

Project Acceptance Test Manual

Joel Milligan

August 17, 2021

Contents

1	Unit Tests	2
1.1	Front End Tests	2
1.2	Back End Tests	2
2	Code Linting	3
3	CI/CD	3
4	Creating Bookings	4
4.1	Accessible Website	4
4.2	Applying for a Booking	4
4.3	PDF Generation	5
5	Admin Functions	6
5.1	Approving an Application	6
5.2	Declining an Application	6
5.3	Creating a Car Park and Bays	7
5.4	Editing a Car Park	7
5.5	Deleting a Car Park	7
5.6	Creating New Admins	8
5.7	Editing Admins	8
5.8	Deleting Admins	8

1 Unit Tests

1.1 Front End Tests

All front end JavaScript tests, using `jest` framework and `react testing library`, should pass. Combination of code unit tests and integration test will be used. This makes sure that all components themselves work and the booking system works correctly. Snapshot testing are used as well, to ensure that the DOM and HTML are correct, and any updates that can't be inspected in code shows in the snapshots correctly.

Test Procedure

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

Expected Results

- All unit tests pass.
- At least 90% test coverage.

1.2 Back End Tests

All back end Python tests, using django's built in testing library, should pass. Combination of code unit tests and integration test will be used. This includes testing endpoints, models and serializers to make sure they work as expected. Backend will use `sqlite3` as a dummy database to test creating, reading, inserting and deleting data. This is to make it easier for code integration, rather than hosting a `mysql` database test server.

Test Procedure

1. Open your CLI to the root of the repository.
2. Run the command `python manage.py test`

Expected Results

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

2 Code Linting

Code linting will be used in both frontend and backend to check for errors and ensure consistent formatting. For frontend we use a combination of `prettier` and `eslint`, and for backend `pycodestyle`. These are industry standards for linting.

Fontend Lint Test

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

Backend Lint Test

1. Open your CLI to the root of the repository.
2. Run the command `pycodestyle app/`

Expected Results

- No linting errors.

3 CI/CD

Continuous integration and deployment with GitHub action is an automatic way do testing and make sure the product can build correctly. Every change to the code base will need to be reviewed, and be put into a pipeline to check all code test and linting.

4 Creating Bookings

4.1 Accessible Website

The booking part of the website should be accessible to anyone with a modern web browser and the two main operating system (Mac and Windows).

Test Procedure

1. Navigate to the main website URL using different modern browsers and operating systems.

Expected Result

- The initial page for booking will be shown.

4.2 Applying for a Booking

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

Test Procedure

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

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 approved.

4.3 PDF Generation

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

Test Procedure

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

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.

5 Admin Functions

5.1 Approving an Application

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

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.

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.

5.2 Declining an Application

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

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.

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.

5.3 Creating a Car Park and Bays

An admin user should be able to create a new carpark, which will then allow normal users to book for it.

Test Procedure

1. Login as an admin
2. Navigate to the car parks section of the interface and click add.
3. Enter details about the car park and add parking bays.

Expected Result

- As a normal user creating a booking, the new carpark and bays should now be options.
- The details entered during the creation of the car park and bays should show to the user.

5.4 Editing a Car Park

An admin use should be able to edit existing car parks.

Test Procedure

1. Login as an admin
2. Navigate to the car parks section of the interface.
3. Click edit on one of the existing car parks.
4. Make changes in the editing page that appears.

Expected Result

- The changes made should be reflected to normal users who view the car park after the edit.

5.5 Deleting a Car Park

An admin user should be able to delete existing car parks.

Test Procedure

1. Login as an admin.
2. Navigate to the car parks section of the interface.
3. Click delete on one of the existing car parks.

Expected Result

- The car park should no longer show as an option to normal users.

5.6 Creating New Admins

An existing admin user should be able to create a new admin account.

Test Procedure

1. Login as an admin
2. Navigate to the users section of the interface and click add.
3. Enter the account details and confirm.

Expected Result

- It should now be possible to login with the new admin account.

5.7 Editing Admins

An existing admin user should be able to edit other admin accounts.

Test Procedure

1. Login as an admin
2. Navigate to the users section of the interface and click edit on an account.
3. Alter the account details and confirm.

Expected Result

- The account details should now be changed.

5.8 Deleting Admins

An existing admin user should be able to delete other admin accounts.

Test Procedure

1. Login as an admin
2. Navigate to the users section of the interface and click delete on an account.

Expected Result

- The account should no longer be able to be logged into.