

EzTravels - Tour Management System

Software Test Plan

CSCI-P465/565 (Software Engineering I)

Project Team

Aditya Padgal

Ava Greene

Avishmita Mandal

Hanna Adams

Owen Harris

Yashaswini Sampath Kumar

1. Overview

The EzTravels Tour Management System has been tested on old and new features developed in component on both the backend and the frontend through API calls and also via the UI.

1.1 Test Objectives

- To make sure that the user can safely authenticate into the service, sign-up for the service. And malicious actors do not have the access if they do not have the right credentials.
- Ensure accounts can be logged into with valid username and password.
- Ensure user information is stored and accessed correctly.
- Ensure the user is able to reset password by validating top.
- Ensure the user is able to Add, delete and view places added to the itinerary.
- To check the backend working of search places.
- To check the backend working of reset password.

1.2. Test Environment

- Typical web browsers: Chrome, Firefox, and Edge on Windows 10, and Safari on ios.
- Postman was used to make the API calls to test the login and signup. Responses were received which indicated the

validity of the flow that was introduced in the development process.

1.3. Test Personnel

- Aditya Padgal was involved in the testing process as developers. Responsibility was to test the basic functionality of authentication flow and Google Oauth implementation with tokenization approach.
- Yashaswini Sampath Kumar, Owen and Avishmita Mandal were involved in the testing process of getting a list of hotels. Also testing the reset password for reentering the old password.
- Owen Harris was involved in testing the bookings apis.

1.4. Acceptance Criteria

- User should be able to login to the system using google OAuth without any errors, handling of jwt token should take place in backend whilst implementing OAuth.
- The Booking page should be up and running and user should be able to book.
- Users should be warned if trying to use their previous password when attempting to reset their password.
- The user should be able to see a list of hotels.
- The user should receive email and sms verification for bookings.
- The user should be able to make booking for flights, accommodations, and activities.

1.5. Noted Omissions

- We test only the basic authentication. We have not imposed any advanced input validations where we check for Regexes expressions for individual fields or other validation constraints. We have made all the fields in the signup as not null.

2. Test Cases

1. LOGIN FEATURE

Test Case 1: User Sign Up – Input validation.

Number :	1
Name :	User Sign Up - Input Validation
Description :	Users should sign up with all the fields present in the form, the fields are not null.
Initial Conditions :	NA
Input Data :	User details - name, email, phone number, password etc.
Specifications :	Successful sign up. The User can now login from the login page.
Procedure :	Enter name, email, phone number, password

Test Case 2: Invalid User Sign Up – Input validation.

Number :	2
Name :	Invalid User Sign Up - Input Validation
Description :	Users should fill all the fields present in the form, the fields are not null. If there are any empty fields during the sign-up, the service will prompt the user to enter the required field.

Initial Conditions :	NA
Input Data :	User details - name, email, phone number, password etc.
Specifications :	Unsuccessful sign up. The user is not signed up for the service. The user cannot login.
Procedure :	Enter the values of the fields leaving one of the fields empty

Test Case 3: Valid User Login

Number :	3
Name :	Valid User Login
Description :	The user should be able to login with correct credentials once they sign-up successfully
Initial Conditions :	Successful signup
Input Data :	User details - username, password
Specifications :	Successful login. Users should now be able to add stuff in itinerary and make bookings. Logout button should appear.
Procedure :	Enter valid username and password.

Test Case 4: Invalid User Login

Number :	4
Name :	Invalid User Login
Description :	The user should not be able to log in if they have not signed up / given wrong credentials.
Initial Conditions :	NA
Input Data :	User details - username, password
Specifications :	Unsuccessful login. A prompt to try again should appear. Logout button should not appear.
Procedure :	Enter invalid username and password.

Test Case 5: User Logout

Number :	5
Name :	User Logout
Description :	Users should be able to logout successfully once logged in.
Initial Conditions :	User must be logged in
Input Data :	NA
Specifications :	Successful logout. The user should be directed to the landing page. Login button should appear on the navigation bar of the home screen
Procedure :	Click on the logout button once logged in

Test Case 6: Forgot Password OTP generation for registered user

Number :	6
Name :	Forgot Password OTP generation
Description :	Users should be request for OTP on providing valid registered email
Initial Conditions :	User must be registered
Input Data :	User details (email).
Specifications :	Successful OTP generation to the registered email ID
Procedure :	Click on the Forgot password button Enter the email id Click on "Request OTP"

Test Case 7: Forgot password OTP generation for unregistered email id

Number :	7
Name :	Forgot Password OTP should not be generated for unregistered emails
Description :	The system only generates otp if the user is registered
Initial Conditions :	User must not be registered
Input Data :	NA
Specifications :	OTP is not generated throws a message user registered

Procedure :	Click on the Forgot password button Enter the email id Click on “Request OTP”
--------------------	---

Test Case 8: Test Forgot password OTP API case1

Number :	8
Name :	Forgot Password OTP generation
Description :	OTP generated on providing valid registered email
Initial Conditions :	Set up POSTMAN parameters
Input Data :	Send a POST request with a valid user’s email
Specifications :	Received status code of 200
Procedure :	Send a POST request to the forgot password API endpoint with valid user email. Verify that the API responds with a status code indicating a successful otp generation(e.g., HTTP 200 OK).Check the logs for messages

Test Case 9: Test Forgot password OTP API case2

Number :	9
Name :	Forgot Password OTP generation
Description :	OTP generated on providing valid not registered email
Initial Conditions :	Set up POSTMAN parameters
Input Data :	Send a POST request with a unregistered user’s email
Specifications :	Received status code of 500
Procedure :	Send a POST request to the forgot password API endpoint with unregistered user email. Verify that the API responds with a status code indicating a failure to otp generation(e.g., HTTP 500 bad server).Check the logs for messages “invalid user”

Test Case 10: Forgot Password update successful

Number :	10
Name :	Forgot Password reset successful
Description :	Users should be able to reset password on validating otp

Initial Conditions :	User must request for OTP
Input Data :	OTP and new password
Specifications :	Successful Password reset
Procedure :	Enter the correct OTP Enter the new password Click on “Confirm”

Test Case 11: Forgot Password update unsuccessful

Number :	10
Name :	Forgot Password reset unsuccessful
Description :	Users should be not able to reset password on validating otp
Initial Conditions :	User must request for OTP
Input Data :	OTP and new password
Specifications :	Unsuccessful Password reset
Procedure :	Enter the wrong OTP Enter the new password Click on “Confirm”

Test Case 12: Test Forgot password update API case1

Number :	12
Name :	Forgot Password update successful
Description :	Should successfully return success status on updating new password to backend
Initial Conditions :	Set up POSTMAN parameters
Input Data :	Send a POST request with a valid user's email,otp,new password
Specifications :	Received status code of 200

Procedure :	Send a POST request to the forgot password API endpoint with valid user email,otp,new password. Verify that the API responds with a status code indicating a successful password update(e.g., HTTP 200 OK).Check the logs for messages
--------------------	--

Test Case 13: Test Forgot password OTP API case1

Number :	13
Name :	Forgot Password update unsuccessful
Description :	Should successfully return failure status on updating new password to backend
Initial Conditions :	Set up POSTMAN parameters
Input Data :	Send a POST request with a invalid user's email,otp,new password
Specifications :	Received status code of 200
Procedure :	Send a POST request to the forgot password API endpoint with invalid user email,otp,new password. Verify that the API responds with a status code indicating a failure to update password(e.g., HTTP 500 OK).Check the logs for messages

2. ITINERARY PAGE

Test Case 14: Add places to Itinerary page from home page

Number :	14
Name :	Add place to Itinerary
Description :	Users should be able to add places to the Itinerary page.
Initial Conditions :	User must must be logged
Input Data :	Select places.
Specifications :	Place should be added to Itinerary.
Procedure :	Click on the save button of the place. Navigate to the Itinerary page and verify the place is saved.

Test Case 15: create a new Itinerary

Number :	15
Name :	Create a new Itinerary
Description :	Users should be able to create Itinerary by specifying dates of trip.
Initial Conditions :	User must must be logged
Input Data :	N/a
Specifications :	Itinerary should be created
Procedure :	Click on the create itinerary button

Test Case 16: create a new Itinerary

Number :	16
Name :	Create a new Itinerary
Description :	Users should be able to create Itinerary by specifying dates of trip.
Initial Conditions :	User must must be logged
Input Data :	N/a
Specifications :	Itinerary should be created
Procedure :	Click on the create itinerary button

Test Case 17: Delete a Itinerary

Number :	17
Name :	Delete a Itinerary
Description :	Users should be able to delete Itinerary
Initial Conditions :	User must must be logged and itinerary should be present
Input Data :	N/a
Specifications :	Itinerary should be delete
Procedure :	Click on the delete itinerary button

Test Case 18: Delete a place from the Itinerary

Number :	18
Name :	Delete a place from the Itinerary
Description :	Users should be able to delete a place from the Itinerary
Initial Conditions :	Users must be logged, itinerary should be present and the place should be present in the itinerary.
Input Data :	N/a
Specifications :	Place from the itinerary should be deleted
Procedure :	Select the place and click on REMOVE

Test Case 19: Delete an activity from the Itinerary

Number :	19
Name :	Delete an activity from the Itinerary

Description :	Users should be able to delete activities from the Itinerary
Initial Conditions :	User must must be logged in, itinerary should be present and the activity should be a part of the itinerary
Input Data :	N/a
Specifications :	Activity from the itinerary should be deleted
Procedure :	Click on the activity and click on the REMOVE button

Test Case 20: Update the time of departure/arrival to a place in the Itinerary

Number :	20
Name :	Update the time of departure/arrival to a place in the Itinerary
Description :	Users should be able to update the time of departure/arrival to a place in the Itinerary
Initial Conditions :	User must must be logged in, itinerary should be present and the place should be present.
Input Data :	N/a
Specifications :	The timing of the place should be modified.
Procedure :	Click on the place and then select from the calendar the arrival and the departure times.

Test Case 21: Update the start time and end time of an activity on the itinerary

Number :	21
Name :	Update the start time and end time of an activity on the itinerary
Description :	Users should be able to update the start time and end time of an activity on the itinerary
Initial Conditions :	User must must be logged, itinerary should be present and the activity should be present in the itinerary
Input Data :	N/a
Specifications :	The start and end times of the activity must be modified
Procedure :	Click on the activity and then select from the calendar the start and end times.

Test Case 22: Sharing an Itinerary

Number :	22
Name :	Sharing an itinerary
Description :	Users should be able to share their itinerary
Initial Conditions :	User must must be logged and itinerary should be present
Input Data :	N/a
Specifications :	Itinerary should be shared and the links should be copied to the clipboard

Procedure :	Click on the COPY SHARE LINK button
--------------------	-------------------------------------

Test Case 23: User should be able to add comments in their itinerary

Number :	23
Name :	User should be able to add comments in their itinerary
Description :	User should be able to add comments in their itinerary
Initial Conditions :	User must must be logged and itinerary should be present
Input Data :	N/a
Specifications :	Itinerary should have the comments that were added.
Procedure :	Write the comment in the text box on the itinerary page and the click SEND COMMENT

Test Case 24: User should be able to view shared itinerary

Number :	24
Name :	User should be able to view shared itinerary
Description :	User should be able to view shared itinerary
Initial Conditions :	User should have the shared link, doesn't necessarily they will have to be logged in
Input Data :	N/a
Specifications :	Itinerary screen should pop up but should not be able to make any comments
Procedure :	Just hit the shared Itinerary URL

Test Case 25: User should be able to comment on the shared itinerary

Number :	25
Name :	User should be able to comment on the shared itinerary
Description :	User should be able to comment on the shared itinerary
Initial Conditions :	User must must be logged and have the shared itinerary URL
Input Data :	N/a
Specifications :	Itinerary screen should pop up and should be able to make any comments, visible on the page.
Procedure :	Just hit the shared Itinerary URL and make sure the user is logged in.

Test Case 26: User should be able to search for places

Number :	26
-----------------	----

Name :	User should be able to search for places
Description :	Users should be able to search for places on the home screen.
Initial Conditions :	N/A
Input Data :	N/a
Specifications :	The places should pop up upon selection
Procedure :	The user needs to type the name of the place they want to explore on the search bar

Test Case 27: Autocompletion should take place for search bar

Number :	27
Name :	Autocompletion should take place for search bar
Description :	Autocompletion should happen once user types on the search bar
Initial Conditions :	N/A
Input Data :	N/a
Specifications :	Drop-down option with relevant suggestions must be visible when trying to search for places
Procedure :	The user needs to type the name of the place they want to explore on the search bar

Test Case 28: Google OAuth Sign In and sign up

Number :	28
Name :	Google O Auth Sign in sign up
Description :	User should be able to sign in or sign up using google credentials
Initial Conditions:	N/A
Input Data :	Google id password
Specifications :	User can login once successful sign up or sign in via google platform
Procedure :	Enter details on to google platform, backend gives call back to google and process authentication verification with googleid and gives jwt token.

Test Case 29: Make booking for flights, accommodations, and activities.

Number :	29
Name :	Booking flights, accommodations, and activities.

Description :	User should be able to book.
Initial Conditions:	N/A
Input Data :	Enter details for booking and perform selections
Specifications :	User can add the booked informations into itinerary.
Procedure :	N/a

Test Case 30: Reset password- Old password check

Number :	30
Name :	Reset password old password check
Description :	User should be prompted to reenter password if its matches with old.
Initial Conditions :	Old Password must exist
Input Data :	Enter the new password
Specifications :	User once entered new password can login into the system using new one.
Procedure :	Enter password that matches the the email of user.

Test Case 31: List hotels

Number :	31
Name :	List Hotels
Description :	User should be able to list hotels
Initial Conditions:	Backend should have hotels in database
Input Data :	Selections
Specifications :	User should be able to see the list of hotels available.
Procedure :	N/A

Test Case 32: Users can filter on hotels’ ratings and price

Number :	32
Name :	Users can filter on hotels’ ratings and price
Description :	Users can filter on hotels’ ratings and price

Initial Conditions:	N/A
Input Data :	N/A
Specifications :	Users should be able to get filtered data from which they can choose.
Procedure :	Click on the filters on the side-bar of the hotels page

Test Case 33: Users can leave reviews and ratings

Number :	33
Name :	Users can leave reviews and ratings
Description :	Users can leave reviews and ratings on the destinations on the homepage
Initial Conditions:	User must be logged in
Input Data :	Review in the form of text can be inserted for a destination
Specifications :	The ratings gets stored and can be seen by other users who wish to checkout the place
Procedure :	Login and press on the “leave review” icon on the destinations on the home screen

Test Case 34: Itinerary gets arranged according to day and location

Number :	34
Name :	Itinerary gets arranged according to day and location
Description :	Itinerary gets arranged according to day and location
Initial Conditions:	User must be logged in and items must be present in the itinerary
Input Data :	N/A
Specifications :	Users can now see their itinerary day wise and location wise
Procedure :	Add items to the itinerary (hotels, flights, activities etc)

Test Case 35: Currency Conversion

Number :	35
Name :	Currency conversion is now possible
Description :	Currency conversion is now possible
Initial Conditions:	N/A

Input Data :	The currency and the amount that we want to convert from and the currency that we want to convert it to.
Specifications :	Users can now make plans for international travel and budget their expenses appropriately
Procedure :	Navigate to the Convert page on the nav bar and select the currency from the dropdown

Test Case 36: Duo Authentication

Number :	36
Name :	Duo Authentication (MFA) - authentication
Description :	Duo Authentication (MFA) - authentication
Initial Conditions:	The user should have logged in with their correct credentials first
Input Data :	Enter the username and password
Specifications :	More stronger authentication
Procedure :	Accept the duo push received on the personal device.

3. Revision History

Revision	Date	Change Description
Version 1	18th Feb 2024	Initial state of the service and the tests done.
Version 2	3rd March 2024	Forgot password, API testing and Itinerary page test done.
Version 3	24th March 2024	Added some more testing of the Itinerary page functionality, Search functionality and the auto-completion on the home screen (TC-18 to TC-27). Minute changes on the features.
Version 4	7th April 2024	Added Google Oauth and booking, listing hotels.
Version 5	21st April 2024	Added testing on hotels, itinerary arrangement, currency conversion, ratings and reviews and duo authentication.

