

# Monday 3:30, Group 1 – Milestone 2 Submission

Name	Student #	Contribution
Abdelmalek, Lawrence	s3656022	20%
Alsagheer, Abrar	s3707180	20%
Tawaf, Elizabeth	s3723812	20%
Yendall, Brodey	s3718834	20%
Wang, Richard	s3720241	20%

## Contributions

### Abdelmalek, Lawrence

- Assisted other front end developers with various tasks and designs
- Developed tests for the front end
- Finalising of Bookings and Hours services

### Alsagheer, Abrar

- Development of the circleCI pipeline
- Deployments of front end and back end services
- Development of the availability service

### Tawaf, Elizabeth

- Implementation of admin create users functionality
- Prototyping and designing the interfaces

### Yendall, Brodey

- Maintenance of back end services
- Increasing and fixing backend tests
- Refactor and improvements to the common library

### Wang, Richard

- Implementation of admin assigning hours functionality
- Implement book appointment functionality
- Investigating how to link back to front end

# Links

Github:

<https://github.com/RMIT-SEPT/majorproject-9-mon-1530-1>

JIRA (Our task board):

<https://sept-assignment.atlassian.net/secure/RapidBoard.jspa?rapidView=5&projectKey=AS>

## CircleCI:

<https://app.circleci.com/pipelines/github/RMIT-SEPT/majorproject-9-mon-1530-1>

Due to running out of credits we were required to make a private repository and developed the CircleCI pipeline there.

Please observe “CircleCI history.pdf” for screenshots of failure and successes from CircleCI and a screenshot the AWS deployed application.

## AWS

Main link:

Use this link to access the application’s front end

<http://sept-env-1.eba-mpb52g22.us-east-1.elasticbeanstalk.com/>

## Back-end URLs

The following URLs are all for backend services. These links do not have a front end interface but use an API instead. Please observe our API documentation for how to use these APIs

Users:

<http://users-env.eba-qkspuxrm.us-east-1.elasticbeanstalk.com>

Hours:

<http://hours-env.eba-mmev4jew.us-east-1.elasticbeanstalk.com/>

Bookings:

<http://bookings-env.eba-kxxpnwmy.us-east-1.elasticbeanstalk.com/>

Availability:

<http://availability-env.eba-5wumcyam.us-east-1.elasticbeanstalk.com/>

## Documentation

You can find our documentation at these links or alternative in the PDFs attached

Meeting minutes:

<https://docs.google.com/document/d/1VwznRiQvTXG2edQyKUvhek1T3K1MYGgKsf5rYva7a2s/edit>

Acceptance tests: <https://docs.google.com/document/d/17i3kjH-Sq0F4iVErP9VUwW7l8hW05puLTY2r6v4cgBM/edit>

Retro notes:

<https://docs.google.com/document/d/1QzsRQ6jkXlGGtWvbo2BbgK1EYT63mxl7qIIjjKftesQ/edit>

Sprint planning notes:

[https://docs.google.com/document/d/1QsxZ\\_l78HfUrBCNNuHI8zo83-UsvzTBsUG0kFy5tYKI/edit](https://docs.google.com/document/d/1QsxZ_l78HfUrBCNNuHI8zo83-UsvzTBsUG0kFy5tYKI/edit)

User stories:

[https://docs.google.com/document/d/1hk5YE-FuwBdbOxs24yind3ak\\_6VOdh7edNO53AaewBA/edit](https://docs.google.com/document/d/1hk5YE-FuwBdbOxs24yind3ak_6VOdh7edNO53AaewBA/edit)

## Additional documentation

API documentation: <https://docs.google.com/spreadsheets/d/11sLVYMP3aJ-I-ocQJlMQttMxrNts48WPDepV-bfG4DY/edit#gid=948496785>

Setting up a basic CRUD service with the common library:

<https://docs.google.com/document/d/1wkaIza8SGxosywtLCUmT8agkewFyaA1RV939RXSbK2g/edit#>

# Testing evidence:

## Hours

Run: sept.major.hours.blackbox in Hours		Tests passed: 22 of 22 tests - 28 s 956 ms
▼ blackbox (sept.major.hours)	28 s 956 ms	"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe"
▼ DeleteBlackBoxTest	1 s 572 ms	19:04:43.190 [main] DEBUG org.springframework
✓ Attempt to delete not existing record	966 ms	19:04:43.208 [main] DEBUG org.springframework
✓ Successfully delete entity	606 ms	19:04:43.251 [main] DEBUG org.springframework
▼ RepositoryBlackBoxTests	2 s 924 ms	19:04:43.273 [main] INFO org.springframework
✓ Boundary testing of findConflictingHours	517 ms	19:04:43.280 [main] DEBUG org.springframework
✓ findAllInRange boundary testing with only one record	435 ms	19:04:43.280 [main] INFO org.springframework
✓ findAllInRange returning multiple records	1 s 972 ms	19:04:43.280 [main] INFO org.springframework
▼ PostBlackBoxTest	4 s 968 ms	19:04:43.281 [main] INFO org.springframework
✓ existing()	1 s 986 ms	19:04:43.448 [main] DEBUG org.springframework
✓ incorrectFieldTypeMap()	342 ms	19:04:43.767 [main] DEBUG org.springframework
✓ missingField()	304 ms	19:04:43.767 [main] INFO org.springframework
✓ incorrectFieldTypeList()	1 s 738 ms	19:04:43.813 [main] INFO org.springframework
✓ valid()	1 s 738 ms	19:04:43.813 [main] INFO org.springframework

Run: sept.major.hours.controller.unit in Hours		Tests passed: 41 of 41 tests - 935 ms
▼ unit (sept.major.hours.controller)	935 ms	"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe"
▼ PatchEndpointTests	440 ms	19:07:34.136 [main] DEBUG org.springframework.t
✓ contextLoads()	196 ms	19:07:34.151 [main] DEBUG org.springframework.t
✓ Update multiple not blank fields to blank	183 ms	19:07:34.185 [main] DEBUG org.springframework.t
✓ Patch attempt on non existing entity	9 ms	19:07:34.205 [main] INFO org.springframework.bo
✓ Successfully patch an entity	15 ms	19:07:34.210 [main] DEBUG org.springframework.t
✓ Provided id isn't an integer	10 ms	19:07:34.211 [main] INFO org.springframework.t
✓ Update not blank field to blank	14 ms	19:07:34.212 [main] INFO org.springframework.t
▶ GetAllEndpointTests	66 ms	19:07:34.262 [main] DEBUG org.springframework.t
▶ PostEndpointTests	146 ms	19:07:34.387 [main] INFO org.springframework.c
▶ DeleteEndpointTests	39 ms	19:07:34.389 [main] INFO org.springframework.bo
▶ GetInRangeEndpointTests	138 ms	19:07:34.679 [main] DEBUG org.springframework.b
▶ GetOnDateEndpointTests	53 ms	19:07:34.680 [main] INFO org.springframework.b

## Bookings

Run: sept.major.bookings.blackbox in Bookings		Tests passed: 22 of 22 tests - 29 s 26 ms
▼ blackbox (sept.major.bookings)	29 s 26 ms	"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe"
▼ DeleteBlackBoxTest	1 s 548 ms	19:14:35.527 [main] DEBUG org.springframework
✓ Attempt to delete not existing record	913 ms	19:14:35.555 [main] DEBUG org.springframework
✓ Successfully delete entity	623 ms	19:14:35.602 [main] INFO org.springframework
▼ PostBlackBoxTest	3 s 189 ms	19:14:35.633 [main] INFO org.springframework
✓ existing()	385 ms	19:14:35.640 [main] DEBUG org.springframework
✓ incorrectFieldTypeMap()	355 ms	19:14:35.641 [main] DEBUG org.springframework
✓ missingField()	320 ms	19:14:35.641 [main] INFO org.springframework
✓ incorrectFieldTypeList()	304 ms	19:14:35.642 [main] INFO org.springframework
✓ valid()	1 s 485 ms	19:14:35.820 [main] DEBUG org.springframework
✓ endBothEndDates()	320 ms	19:14:35.821 [main] INFO org.springframework
▼ PatchBlackBoxTest	14 s 416 ms	19:14:36.094 [main] DEBUG org.springframework
✓ Provided value for a field as a map	4 s 387 ms	19:14:36.095 [main] INFO org.springframework
✓ Attempt to update not existing entity	1 s 978 ms	19:14:36.145 [main] DEBUG org.springframework

Run: sept.major.bookings.controller.unit in Bookings		Tests passed: 41 of 41 tests - 888 ms
▼ unit (sept.major.bookings.controller)	888 ms	"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe"
▼ GetAllEndpointTests	294 ms	19:17:19.822 [main] DEBUG org.springframework
✓ contextLoads()	209 ms	19:17:19.846 [main] DEBUG org.springframework
✓ No record found	42 ms	19:17:19.897 [main] INFO org.springframework
✓ Successfully retrieve one record	16 ms	19:17:20.003 [main] INFO org.springframework
✓ Successfully retrieve many records	17 ms	19:17:20.008 [main] DEBUG org.springframework
✓ Invalid usernames provided	10 ms	19:17:20.008 [main] INFO org.springframework
▶ GetOnDateEndpointTests	93 ms	19:17:20.009 [main] INFO org.springframework
▶ UnitTestHelper	6 ms	19:17:20.010 [main] INFO org.springframework
▶ GetInRangeEndpointTests	94 ms	19:17:20.062 [main] DEBUG org.springframework
▶ DeleteEndpointTests	54 ms	19:17:20.188 [main] DEBUG org.springframework
▼ PostEndpointTests	193 ms	19:17:20.189 [main] INFO org.springframework
✓ contextLoads()	6 ms	19:17:20.436 [main] INFO org.springframework
✓ No start date provided	110 ms	19:17:20.437 [main] INFO org.springframework

## Users

Run: sept.major.users.blackbox in users		Tests passed: 17 of 17 tests - 25 s 548 ms
▼ blackbox (sept.major.users)	25 s 548 ms	"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe"
▼ DeleteBlackBoxTest	1 s 515 ms	19:19:03.398 [main] DEBUG org.springframework
✓ Attempt to delete not existing record	914 ms	19:19:03.411 [main] DEBUG org.springframework
✓ Successfully delete entity	601 ms	19:19:03.455 [main] INFO org.springframework
▼ PostBlackBoxTest	9 s 921 ms	19:19:03.473 [main] INFO org.springframework
✓ existing()	408 ms	19:19:03.479 [main] DEBUG org.springframework
✓ incorrectFieldTypeMap()	304 ms	19:19:03.479 [main] DEBUG org.springframework
✓ missingField()	4 s 535 ms	19:19:03.480 [main] INFO org.springframework
✓ incorrectFieldTypeList()	406 ms	19:19:03.482 [main] INFO org.springframework
✓ valid()	4 s 584 ms	19:19:03.647 [main] DEBUG org.springframework
▼ GetBlackBoxTests	2 s 584 ms	19:19:03.648 [main] INFO org.springframework
✓ GET all endpoint successfully gets record with	1 s 984 ms	19:19:03.901 [main] DEBUG org.springframework
✓ Successfully retrieve record by username	600 ms	19:19:03.902 [main] INFO org.springframework
▼ GetBlackBoxTests	1 s 745 ms	19:19:03.955 [main] INFO org.springframework
✓ valid()	1 s 745 ms	19:19:03.964 [main] DEBUG org.springframework

Run: sept.major.users.controller.unit in users		Tests passed: 30 of 30 tests - 5 s 232 ms
▼ unit (sept.major.users.controller)	5 s 232 ms	"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe"
▼ GetByUsernameEndpointTests	405 ms	19:21:32.389 [main] DEBUG org.springframework
✓ contextLoads()	262 ms	19:21:32.408 [main] DEBUG org.springframework
✓ No record found with given username	45 ms	19:21:32.443 [main] DEBUG org.springframework
✓ Successfully retrieve record by username	96 ms	19:21:32.462 [main] INFO org.springframework
▼ PatchEndpointTests	726 ms	19:21:32.470 [main] DEBUG org.springframework
✓ contextLoads()	8 ms	19:21:32.472 [main] DEBUG org.springframework
✓ Update multiple not blank fields to blank	8 ms	19:21:32.473 [main] INFO org.springframework
✓ Patch attempt on non existing entity	209 ms	19:21:32.476 [main] INFO org.springframework
✓ Successfully patch an entity	148 ms	19:21:32.528 [main] DEBUG org.springframework
✓ Successfully patch with no patch values	17 ms	19:21:32.678 [main] DEBUG org.springframework
✓ Update not blank field to blank	144 ms	19:21:32.679 [main] INFO org.springframework
▼ GetAllEndpointTests	2 s 861 ms	19:21:32.818 [main] DEBUG org.springframework
✓ contextLoads()	5 ms	19:21:32.819 [main] INFO org.springframework
✓ valid()	5 ms	19:21:32.855 [main] INFO org.springframework

## Availability

Run: AvailabilityServiceTest		Tests passed: 9 of 9 tests - 163 ms
▼ Test Results	163 ms	"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe"
▼ AvailabilityServiceTest	163 ms	19:03:21.025 [main] DEBUG org.springframework
✓ checkAllAvailabilities_bookingWithingHours()	107 ms	19:03:21.055 [main] DEBUG org.springframework
✓ checkAllAvailabilities_fullOverlap()	17 ms	19:03:21.093 [main] DEBUG org.springframework
✓ checkAllAvailabilities_partialEndOverlap()	3 ms	19:03:21.111 [main] INFO org.springframework
✓ checkAllAvailabilities_noOverlap()	7 ms	19:03:21.118 [main] DEBUG org.springframework
✓ checkAllAvailabilities_bookingsOnly()	3 ms	19:03:21.119 [main] DEBUG org.springframework
✓ checkAllAvailabilities_partialStartOverlap()	5 ms	19:03:21.119 [main] INFO org.springframework
✓ checkAllAvailabilities_complex()	5 ms	19:03:21.120 [main] INFO org.springframework
✓ checkAllAvailabilities_empty()	5 ms	19:03:21.188 [main] DEBUG org.springframework
✓ checkAllAvailabilities_hoursOnly()	11 ms	19:03:21.321 [main] DEBUG org.springframework
✓ valid()	11 ms	19:03:21.323 [main] INFO org.springframework
✓ valid()	11 ms	19:03:21.512 [main] DEBUG org.springframework
✓ valid()	11 ms	19:03:21.522 [main] INFO org.springframework
✓ valid()	11 ms	19:03:21.604 [main] INFO org.springframework

# Front end

```
Watch Usage
> Press a to run all tests.
> Press f to run only failed tests.
> Press q to quit watch mode.
> Press p to filter by a filename regex pattern.
> Press t to filter by a test name regex pattern.
> Press Enter to trigger a test run.
PASS src/Components/Login/Login.test.js (5.229s)
PASS src/Components/Dashboard/Bookings/BookingComponents.test.js (6.361s)
PASS src/Components/Dashboard/Dashboard.test.js
PASS src/Contexts/BookingContext.test.js
FAIL src/Components/Dashboard/DashboardComponents.test.js
  ● DashboardComponents > UpcomingAppointmentCard > should render an UpcomingAppointmentCard with booking information passed from props
    Unable to find an element with the text: 13/01/2022. This could be because the text is broken up by multiple elements. In this case,
    use `findAllByText` to find all elements with the text.

    <body>
      <div>
        <div
          class="sc-fzq2Es jyhIAw"
        >
          
          <div
            class="sc-fzq0ul jfQ0JO"
          >
            <div>
              <svg
                class="upcomingAppointmentsCardIcon"
                fill="none"
                height="16px"
                stroke="currentColor"
                stroke-linecap="round"
                stroke-linejoin="round"
                stroke-width="2"
                viewBox="0 0 24 24"
                width="16px"
                xmlns="http://www.w3.org/2000/svg"
              >
                <path
                  d="M20 21v-2a4 4 0 0 0-4-4h8a4 4 0 0 0-4 4v2"
                />
                <circle
                  cx="12"
                  cy="7"
                  r="4"
                />
              </svg>
              <span
                class="sc-fzoCCn gIHecX"
              >
                testWorker
              </span>
            </div>
          </div>
        </div>
      </div>
    </body>

    at Object.getElementError (node_modules/@testing-library/dom/dist/config.js:34:12)
    at node_modules/@testing-library/dom/dist/query-helpers.js:71:38
    at getByText (node_modules/@testing-library/dom/dist/query-helpers.js:54:17)
    at Object.<anonymous> (src/Components/Dashboard/DashboardComponents.test.js:64:14)

FAIL src/Components/Form/Form.test.js
  ● Test suite failed to run
    Cannot find module './Components/Form' from 'Form.test.js'

    2 | import 'mutationobserver-shim';
    3 | import { render } from '@testing-library/react';
  > 4 | import Form from './Components/Form';
      |     ^
    5 |
    6 | global.MutationObserver = window.MutationObserver;
    7 |

    at Resolver.resolveModule (node_modules/jest-resolve/build/index.js:259:17)
    at Object.<anonymous> (src/Components/Form/Form.test.js:4:1)

PASS src/Components/Toolbar/Toolbar.test.js
PASS src/Components/Main/Main.test.js
  ● Console

    console.warn node_modules/react-dom/cjs/react-dom.development.js:88
      Warning: componentWillReceiveProps has been renamed, and is not recommended for use. See https://fb.me/react-warning-component-will-receive-props
    * Move data fetching code or side effects to componentDidUpdate.
    * If you're updating state whenever props change, refactor your code to use memoization techniques or move it to componentDidMount after the first render.
    * Rename componentWillReceiveProps to UNSAFE_componentWillReceiveProps to suppress this warning in non-strict mode. You can learn more about https://fb.me/react-unsafe-component-lifecycles in your project source folder.

    Please update the following components: TransitionMotion

PASS src/App.test.js
  ● Console

    console.warn node_modules/react-dom/cjs/react-dom.development.js:88
      Warning: componentWillReceiveProps has been renamed, and is not recommended for use. See https://fb.me/react-warning-component-will-receive-props
    * Move data fetching code or side effects to componentDidUpdate.
    * If you're updating state whenever props change, refactor your code to use memoization techniques or move it to componentDidMount after the first render.
    * Rename componentWillReceiveProps to UNSAFE_componentWillReceiveProps to suppress this warning in non-strict mode. You can learn more about https://fb.me/react-unsafe-component-lifecycles in your project source folder.

    Please update the following components: TransitionMotion

Test Suites: 2 failed, 7 passed, 9 total
Tests: 1 failed, 24 passed, 25 total
Snapshots: 0 total
Time: 9.558s
```

## Work done since Milestone 1

- Implementation of availability service MVP
  - The availability service is a convenience API for front end that formats Hours and Bookings data into a single payload. Saving the front end from doing extra calculations
  - Back/Availability directory contains the project for this service.
- Finalisation of the Bookings and Hours services
  - Changes the controllers, services, repositories and junit tests. Each of these can be found in the relevant BackEnd folders for each service
  - As of milestone 1 the Bookings and Hours services were barebones and have bugs a plenty. This milestone we finalised these services by completing the functionality, fixing the bugs and a great improvement to the testing
- Blackbox testing with a temporary dockerized database
  - Allows for junit tests for backend services to access a database that will be closed and reopened for every test. This functionality allows us to do complex and isolated tests without the use of mocks, allowing us to test the full scope of the application.
  - BackEnd\Common\src\main\java\sept\major\common\testing for all classes used in the implementation of this.
  - Bookings, Hours and Users implement an application-blackbox.properties file and include a startup .sql file in the resources folder relevant to the service.
- Improvements to User service tests
  - The user service lacked any sort of blackbox testing. This implemented this functionality as well as general improvements to the unit tests.
  - See BackEnd\Users\src\test\java\sept\major\users
- Implement book appointment functionality
  - User story 8
  - In order to accurately handle requests sent to the back-end, we must ensure that every request sent has all the information required. There is no point posting a query to the back-end without the correct parameters. The requests should be handled client-side.
  - Support Firefox date and time selection for making a booking ensure that our application runs smoothly on major supported browsers so that any user can successfully use the app. We need to ensure all components we use are supported, and where they are not, we implement workarounds to address these issues.
  - Handle return status from booking an appointment for the user dashboard, we can start sending requests, which is great and all, but how do we know a request worked? This story will explore updating the UI and view to reflect whether a request made by a user was successful or not.
  - Changes are in FrontEnd/src/Components/Dashboard/Bookings/BookingComponents.js
  - FrontEnd/src/Components/Dashboard/DashboardComponents.js
  - FrontEnd/src/Components/Dashboard/Dashboard.js
- Linking back end to the front
  - We have decided that axios is the best option to link our react based front end to our Java backend.
  - This was done easily by adding the axios library to our react library
  - In order to post or get data; we make an axios object that identifies if it is a get or a post request, then in that method we pass in the data to the backend if we are doing a post request.
  - If we need to make a get request we pass in the key id for the specific data base
- Admin create worker
  - User story 7
  - Implement a page that allows an admin user to create a new employee, linking with relevant back end services. This involves creating a component to display a form to capture employee information, and when the form is submitted, it will send a POST request to /user.
  - This contains the Admin page /FrontEnd/src/Components/Dashboard/Admin/Admin.js

- This contains a Form that could be reused to signup the user  
/FrontEnd/src/Components/Form/FormDetails.js
- Admin assign hours to work
  - User story 3
  - Allow an admin to assign hours to an employee, linking with relevant back end service.
  - Have the UI to allow the Admin to set hours for workers.
- Implemented front end testing
  - Add unit tests for Dashboard/DashboardComponents
  - Add unit tests for Booking/BookingComponents
  - Update front-end tests for current page set (sign up)
  - Add unit tests for static front-end pages
  - Add unit tests for Login
- Implemented circleCI pipeline
  - The circleCI pipeline allows for testing and evaluating commits to the github repository, increasing the quality of code that is merged into develop and consequently release and master.
  - See the circleCI pipeline link given above
- Deployed services to AWS
  - We now have the front end and back end services deployed to AWS so that they may be accessed without setting up in your local environment
  - See AWS links provided above
- Moved to google sheets for database documentation
  - Our previous API documentation was done with Swagger however this was too difficult to maintain and read so we migrated to a google sheet.
  - <https://docs.google.com/spreadsheets/d/11sLVYMP3aJ-l-ocQJIMQttMxrNts48WPDepV-bfG4DY/edit?usp=sharing>