Sprint 4 Review

Date: 15/10/2020

Sprint: 4

Scrum Master: Vincent Villaflores (s3728807)

Development Team: Meng Kheang Leng (s3704080), Chhayhy Kourn (s3699618), Hue Phuong Le (s3687477), William Bossen (s3658961)

Sprint Goals

Since this is the last sprint, the team will implement all the functionalities that are left to complete the full functionality of our system. This consists of: admins being able to view and change the available sessions, customers being able view and edit their information and password, as well as cancel their appointments. As well as workers being able to view their assigned hours and services. Additionally, we will be completing the sprint update, product backlog and refactoring. As well as automatic deployment using Kubernetes. As mentioned before, we will also update the product backlog accordingly based on the Definition of Done.

Status Overview

A table below shows the information of product backlog items will be discussed.

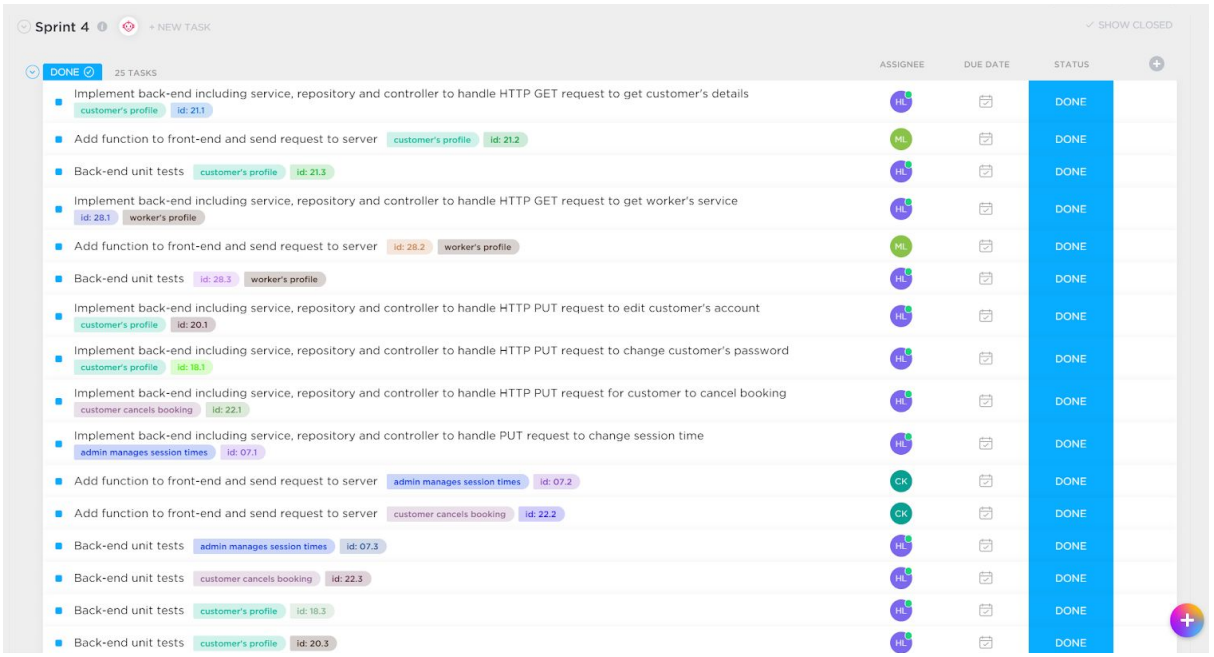
ID	Task name	User story	Status	Demo
7.1	Implement back-end including service, repository, controller & spring boot security to handle HTTP PUT request to change session times for admin	As an admin, I want to be able to change the available session times so that customers don't book appointments no worker can go to	Done	Yes
7.2	Add function to front-end and send request to server		Done	Yes
7.3	Back-end unit tests		Done	Yes
7.4	Front-end unit tests		Done	Yes
12.1	Implement back-end including service, repository, controller,	As an admin, I want to view all workers'	Done	Yes

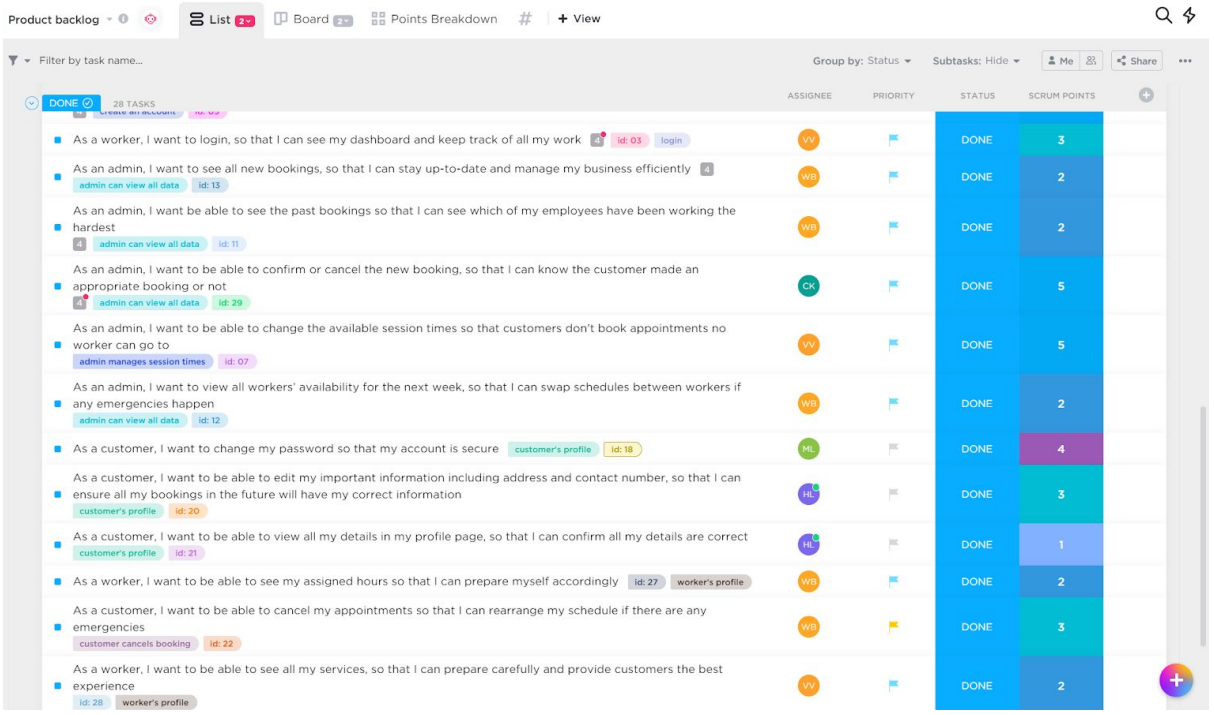
	security to handle HTTP GET request to view all worker	availability for the next week, so that I can swap schedules between workers if any emergencies happen		
12.2	Add function to front-end and send request to server		Done	Yes
12.3	Back-end unit tests		Done	Yes
12.4	Front-end unit tests		Done	Yes
18.1	Implement back-end including service, repository, controller, security to handle HTTP PUT request to change password for customer	As a customer, I want to change my password so that my account is secure	Done	Yes
18.2	Add function to front-end and send request to server		Done	Yes
18.3	Back-end unit tests		Done	Yes
18.4	Front-end unit tests		Done	Yes
20.1	Implement back-end including service, repository, controller, security to handle HTTP PUT request to change customer information	As a customer, I want to be able to edit my important information including address and contact number, so that I can ensure all my bookings in the future will have my correct information	Done	Yes
20.2	Add function to front-end and send request to server		Done	Yes
20.3	Back-end unit tests		Done	Yes
20.4	Front-end unit tests		Done	Yes
21.1	Implement back-end including service, repository, controller, security to handle HTTP GET request to view customer details	As a customer, I want to be able to view all my details in my profile page, so that I can confirm all my details are correct	Done	Yes
21.2	Add function to front-end and send request to server		Done	Yes
21.3	Back-end unit tests		Done	Yes
21.4	Front-end unit tests		Done	Yes
22.1	Implement back-end including service, repository and controller to handle HTTP GET request to view	As a worker, I want to be able to see my assigned hours so	Done	Yes

	assigned hour for worker	that I can prepare myself accordingly		
22.2	Add function to front-end and send request to server		Done	Yes
22.3	Back-end unit tests		Done	Yes
22.4	Front-end unit tests		Done	Yes
27.1	Implement back-end including service, repository and controller to handle HTTP PUT request to cancel booking for customer	As a customer, I want to be able to cancel my appointments so that I can rearrange my schedule if there are any emergencies	Done	Yes
27.2	Add function to front-end and send request to server		Done	Yes
27.3	Back-end unit tests		Done	Yes
27.4	Front-end unit tests		Done	Yes
28.1	Implement back-end including service, repository and controller to handle HTTP GET request to see all service for worker	As a worker, I want to be able to see all my services, so that I can prepare carefully and provide customers the best experience	Done	Yes
28.2	Add function to front-end and send request to server		Done	Yes
28.3	Back-end unit tests		Done	Yes
28.4	Front-end unit tests		Done	Yes
40	Automatic deployment using Kubernetes	N/A	Done	Yes

A screenshot of Clickup

We moved all tasks from Sprint 4 Backlog to “Done” and all PBIs from Product Backlog to “Done”.

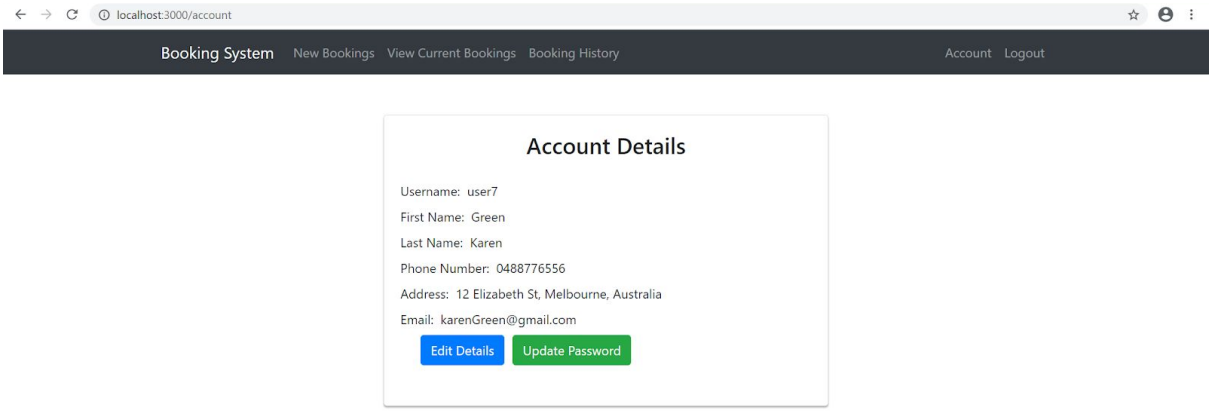




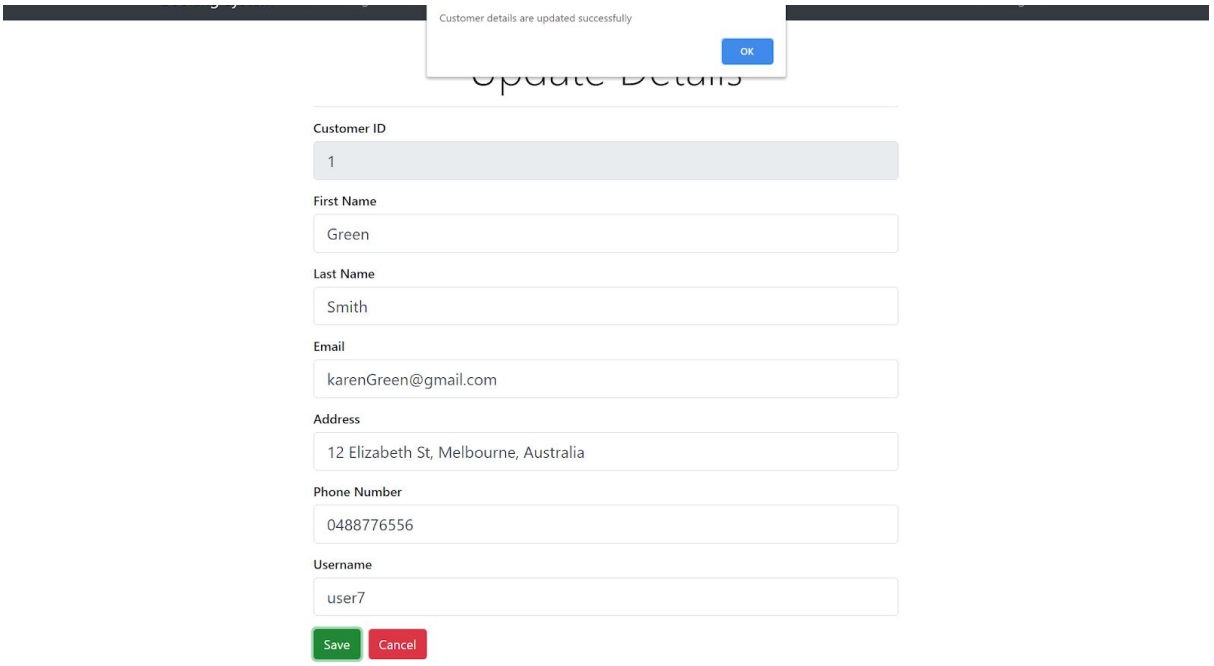
Screenshots

- **Customer Manages Profile**

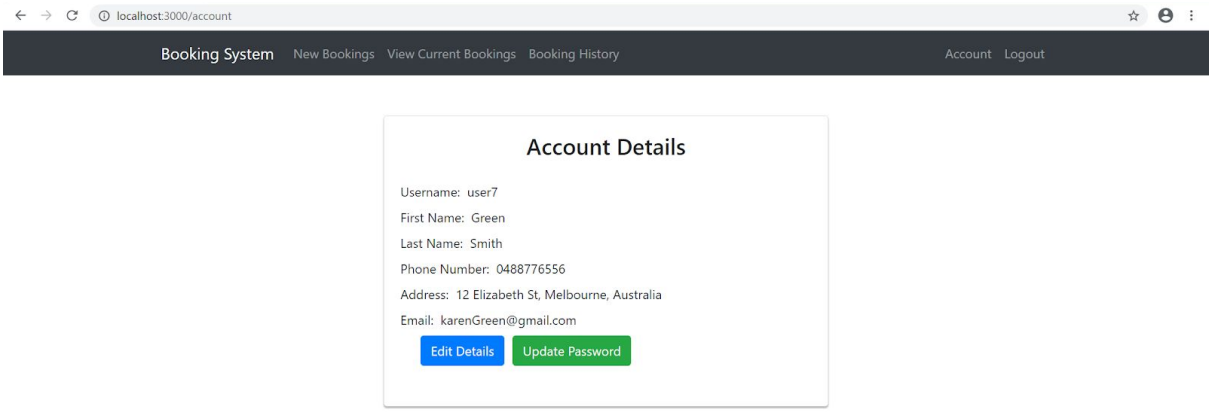
After logging in a customer profile and clicking on the “Account” button in the top navigation bar, the customer is redirected to a page displaying all their details with two buttons “Edit Details” and “Update Password”.



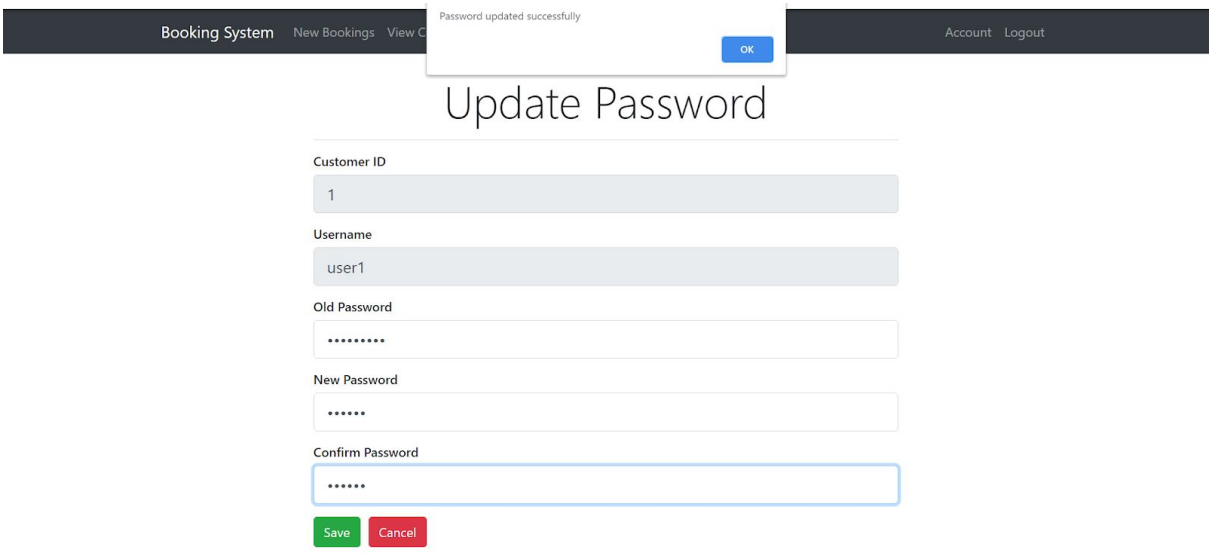
To update the account details, customers can click on the “Edit Details” button and he/she will be redirected to a page displaying a “Update Details” form containing all details which are editable. Customer then makes changes to the last name from “Karen” to “Smith” and clicks the “Save” button.



A message “Customer details are updated successfully” displays and the customer is redirected back to the account page with the updated last name.

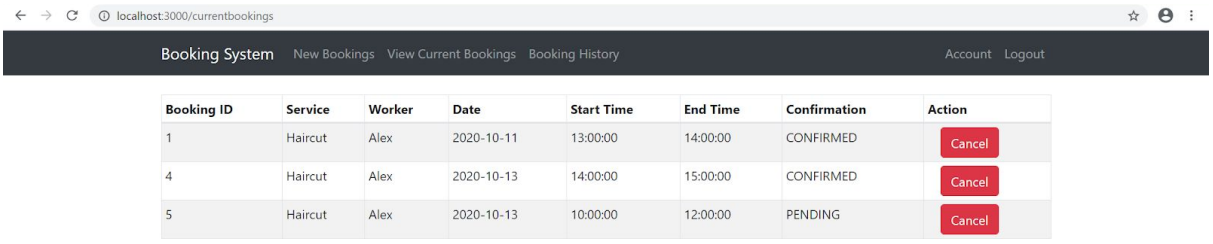


Now to change the password, the customer can click on the “Update Password” button and he/she is redirected to a page prompting the customer to enter old, new and confirm password. By entering all valid information and clicking on the “Save” button, the message “Password updated successfully” pops up to inform that the password is updated.

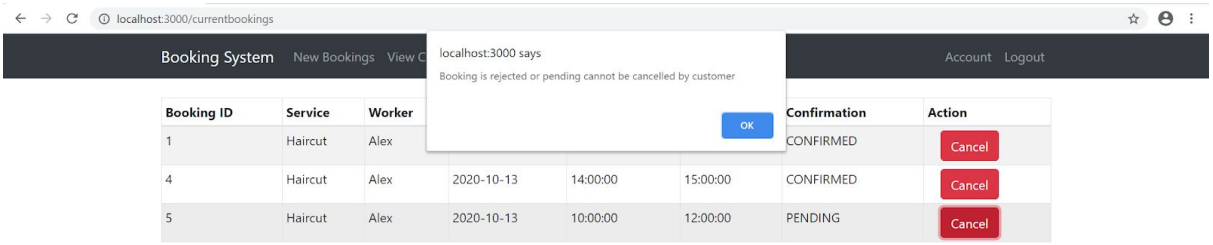


- **Customer Cancels Booking**

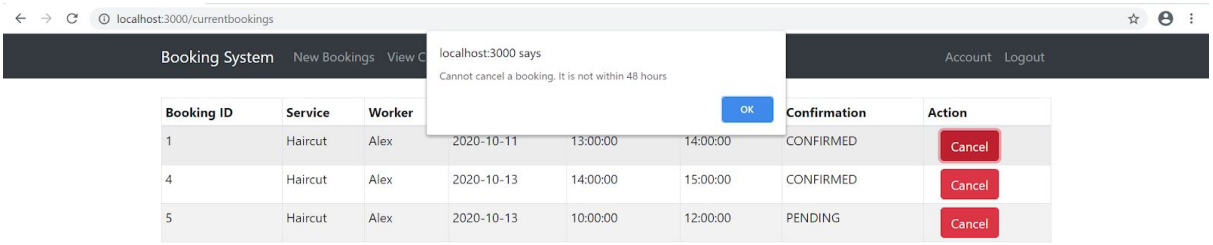
Now customers can click on the “Current Bookings” button and he/she will be redirected to a page displaying all the new bookings including the “Cancel” button.



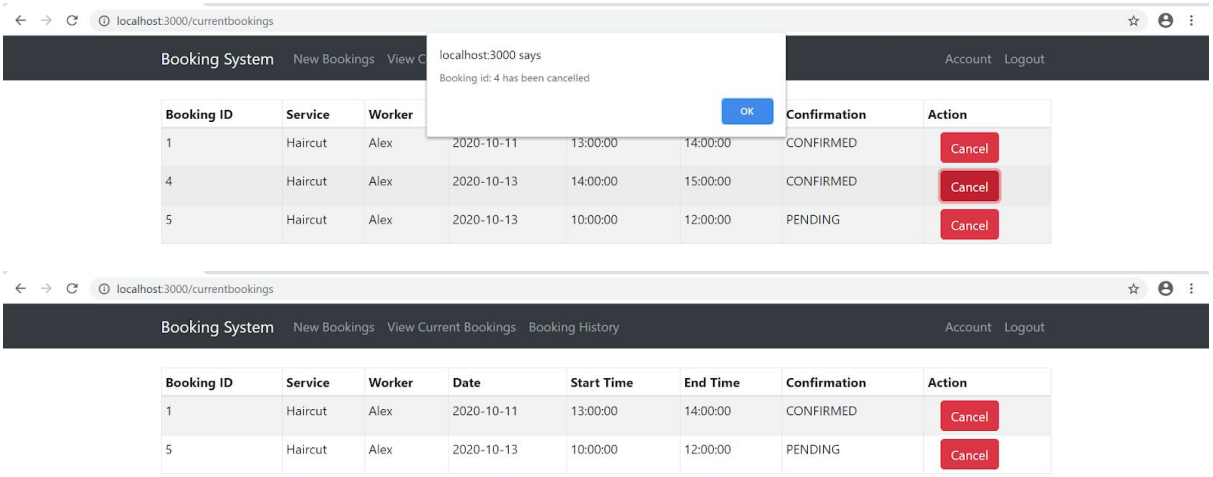
If they cancel a pending booking, a message to notify about the mistake will display.



Similar thing happens if they try to cancel a booking which is within 48 hours.

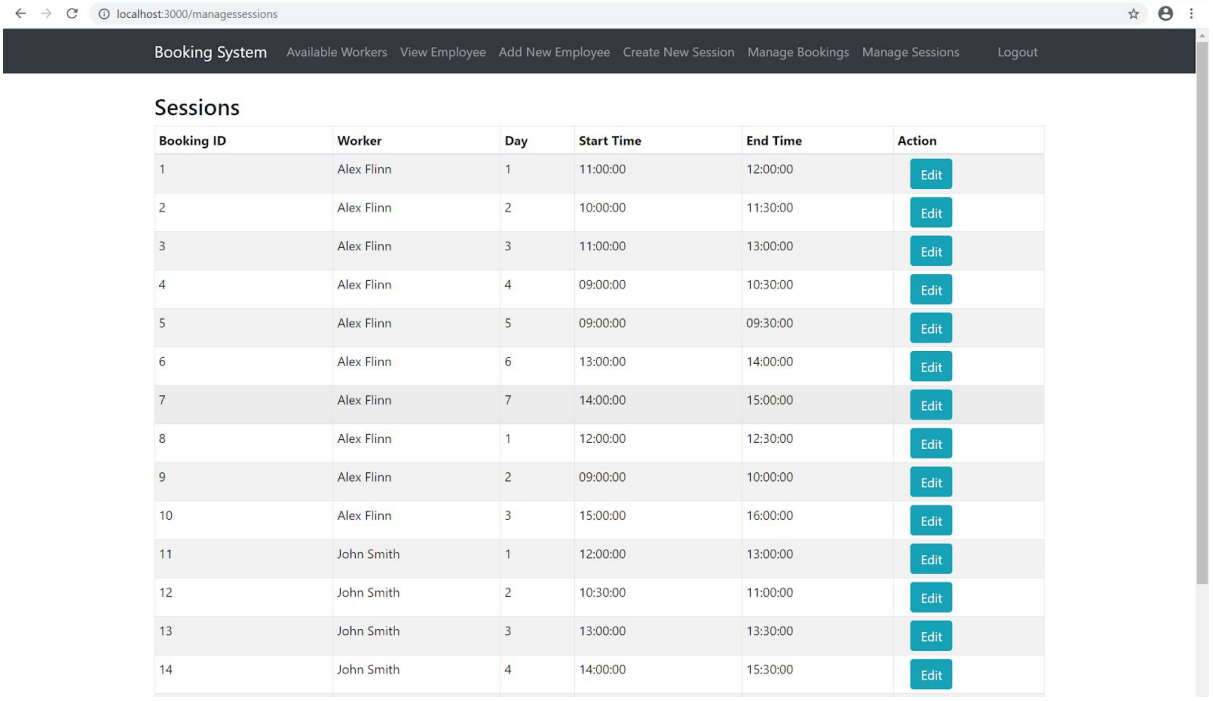


A message “Booking has been cancelled” is displayed if they cancel a valid booking and the booking will be removed from the current bookings page.

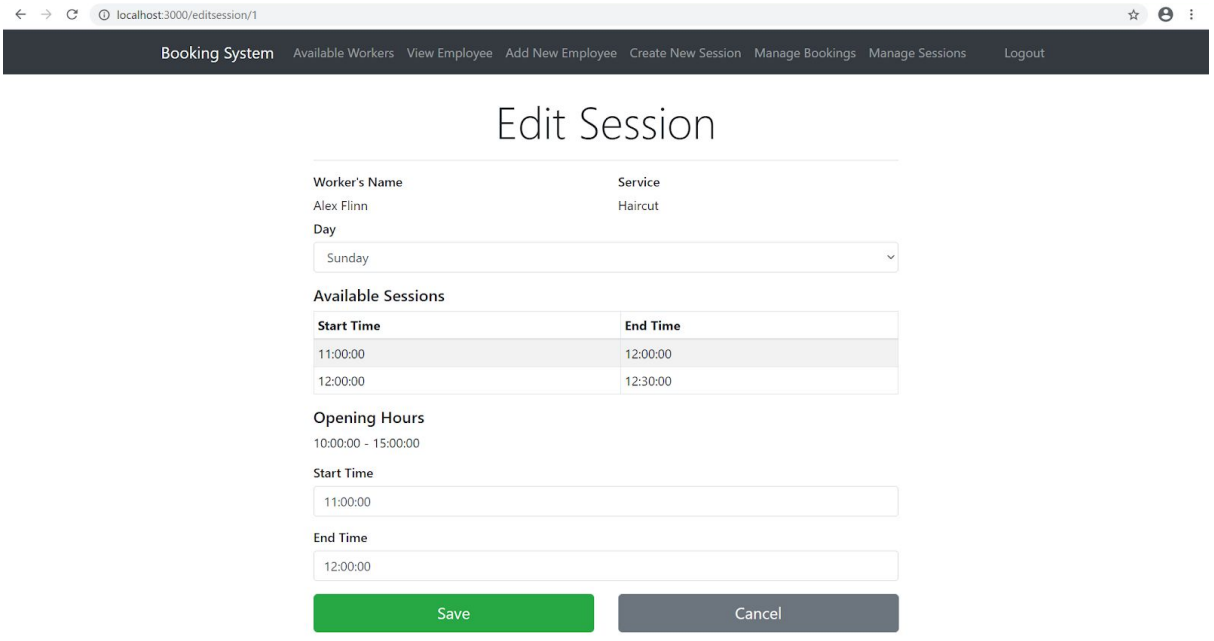


- **Admin Manages Sessions**

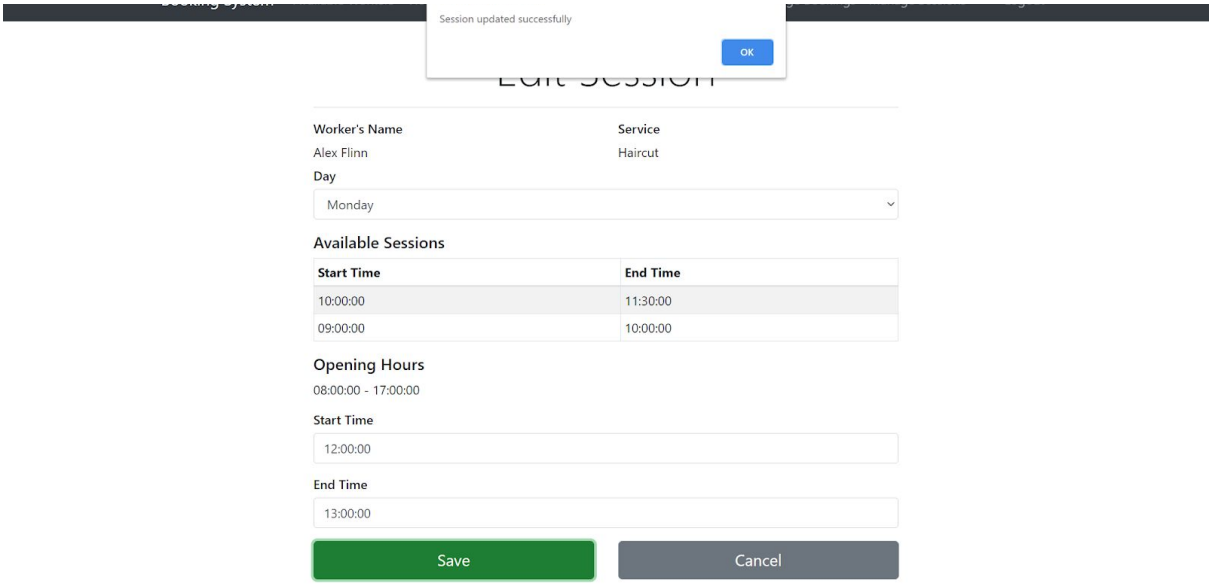
By logging in to the admin account and selecting the “Manage Sessions” button in the navigation bar, an admin is redirected to a page displaying all his/her available sessions with the “Edit” button for each session.



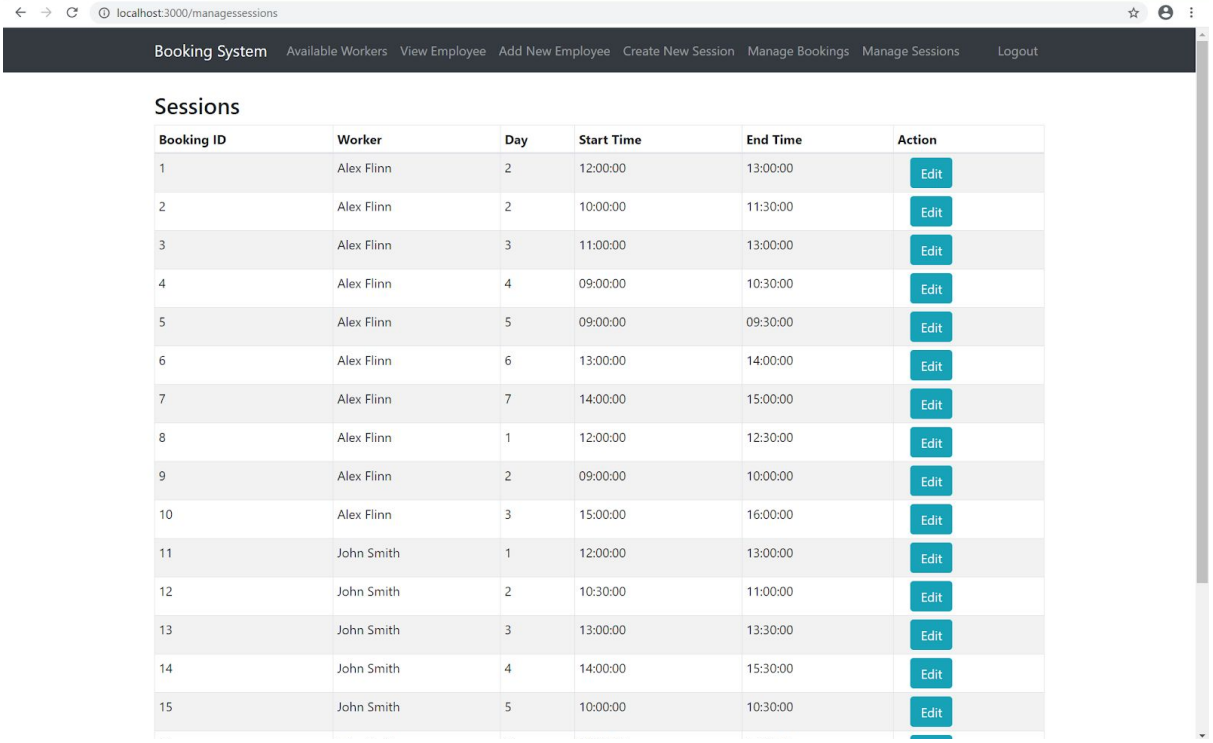
Picking a session and choosing the appropriate “Edit” button, he/she is redirected to a page showing all sessions details which are editable (consist of day, start time and end time).



The “Available Sessions” and “Opening Hours” will be filtered according to the day that the admin picks to help them to change the time correctly. Admin now change the session from “Sunday 11:00:00-12:00:00” to “Monday 12:00:00-13:00:00” and click the “Save” button. Then, a message “Session updated successfully” pops up to notify them.

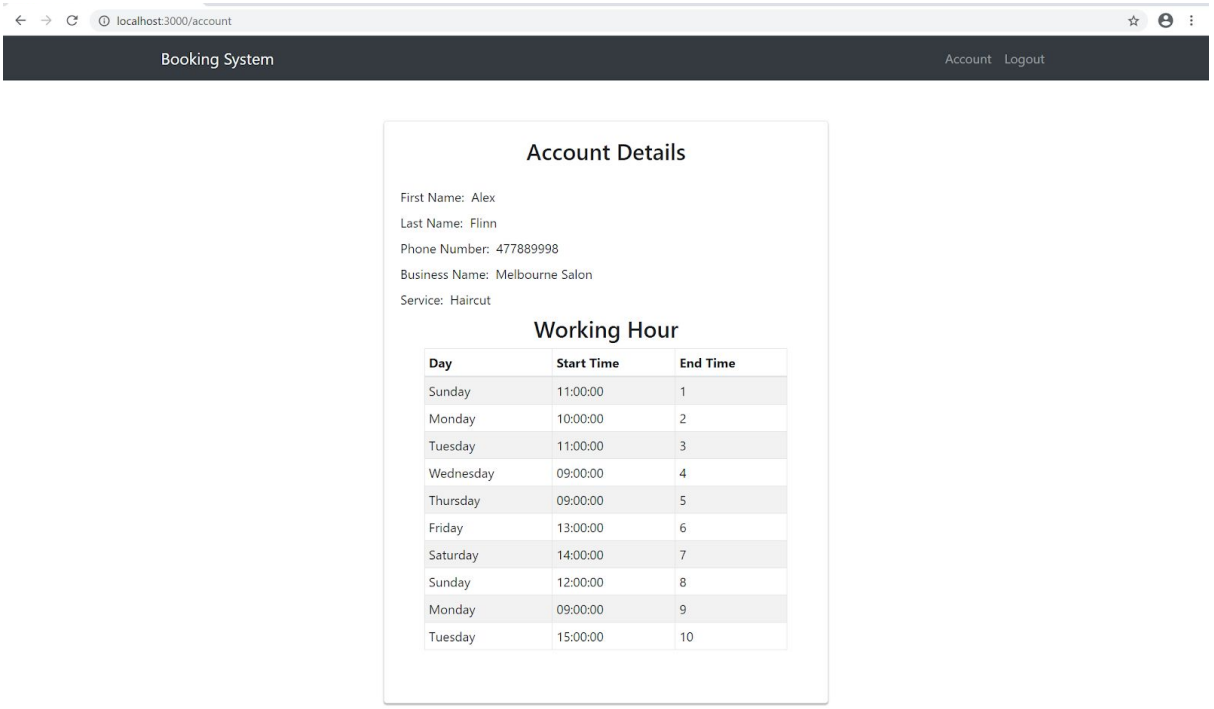


Admin is redirected back to the “Manage Sessions” page with the session already changed.



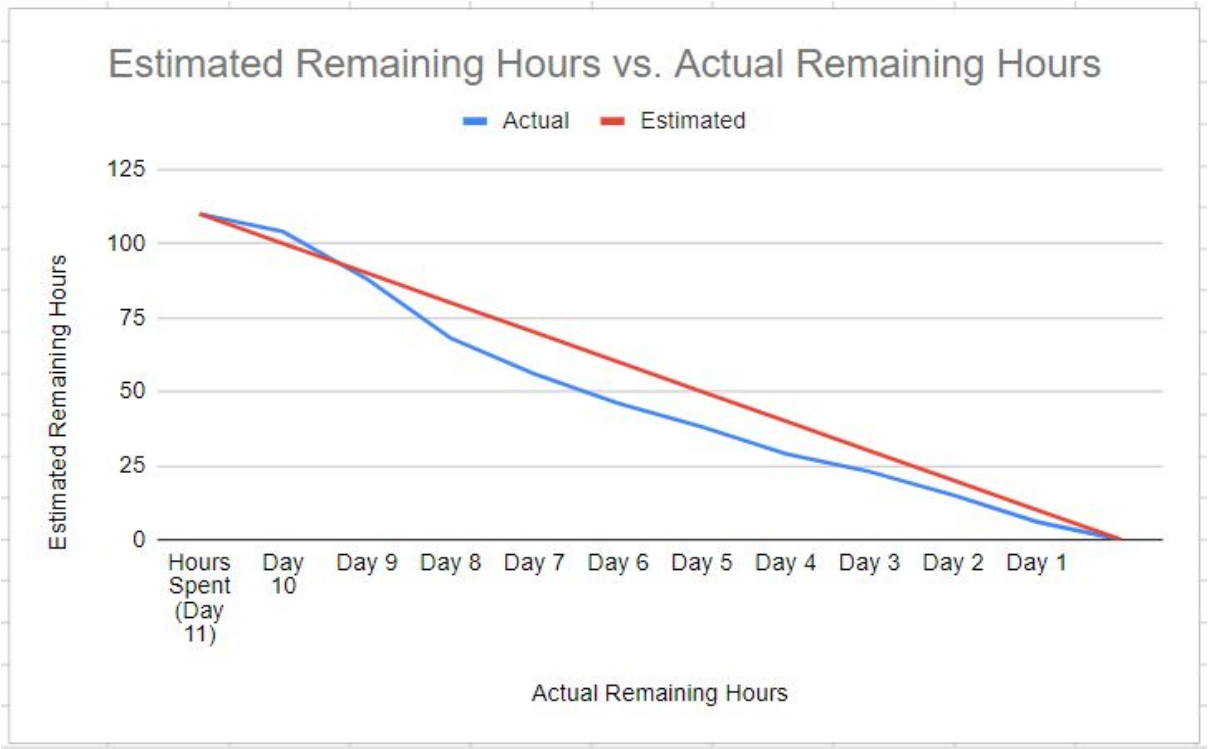
- **Display Worker's Profile**

By logging in the worker account and clicking on the "Account" button in the navigation bar. The worker is redirected to a page displaying all their details including service and assigned working hours.



Sprint Statistics

Task	Start Hours	Hours Spent (Day 11)	Day 10	Day 9	Day 8	Day 7	Day 6	Day 5	Day 4	Day 3	Day 2	Day 1	Total Hours
7.1	4	0	0	0	0	0	1	3	0	0	0	0	4
7.2	5	0	0	0	0	0	0	0	1	3	1	0	5
7.3	3	0	0	0	0	0	0	0	0	0	2	1	3
7.4	8	0	0	0	0	0	0	0	0	3	3	2	8
12.1	3	0	1	2	0	0	0	0	0	0	0	0	3
12.2	2	0	0	2	0	0	0	0	0	0	0	0	2
12.3	2	0	0	1	0	0	0	1	0	0	0	0	2
12.4	4	0	0	0	1	2	1	0	0	0	0	0	4
18.1	3	1	2	0	0	0	0	0	0	0	0	0	3
18.2	2	0	2	0	0	0	0	0	0	0	0	0	2
18.3	3	0	0	2	1	0	0	0	0	0	0	0	3
18.4	8	0	0	2	2	2	2	0	0	0	0	0	8
20.1	2	2	0	0	0	0	0	0	0	0	0	0	2
20.2	2	0	1	1	0	0	0	0	0	0	0	0	2
20.3	2	0	2	0	0	0	0	0	0	0	0	0	2
20.4	6	0	0	3	2	1	0	0	0	0	0	0	6
21.1	2	0	0	0	0	0	1	1	0	0	0	0	2
21.2	2	0	0	0	0	0	0	0	1	1	0	0	2
21.3	2	0	0	0	0	0	0	0	2	0	0	0	2
21.4	2	0	0	0	0	0	0	0	0	0	1	1	2
22.1	2	2	0	0	0	0	0	0	0	0	0	0	2
22.2	2	0	1	1	0	0	0	0	0	0	0	0	2
22.3	2	0	2	0	0	0	0	0	0	0	0	0	2
22.4	6	0	0	2	3	1	0	0	0	0	0	0	6
27.1	2	0	2	0	0	0	0	0	0	0	0	0	2
27.2	2	0	1	1	0	0	0	0	0	0	0	0	2
27.3	3	0	0	1	0	1	0	0	0	1	0	0	3
27.4	4	0	0	0	2	1	0	0	1	0	0	0	4
28.1	2	1	1	0	0	0	0	0	0	0	0	0	2
28.2	2	0	1	1	0	0	0	0	0	0	0	0	2
28.3	3	0	0	1	0	0	0	1	0	0	0	1	3
28.4	5	0	0	0	0	0	1	1	1	0	2	0	5
40	8	0	0	0	1	2	2	2	0	0	0	1	8
Actual Remaining Hours	110	104	88	68	56	46	38	29	23	15	6	0	110
Estimated Remaining Hours	110	100	90	80	70	60	50	40	30	20	10	0	



The last sprint had a total of 33 tasks, completed over 11 days. We spent a total of 110 hours to achieve this sprint. This was based on having 10 working hours a day. The red line depicts the ideal situation, with the estimated hours distributed to show a steady steep on the graph. In comparison to this, we spent our hours exceptionally, as displayed by the blue line.

After the first day, the estimated hours were much higher than anticipated, especially from the third day to the ninth day, with the difference between the estimated remaining hours and the actual remaining hours, being approximately 5 hours each day. Therefore in saying this, we were able to complete the sprint by the deadline and we are all happy with the outcome of this sprint