

PROJECT REPORT

Hatfield Junior Swimming School



Muhammad Ammad Hassan
(23034939)

7COM1025
Programming for Software Engineers

Mr. Hui Cheng

Mr. Michael Watkins

Mr. Yar Muhammad

ABSTRACT

This report is about a system developed for Hatfield Junior Swimming School (HJSS). It is a console-based Java application. The application's main objective is to manage learner's swimming lessons. Alongside handling the lesson bookings, it also registers new learners and generates monthly reports. It is designed according to the learner's perspective. Additionally, it is developed by not using any external database and stores the required data in a simple file using JSON format.

INTRODUCTION

The Hatfield Junior Swimming School (HJSS) required software to maintain the bookings of swimming lessons made by the learners. The school offers swimming lessons for Grades 1, 2, 3, 4, and 5. The application should allow the learners to book a swimming lesson, change a swimming lesson, cancel a swimming lesson, and mark a swimming lesson attended with the feedback. The application should register a new learner. The application must print monthly reports of the learners and coaches.

A learner who wants to book a swimming lesson can view the timetable in three ways i.e., day, lesson grade, and coach name. A learner should be allowed to either change or cancel the booked lesson. After attending a lesson, the learner should give feedback by rating the lesson and writing a review. An application should maintain the record of learners and also be able to register a new learner. An application must provide an option to generate reports about the learners and the coaches based on the month.

The school runs on Monday, Wednesday, Friday, and Saturday. Monday, Wednesday, and Friday contain 3 lessons each day respectively with the time slots of (4 pm to 5 pm), (5 pm to 6 pm), and (6 pm to 7 pm). Saturday contains two lessons with the time slots (2 pm to 3 pm) and (3 pm to 4 pm). Each swimming lesson can accommodate a maximum of 4 learners. Detailed requirements about the system are mentioned in the System Requirements section.

SYSTEM REQUIREMENTS

The basic requirements of the system are already defined. The developed system provides at least 7 major functions in the command line interface.

1. Book a swimming lesson
2. Change / Cancel a swimming lesson
 - 2.1. Change a swimming lesson
 - 2.2. Cancel a swimming lesson
3. Attend a swimming lesson
4. Generate monthly reports
 - 4.1. Monthly report of Learners
 - 4.2. Monthly report of Coaches
5. Register a new learner

Other requirements of the system that are defined are also implemented. They are listed below:

- 1) The system must display the timetable of lessons in three ways i.e., day, grade and coach name.
- 2) Each swimming lesson can accommodate a maximum of 4 learners.
- 3) The duration of each lesson is 1 hour.
- 4) A learner can only book the lesson of the same grade or one grade higher.
- 5) A learner can't book the same lesson twice.
- 6) A learner can't book the lesson if slots are not available.
- 7) When a learner books a lesson, reduce one slot from that lesson.
- 8) A learner is allowed to change a booked lesson.
- 9) A learner can cancel a booked lesson.
- 10) A learner can write a review and rate the lesson they have attended.
- 11) A review will be stored in a system.
- 12) The system should generate a monthly report of learners which provides information about booked, attended and cancelled lessons.
- 13) The system should generate a monthly report of coaches which provides information about the average rating they received.
- 14) Ask a learner to select the month to generate a report of that month.
- 15) The system should record a learner's name, gender, age, emergency contact number, and current grade level.
- 16) A learner's age must be between 4 and 11.
- 17) A new learner can be registered in a system.

ADDITIONAL FEATURES

The Hatfield Junior Swimming School system also contains some extra functionality not described before. These additional features make the system more efficient, accurate, robust, reliable, usable, understandable, and maintainable. They are listed below:

- 1) Display a main menu, to select a major function.
- 2) The system always validates that the learner selects the correct option to proceed next.
- 3) The system should display an error when a learner makes a mistake.
- 4) Display a try-again menu when a learner when a learner makes a mistake.
- 5) The system asks for a learner ID once either before booking, changing, cancelling, or attending a lesson.
- 6) Once a learner ID is entered, the system does not require to enter a learner ID again until the learner exits the system.
- 7) The timetable shows the lesson from the current date to the upcoming seven days.
- 8) The system automatically generates a booking ID for a new booking.
- 9) Display a yes-no menu to confirm whether the learner wants to select that particular lesson.
- 10) Display the information when a lesson is booked, changed, cancelled, or attended.
- 11) Validates that the booked lesson cannot be changed or cancelled after the lesson is delivered.
- 12) Validates that the learner cannot mark the booked lesson attended before the lesson is delivered.
- 13) When a new learner is registered, the system validates all the fields.
- 14) Display a yes-no menu to get confirmation that the provided details are correct.
- 15) The system automatically generates a learner ID for a new learner.
- 16) The system stores the data of Learners, Lessons, and Coaches in JSON file format.
- 17) Display a system menu, after completing a particular job.
- 18) The system menu asks a learner either to repeat the job, go to the main menu, or exit the system.

DESIGN & STRUCTURE

The system contains 10 Java classes of which 5 classes are major classes of the program. The list of classes and their use is given below:

1) HJSSApplication

It is the main class that starts the application.

2) Main Menu

The class is responsible for displaying major functions and triggering action on those functions.

3) Learner

It represents a learner and contains important functions related to a learner.

4) Lesson

It represents a lesson and contains important functions related to a lesson.

5) Timetable

This class is responsible for generating and displaying the timetable.

6) Reusable Methods

This class contain different methods related to validation or I/O file functions. And it is used by other classes.

7) Report

This class is responsible for generating different reports.

8) Review

This class is responsible for taking feedback when the learner attends a lesson.

9) Coach

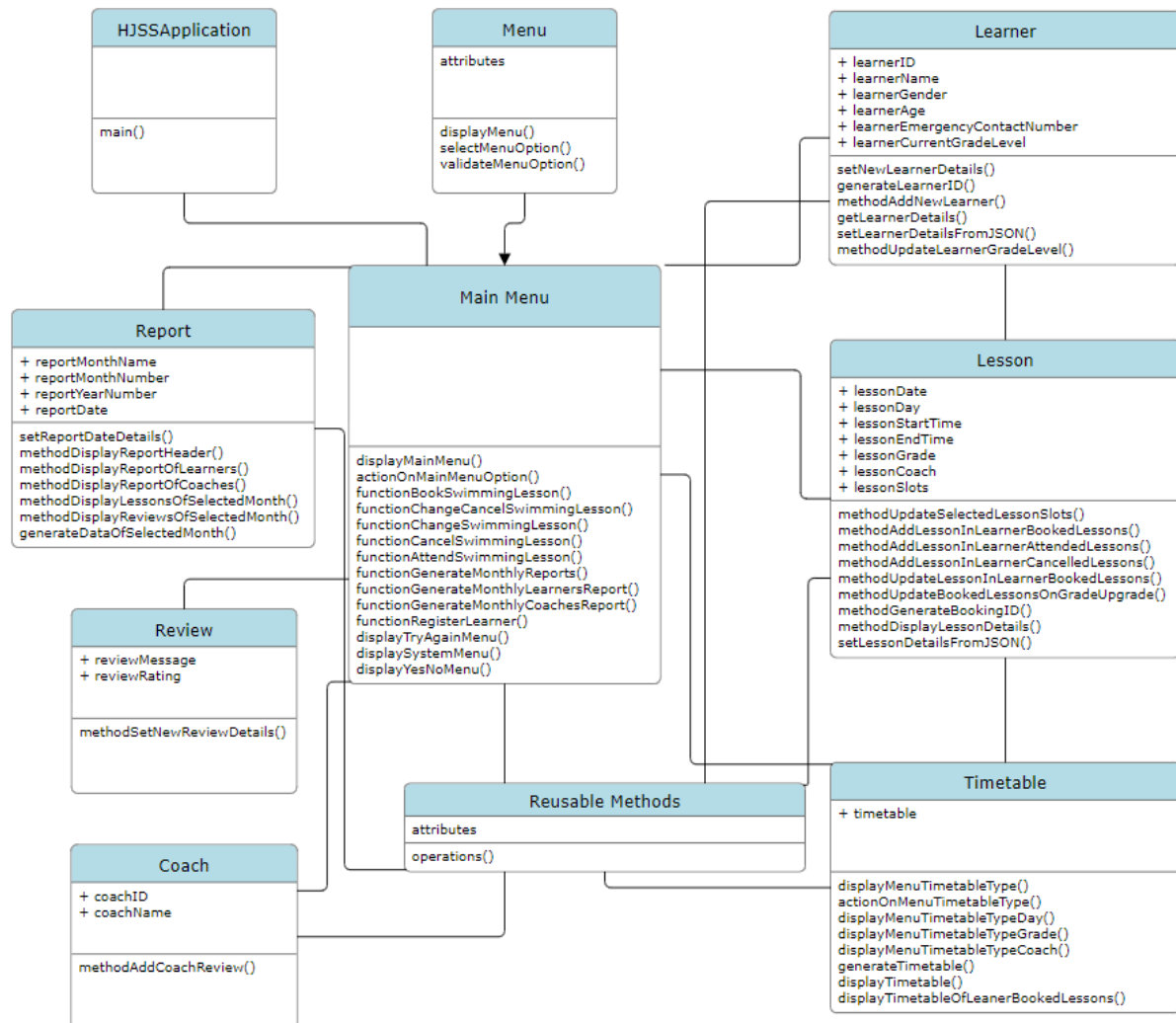
It represents a coach and adds learner feedback to a specified coach.

10) Menu

It is a parent class of the Main Menu. It contains methods how to display and deal with menus.

Class Diagram

The class diagram of the system is shown below.



Data Format

The system stores the data in JSON format. It contains three data files.

1) Learners Data

```
LearnersData.json x
1  [
2    { "learnerID": "SWL1", "learnerName": "Learner A", "learnerGender": "male", "learnerAge": 6, "learnerEmail": "swl1@gmail.com",
3      "learnerLessons": {
4        "booked": [
5          {
6            "bookingID": "s35bau0m", "lessonDate": "2024-04-05", "lessonDay": "Friday", "lessonGrade": 2,
7            "lessonStartTime": "5pm", "lessonEndTime": "6pm", "coachName": "Coach B", "coachID": "HJSSC2"
8          }
9        ],
10      "attended": [
11        {
12          "bookingID": "e95ba6aa", "lessonDate": "2024-04-01", "lessonDay": "Monday", "lessonGrade": 1,
13          "lessonStartTime": "4pm", "lessonEndTime": "5pm",
14          "coachName": "Coach A", "coachID": "HJSSC1", "lessonReviewRating": 5,
15          "lessonReviewMessage": "Nice"
16        }
17      ]
18    }
19  ]
```

2) Lessons Data

```
LessonsData.json x
1  [
2    { "lessonDate": "2024-04-01", "lessonDay": "Monday", "lessonStartTime": "4pm", "lessonEndTime": "5pm", "lessonGrade": 1, "coachID": "HJSSC1", "coachName": "Coach A", "lessonReviewRating": 5, "lessonReviewMessage": "Nice" },
3    { "lessonDate": "2024-04-01", "lessonDay": "Monday", "lessonStartTime": "5pm", "lessonEndTime": "6pm", "lessonGrade": 2, "coachID": "HJSSC2", "coachName": "Coach B", "lessonReviewRating": 4, "lessonReviewMessage": "Good" },
4    { "lessonDate": "2024-04-01", "lessonDay": "Monday", "lessonStartTime": "6pm", "lessonEndTime": "7pm", "lessonGrade": 3, "coachID": "HJSSC3", "coachName": "Coach C", "lessonReviewRating": 3, "lessonReviewMessage": "Average" },
5    { "lessonDate": "2024-04-03", "lessonDay": "Wednesday", "lessonStartTime": "4pm", "lessonEndTime": "5pm", "lessonGrade": 1, "coachID": "HJSSC1", "coachName": "Coach A", "lessonReviewRating": 5, "lessonReviewMessage": "Nice" },
6    { "lessonDate": "2024-04-03", "lessonDay": "Wednesday", "lessonStartTime": "5pm", "lessonEndTime": "6pm", "lessonGrade": 2, "coachID": "HJSSC2", "coachName": "Coach B", "lessonReviewRating": 4, "lessonReviewMessage": "Good" },
7    { "lessonDate": "2024-04-03", "lessonDay": "Wednesday", "lessonStartTime": "6pm", "lessonEndTime": "7pm", "lessonGrade": 3, "coachID": "HJSSC3", "coachName": "Coach C", "lessonReviewRating": 3, "lessonReviewMessage": "Average" },
8    { "lessonDate": "2024-04-05", "lessonDay": "Friday", "lessonStartTime": "4pm", "lessonEndTime": "5pm", "lessonGrade": 1, "coachID": "HJSSC1", "coachName": "Coach A", "lessonReviewRating": 5, "lessonReviewMessage": "Nice" },
9    { "lessonDate": "2024-04-05", "lessonDay": "Friday", "lessonStartTime": "5pm", "lessonEndTime": "6pm", "lessonGrade": 2, "coachID": "HJSSC2", "coachName": "Coach B", "lessonReviewRating": 4, "lessonReviewMessage": "Good" },
10   { "lessonDate": "2024-04-05", "lessonDay": "Friday", "lessonStartTime": "6pm", "lessonEndTime": "7pm", "lessonGrade": 3, "coachID": "HJSSC3", "coachName": "Coach C", "lessonReviewRating": 3, "lessonReviewMessage": "Average" },
11   { "lessonDate": "2024-04-06", "lessonDay": "Saturday", "lessonStartTime": "2pm", "lessonEndTime": "3pm", "lessonGrade": 1, "coachID": "HJSSC1", "coachName": "Coach A", "lessonReviewRating": 5, "lessonReviewMessage": "Nice" },
12   { "lessonDate": "2024-04-06", "lessonDay": "Saturday", "lessonStartTime": "3pm", "lessonEndTime": "4pm", "lessonGrade": 2, "coachID": "HJSSC2", "coachName": "Coach B", "lessonReviewRating": 4, "lessonReviewMessage": "Good" },
13   { "lessonDate": "2024-04-08", "lessonDay": "Monday", "lessonStartTime": "4pm", "lessonEndTime": "5pm", "lessonGrade": 1, "coachID": "HJSSC1", "coachName": "Coach A", "lessonReviewRating": 5, "lessonReviewMessage": "Nice" },
14   { "lessonDate": "2024-04-08", "lessonDay": "Monday", "lessonStartTime": "5pm", "lessonEndTime": "6pm", "lessonGrade": 2, "coachID": "HJSSC2", "coachName": "Coach B", "lessonReviewRating": 4, "lessonReviewMessage": "Good" },
15   { "lessonDate": "2024-04-08", "lessonDay": "Monday", "lessonStartTime": "6pm", "lessonEndTime": "7pm", "lessonGrade": 3, "coachID": "HJSSC3", "coachName": "Coach C", "lessonReviewRating": 3, "lessonReviewMessage": "Average" },
16   { "lessonDate": "2024-04-10", "lessonDay": "Wednesday", "lessonStartTime": "4pm", "lessonEndTime": "5pm", "lessonGrade": 1, "coachID": "HJSSC1", "coachName": "Coach A", "lessonReviewRating": 5, "lessonReviewMessage": "Nice" }
17 ]
```

3) Coaches Data

```
CoachesData.json x
1  [
2    {
3      "coachID": "HJSSC1",
4      "coachName": "Coach A",
5      "coachReviews": [
6        {
7          "bookingID": "e95ba6aa", "learnerID": "SWL1", "learnerName": "Learner A", "lessonDate": "2024-04-01", "lessonDay": "Monday", "lessonGrade": 1,
8          "lessonStartTime": "4pm", "lessonEndTime": "5pm",
9          "lessonReviewRating": 5, "lessonReviewMessage": "Nice"
10         },
11         {
12          "bookingID": "5k5ba6aa", "learnerID": "SWL4", "learnerName": "Learner D", "lessonDate": "2024-04-01", "lessonDay": "Monday", "lessonGrade": 1,
13          "lessonStartTime": "4pm", "lessonEndTime": "5pm",
14          "lessonReviewRating": 4, "lessonReviewMessage": "Nice"
15         }
16       ]
17     }
18   ]
```


DEVELOPMENT STRATEGIES

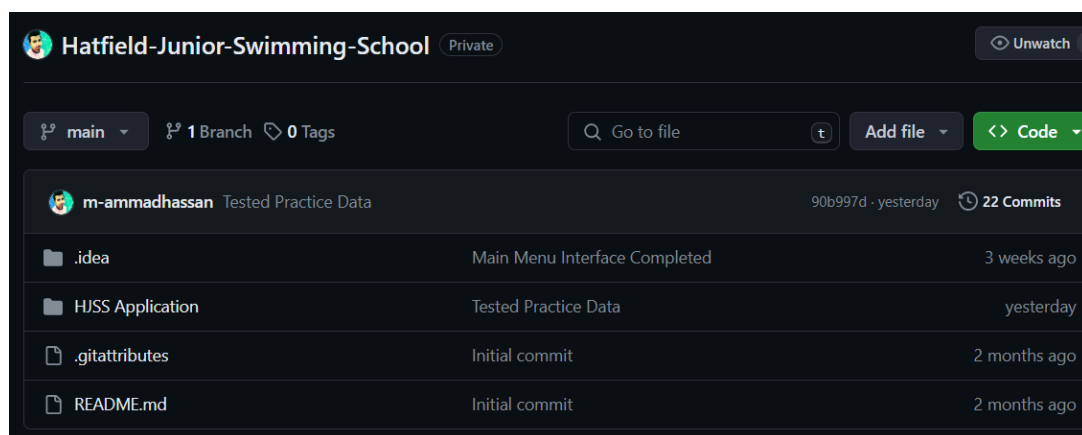
The system is developed in increments. Each new increment is integrated into the previous version of the software.

Version Control System

The system uses the version control system. The version control system is implemented locally and remotely. The complete code is stored in a repo on GitHub.

The link to the GitHub repository is:

<https://github.com/m-ammadhassan/Hatfield-Junior-Swimming-School>



The commits log file is available in the project. Its one screenshot is shown below.

```
Application>git log
commit 90b997dbb692e8ed4002505865d53864601ccc1e (HEAD -> main, origin/main, origin/HEAD)
Author: Muhammad Ammad Hassan <m.ammadhassan@yahoo.com>
Date: Sun Apr 21 21:48:06 2024 +0100

    Tested Practice Data

commit 523bdfe33a934a679bb3f7ca3e2c2293b15f812e
Author: Muhammad Ammad Hassan <m.ammadhassan@yahoo.com>
Date: Sun Apr 21 21:47:50 2024 +0100

    Modified Review Class

commit 4294db27da28d8b76c8c7d8eb09acbac52f0ba3e
Author: Muhammad Ammad Hassan <m.ammadhassan@yahoo.com>
Date: Sun Apr 21 21:47:35 2024 +0100

    Modified Reusable Methods Class

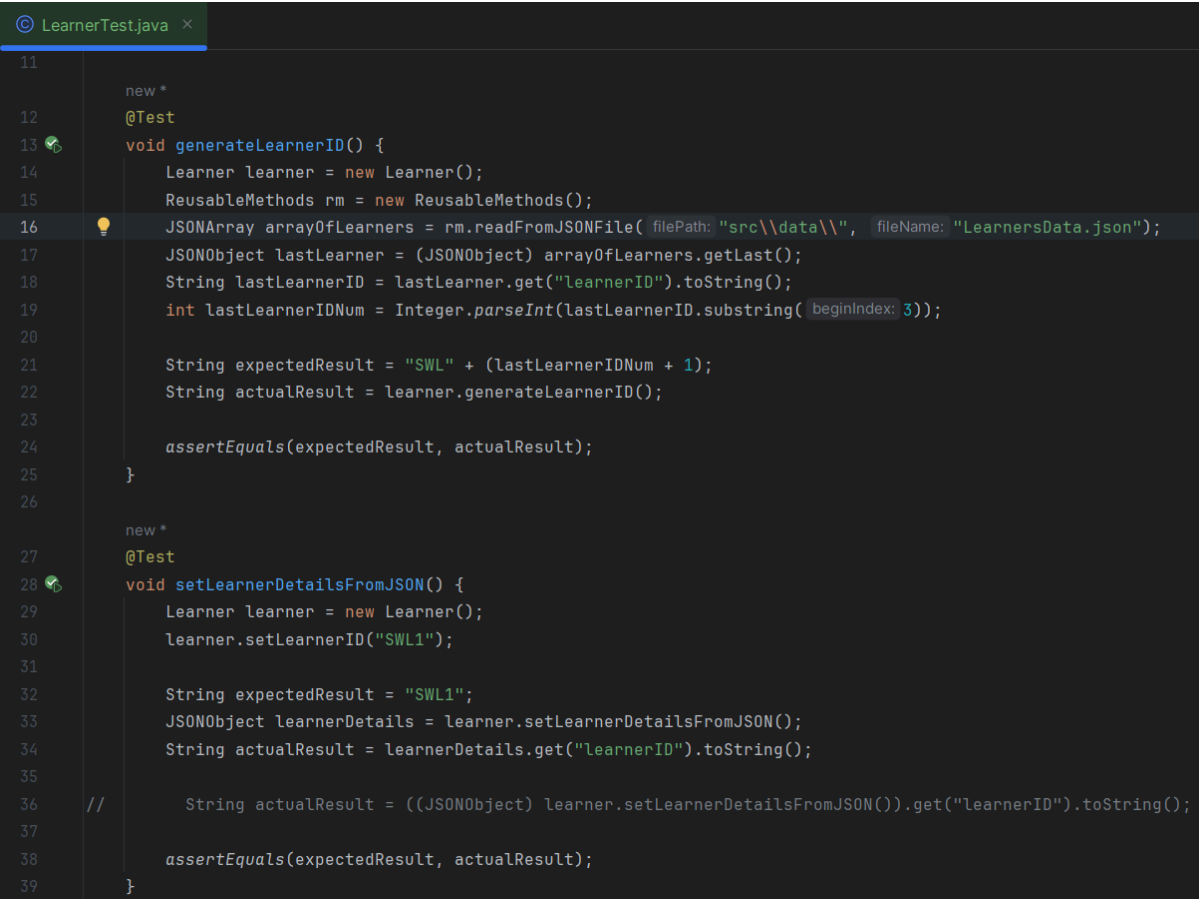
commit 1c1426d5a9e7af01e1119472f6455fae908c9ab7
Author: Muhammad Ammad Hassan <m.ammadhassan@yahoo.com>
Date: Sun Apr 21 21:47:13 2024 +0100

    Modified Report Class

commit d995b5cce41e68454908889c65eb414acd820901
Author: Muhammad Ammad Hassan <m.ammadhassan@yahoo.com>
Date: Sun Apr 21 21:46:49 2024 +0100
```


TESTING

The system is thoroughly tested using Black Box Testing. Some of its components are also unit-tested. For unit testing, version 5 of jUnit is used. Some of the screenshots are displayed below.



```
11
12 new *
13 @Test
14 void generateLearnerID() {
15     Learner learner = new Learner();
16     ReusableMethods rm = new ReusableMethods();
17     JSONArray arrayOfLearners = rm.readFromJSONFile( filePath: "src\\data\\", fileName: "LearnersData.json");
18     JSONObject lastLearner = (JSONObject) arrayOfLearners.getLast();
19     String lastLearnerID = lastLearner.get("learnerID").toString();
20     int lastLearnerIDNum = Integer.parseInt(lastLearnerID.substring( beginIndex: 3));
21
22     String expectedResult = "SWL" + (lastLearnerIDNum + 1);
23     String actualResult = learner.generateLearnerID();
24
25     assertEquals(expectedResult, actualResult);
26 }
27
28 new *
29 @Test
30 void setLearnerDetailsFromJSON() {
31     Learner learner = new Learner();
32     learner.setLearnerID("SWL1");
33
34     String expectedResult = "SWL1";
35     JSONObject learnerDetails = learner.setLearnerDetailsFromJSON();
36     String actualResult = learnerDetails.get("learnerID").toString();
37
38     // String actualResult = ((JSONObject) learner.setLearnerDetailsFromJSON()).get("learnerID").toString();
39
40     assertEquals(expectedResult, actualResult);
41 }
```

```

MenuTest.java x
8
new *
9
@Test
10 void validateMenuOption() {
11     Menu menu = new Menu();
12
13     // If menu option is a number and is greater than or equal to minimum value and is less than or equal to maximum value
14     boolean expectedResult = true;
15     boolean actualResult = menu.validateMenuOption(menuOption: "2", min: 1, max: 5);
16     assertEquals(expectedResult, actualResult);
17
18     // If menu option is not a number
19     boolean expectedSecondResult = false;
20     boolean actualSecondResult = menu.validateMenuOption(menuOption: "alpha", min: 1, max: 5);
21     assertEquals(expectedSecondResult, actualSecondResult);
22
23     // If menu option is less than minimum value
24     boolean expectedThirdResult = false;
25     boolean actualThirdResult = menu.validateMenuOption(menuOption: "-1", min: 1, max: 5);
26     assertEquals(expectedThirdResult, actualThirdResult);
27
28     // If menu option is greater than maximum value
29     boolean expectedFourthResult = false;
30     boolean actualFourthResult = menu.validateMenuOption(menuOption: "7", min: 1, max: 5);
31     assertEquals(expectedFourthResult, actualFourthResult);
32 }
33 }

```

```

ReusableMethodsTest.java x
17
new *
18 @Test
19 void validateName() {
20     ReusableMethods rm = new ReusableMethods();
21
22     // If name contain only two or more alphabets
23     boolean expectedResult = true;
24     boolean actualResult = rm.validateName("Learner Name");
25     assertEquals(expectedResult, actualResult);
26
27     // If name is empty
28     boolean expectedSecondResult = false;
29     boolean actualSecondResult = rm.validateName("");
30     assertEquals(expectedSecondResult, actualSecondResult);
31
32     // If name contains only single alphabet
33     boolean expectedThirdResult = false;
34     boolean actualThirdResult = rm.validateName("A");
35     assertEquals(expectedThirdResult, actualThirdResult);
36
37     // If name contain numbers
38     boolean expectedFourthResult = false;
39     boolean actualFourthResult = rm.validateName("Learner123");
40     assertEquals(expectedFourthResult, actualFourthResult);
41 }

```

RESULTS

```
---> Hatfield Junior Swimming School <---
```

- 1) Book a Swimming Lesson
- 2) Change / Cancel Booking
- 3) Attend a Swimming Lesson
- 4) Monthly Reports
- 5) Register a New Learner
- 6) Exit

```
Enter your choice [1-6] :
```

```
||=====||=====||=====|| TIMETABLE ||=====||=====||=====||
```

```
1 - DATE --> 2024-04-26      DAY --> Friday      TIME --> 4pm-5pm
    GRADE --> 1              COACH --> Coach B
    AVAILABLE SLOTS --> 4
```

```
2 - DATE --> 2024-04-26      DAY --> Friday      TIME --> 5pm-6pm
    GRADE --> 2              COACH --> Coach A
    AVAILABLE SLOTS --> 4
```

```
3 - DATE --> 2024-04-26      DAY --> Friday      TIME --> 6pm-7pm
    GRADE --> 3              COACH --> Coach D
    AVAILABLE SLOTS --> 4
```

```
Do you want to select a Lesson?
```

- 1) Yes
- 2) No

```
Enter your choice [1-2] : |
```

=====

Monthly Report of Learners

=====

Month Name: April

Year: 2024

=====

Learner ID: SWL1

Name: Learner A

Gender: male

Current Grade: 3

Booked Lesson Details:

1 - Lesson Date: 2024-04-05

Lesson Time: 5pm-6pm

Booking ID: s35bau0m

Lesson Grade: 2

Coach Name: Coach B

Attended Lesson Details:

1 - Lesson Date: 2024-04-01

Lesson Time: 4pm-5pm

Booking ID: e95ba6aa

Lesson Grade: 1

Coach Name: Coach A

2 - Lesson Date: 2024-04-03

Lesson Time: 4pm-5pm

Booking ID: s35bau75

Lesson Grade: 1

Coach Name: Coach A

3 - Lesson Date: 2024-04-10

Lesson Time: 5pm-6pm

Booking ID: nj8ba6aa

Lesson Grade: 2

Coach Name: Coach B

4 - Lesson Date: 2024-04-22

Lesson Time: 5pm-6pm

Booking ID: yu7ba6aa

Lesson Grade: 3

Coach Name: Coach D

Cancelled Lesson Details:

No lessons cancelled in this month

Booked Lessons = 1

Attended Lessons = 4

Cancelled Lessons = 0

CONCLUSION

The system performs all the required functions and also includes some extra functionality. It also needs some more improvements in terms of security protocols, authorization and authentication, registering new lessons, interface for coaches, etc.

REFERENCES

1. 7COM1025 Coursework Explanation Slides 2024 by Dr. Hui Cheng
2. 7COM1025 Coursework Briefing Sheet (2023/24 Academic Year) by Dr. Hui Cheng