Priority	Middle
			Input	abcd
			Output	Password should be more than 6 characters
			Step	If the input is illegal, pop a warning
			Method	MT
UR-02-Login	TR-02-01	TR-02-01-01	Name	Detect if the e-mail address is legal
			Object	Login Page
			Priority	Middle
			Input	abcd/(EMPTY)/abc@d
			Output	Not a legal address
			Step	If the address is illegal, pop a warning
			Method	MT/WB
		TR-02-01-02	Name	Check if the e-mail fits the password
			Object	Login Page
			Priority	High
			Input	abcd@163.com+1234567
			Output	e-mail or password not correct
			Step	If the e-mail does not fit the password in our database, pop a warning
			Method	MT/BB

Table 4: Testing Use Case Table (1)

Required Item	Index	Use Case Index	Use Case	
UR-03-Recommend	TR-03-01	TR-03-01-01	Name	Check if the itinerary can be returned with any like input.
			Object	Recommendation Page
			Priority	High
			Input	All possible combinations of input (12*4^6=49,152 kinds of input)
			Output	The Itinerary page with properly displayed itinerary
			Step	Use auto-generated test inputs andcheck if the return of database is non-empty
			Method	AT/BB
UR-04-Customization	TR-04-01	TR-04-01-01	Name	Check if any hotel can be selected
			Object	Customization kPage
			Priority	Middle
			Input	Selection of a hotel
			Output	Selected
			Step	If a hotel is selected, other hotels will be removed from the map
			Method	MT/BB
		TR-04-01-02	Name	Check if any sight can be selected
			Object	Customization kPage
			Priority	Middle
			Input	Selection of a sight
			Output	Selected, route displayed
			Step	If a sight is selected, a route going there will be displayed as well as this sight
			Method	MT/BB

Table 5: Testing Use Case Table (2)

Required Item	Index	Use Case Index	Use Case	
UR-05-ViewItinerary	TR-05-01	TR-05-01-01	Name	Check if the itinerary can be properly displayed
			Object	DisplayItinerary Page
			Priority	High
			Input	An itinerary from database
			Output	The itinerary on the map
			Step	Send the itinerary from database and on the phone an itinerary on the map will be shown
			Method	MT/WB
		TR-05-01-02	Name	Check if the spot detail page can be properly displayed
			Object	DisplayItinerary Page
			Priority	Middle
			Input	A spot
			Output	The spot detail page
			Step	When click on a spot, the spot detail page will be displayed.
			Method	MT/BB
	TR-05-02	TR-05-02-01	Name	Check if the function works in recommendation part
			Object	DisplayItinerary Page
			Priority	High
			Input	Recommendation
			Output	Itinerary
			Step	User select the likes and an itinerary will be displayed.
			Method	MT/BB
		TR-05-02-02	Name	Check if the function works in customization part
			Object	DisplayItinerary Page
			Priority	High
			Input	Customization
			Output	Itinerary
			Step	User inputs his/her customization and the selection will be properly displayed
			Method	MT/BB
		TR-05-02-03	Name	Check if the function works in recovery part
			Object	DisplayItinerary Page
			Priority	High
			Input	RecoveryID
			Output	Itinerary
			Step	The itinerary is saved in the database with a specific recovery ID. The frontend sends the recovery ID to backend and then database sends itinerary to frontend. Finally the itineray should be displayed properly.
			Method	AT/WB

Table 6: Testing Use Case Table (3)

Stage	Workload (Person*Day)	Percentage
Test Plan	2	14%
Test Design	5	33%
Test Preparation	1	6%
Test Performance	5	33%
Test Evaluation	2	14%

Table 7: Workload Estimation

Role	Personnel	Job
Test Manager	Song Yiwen	Manage the whole process of testing. Adjust the testing process according to real situations.
Test Planner	Xie Zhihui	Plan the details of testing process according to technical details and user requirements.
Test Performer	All others	Perform testing as planned. Report the test result and evaluate the possible faults that leads to failure.

Table 8: Personnel Requirement

Milestone	Start	End	Remark
Planning	2019.6.14	2019.6.15	Complete the test planning.
Designing	2019.6.15	2019.6.16	Complete designing test procedure.
Preparation	2019.6.16	2019.6.16	Prepare for testing.
Execution	2019.6.16	2019.6.18	Complete all the planned testing procedures.
Evaluation	2019.6.18	2019.6.20	Evaluate the test results and complete the test summary.

Table 9: Progress Plan

Risk	Percentage	Negative Effect
Serious failure happens that the software can't run normally	5	10
Use cases are not designed completely	35	8
Too many use cases that the test time costs more than predicted	10	6
Not enough personnel	20	6

Table 10: Risk Management