TEST PLAN SPECIFICATION

KINGO COFFEE

Introduction to Software Engineering_SWE3002_41

TEAM 10
TEAM LEADER | JISU RYOU
TEAM MEMBER | CHANHO KIM
TEAM MEMBER | KANGIN PARK
TEAM MEMBER | YELIM SO
TEAM MEMBER | JUNSUNG LEE
TEAM MEMBER | SUYOUNG BAE

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 PURPOSE

This Test Plan document specifies the specific test of the KINGO COFFEE application. The main reader of this document is Team 10, and stakeholders such as the professor and TA in 'Introduction to software engineering' class can be the main reader. This document covers specific methods, test cases, and strategies for testing the application.

1.2 SCOPE

This Test Plan document describes the units identified in SDS and the tests to be performed on the interface. Test elements include functional requirements, nonfunctional requirements identified in SRS.

The following units are tested:

- Authentication
- Profile
- Map View
- Café
- Order
- Payment

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATION

The following table lists the abbreviations used in this document.

Acronyms & Abbreviations	Explanation
SRS	Software Requirement Specification
SDS	Software Design Specification
UI	User Interface

UX	User Experience
SKKU	SungKyunKwan University
AWS	Amazon Web Services
GPS	Global Positioning System
API	Application Programming Interface
EC2	Elastic Compute Cloud
JWT	JSON Web Token
SSH	Secure Shell

Table 1 Table of Abbreviations

The following table defines the specific technical terms used in this document.

Terms	Explanation
User	It means the user who uses this application
Customer	It is used in the same sense as User and refers to a person who orders through an application
Front-end	It refers to the interface used by User, including UI/UX
Back-end	Area that users can not directly use, including Database and Server
KINGO COFFEE	Name of the application covered by this document
Waiting time	Estimated wait time for order processing based on order status at the cafe
Arrival time	The time it takes to get to the cafe from the current location.

Table 2 Table of Technical Terms

1.4 References

- https://www.guru99.com/test-plan-for-project.html
- https://www.guru99.com/unit-testing-guide.html
- https://www.guru99.com/interface-testing.html

1.5 Overview

This test plan document contains four chapters, excluding the current chapter. Chapter 2 covers the overall coverage of all tests covered in documents that include the unit test and the criteria for success. Chapter 3 covers the details of the unit test. Each unit contains a detailed test case of test input, expected results. Chapter 4 covers the details of the interface test. Chapter 5 covers document history.

2. Approach

2.1 Test Method

2.1.1 Software unit test methods

2.1.1.1 Overview

Objective	Test the KINGO COFFEE units independently to make sure that each unit works perfectly by itself
Required Tool	The tools required to achieve the unit testing: • JUnit
Success Criteria	The success criteria will be tested for: • Use cases of the system • Features
Special Considerations	Unit testing tests the functionalities only and cannot catch any errors at the integrity level, which should be taken into consideration to back up the unit tests with other types of testing.

Table 3 Table of Software Unit Test Overview

2.1.1.2 Success Criteria

Use Case	Feature	Description
Register	Required Fields	Check user should Register by filling all the required fields
	Required Fields	Check the required fields by not filling any data
	Email Validation	Check there's not an existing email
	Password Validation	Check the valid password in 8~11 limit
	Phone Number Validation	Check user can verify its phone number
Login	Required Fields	Check user should login by filling all the required fields
	Required Fields	Check the required fields by not filling any data
	User Login	Check when passing a correct email and invalid password
	User Login	Check when passing correct email and password
	Register Option	Check whether the register link for the new user is working
Logout	User Logout	Check user can successfully log out
Get Profile	Get Profile	Check user's every information is shown
Set Profile	Required Fields	Check user should set profile by filling all the required fields
	Required Fields	Check the required fields by not filling any data
	Email Validation	Check there's not an existing email
	Password Validation	Check the valid password in 8~11 limit

	Phone Number Validation	Check user can verify its phone number
Map View	Show Map	Check the map is shown according to user's current location
	Switch Campus	Check the map is switched according to the selected campus
	Select Cafe	Check when a cafe is selected
	Get Cafe Info	Check the selected cafe's information(name, location, expected waiting time, expected arrival time) is shown correctly
	Recommendation	Check whether a cafe is properly recommended
Cafe	Required Fields	Check cafe manager should register a cafe by filling all the required fields
	Required Fields	Check the required fields by not filling any data
	Get Order List	Check whether a list of orders so far is shown
	Update Cafe Status	Check whether a cafe's status is successfully updated
	Delete Order	Check whether an order is successfully deleted
Order	Make Order	Check whether an order is successfully made
	Get Order Info	Check whether an order's every information is shown
	Cancel Order	Check whether the order is successfully cancelled
Payment	Payment	Check user should request a payment with a total amount
	Payment	Check when requesting a payment without any cost
	Cancel Payment	Check whether a payment is successfully cancelled

TABLE 4 TABLE OF SOFTWARE UNIT TEST

2.1.2 *Software Interface test methods*

2.1.2.1 Overview

Objective	Detect faults due to interface errors or invalid assumptions about interfaces
Required Tool	The tools required to achieve the interface testing: • MockMvc
Success Criteria	The success criteria will be tested for: • AWS • MySQL • Authentication • SpringBoot
Special Considerations	N/A

Table 5 Table of Software Interface Test Overview

2.1.2.2 Success Criteria

Interface	Test	Description
AWS	Connection Test	Check whether the connection to the aws server is normal
	API Test	Check whether the API meets expectations for functionality, reliability, performance, and security
MySQL	Connection Test	Check whether the connection to the mySQL server is normal
Authentication (JWT Token)	Return Token Test	Check whether JWT token is created successfully
	Create Cafe Test	Check if the authorization can be used successfully with JWT token

SpringBoot	API Test	Check whether the API meets expectations for functionality, reliability, performance, and security

TABLE 6 TABLE OF SOFTWARE INTERFACE TEST

3. SOFTWARE UNIT TEST

3.1 AUTHENTICATION

Test basic information for register, positive and negative case for login and logout function.

3.1.1 REGISTER

3.1.1.1 Required Fields

- success

Test case object	To verify that the user has filled in the required fields
Test Inputs	User's information (Email, Password, Name, Phone Number, Campus)
Expected Results	A success message will be displayed for the correct credential

TABLE 7 TABLE OF REGISTER'S REQUIRED FIELDS TEST – SUCCESS

- failure

Test case object	To check the required fields by not filling any data
Test Inputs	User information that does not meet all requirements
Expected Results	An error message will be displayed for the wrong credential

TABLE 8 TABLE OF REGISTER'S REQUIRED FIELDS TEST – FAILURE

3.1.1.2 Email Validation

- success

Test case object	To check that there's an existing email
Test Inputs	Correct email address
Expected Results	A success message will be displayed

TABLE 9 TABLE OF REGISTER'S EMAIL VALIDATION TEST – SUCCESS

- failure

Test case object	To check that there's not an existing email
Test Inputs	Wrong email address
Expected Results	An error message will be displayed

TABLE 10 TABLE OF REGISTER'S EMAIL VALIDATION TEST – FAILURE

3.1.1.3 Password Validation

- success

Test case object	To check the password in 8~11 limit
Test Inputs	8-11 passwords, including uppercase, lowercase and numbers
Expected Results	A success message will be displayed

TABLE 11 TABLE OF REGISTER'S PASSWORD VALIDATION TEST – SUCCESS

- failure

Test case object	To check the password in 8~11 limit
Test Inputs	Number not met by password criteria
Expected Results	An error message will be displayed

TABLE 12 TABLE OF REGISTER'S PASSWORD VALIDATION TEST – FAILURE

3.1.1.4 Phone Number Validation

- success

Test case object	To check user can verify its phone number
Test Inputs	Valid phone number
Expected Results	A success message will be displayed

TABLE 13 TABLE OF REGISTER'S PHONE NUMBER VALIDATION TEST – SUCCESS

– failure

Test case object	To check user can verify its phone number
Test Inputs	Invalid phone number
Expected Results	An error message will be displayed

TABLE 14 TABLE OF REGISTER'S PHONE NUMBER VALIDATION TEST – FAILURE

3.1.2 LOGIN

3.1.2.1 Required Fields

– success

Test case object	To verify that the user has filled in the required fields
Test Inputs	User's information (Email, Password)
Expected Results	A success message will be displayed for the correct credential

TABLE 15 TABLE OF LOGIN'S REQUIRED FIELDS TEST – SUCCESS

– failure

Test case object	To check the required fields by not filling any data
Test Inputs	User information that does not meet all requirements
Expected Results	An error message will be displayed for the wrong credential

TABLE 16 TABLE OF LOGIN'S REQUIRED FIELDS TEST – FAILURE

3.1.2.2 User Login

- success

Test case object	To check when passing a correct email and password
Test Inputs	User login info (Email, Password)
Expected Results	Login success message

TABLE 17 TABLE OF LOGIN'S USER LOGIN TEST – SUCCESS

- failure

Test case object	To check when passing a correct email and invalid password
Test Inputs	User login info (Email, Password)
Expected Results	Login failed message

TABLE 18 TABLE OF LOGIN'S USER LOGIN TEST – FAILURE

3.1.2.3 Register Option

Test case object	Check whether the register link for the new user is working
Test Inputs	Logic Data, button
Expected Results	Display Register screen

TABLE 19 TABLE OF LOGIN 'S REGISTER OPTION TEST

3.1.3 LOGOUT

Test case object	To check user can successfully log out
Test Inputs	Logic Data, button
Expected Results	Logout success message with screen with log –out status.

TABLE 20 TABLE OF LOGOUT TEST

3.2 PROFILE

Tests for basic information, such as user's name, phone number, ID, password, and default campus selection.

3.2.1 GET PROFILE

Test case object	To check user's information is shown
Test Inputs	Logic Data, button
Expected Results	Displays the user's profile screen and shows the user's information.

TABLE 21 TABLE OF GET PROFILE TEST

3.2.2 SET PROFILE

3.2.2.1 Required Fields

- success

Test case object	To verify that the user has filled in the required fields
Test Inputs	User's information (Email, Password, Name, Phone Number, Campus)
Expected Results	A success message will be displayed for the correct credential

TABLE 22 TABLE OF SET PROFILE'S REQUIRED FIELDS TEST - SUCCESS

- failure

Test case object	To check the required fields by not filling any data
Test Inputs	User information that does not meet all requirements
Expected Results	An error message will be displayed for the wrong credential

TABLE 23 TABLE OF SET PROFILE'S REQUIRED FIELDS TEST - FAILURE

3.2.2.2 Email Validation

- success

Test case object	To check that there's an existing email
Test Inputs	Correct email address
Expected Results	A success message will be displayed

Table 24 Table of Set Profile's Email Validation Test – Success

- failure

Test case object	To check that there's not an existing email
Test Inputs	Wrong email address
Expected Results	An error message will be displayed

Table 25 Table of Set Profile's Email Validation Test – Failure

3.2.2.3 Password Validation

- success

Test case object	To check the password in 8~11 limit
Test Inputs	8-11 passwords, including uppercase, lowercase and numbers
Expected Results	A success message will be displayed

Table 26 Table of Set Profile's Password Validation Test – Success

- failure

Test case object	To check the password in 8~11 limit
Test Inputs	Number not met by password criteria
Expected Results	An error message will be displayed

Table 27 Table of Set Profile's Password Validation Test – Failure

3.2.2.4 Phone Number Validation

- success

Test case object	To check user can verify its phone number
Test Inputs	Valid phone number
Expected Results	A success message will be displayed

Table 28 Table of Set Profile's Phone Number Validation Test - Success

- failure

Test case object	To check user can verify its phone number
Test Inputs	Invalid phone number
Expected Results	An error message will be displayed

Table 29 Table of Set Profile's Phone Number Validation Test - Failure

3.3 Map

Tests maps for your current location, information from selected cafes, and recommendations for accuracy.

3.3.1 Show Map

Test case object	To check the map is shown according to user's current location
Test Inputs	Map View button, Google map API
Expected Results	Displays the map screen and shows the user's current location.

Table 30 Table of Show Map Test

3.3.2 Switch Campus

- Humanities and Social Sciences (Seoul) Campus

Test case object	To check the map is switched according to the selected campus
Test Inputs	Select button, Google map API

Expected Results	Displays a map of Seoul Campus

TABLE 31 TABLE OF SWITCH CAMPUS TEST – HS CAMPUS

– Natural Sciences (Suwon) Campus

Test case object	To check the map is switched according to the selected campus
Test Inputs	Select button, Google map API
Expected Results	Displays a map of Suwon Campus

TABLE 32 TABLE OF SWITCH CAMPUS TEST – NS CAMPUS

3.3.3 SELECT CAFE

Test case object	To check when a café is selected
Test Inputs	Café button
Expected Results	Displays a screen containing information from that café

TABLE 33 TABLE OF SELECT CAFE TEST

3.3.4 GET CAFE INFO

Test case object	To check that the selected café's information is shown correctly
Test Inputs	Café button
Expected Results	Shows correct information (name, location, expected waiting time, expected arrival time) of the café

TABLE 34 TABLE OF GET CAFE INFO TEST

3.3.5 RECOMMENDATION

Test case object	To check whether a café is properly recommended

Test Inputs	User's location, café location, café waiting time, expected arrival time
Expected Results	Displays the recommended café list screen

TABLE 35 TABLE OF RECOMMENDATION TEST

3.4 CAFÉ

3.4.1 REGISTER CAFE

- success

Test case object	Check cafe manager should register a cafe by filling all the required fields
Test Inputs	Every Cafe key table
Expected Results	success message

TABLE 36 TABLE OF REGISTER CAFE TEST – SUCCESS

- failure

Test case object	To check the required fields by not filling any data
Test Inputs	Cafe information that does not meet all requirements
Expected Results	error message

TABLE 37 TABLE OF REGISTER CAFE TEST – FAILURE

3.4.2 GET ORDER LIST

Test case object	Check whether a list of orders so far is shown
Test Inputs	Item name and price that customer click expected order page ui
Expected Results	match success message

TABLE 38 TABLE OF GET ORDER LIST TEST

3.4.3 UPDATE CAFE STATUS

Test case object	Check whether a cafe's status is successfully updated
Test Inputs	customer's order table cafe table
Expected Results	update success message

TABLE 39 TABLE OF UPDATE CAFE STATUS TEST

3.4.4 DELETE ORDER

Test case object	Check whether an order is successfully deleted
Test Inputs	Café button
Expected Results	Shows correct information (name, location, expected waiting time, expected arrival time) of the café

TABLE 40 TABLE OF DELETE ORDER TEST

3.5 ORDER

3.5.1 MAKE ORDER

Test case object	Check whether an order is successfully made
Test Input	Item name and price Expected total price
Expected Results	Order pass success message

TABLE 41 TABLE OF MAKE ORDER TEST

3.5.2 GET ORDER INFO

Test case object	Check whether an order's every information is shown
Test Input	Item name and price that customer click expected cart page ui
Expected Results	match success message

TABLE 42 TABLE OF GET ORDER INFO TEST

3.5.3 CANCEL ORDER

Test case object	Check whether the order is successfully cancelled
Test Input	canceled item expected cart page ui, total price
Expected Results	cancel success message

TABLE 43 TABLE OF CANCEL ORDER TEST

3.6 PAYMENT

3.6.1 PAYMENT

- success

Test case object	Check user should request a payment with a total amount
Test Input	success total payment value
Expected Results	payment success message

TABLE 44 TABLE OF PAYMENT TEST – SUCCESS

- failure

Test case object	Check user should request a payment with a total amount
Test Input	success total payment value

Expected Results	payment failure message

Table 45 Table of Payment Test – Failure

3.6.2 Cancel Payment

- success

Test case object	Check whether a payment is successfully cancelled
Test Input	Customers' cancel request
Expected Results	Cancel success message

Table 46 Table of Cancel Payment Test – Success

- failure

Test case object	Check whether a payment is successfully cancelled
Test Input	Customers' cancel request
Expected Results	Cancel success message

Table 47 Table of Cancel Payment Test – Failure

4. Software Interface test

4.1. AWS : EC2

Basically, we decided to select ec2 instances as back-end server environments in this task. Therefore, the EC2 instance interface test attempts to conduct two tests: It will conduct a simple API test with the domain after being assigned an elastic ip address(fixed IPV4 address) and a test to verify that the server is normally connected with the ssh command

4.1.1 Connection test

Test case object	To check external software interfaces of AWS

Test Input	ssh -i "KeyPair.pem" ubuntu@ec{ip}.us-east-2.compute.amazonaws.com
Expected Results	Server Connect Success message

TABLE 48 TABLE OF AWS CONNECTION TEST

4.1.2 SIMPLE CONTROLLER(API) TEST

Test case object	Hello Controller
Test Input	http://{ip}/hello
Expected Results	“hello”

TABLE 49 TABLE OF AWS API TEST

4.2. EC2 : MYSQL

In this task, we will utilize Spring Data JPA framework to communicate with Mysql and manage the data. This method will assign 3306 ports to ec2 servers and use them as Mysql servers. Therefore, an interface test is required to ensure that the Mysql server is connected well. Verify that the ip address assigned in 4.1 and port 3306 are connected to the server

4.2.1 CONNECTION TEST

Test case object	Mysql Server Connect Object.
Test Input	spring.datasource.url=jdbc:mysql://{ip}:3306/{project_name}?useUnicode=true&characterEncoding=utf8
Expected Results	Connection Success message

TABLE 50 TABLE OF MYSQL CONNECTION TEST

4.3 AUTHENTICATION, JWT TOKEN

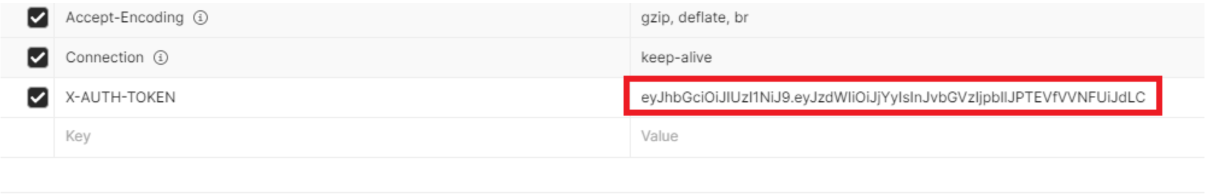
In this task, the issue of user authentication is a very important issue because the permissions of Cafe Manager and User are designed differently. When implemented using Oauth2, there was a disadvantage of high latency and difficulty in designing tests because each server had to go through an external certificate server. On the other hand, in the case of JWT token, user authentication information can be easily retrieved from the database because it contains user authentication information. Therefore, a JWT token was used, and two tests were designed to certify its functionality. Two tests were designed to ensure that JWT token was successfully returned at the time of User Signup, and that permissions could be used at the request header.

4.3.1 RETURN TOKEN TEST

Test case object	JWT token provider
Test Input	User Object
Expected Results	X-AUTH-TOKEN

TABLE 51 TABLE OF AUTHENTICATION RETURN TOKEN TEST

FIGURE 1 X-AUTH-TOKEN IN REQUEST HEADER



☑	Accept-Encoding ⓘ	gzip, deflate, br
☑	Connection ⓘ	keep-alive
☑	X-AUTH-TOKEN	eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiJjYyIsInJvbGVzIjpbIlJPTEVfVVNFUiJdLC
	Key	Value

4.3.2 CREATE CAFE TEST

Test case object	Cafe Management Object
Test Input	Cafe Object, X-AUTH-TOKEN in request header
Expected Results	Success message

TABLE 52 TABLE OF AUTHENTICATION CREATE CAFE TEST

4.4. SPRINGBOOT API TEST

API Controller test will be conducted using MockMvc, the most representative method of testing the SpringBoot framework. MockMvc is an object that means request and response of a browser, and it can specify the request of that API, and it is also easy to analyze response. In order to proceed with the test, we will put one User in the User repository before the test and conduct the test using @WithUserDetail(username=”predefined user”).

4.4.1 CREATE TEST

Test case object	MockMvc Object

Test Input	@WithUserDetail(username=”predefined_user”)
	URL = “http://{ip}:8080/{domain_name}/save”, method = POST
	List<Object> all = ObjectRepository.findall()
	assertThat(all.get(0).get{attribute}).isEqualTo({test_attribute})

Expected Results	Test success

TABLE 53 TABLE OF SPRINGBOOT API TEST – CREATE

4.4.2 READ TEST

Test case object	MockMvc Object, ObjectMapper
Test Input	@WithUserDetail(username=”predefined_user”) URL = “http://{ip}:8080/{domain_name/read/{page_no}/{size}”, method = GET
Expected Results	Test success

TABLE 54 TABLE OF SPRINGBOOT API TEST – READ

4.4.3 UPDATE TEST

Test case object	MockMvc Object
Test Input	@WithUserDetail(username=”predefined_user”)

	URL = "http://{ip}:8080/{domain_name}/update/ObjectId",
	method = POST
	List<Object> all = ObjectRepository.findall()
	assertThat(all.get(0).get{attribute}).isEqualTo({update_attribute})

Expected Results	Test success

TABLE 55 TABLE OF SPRINGBOOT API TEST – UPDATE

4.4.4 DELETE TEST

Test case object	MockMvc Object

Test Input	@WithUserDetail(username="predefined_user")
	URL = "http://{ip}:8080/{domain_name}/delete", method = Delete
	List<Object> all = ObjectRepository.findall()
	assertThat(all.size()).isEqualTo(0)

Expected Results	Test success

TABLE 56 TABLE OF SPRINGBOOT API TEST – DELETE

5. SUPPORTING INFORMATION

5.1 DOCUMENT HISTORY

Date	Version	Description	Writer
2021/05/21	0.1	Style and overview	Jisu Ryou
2021/05/21	1.0	Addition of 1.1, 1.2	Kangin Park
2021/05/22	1.1	Addition of 4.1, 4.2	Chanho Kim

2021/05/22	1.2	Addition of 2	Jisu Ryou
2021/05/24	1.3	Addition of 3.1	Junsung Lee
2021/05/24	1.4	Addition of 3.4, 3.5, 3.6	Suyoung Bae
2021/05/25	1.5	Addition of 1.3, 1.4, 1.5	Kangin Park
2021/05/25	1.6	Addition of 4.3, 4.4	Chanho Kim
2021/05/25	1.7	Revision of 1.4	Kangin Park
2021/05/26	1.8	Addition of 3.2, 3.3	Yelim So
2021/05/26	1.9	Revision of 4	Chanho Kim
2021/05/27	2.0	Revision of structure	Jisu Ryou

Table 57 Table of Document History