

Documentation

Meeting Minutes

P

Meeting Arrangements

P

Lean Testing

P

RESOURCES

SETTING UP HOMESTEAD

13

3

Dusk on Laravel 5.4

Project Installation

Deliverables

Identify User Stories

Define a series of Acceptance Tests

Basic user documentation

Class diagram

1

Database diagram

1

Test Logs

1

Release Notes

Project Post-Mortem

Project pitch video

User Stories

UC_01: User wants to login to the system

2

P

UC_02: Customer wants to register an account

UC_03: Customer wants to make bookings

P

UC_04: Customer wants to view available slots

P

UC_05: Business owner adds a new employee

P

UC_06: Business owner rosters employees

P

UC_07: Business owner checks booking summaries

P

UC_08: Business owner checks employee's availability for the next 7 days

P

UC_09: Business owner makes booking for a registered customer

P

UC_10: User updates his/her profile

P

Product Backlog

Landing Page

User Registration

Dashboard (Customer)

Dashboard (Business Owner)


Database Design

Sprint Backlog

In Progress

Release Notes

🕒 May 25





P

Project Post-Mortem

🕒 May 23

💬 2



P

Project pitch video

May 25

P

Blocked

Completed (Part C)

Business Registration

1

11/11

Name for the app - BookMe

4

P

Allow users to update their profiles

2

Allow business to customise GUI

1

Allow customer to view booking summary

Lock account after 5 times wrong password

1

Allow business owner and customer to cancel booking

paginate booking summary page

Validate name and address

Improve button designs and hover transitions

Improve table design

Redesign dashboard

Provide automated builds

1

P

More test cases

P

Further user stories

P

Opt for scaled marks for assessment 3

🕒 May 16

Completed (Part B)

Paginate availability page

🐼

Refactor and Comment code

✅ 4/4

🐼

Roster issues

🐼

Business Owner create bookings for customer using username/email

🐼

Customer create bookings

✅ 2/2

🐼

Implement authentication using innate Auth for security check

✅ 4/4

🐼

Write new user stories

P

Tests cases new functionality

P

Write project documentation in README file

👁

🐧

Gather all required deliverables for submission

👁

🐧

Set datepicker for making bookings 1 day ahead of current date

👁

🐧

Add working time should check employee's availability

1

3/3

Add a LOGOUT button on top right

Upload individual logging files to Google Drive

2

P

Booking availability should take into account num_of_slots

2

Add/update staff working days/times later

Missing businessID in getBookingsByDate in BookingController

3/3

Redirect to login page whenever try to access pages without logging

1

P

Add config: default slot period (30 min in this case)

1

Change table bookings: end_time -> num_of_slots

Add column emp_id to table bookings

Define activities that take different times

1

Employee no longer has role but the type of activity in charge

2/2

SOLVE: how to keep track of tables

1

Order booking summary by date

Merging old changes from master to dev branch

Logging capability

1

A page that allow business owner to add bookings for customers (using username/email)

Implemented session

P

Refactored tests

P

Completed (Part A)

Integrate business owner functionalities

1

7/7

Integrate customer functionalities

1/1

Implement go back to dashboard button for business owner

Implement showing all workers' availability for the next 7 days (view by business owner)

Present user stories in a PDF

Write up user documentation

P

Set up project from clean install

P

Design class diagram

P

Update Roster Tests cases

P

Design business owner dashboard

Design booking summary page (business owner)

2

Design add new staff page (business owner)

Design view roster page (business owner)

Design add staff working time page (business owner)

4

Design view available slots page/section (customer)

2

Write seeders for employees and rosters tables

2/2

Build test units based on user story "Customer wants to register for an account"

1

Build test units based on user story "user wants to login to the system"

P

set up Laravel environment

3/3

P

set up git environment

3/3

P

Implement user registration

3/3

Implement user authentication

Design and implement database for customers, business owners and bookings

3

4/4

Build UI for login/registration page

4

Create brief that denotes user flow between pages

Create prototypes for customer and business dashboards

🕒 Mar 24

Add radio buttons for selecting between customer or business accounts on sign in form

Move login error messages to appropriate places

Improve authentication according to channel

Add customer details to booking summary page

Implement Booking related functions

Add seeders needed for demo wk5

✅ 3/3

Design customer dashboard

1

Modify user-type to be determined by account name or email

✅ 2/2

Design and implement database for employee and its working time/dates

P

Implement adding new employee (by business owner)

Implement adding working time/dates for employee for the next month (by business owner) -> rostering

Fix testing framework. Adopt LeanTesting

P

Tests for Employee related functions

P