2020

SEPT Milestone 3 Report

SEBASTIAN WISIDAGAMA, SHAUNAK, RUCHELLE, AMNA, JULIAN
S3769969, S3782215, S3781183, S3778713, S3690935

RMIT|COSC 2299

Contents

Vision Statement

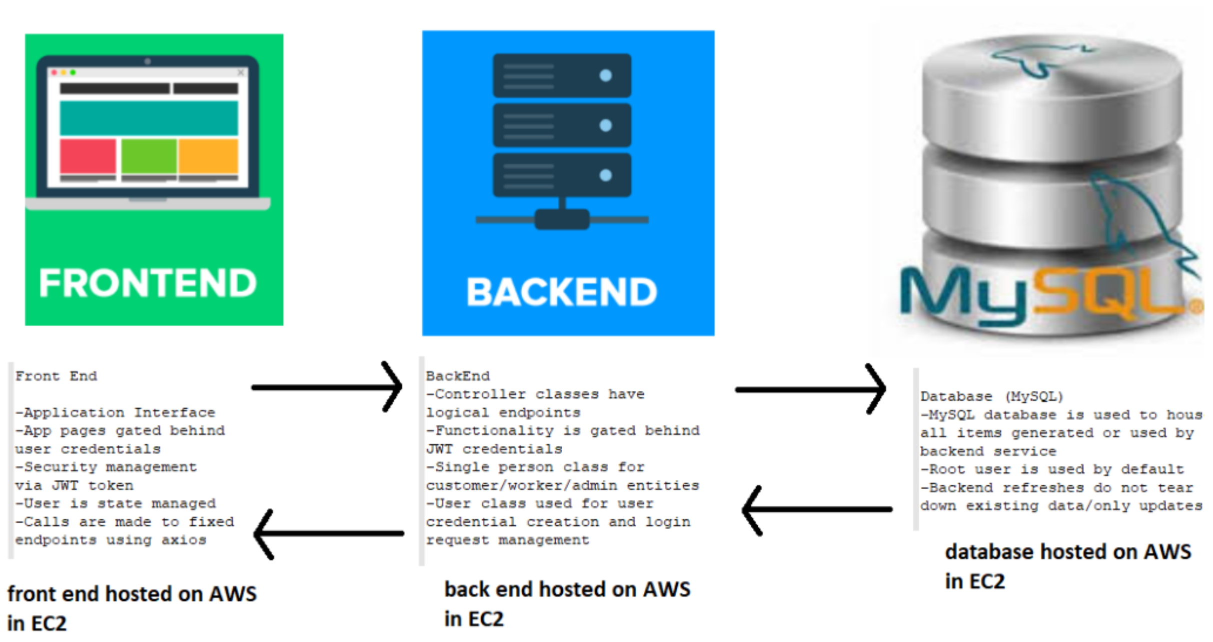
Our group project is to create a booking website that a business can use to manage customer bookings and employee weekly roster. Through their own account, employees can view their roster and customers can also make bookings based on the employees' availabilities. We also designed our website so that any type of user can create bookings for themselves as a customer. This makes sense as an admin or an employee can also be a customer in their own workplace.

We envisioned this project to be as user-friendly as possible. Hence for our website, users can login with just their username and password because determining their account type is handled by the API calls in the login component. Usability has also been greatly considered in the easy registration that redirects the page to user login once the user has successfully signed up, a one-click edit profile that auto fills the fields so only the ones need modifications will be edited, a search function based on either service or worker and an interactive calendar for bookings on worker availabilities.

We also wanted our website to have good security. Hence, we made use of the JWT authentication for a more secure log-in. A secure routing on our front-end functionality was also implemented during the last milestone of our project. With this, users will not be able to reach the other links without being logged in. Additionally, we've also hosted our project on AWS so people will be able to access the site securely.

Application Architecture

Application architecture overview



```
Front End

-Application Interface
-App pages gated behind
user credentials
-Security management
via JWT token
-User is state managed
-Calls are made to fixed
endpoints using axios
```

front end hosted on AWS in EC2

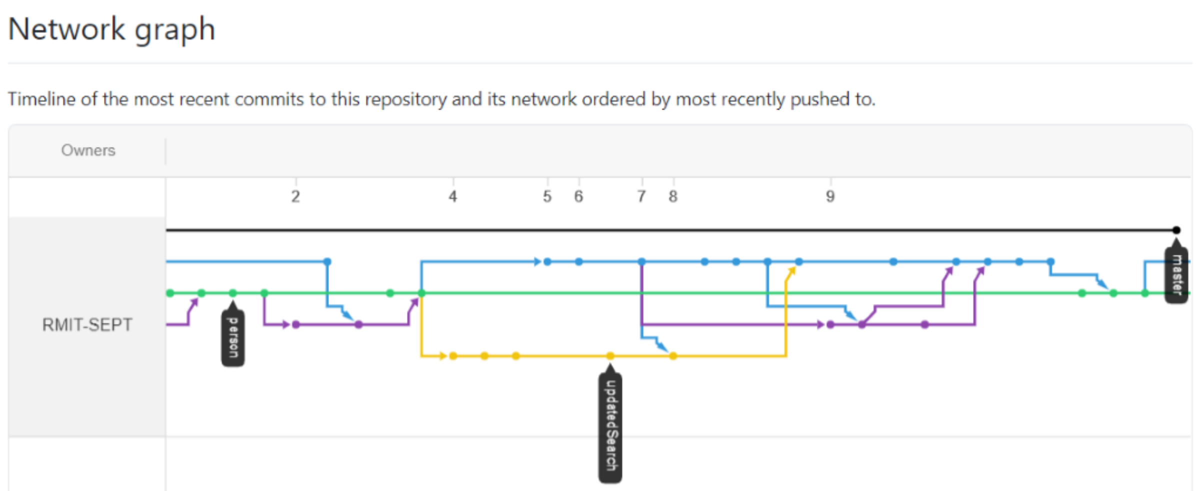
```
BackEnd
-Controller classes have
logical endpoints
-Functionality is gated behind
JWT credentials
-Single person class for
customer/worker/admin entities
-User class used for user
credential creation and login
request management
```

back end hosted on AWS in EC2

```
Database (MySQL)
-MySQL database is used to hous
all items generated or used by
backend service
-Root user is used by default
-Backend refreshes do not tear
down existing data/only updates
```

database hosted on AWS in EC2

GitFlow Organisation



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

The above image shows a snippet from our GitHub network graph. Each of the colored lines represents a specific feature branch which came of the main development branch(green) that was branched off and tagged for releases at the end of all major milestones.

Commit frequency ranged from once per day during early planning/development to 10-15 commits as release was closing and bugs were ironed out.

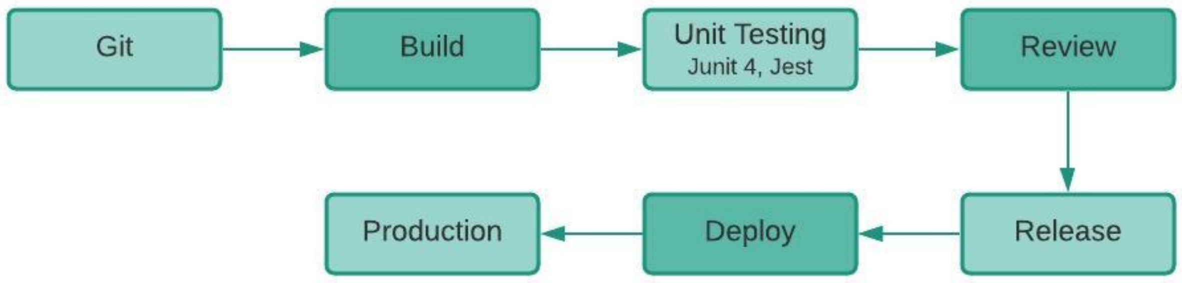
Scrum Process

Scrum master assigned was Ruchelle, with assistant scrum master being Shaunak. This was done to divide the additional work required in maintaining scrum documentation, as it felt unreasonable to assign a considerable extra workload for paperwork and limit each member's involvement in development and learning.

Meetings were conducted three times a week on fixed days, with an agenda of presenting changes/updates since the last meeting and this facilitated the groups learning for areas in development which members may have not been actively involved in. Additionally, Teams was used to tag other members as specific request were made to update front and back end as needed. This ensured continuity between endpoints even during refactoring periods.

Notion was used to keep the formal assignments of any work and meetings logged. Moreover, notion was used as a task management tracker to check the status of work as tasks were completed. The periodic updates during scrum planning sessions and scrum poker of new tasks allowed all the backlog items to be addressed and prioritized as needed. The used of Notion as a task tracker was somewhat redundant due to the frequency of communication which made all members aware of the state of every aspect/task in the project.

Deployment Pipeline Diagram



Refactoring Report

Major refactoring was required during the creation of a Person entity as a replacement for the separate entities for a Customer, Administrator and Employee. This change was done for model classes, repository interfaces, controller classes, service classes and test classes.

The Person model class essentially had the same fields and methods previously present, with the addition of an extra field (**userType**) that contained a string representation of whether the Person was an admin, customer, or employee. The Person controller essentially was a renamed version of the controllers previously present along with appropriate paths and error messages.

The Person Service changed a fair bit from the previous repositories as there were now more concise queries to find and get lists of users. The Person Service was much like the controller, in that methods and fields were renamed appropriately to be consistent with the new structure, with there being no additional methods added. The Person test class combined the tests that were present before, with the removal of duplicate tests, the renaming of tests and fields to stay consistent along with minor changes to paths and error message strings. Finally, the last part of refactoring was the refactoring of test classes to have more meaningful test names that describe the test well enough without there being a need for a foreign programmer to analyse functionality.

Back-end Acceptance Test Report

Assigned Service Controller Tests:

Date Performed:	**Function Being Tested:**	**Expected Result:**	**Actual Result:**
12/10/20	Posting a valid Assigned Service object	Created HTTP Status is returned	Created HTTP Status is returned
12/10/20	Posting a duplicate Assigned Service object	Conflict HTTP Status is returned with the error message: "Assigned Service Object Could Not Be Created"	Conflict HTTP Status is returned with the error message: "Assigned Service Object Could Not Be Created"
12/10/20	Posting an Assigned Service object with a NULL username	Bad Request HTTP Status is returned with the error message: "Invalid Assigned Service Object"	Bad Request HTTP Status is returned with the error message: "Invalid Assigned Service Object"
12/10/20	Getting a non-empty list of all Assigned Service objects	Ok HTTP Status is returned	Ok HTTP Status is returned
12/10/20	Getting an empty list of all Assigned Service objects	Not Found HTTP Status is returned with the error message: "No Assigned Service Objects"	Not Found HTTP Status is returned with the error message: "No Assigned Service Objects"
12/10/20	Getting a list of Assigned Service objects allocated to an existing employee	Ok HTTP Status is returned	Ok HTTP Status is returned
12/10/20	Getting a list of Assigned Service objects allocated to a non-existent employee	Not Found HTTP Status is returned with the error message: "No Assigned Service Objects"	Not Found HTTP Status is returned with the error message: "No Assigned Service Objects"
12/10/20	Getting a list of people by a valid Assigned Service	Ok HTTP Status is returned	Ok HTTP Status is returned
12/10/20	Getting a list of people by an invalid Assigned Service	Not Found HTTP Status is returned with the error message: "No Person Objects"	Not Found HTTP Status is returned with the error message: "No Person Objects"

Booking Controller Tests:

Date Performed:	**Function Being Tested:**	**Expected Result:**	**Actual Result:**
11/10/20	Posting a valid Booking object	Created HTTP Status is returned	Created HTTP Status is returned
11/10/20	Posting a Booking object with a null employee ID	Bad Request HTTP Status is returned with the error message: "Invalid Booking Object"	Bad Request HTTP Status is returned with the error message: "Invalid Booking Object"
11/10/20	Getting a single, existing Booking object	Ok HTTP Status is returned	Ok HTTP Status is returned
11/10/20	Getting a single, non-existent Booking object	Not Found HTTP Status is returned with the error message: "No Booking Object"	Not Found HTTP Status is returned with the error message: "No Booking Object"
11/10/20	Getting a non-empty list of all Booking objects for a customer	Ok HTTP Status is returned	Ok HTTP Status is returned
11/10/20	Getting an empty list of all Booking objects for a customer	Not Found HTTP Status is returned with the error message: "No Booking Objects"	Not Found HTTP Status is returned with the error message: "No Booking Objects"
11/10/20	Getting a non-empty list of all past Booking objects	Ok HTTP Status is returned	Ok HTTP Status is returned
11/10/20	Getting an empty list of all past Booking objects	Not Found HTTP Status is returned with the error message: "No Booking Objects"	Not Found HTTP Status is returned with the error message: "No Booking Objects"
11/10/20	Getting a non-empty list of all upcoming Booking objects	Ok HTTP Status is returned	Ok HTTP Status is returned
11/10/20	Getting an empty list of all past upcoming objects	Not Found HTTP Status is returned with the error message: "No Booking Objects"	Not Found HTTP Status is returned with the error message: "No Booking Objects"

Person Controller Tests:

Date Performed:	**Function Being Tested:**	**Expected Result:**	**Actual Result:**
11/10/20	Posting a valid Person object	Created HTTP Status is returned	Created HTTP Status is returned

11/10/20	Posting a Person object with a 4-character password	Created HTTP Status is returned	Created HTTP Status is returned
11/10/20	Posting a Person object with a 25-character password	Created HTTP Status is returned	Created HTTP Status is returned
11/10/20	Posting a Person object with a null username	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a null first name	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a null last name	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a null address	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a null phone number	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a blank username	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a blank password	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a blank first name	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a blank last name	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”

11/10/20	Posting a Person object with a blank address	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a blank phone number	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a 3-character password	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a 26-character password	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a 9-character phone number	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with a 11-character phone number	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with numbers in the first name	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with numbers in the last name	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Posting a Person object with alphabetic letters phone number	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”	Bad Request HTTP Status is returned with the error message: “Invalid Person Object”
11/10/20	Updating an existing Person object with valid fields	Ok HTTP Status is returned	Ok HTTP Status is returned
11/10/20	Updating an existing Person object with a null password	Bad Request HTTP Status is returned with the error message:	Bad Request HTTP Status is returned with the error message:

		"Invalid Person Object"	"Invalid Person Object"
11/10/20	Getting an existing Person object	Ok HTTP Status is returned	Ok HTTP Status is returned

<u>Service Controller Tests:</u>

Date Performed:	**Function Being Tested:**	**Expected Result:**	**Actual Result:**
13/10/20	Posting a valid Service object	Created HTTP Status is returned	Created HTTP Status is returned
13/10/20	Posting a Service object with a null description	Bad Request HTTP Status is returned with the error message: "Invalid Service Object"	Bad Request HTTP Status is returned with the error message: "Invalid Service Object"
13/10/20	Getting a non-empty list of Services	Ok HTTP Status is returned	Ok HTTP Status is returned
13/10/20	Getting an empty list of Services	Bad Request HTTP Status is returned with the error message: "No Service Objects"	Bad Request HTTP Status is returned with the error message: "No Service Objects"
13/10/20	Getting a single Service object	Bad Request HTTP Status is returned with the error message: "Invalid Service Object"	Bad Request HTTP Status is returned with the error message: "Invalid Service Object"

<u>Working Hours Controller Tests:</u>

Date Performed:	**Function Being Tested:**	**Expected Result:**	**Actual Result:**
11/10/20	Posting a valid Working Hours object	Created HTTP Status is returned	Created HTTP Status is returned
11/10/20	Posting a Working hours object with null employee ID	Bad Request HTTP Status is returned with the error message: "Invalid Working Hours Object"	Bad Request HTTP Status is returned with the error message: "Invalid Working Hours Object"
11/10/20	Posting a Working hours object with a 2500 end time	Bad Request HTTP Status is returned with the error message: "Invalid Working Hours Object"	Bad Request HTTP Status is returned with the error message: "Invalid Working Hours Object"
11/10/20	Updating a Working Hours object with a null employee ID	Bad Request HTTP Status is returned with the error message: "Invalid Working Hours Object"	Bad Request HTTP Status is returned with the error message: "Invalid Working Hours Object"

11/10/20	Getting an existing Working Hours object	Ok HTTP Status is returned	Ok HTTP Status is returned
11/10/20	Getting a list of Working Hours objects for a non-existent employee	Bad Request HTTP Status is returned with the error message: “No Working Hours Objects”	Bad Request HTTP Status is returned with the error message: “No Working Hours Objects”
11/10/20	Getting a list of Working Hours objects for a non-existent employee and invalid date	Bad Request HTTP Status is returned	Bad Request HTTP Status is returned

Front-end Acceptance Test Report

Admin Functionalities:

ID	Date Performed	Function Being Tested:	Expected Result:	Actual Result
12.4	16/10/20	Clicking Add Employee button on the admin page	redirected to the Add New Employee page	redirected to the Add New Employee page
12.5	16/10/20	Adding an employee without any of the required fields filled out	“fill out this field” text can be seen above the empty field	“fill out this field” text can be seen above the empty field
12.6	16/10/20	Adding an employee assuming that all the fields are filled out with phone number only having 5 digits.	an error message saying registration failed appear on the page	an error message saying registration failed appear on the page
12.7	16/10/20	Adding an employee with filled out required fields with password having less than five characters	an error message saying registration failed should appear on the page	an error message saying registration failed should appear on the page
12.8	16/10/20	Adding an employee with completed correct fields	an employee is added in the database	an employee is added in the database
35.1	16/10/20	Clicking Add Service button in the admin page	redirected to the Add Service page	redirected to the Add Service page
35.2	16/10/20	Adding a service with incomplete required fields	“fill out this field” text can be seen above the empty field	“fill out this field” text can be seen above the empty field
35.3	16/10/20	Adding a service with completed fields with non-numeral service ID	“add number” can be seen above the service ID field	“add number” can be seen above the service ID field

35.4	16/10/20	Adding a service with completed fields with non-numeral duration	“add number” can be seen above the service ID field	“add number” can be seen above the service ID field
35.5	16/10/20	Adding a service with completed correct fields	a new service is added in the database	a new service is added in the database
14.3	16/10/20	Clicking Add Working Hours button in the admin page	redirected to the Add Working Hours page	redirected to the Add Working Hours page
14.4	16/10/20	Adding working hours with filled out required fields with a past date	an error message saying adding working hours failed appear on the page	an error message saying adding working hours failed appear on the page
14.5	16/10/20	Adding working hours with completed correct fields	Employee’s working hours added in the database	Employee’s working hours added in the database
16.2	16/10/20	Clicking View All Past Bookings button in the admin page	redirected to the Past Bookings page	redirected to the Past Bookings page
16.3	16/10/20	Viewing past bookings with no past bookings in the database	A text saying there are no past bookings can be seen in the Past Bookings page	A text saying there are no past bookings can be seen in the Past Bookings page
16.4	16/10/20	Viewing past bookings with past bookings in the database	all the past bookings from the database can be seen in the Past Bookings page	all the past bookings from the database can be seen in the Past Bookings page
16.5	16/10/20	Clicking View All Upcoming Bookings button in the admin page	redirected to the Upcoming Bookings page	redirected to the Upcoming Bookings page
16.6	16/10/20	Viewing past bookings with past bookings in the database	all the upcoming bookings from the database can be seen in the Past Bookings page	all the upcoming bookings from the database can be seen in the Past Bookings page
36.1	16/10/20	Clicking Assign Service button in the admin page	redirected to the Assign Service page	redirected to the Assign Service page
36.2	16/10/20	Assigning service with selected employee and service	Service should be added to the list of services assigned to the employee in the database.	Service should be added to the list of services assigned to the employee in the database.