

Project Acceptance Test Manual

Joel Milligan

Contents

1 Objectives	2
2 Test Summary	2
3 Testing Strategy	2
4 Front End Unit Tests	3
4.1 Test Procedure	3
4.2 Expected Results	3
5 Back End Unit Tests	3
5.1 Test Procedure	3
5.2 Expected Results	3
6 Accessible Website	4
6.1 Test Procedure	4
6.2 Expected Result	4
7 Applying for a Booking	4
7.1 Test Procedure	4
7.2 Expected Result	4
8 Approving an Application	5
8.1 Test Procedure	5
8.2 Expected Result	5
9 Declining an Application	5
9.1 Test Procedure	5
9.2 Expected Result	5
10 PDF Generation	6
10.1 Test Procedure	6
10.2 Expected Result	6

1 Objectives

In this section, describe the purpose of this document. Forecast the testing strategy, the unit tests, integration test and system tests listed in this document.

2 Test Summary

In this section, describe the functions of the system tested in this document.
Refer to Requirements Analysis Document and Problem Statement.

3 Testing Strategy

In this section, define the subsystem or subsystems to be tested, the system integration strategy and how, where, when, and by whom the tests will be conducted.

You may want to include drawings depicting relationships among the major classes of the subsystem or the subsystem decomposition, if you feel this is appropriate.

4 Front End Unit Tests

All front end JavaScript unit tests, using the `jest` framework, should pass.

4.1 Test Procedure

1. Open your CLI to the root of the repository.
2. Run the command `yarn test`.

4.2 Expected Results

- All unit tests pass.
- At least 50% test coverage.

The reason for the relatively low coverage is that most tests for the front end will be integration tests.

5 Back End Unit Tests

All back end Python unit tests, using `pytest`, should pass.

5.1 Test Procedure

1. Open your CLI to the root of the repository.
2. Run the command `yarn test`.

5.2 Expected Results

- All unit tests pass.
- At least 75% test coverage.

6 Accessible Website

The booking part of the website should be accessible to anyone with a modern web browser.

6.1 Test Procedure

1. Navigate to the main website URL in either Google Chrome or Mozilla Firefox.

6.2 Expected Result

The initial page for booking will be shown.

7 Applying for a Booking

Once a user has access to the booking site, they can create a booking application.

7.1 Test Procedure

1. Choose a date.
2. Choose a carpark.
3. Choose a bay and time.
4. Review the booking details before submitting.
5. Submit application.

7.2 Expected Result

- Booking details before submitting should be correct.
- The user should be sent an email confirming the application is submitted.
- The admin team should get an email notifying them of a new application that needs to be actioned.

8 Approving an Application

An admin user should be able to approve an existing application.

8.1 Test Procedure

1. As a normal user, create a booking application.
2. As an admin, navigate to the approvals page.
3. Click approve on the application.

8.2 Expected Result

- User should be automatically emailed to notify them that their application was approved.
- The application in the admin panel should now show as approved.

9 Declining an Application

An admin user should be able to decline an existing application.

9.1 Test Procedure

1. As a normal user, create a booking application.
2. As an admin, navigate to the approvals page.
3. Click decline on the application.

9.2 Expected Result

- User should be automatically emailed to notify them that their application was declined.
- The application in the admin panel should now show as declined.

10 PDF Generation

A PDF containing details about the booking should be automatically generated when a booking is approved.

10.1 Test Procedure

1. Apply for a booking.
2. An admin approves the booking.

10.2 Expected Result

- A PDF should be available for download or viewing by the admin upon approval of the booking.
- The client should be sent an email with the same PDF attached.