IT314 – Software Engineering
G27 Flight Booking System
Acceptance Testing

Document Purpose: This document outlines the acceptance criteria, scenarios and results for our Flight Booking System, SkyLynx. The purpose is to validate the system against the requirements and confirm that it is ready for production use.

Scope of Testing:

- **Inclusion Criteria:**
  Acceptance testing will cover all features and functionalities outlined in the requirements document

- **Exclusion Criteria:**
  The following areas are excluded from acceptance testing, as they have been validated in other testing phases: Performance Testing (Load and Stress Testing) and Security Testing

Features Covered:

1. User Registration
2. User Login
3. View User Profile
4. Edit Profile Information
5. Flight Search Functionality
6. Flight Sorting and Filtering
7. Flight Booking Process
8. Booking Payment and Confirmation

Testing Approach:

The acceptance testing will adopt a black-box testing methodology. This approach will ensure that the software meets the expectations of end-users by validating its behavior against the specified requirements.

Test Data: Test data will be carefully crafted to simulate real world user interactions including both valid and invalid inputs.

Testing Environment:
Devices: Laptop,Mobile
Browsers: Google Chrome,Edge
Operating Systems: Windows,macOS
Database: MongoDB

TESTING:

Feature 1: User Registration
Allows a user to register on the website to create an account and gain access to login functionality.

Success Scenario:
If all details (Name, Email, Password and Confirm Password) entered by the user are valid then the user is successfully registered and redirected to the login page.

Failure Scenarios:
Error message is displayed for the following cases:

1. Email format is invalid
2. Username already exists in the system
3. Email already exists in the system

4. Confirm password field does not match with the password field
5. Password does not meet length requirement

Test Cases:

Valid Input Tests:

1. Enter valid details for the registration and click Sign up button.

Invalid Input Tests:

1. Use an existing username or email.
2. Input an invalid email address.
3. Enter a password shorter than the minimum required length.
4. Enter a Confirm Password that does not match the Password field.
5. Leave some or all fields blank.
6. Check whether the website is redirected to the login page after clicking ‘Sign Up’.

Expected Outputs:

1. Display error messages for the following:
   - Fields left empty. (“*All fields are required*”)
   - Invalid format of email . (“*Please enter valid email*”)
   - Username or email already in use. (“*Email already in use*”)
   - Passwords do not meet the criteria (length, content). (“*password length must be at least x characters long*”)

- Mismatch between Password and Confirm Password.(“*Passwords do not match*”)

2. Redirect to the login page upon successful registration with valid details.

Acceptance Criteria:

This feature meets the acceptance criteria.

- Users can register with valid details and are redirected to the login page with a success message.
- Clear error messages are displayed for invalid inputs.
- User details are securely stored, avoiding duplicates, and all test cases pass without issues.

This feature meets the acceptance criteria.

Feature 2: User Login

Allows users to log in to their profiles from different devices to access the website and its features.

Success Scenario:

If all the details (email,password) entered by the user are correct, successfully authenticated and the user is redirected to the home page.

Failure Scenarios:

1. If the email does not exist or is not registered in the system then the error message displayed: “*Auth failed: email or password is wrong*”.

2. If the password entered does not match with the email then the error message displayed: *“Auth failed: email or password is wrong”*.
3. If the password does not meet the minimum length of the password then displays an error message.
4. Either email or password or both fields left blank.

Test Cases:

Valid Input Tests:

1. Enter a valid registered email and the corresponding correct password.
2. Login from a different device using valid credentials.

Invalid Input Tests:

1. Enter an email that is not registered in the system.
2. Input a correct registered email but an incorrect password.
3. Input a password shorter than the minimum length requirement.
4. Leave the email or password blank and attempt to log in.

Expected Outputs:

Display error messages for the following:

- *"Auth: failed email or password is wrong"* if the email is not registered.
- *"Auth: failed email or password is wrong"* if the password does not match the email.

- Prompt an error (*“password length must be at least x characters long”*) if either the email or password field is left blank.
- Prompt an error (*“Email and password are required”*) if the password does not meet the minimum length of the password.

Successful login redirects the user to the Home page when valid credentials are provided.

Acceptance Criteria:

This feature meets the acceptance criteria.

- Users can successfully log in with valid credentials from any device.
- Appropriate error messages are displayed for invalid or incomplete inputs.
- Users are redirected to the home page upon successful authentication.

This feature meets the acceptance criteria.

Feature 3: View User Profile

Allows logged-in users to view their profile information on the website, including personal and account related details.

Success Scenario:

After successfully logging in, users can navigate to their profile page and view details such as their name, email, phone number, and other relevant details.

Test Cases:

Valid Scenario:

Login with valid credentials and navigate to profile page (All profile information including Username, Email, Phone Number, Booking History and Address should be visible)

Invalid Scenario:

Attempt to directly access the profile page using URL without logging in

Expected Outputs:

- For logged-in users: Profile details are displayed correctly, including fields like name, email, address and phone number. The page loads seamlessly with a user-friendly interface.
- The user is redirected to the login page with a message indicating the need to login to access their profile. (Invalid scenario of testcase)

Acceptance Criteria:

This feature meets the acceptance criteria.

- Logged-in users can view their profile information accurately on the profile page.
- The interface remains user-friendly and functional.

This feature meets the acceptance criteria.

Feature 4: Edit User Profile

Allows users to modify their profile information, including name, email, address, phone number and other personal details, ensuring the account reflects accurate and updated information.

Success Scenario:

- The user successfully updates their profile details, and the changes are saved in the database
- A confirmation message, *“Profile updated successfully”*, is displayed upon completion.

Failure Scenarios:

1. Required fields left blank
2. Invalid Phone Number Format
3. Invalid Email Format

Test Cases:

Valid Input Tests:

1. Update all fields with valid information and save changes.
2. Modify specific fields (Ex: Phone Number,Address) while leaving others unchanged.

Invalid Input Tests:

1. Leave one or more required fields empty and attempt to save.
2. Enter a phone number with less than 10 digits and attempt to save.

3. Enter an invalid email format (Ex: Without @) and attempt to save.

Expected Outputs:

- For Valid Inputs:
    - Profile Details are updated successfully in the database.
    - A success message is displayed: *“Profile Updated Successfully”*
- For Invalid Inputs: Error messages displayed are
    - *“This <name of field> cannot be empty”*
    - *“Phone number must be a 10 digit number”*
    - *“Please enter a valid email address”*

Acceptance Criteria:

This feature meets the acceptance criteria.

- Users can modify their profile details without issues.
- The system validates inputs appropriately and displays clear error messages for invalid data.
- Changes to the profile are saved successfully, and a confirmation message is shown.

This feature meets the acceptance criteria.

Feature 5: Flight Search Functionality

Users should be able to search for flights by entering relevant details such as origin, destination, departure date. Search results should display available flights matching the criteria.

Success Scenarios:

1. The user enters valid search criteria (Eg: valid origin, destination, and dates) and receives a list of available flights.
2. Results are displayed with flight details such as airline name, flight number, departure and arrival times, duration, price, and availability.
3. If no flights are available for the entered criteria, message displayed is: *"No flights found"*

Failure Scenarios:

1. Empty Fields: If required fields (Eg: origin, destination) are left blank, display an error message.
2. Invalid Date Selection: If a user enters invalid dates (Eg: departure date before today or return date earlier than the departure date.), display an error message

Test Cases:

Valid Input Tests:

1. Enter a valid origin, destination, departure date, and passenger count for a one-way trip.

Invalid Input Tests:

1. Leave one or more required fields empty and attempt to search.
2. Enter a departure date in the past and try searching.
3. Simulate a system error during the search process.

Expected Outputs:

For Valid Inputs:

- Display a list of matching flights with relevant details:
    - Airline name
    - Flight number
    - Departure and arrival times
    - Duration
    - Price
    - Class:Economy/Business

For Invalid Inputs:

- Provide clear error messages for incorrect or incomplete input such as:
    - *“Please enter both origin and destination cities”*

Acceptance Criteria:

This feature is considered complete when:

- Users can successfully search for flights with valid inputs.
- Flight search results are accurate and displayed clearly.

- The user interface is responsive and intuitive, allowing seamless interaction across devices.

This feature meets the acceptance criteria.

Feature 6: Flight Sorting and Filtering
Users should be able to sort and filter flight search results to refine their options. Sorting can be done based on parameters like price,duration,etc. Filters include airlines,flight class (Economy/Business), and stops.

Success Scenarios:

1. Users can sort flight results by selected criteria such as:
   a. Price (Low to High, High to Low)
   b. Duration (Lowest to Highest, Highest to Lowest)
2. Users can apply filters to narrow down results based on:
   a. Airline Preferences
   b. Flight Class (Business/Economy)
   c. Stops in between the flight (One stop,No stops,etc)
3. Results update based upon the sort and filter criteria

Failure Scenarios:

1. Sorting errors: Sorting options do not reorder the flights correctly
2. No results received after filtering

3. System errors: Failure in applying filters due to server issues

Test Cases:

Valid Input Tests:

1. Sort flights by price and duration individually.
2. Filter results by airline and verify that only selected airlines are displayed.
3. Combine multiple filters (Ex:Class and No stops) and check for accuracy.

Invalid Input Tests:

1. Applying filters that exclude all flights.
2. Test sorting functionality when no flights are available.

Expected Outputs:

- **For Valid Inputs:**
    - Results are updated based on required criteria.
    - Sorting options reorder the flights correctly.
    - Filters refine the search based on user selections.
- **For Invalid Inputs:**
    - Display appropriate message when no flights match the filter(“*No flights match your criteria please modify your filters*”)
    - Handle system errors carefully.

Acceptance criteria:

This feature is considered complete when:

- Sorting and filtering functionality works as intended across all parameters.
- User inputs for sorting and filtering produce accurate results.
- Clear and user-friendly error messages are displayed for invalid inputs or system issues.
- The interface is responsive and performs seamlessly across devices.

This feature does not meet the acceptance criteria as it does not display error messages when flights are not found after sorting and filtering.

Feature 7: Flight Booking Process

This feature enables users to book a flight by adding passenger details and selecting optional extras to enhance their travel experience. The flow includes entering mandatory passenger details, selecting saved passengers, and customizing the booking with addons like baggage, seats, meals, and insurance.

Success Scenarios:

- Users can successfully:
  - Add passenger details (including designation, name, date of birth, and contact information).
  - Select saved passenger profiles for convenience.
  - Proceed to the extras page after filling out mandatory fields.

- Optional extras (baggage, seats, meals, insurance, passenger assistance, and airport services) are selectable, with prices displayed accurately.
- The total cost, including addons, is calculated and displayed correctly.
- Users receive a booking confirmation with all selected details and a reference number.

Failure Scenarios:

- If mandatory fields are not filled (Eg: name or phone number) an error message is displayed: *"Please complete all required fields to continue."*
- If invalid data is entered (Eg: Invalid phone number), an error message is displayed: *"Please enter a valid phone number."*
- If an add-on is selected but cannot be applied (Eg: due to system issues), display an error: *"Unable to add selected extras. Please try again later."*
- If the total price calculation fails, display an error message: *"Total price could not be updated. Please refresh the page."*

Test Cases:

Valid Input Tests

- Enter All Mandatory Details: Input valid name, email, phone, DOB, and proceed to extras page.
- Select Saved Passenger: Choose an existing saved passenger and auto-fill details.
- Select Addons: Choose add ons like baggage, seats, meals, etc.

- Deselect Addons: Remove selected add ons and update the price.
- Complete Booking: Enter valid details, select addons, and proceed to payment.

Invalid Input Tests

- Leave required fields blank.
- Input incorrect DOB format (Eg: “dd-mm-yyyy”).
- Enter an invalid email format.
- Enter a phone number that’s not 10 digits.
- Fail to select an addon due to a system issue.
- Price fails to update after addon selection.

Expected Outputs:

- For Valid Input Test Cases
  - Enter All Mandatory Details: Redirect to extras page with correct details.
  - Select Saved Passenger: Auto-fill saved details and continue.
  - Select Addons: Add selected addons to total price.
  - Deselect Addons: Deduct the price of removed addons.
  - Complete Booking: Display booking confirmation with details and reference number.
- For Invalid Input Test Cases errors displayed will be:
  - All fields required.
  - Enter a valid DOB.
  - Enter a valid email address.
  - Enter a valid 10-digit phone number.

    - Unable to add addon. Try again.
    - Price calculation failed. Refresh.

Acceptance criteria:

This feature is considered complete when:

- The system handles valid inputs seamlessly, allowing progression through the booking flow.
- Invalid inputs trigger clear error messages, preventing further actions until corrected.
- Total cost calculations and add-on selections function accurately, providing a smooth user experience.

This feature meets the acceptance criteria.

Feature 8: Booking Payment and Confirmation

This feature allows users to complete their booking payment using various payment methods (Credit Card, UPI) and confirm their booking. The payment process includes entering card details or selecting UPI as a payment method. The system confirms successful payments and navigates to the booking confirmation page.

Success Scenarios:

- Valid Payment Method Selection:
    - Users can select a valid payment method (Credit Card or UPI).
    - Users successfully complete the payment process and are redirected to the booking confirmation page.
- Successful Payment

  - Users successfully enter valid payment details and complete the transaction without errors.

Failure Scenarios:

- Invalid Payment Method
  - Users select a payment method but fail to provide valid details (Eg: an invalid credit card number or expired card).
  - The payment fails, and an error message is displayed.
- Incomplete Payment Details
  - Users do not enter all the required fields (Eg: card number, CVV, or postal code) and are unable to submit the payment.
  - The system displays an error message requesting the missing information.
- Payment Failure
  - The payment fails due to incorrect details, insufficient funds, or payment gateway errors.
  - Users are notified with an error message.

Test Cases:

Valid Input Tests:

- Credit Card Payment: Enter valid credit card details (card number, expiry date, CVV, and postal code).
- UPI Payment: Choose UPI as the payment method, enter valid UPI details, and complete the payment.
- Successful Transaction with Card: Enter valid credit card information and submit payment.

- Successful Transaction with UPI: Enter valid UPI details and complete the payment.

Invalid Input Tests:

- Invalid Credit Card Number: Enter an invalid card number.
- Expired Credit Card:Enter an expired credit card.
- Incorrect CVV Code: Enter an incorrect CVV code.
- Invalid Postal Code: Enter an incorrect postal code.
- UPI Payment Failure: Enter invalid UPI details or encounter a UPI payment failure.
- Incomplete Payment Details: Leave required payment fields (card number, CVV, etc.) empty and attempt to submit.
- Payment Method Not Selected: Attempt to submit the form without selecting a payment method.

Expected Outputs:

- **For Valid Input Test Cases**
  - Users are redirected to the booking confirmation page.
  - The payment is processed successfully, and a confirmation message is displayed with booking details.
- **For Invalid Input Test Cases**
  - Appropriate error messages are displayed based on the invalid input, such as invalid card number, expired card, incorrect CVV, invalid UPI details, and incomplete fields.

Acceptance Criteria:

This feature is considered complete when:

- The system must process payments successfully for valid credit card and UPI details, and redirect the user to the booking confirmation page.
- The system must handle errors gracefully, displaying appropriate messages for invalid or incomplete payment details.
- The payment page must be intuitive, responsive, and provide clear error messages for invalid inputs.
- After unsuccessful payment, users should be notified that the payment was unsuccessful and should be able to try again for payment.
- After successful payment, users should be redirected to the booking confirmation page with all relevant booking details.

This feature does not meet the acceptance criteria as the payment failure scenario has not been implemented.