
Software Functionality Test Plan Document

for

Coffee-OT Project

Prepared by Kelvin David

University of Waikato

05/05/2020

Contents

1.Introduction.....	3
2.Detailed Testing Strategy.....	3
2.1 <i>Test plan table</i>	3
2.2 <i>TestPlan1</i>	3
2.3 <i>TestPlan2</i>	4
2.4 <i>TestPlan3</i>	5
2.5 <i>TestPlan4</i>	5
2.6 <i>TestPlan5</i>	6
2.7 <i>TestPlan6</i>	6
2.8 <i>TestPlan7</i>	7
2.9 <i>TestPlan8</i>	7
2.10 <i>TestPlan9</i>	8
2.11 <i>TestPlan10</i>	9
2.12 <i>TestPlan11</i>	9

1. Introduction

This document provides the test documentation that facilitates the functional tasks involved with the development of the Coffee-OT project. It will include detailed test plans and cases that will cover all 11 requirements.

2. Detailed Testing Strategy

2.1 Test Plan Summary

TEST PLAN TABLE			
Test Plan	Requirement	Level	Components
TP1	Unauthorized Site Landing	Unit	App-Server
TP2	Unauthorized Sign Up	Integration	App-Server, Database
TP3	Unauthorized Log in	Integration	App-Server, Database
TP4	User Logout	Integration	App-Server, Database
TP5	New Coffee Machine Record	Integration	App-Server, Database
TP6	Delete Coffee Machine Record	Integration	App-Server, Database
TP7	View Calendar	Integration	App-Server, Database
TP8	Browse Calendar	Integration	App-Server, Database
TP9	Add New Activation Event	Integration	App-Server, Database
TP10	Delete Event	Integration	App-Server, Database
TP11	Polling	Integration	App-Server, Database

Shared Test

Requirements involved with TestPlan4-TestPlan10 all require authorization check before the function request can be accepted. The same authorization test can be used for all these test plans by changing the type and URI.

Valid -> if request has appropriate authorization.

Invalid -> if request has no authorization.

2.2 Test Plan 1

Preconditions:

- User requested any of the website's pages
- User is not authorized

Equivalent Classes:

- Invalid input is request with authorization
- Valid input is request without authorization

Testing Methodology: Automatic

Test Case	Input	Output	Description
01	GET request without user authorization for any page.	401 Unauthorized Return login page	Users without authorization should be redirected to the login page. Expected to succeed.
02	GET request with appropriate user authorization for any page	200 OK Return '/{pageURI}'	Users with authorization should be redirected to their homepage. Expected to fail.

2.3 Test Plan 2

Preconditions:

- User is not authorized
- User is on the login page

Equivalent Classes:

- Valid input when username is available
- Invalid input when username/password length is below 5
- Invalid input when username length is more than 15

Boundary values:

- Username length 5 – 15
- Password length < 5

Testing Methodology: Automatic

Test Case	Input	Output	Description
01	PUT request to {URL}/signup with contents of Valid username & password	200 OK Account added	This tests if accounts can be added to the DB. Expected to succeed.
02	2 PUT requests to {URL}/signup with the same username & password as content.	409 Username already exists	This tests for stopping username duplicates. second request is expected to fail.
03	PUT requests to {URL}/signup with a username with less than 5 or more than 15 length and valid password as content	403 Username doesn't meet requirements	This tests for stopping invalid usernames & passwords. Expected to fail.
04	PUT requests to {URL}/signup with a password with less than 5 length and valid username as content	403 Password doesn't meet requirements	

2.4 Test Plan 3

Preconditions:

- User is not authorized
- User on login page

Equivalent Classes:

- Valid when username/password match
- Invalid when username not found
- Invalid when password doesn't match

Testing Methodology: Automatic

Test Case	Input	Output	Description
01	GET request {URL}/login with valid username/password	Give user appropriate authentication. Redirect to /calendar	Testing for if login works with valid credentials. Expected to succeed.
02	GET request {URL}/login with empty content	404 username not found	Testing for logins to bounce with invalid credentials. Expected to fail.
03	GET request {URL}/login with invalid password	401 Authorization failed	
04	GET request {URL}/login with invalid username	404 Username not found	

2.5 Test Plan 4

Preconditions:

- User is not authorized

Equivalent Classes:

- Valid input if user is authenticated
- Invalid input if user is unauthenticated

Testing Methodology: Manual

Test Case	Input	Output	Description
01	GET request for {URL}/logout with appropriate authentication	Remove user's authentication	Testing for logout as authorized user. Expected to succeed.
02	GET request {URL}/logout without authentication.	401 authorization failed	Testing for logout as unauthorized user Expected to fail.

2.6 Test Plan 5

Preconditions:

- User is not authorized
- User is on manage coffee page

Equivalent Classes:

- Valid input when user is authenticated
- Invalid input when user is unauthenticated
- Invalid input when CoffeeID already exists

Boundary Values

- Invalid < 0 ID in database

Test Methodology: Automatic

Test Case	Input	Output	Description
01	PUT request to /coffeemachines/{id} with appropriate authorization, userID in header	Add {id} to the database	Testing adding coffeeMachineRecords with authorization. Expected to succeed.
02	PUT request to /coffeemachines/{id} without appropriate authorization, userID in header	401 authorization failed	Testing adding coffeeMachineRecords without authorization. Expected to fail.
03	PUT request to /coffeemachines/{id} with appropriate authorization, userID in header (twice with the same id)	409 Id already exists	Testing adding the same {id} twice on one user. Expected to fail

2.7 Test Plan 6

Preconditions:

- User is authorized
- User is on manage coffee page

Equivalent Classes:

- Valid input if > 0 matches of the coffeeID
- Invalid input if < 0 matches of the coffeeID
- Invalid input if user is unauthenticated.

Boundary Values:

- Invalid < 0 matches
- Valid > 0 matches

Testing Methodology: Manual

Test Case	Input	Output	Description
01	DELETE request of an existing coffeemachineld to /coffeemachines/{coffeeMachineld} with appropriate user authentication.	Delete Record from database	Testing if we can delete coffee machines from the DB Expected to Succeed
02	DELETE request of a non-existing coffeemachineld to /coffeemachines/{coffeeMachineld} with appropriate user authentication.	404 Coffeemachineld not found	Testing if we bounce DELETE requests for machines that don't exist in our DB Expected to fail.

2.8 Test Plan 7

Preconditions:

- User is authorized

Equivalent Classes:

- Valid input when user is authenticated
- Invalid input when user is unauthenticated

Testing Methodology: Manual

Test Case	Input	Output	Description
01	GET request to /calendar/week with appropriate authorization, userID in header	200 OK Add {id} to the database	Testing if authorized user can view their calendar Expected to succeed.
02	GET request to /calendar/week without appropriate authorization	401 authorization failed	Testing if unauthorized user cannot view calendars Expected to fail.

2.9 Test Plan 8

Preconditions:

- User is authorized
- User on calendar page

Equivalent Classes:

- Valid input if we user is authorized
- Invalid input if user is unauthorized

Testing Methodology: Manual

{page} definition for Shared TestCase01-04 & TestCase05-08: year,month,week,day

Test Case	Input	Output	Description
01-04	GET request to /calendar/{page} with appropriate authorization, userID in header	200 OK Get calendar/date page from DB	Testing if authorized user can change view of their calendar Expected to succeed.
05-08	GET request to /calendar/week without appropriate authorization	401 Authorization failed	Testing if unauthorized user can change view of their calendar. Expected to fail.

2.10 Test Plan 9

Preconditions:

- User is authorized
- User on calendar page

Equivalent Classes:

- Valid input when user is authenticated (shared testing)
- Invalid input when activationEvent for coffeeMachineId exists in dateTime

Testing Methodology: Automatic

Test Case	Input	Output	Description
01	PUT request to /activationEvents/{coffeeMachineId}/{dateTime} with authentication, userID in the header	200 OK Add event to DB	Testing to add activationEvent with authorization. Expected to succeed.
02	PUT request to /activationEvents/{coffeeMachineId}/{dateTime} without authentication, userID in the header	401 authorization failed	Testing to add activationEvent with authorization. Expected to fail.
03	PUT request to /activationEvents/{coffeeMachineId}/{dateTime} with authentication, userID in the header (twice with the same Id and dateTime)	409 DateTime already exist for {coffeeMachineId}	Testing to add event duplicate in DB. Expected to fail.

2.11 Test Plan 10

Preconditions:

- User is authorized
- User on calendar page

Equivalent Classes:

- Valid input if > 0 matches of the eventID
- Invalid input if < 0 matches of the eventID
- Invalid input if user is unauthenticated

Boundary Values:

- Invalid < 0 matches
- Valid > 0 matches

Testing Methodology: Manual

Test Case	Input	Output	Description
01	DELETE request of existing eventID to /activationEvents/{eventID} with authentication, userID in header	200 OK Delete activation event from the database	Testing if we can delete eventID from DB Expected to Succeed.
02	DELETE request of non-existent eventID to /activationEvents/{eventID} with authentication, userID in header	404 Activation events not found	Testing if we can delete non-existent eventID Expected to fail.

2.12 Test Plan 11

Equivalent Classes:

- Valid when userProvidedID&key exists in DB
- Invalid when userProvidedID exists in DB
- Invalid when key exists in DB

Testing Methodology: Automatic

Test Case	Input	Output	Description
01	GET request to {URL}/remoteCoffeeMachine/{userProvidedId}/{key} (within 30seconds of activation)	Return "BREW!"	Testing for successful activation. Expected to succeed.

02	GET request to {URL}/remoteCoffeeMachine/{userProvidedId}/{key} (not 30seconds of activation)	Checked response	Testing for successful polling. Expected to succeed
02	GET request to {URL}/remoteCoffeeMachine/{userProvidedId}/{key} (not 30seconds of activation) with Invalid userProvidedID	404 Not found	Testing polling with invalid ID. Expected to fail
03	GET request to {URL}/remoteCoffeeMachine/{userProvidedId}/{key} (not 30seconds of activation) with invalid key	404 Not found	Testing polling with invalid key. Expected to fail