Case Study 3: Meeting Scheduling

A PROJECT REPORT

Submitted by

NAME: Olan Healy
StudentID: 21318204

NAME: Kevin Collins

StudentID: 21344256

NAME: Sean Capilis
StudentID: 213423342

in partial fulfilment for the award of the degree

Of

(i)

Computer Systems

in

CS4004: Software Testing and Inspection

TABLE OF CONTENTS

1.0 Cover Page

2.0 Table Of Contents

3.0 **Task 1: Bug Reporting**

3.1 Log 7

3.2 Reflection 8

3.3 Bug Reports 9-18

4.0 Task 2: Test Driven Development

4.1 Log 19

4.2 Reflection 20

4.3 Tests 21-50

5.0 Github Actions CI/CD

5.1 What is github Actions CI/CD? 51-55

5.2 Reflection 55-56

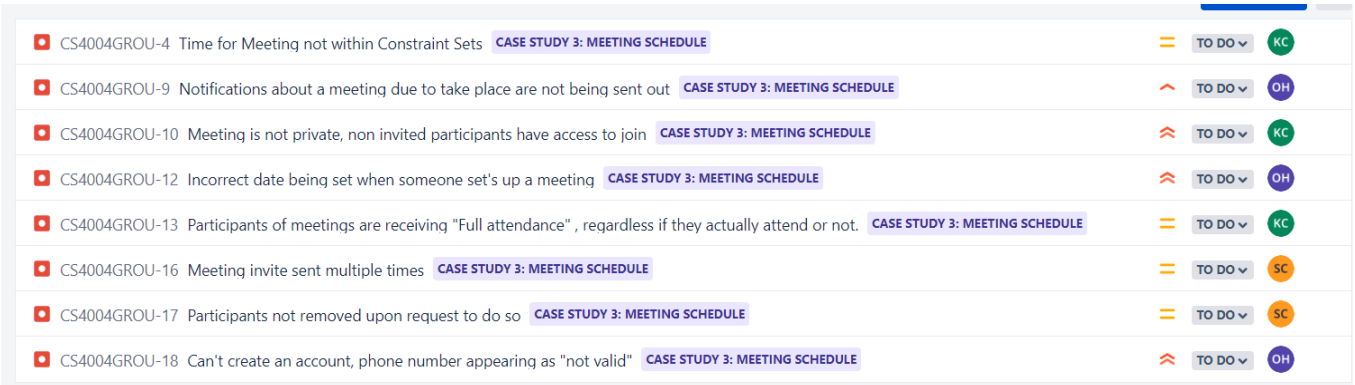
6.0 Summary 56

Task1: Bug Reporting

3.1: Log

List of subtasks:

1. Spelling mistake in security in the settings page
2. Meeting is not private, non invited participants have access to join
3. Notifications about a meeting due to take place are not being sent out
4. Time for Meeting not within Constraint Sets
5. Participants of meetings are receiving "No attendance" , regardless if they actually attend or not.
6. Can't create an account, phone number not valid
7. Participants not being removed from meeting list
8. Meeting invite sent multiple times to participants



3.2: Reflection

Overall, we learnt the process of writing an effective bug. We learnt that you need to be condense and ambiguous in the writing of a bug. It needs to be easy to understand so the software tester can easily know what to look for. It is also important to have a test manager to help easily lead the testing team and make sure the bugs are to a good standard, following the 8 steps of writing a good bug report. We also wrote our bugs using Jira, which we all felt was very beneficial to use as aspiring software developers as big companies such as AER Lingus and Indeed use Jira to help aid their software development process in an agile environment.

3.3: Bug Reports

3.3.1: Bug report 1: Spelling mistake in security in the settings page

Projects / Cs4004GroupProject / CS4004GROU-1 / CS4004GROU-11

Spelling mistake in security in the settings page

Attach | Add a child issue | Link issue

Description

In the settings page, the "security" button is shown as "secuirity"

Steps to reproduce:

1. From the home page, navigate to settings icon in the top left hand side of the screen
2. Navigate down 3 columns until you see a key icon and hover over it
3. Notice the pop-up spelling is spelt incorrectly

Expected result:

Once you hover over the key icon, a pop up should appear called "security".

Actual result:

The pop up is appearing with a spelling "secuirity"

Priority: Lowest

Written by: Olan Healy

What I did: I wanted to start off with a basic bug with low priority and low severity so we could get into the swing of the project. Spelling mistakes are quite common amongst programmers and can be annoying for user's of an application

What I learned: From this bug, I learned that literally anything you can think of could be a bug, whether it be a missing full stop in a sentence, to a blue screen error, they all vary in priority. This bug was very straight forward so I needed to make sure I condensed it efficiently to keep it basic.

3.3.2: Bug Report 2: Meeting is not private, non invited participants have access to join

Meeting is not private, non invited participants have access to join

Attach | Add a child issue | Link issue

Description

In various meetings, user's who have not been invited can simply join someone's private meeting by clicking on their name and then there meeting status

Steps to reproduce:

1. Click the search bar
2. Search for a persons name who is in a meeting eg John Hayes
3. Click on the ""meeting status" button which should appear under their name (green icon indicating they are in a meeting)
4. An option to join should appear

Expected result:

Should not be given a option to join someone's private meeting

Actual result:

You are able to join someone's private meeting

Priority Highest

Written by: Kevin Collins

What I did: Privacy is key in meetings. This is a bug of high severity and high priority as it affects user's privacy and you never want non intended people in any type of meeting. I found this bug very interesting as I knew for the test driven development part, it could be difficult to write

What I learnt: The importance of trying to condense and making my bug as clear as possible. At first, this bug was quite long winded and when showing it to the test manager it didn't

really make sense on what I was looking for and he asked me to go back and improve it.

3.3.3: Bug Report 3: Notifications about a meeting due to take place are not being sent out

Projects / Cs4004GroupProject / CS4004GROU-1 / CS4004GROU-9

Notifications about a meeting due to take place are not being sent out

Attach | Add a child issue | Link issue

Description • Unsaved changes

In the notifications page, user's are not receiving a notification about an upcoming meeting which they should have been invited too

Steps to reproduce:

1. In the home page, select "Notifications".
2. Click and the sort button and select "Newest to Oldest"

Expected result:

Should got notification of upcoming meeting

Actual result:

Not getting a notification of an upcoming meeting

Priority High

Written by: Kevin Collins

What I did: A bug in which a meeting is due to take place but is not sending is not of high severity but high enough priority. A user not receiving a notification about an upcoming meeting is quite bad for them as they could miss out on meetings of high importance. This bug is very reproducible as it would happen again and again and again for every meeting being setup

What I learned: I learnt that as this bug is quite reproducible, we could use it to our advantage. In us working on our project for CS4013 in building a restaurant management system, we found a bug with the menu that was reproducible. As this bug is reproducible I realised that if it was a program we were debugging, we could have gotten a lot of information for the software tester to develop their test.

3.3.4: Bug Report 4: Time for Meeting not within Constraint Sets

Projects / Cs4004GroupProject / CS4004GROU-1 / CS4004GROU-4

Time for Meeting not within Constraint Sets

Attach | Add a child issue | Link issue

Description

Time constraints for available times set. In the meetings page, user's are not able to set a meeting time within the time constraints

Steps to reproduce:

1. Select time from available times.
2. Confirm meeting time.

Expected result:

Time is within available times.

Actual result:

Time selected is not within available constraints.

.

Priority = Medium

Written by: Kevin Collins

What I did: This is a severe bug but of medium priority as it would be easy enough to edit the meeting afterwards to the correct date, but the functionality of the program is affected.

What I learned: That the summary is by far the most important part of the bug report. It needed to be clear and concise on what the actual bug is so the software testers could understand it.

3.3.5: Bug Report 5: Participants of meetings are receiving "No Attendance", regardless if they actually attend or not.

Projects / Cs4004GroupProject / CS4004GROU-1 / CS4004GROU-13

Participants of meetings are receiving "No attendance" , regardless if they actually attend or not.

Attach | Add a child issue | Link issue

Description

When a meeting end, The assignee has access to an "Attendance Report" and in this, Every participant's attendance is set to "No attendance".

Steps to reproduce:

1. When an assignee concludes a meeting, they should click on "Attendance report"
2. Every participant invited to the meeting will be displayed here and notice everyone's attendance is set to "No attendance" even though that information is false

Expected result:

Every participant's attendance should correspond to their participation in the meeting eg "attended or no attendance"

Actual result:

Every participant's attendance is set to "No attendance"

Priority = Medium

Written by: Olan Healy

What I did: This bug is of low severity and medium priority. It can be annoying for a meeting organiser to receive an incorrect status report for a meeting. It is not a serious bug as it doesn't affect the actual functionality of the meeting running and is of medium priority as attendance could be taken manually.

What I learned: That bug's can incorporate other bugs. This can link easily to non intended participants who can join a meeting as the methods for the program can easily cross each other. If that bug is to fix if only intended people can join, my bug can use that to incorporate maybe an attendance method. It is also very important in how you word the bug so the software testers can understand it.

3.3.6: Bug Report 6: Can't create an account, phone number appearing as "not valid"

Projects / Cs4004GroupProject / CS4004GROU-1 / CS4004GROU-18

Can't create an account, phone number not valid

Attach | Add a child issue | Link issue

Description

A user is unable to set up their account on the meeting website with their phone number even though they are using a valid phone number

Steps to reproduce:

1. Click "Create an account"
2. Enter Phone number and desired password

Expected Result:

User should be prompted with "Verify Phone number" message

Actual Result:

User is receiving "Invalid Phone number, please try again"

Priority Highest

Activity

OH Add a comment...

Pro tip: press M to comment

Written by: Olan Healy

What I did: Sometimes, User’s have difficulty in setting up an account for various websites, whether it be a tedious captcha or otherwise. Imagine inputting a correct Irish phone number in setting up an account only for the application to say “Incorrect phone Number, please try again. It is a very serious bug and of high priority as if the website can’t generate any new users, it can’t possibly generate any new income

What I learned: That a bug can be improved on as you are testing. It could start off as something small but there are many improvements that you can make to the actual program by first starting off with the bug and test. I also learned the importance of reviewing a bug as originally, I forgot to include the part about it being an attendance report and my group were very confused on what I meant.

3.3.7: Bug Report 7: “Unable to remove participants from meeting list”

Projects / Cs4004GroupProject / CS4004GROU-1 / CS4004GROU-17

Participants not removed upon request to do so

Attach | Add a child issue | Link issue

Description

Participants who decline the meeting invite are not being removed from the meeting group

Reproduce Bug

1. Send meeting invites
2. Participants(s) decline meeting invite

Expected Result

Participant is removed from meeting list

Actual Result

Participant is not removed from meeting list

Priority = Medium

Written By: Sean Caplis

What I did: Participants should be able to have themselves removed from a meeting they do not wish to attend. I created a test to check the meeting list for a participant that should have been removed. I then changed the code and was able to create code that could successfully remove the participant and pass the test.

What I learnt: I learnt that creating tests first guarantees at least a degree of functionality for the user. Even if other bugs exist, developing tests first allows for at least the most critical of functionality.

3.3.8: Bug Report 8: "Invite sent multiple times invitees"

Meeting invite sent multiple times

Attach | Add a child issue | Link issue

Description

Meeting confirmations are being sent multiple times to attendees

Steps to reproduce

1. Create meeting
2. Add participants
3. Add time and date
4. Confirm meeting

Expected Result

Meeting confirmation details sent to intended recipients once

Actual Result

Meeting confirmation details sent multiple times to intended recipients

Priority = Medium

Written By: Sean Caplis

What I Did: I created a test to verify that the meeting list only contains one instance of each participant's email address to be used. The test case compares an expected list of emails to the actual list of emails from the MeetingList.

What I learnt: I learned that Test Driven Development can be used to improve user experience. The program is still functional if meeting invites are sent multiple times, however it would be incredibly frustrating for users. Ensuring Invites are only sent once allows for greater user experience.

4: Task 2: Test Driven Development:

4.1: log

List of subtasks:

1. Test for Spelling mistake in security in the settings page
2. Test For Meeting is not private, non invited participants have access to join
3. Test for Notifications about a meeting due to take place are not being sent out
4. Test for Time for Meeting not within Constraint Sets
5. Test for Participants of meetings are receiving "No attendance" , regardless if they actually attend or not.
6. Test for Can't create an account, phone number not valid
7. Test for Participants not being removed from meeting list
8. Test for Meeting invite sent multiple times to participants

4.2: Reflection: Overall, we learnt that although TDD is an old technique, it is very useful for reducing the density of unit test failures. It is a unique and also challenging way of building a program but gave us a lot of insight on how to actually look for bugs. Coverage based test design gave us a good understanding on how to properly organise a test. You need to look at what you are actually testing which needs to be satisfied and covering all possible outcomes e.g. looking at below or above the requirements in the test and that they subsume each other. It was a very interesting project and we feel will definitely benefit us in our future college and working careers.

4.3: Tests

4.3.1.1 Test for Spelling mistake in security in the settings page

This test was all about the basics and the method used was unit testing as when writing the test, I realised it was just a problem with the way security was spelt. This was unit testing as I was just testing the security unit.

4.3.1.2 code developed for test

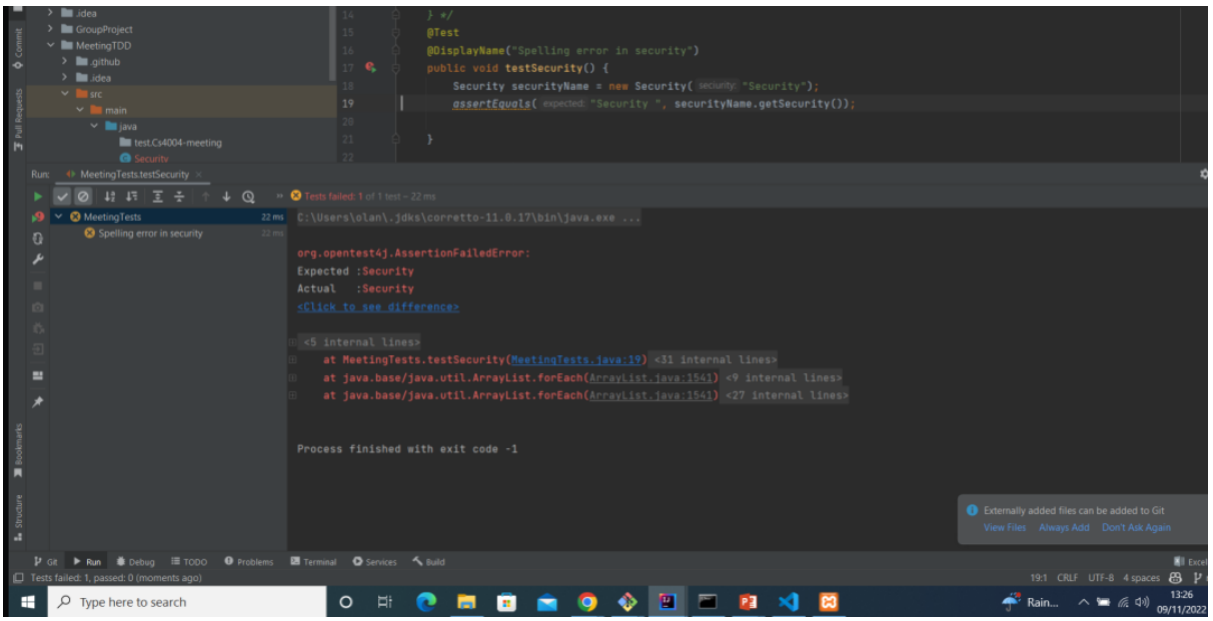
```java
    @Test
    @DisplayName("Spelling error in security")
    public void testSecurity() {
        Security securityName = new Security( security: "Security");
        assertEquals( expected: "Security ", securityName.getSecurity());

    }
```

4.3.1.3 code developed for program

```java
public class Security {

    2 usages
    private String security;

    1 usage   Olan *
    public Security(String Security) {
        this.security = Security;
    }

    Olan
    public String getSecurity() {
        return security;
    }
}
```

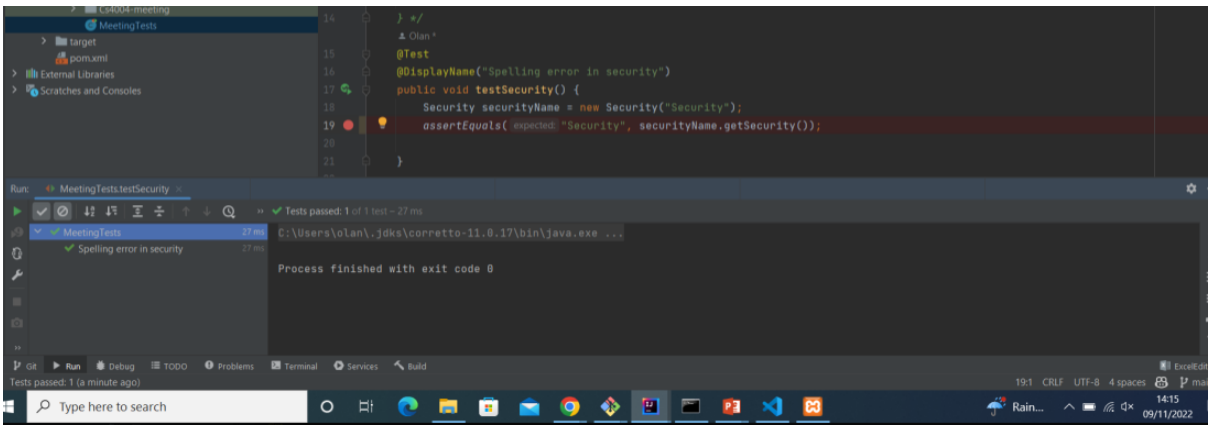
4.3.1.4 Screenshots of success and failure

First test failing:

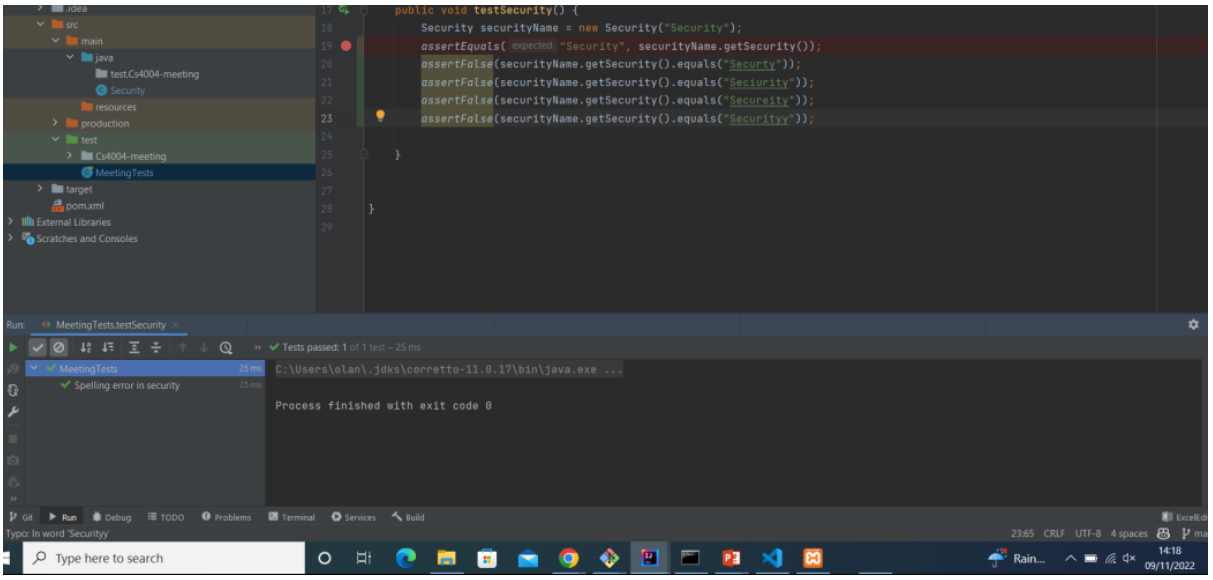


Code refactoring:

Test passing:



Adding more coverage:



4.3.2.1 Test For Meeting is not private, non invited participants have access to join

Privacy is key in meetings. I set up that if a user doesn't have a matching name, email and phone number to one already within the meeting list they cannot join the meeting. Originally I ran into a bug in which the program was not telling the difference between two different meetings. After refactoring I got it to pass. I then realised it should not be possible to join the meeting unless the meeting was underway.

I used LocalTime to compare the current time to the time of the meeting. This took some trial and error to implement as I wasn't too familiar with LocalTime methods. I learned that thinking of tests that can make your code fail will ultimately lead you to coming up with methods you may not have thought needed initially. For example adding the time restrictions. The cycle of Test Driven Development allows for a better understanding of the processes involved.

4.3.2.2 Code developed for test

```
public class nonIntendedParticipant {
    @Test
    @DisplayName("Meeting is private, invited participants have access to join")
    public void IntendedParticipantsCanJoinOnlineMeeting() {
        Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "00:00",  endTime: "23:59",  room: "Room 1");
        //meeting.getAttendees();
        meeting.addAttendees(meeting.getMeetingList());
        assertTrue(meeting.joinMeeting(new People( name: "Kevin",  email: "kev@hotmail.com",  phone: "+353 0868604574")));
    }
```

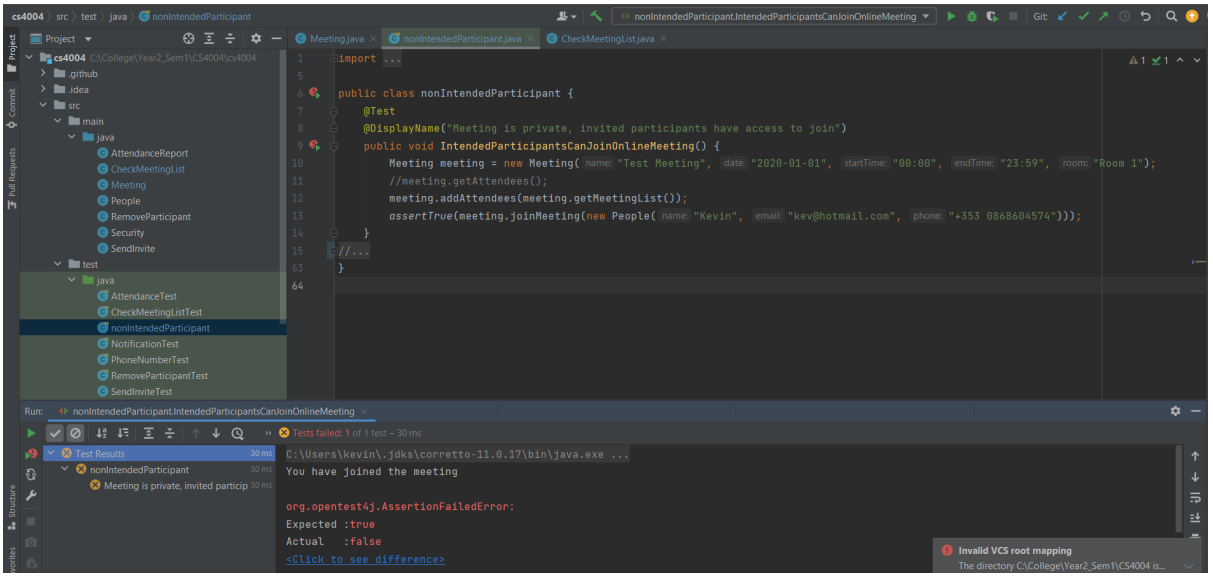
4.3.2.3 Code developed for program

```
public boolean joinMeeting(People people) {
    boolean joined = false;

    //check current time between start and end time
    if(checkAttendees(people)){
        System.out.println("You have joined the meeting");
        joined = false;
    } else {
        System.out.println("You are not invited to this meeting");
    }
    return joined;
}
```

```
public boolean checkAttendees( People people) {

    //if the meeting list contains the people

    for(People p : meetingList) {
        if(p.getName().equals(people.getName()) && p.getEmail().equals(people.getEmail())
                && p.getPhone().equals(people.getPhone())) {
            return true;
        }
    }
    return false;
}
```

4.3.2.4 Screenshot of success and failure

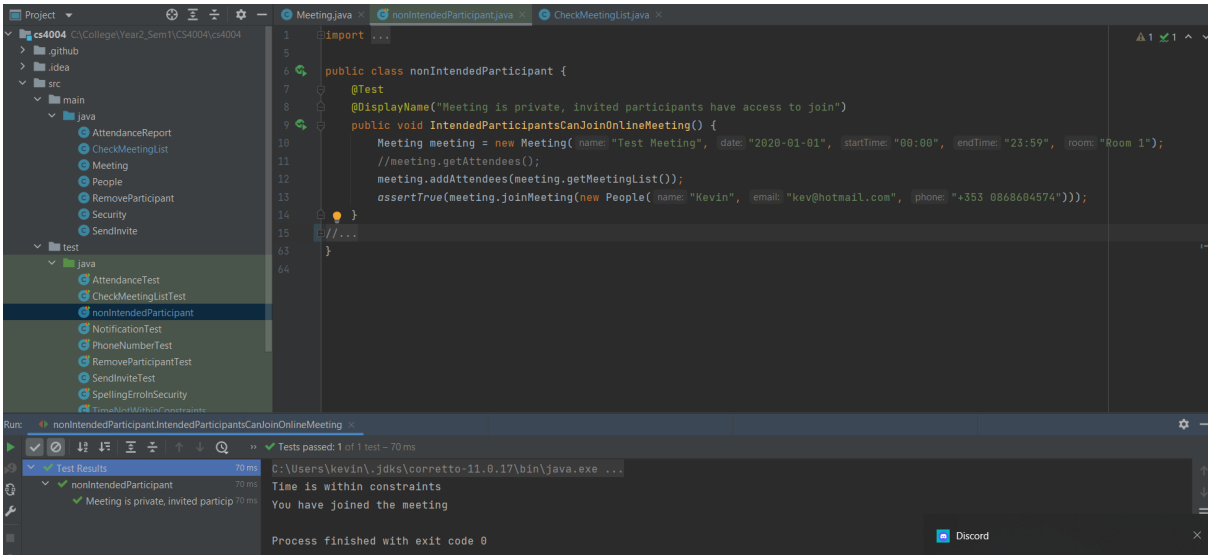
First test failing



Refactoring of code

```
public boolean joinMeeting(People people) {
    boolean joined = false;

    //check current time between start and end time
    if(checkAttendees(people)){
        System.out.println("You have joined the meeting");
        joined = true;
    } else {
        System.out.println("You are not invited to this meeting");
    }
    return joined;
}

public boolean checkAttendees( People people) {

    //if the meeting list contains the people

    for(People p : meetingList) {
        if(p.getName().equals(people.getName()) && p.getEmail().equals(people.getEmail())
                && p.getPhone().equals(people.getPhone())) {
            return true;
        }
    }
    return false;
}
```

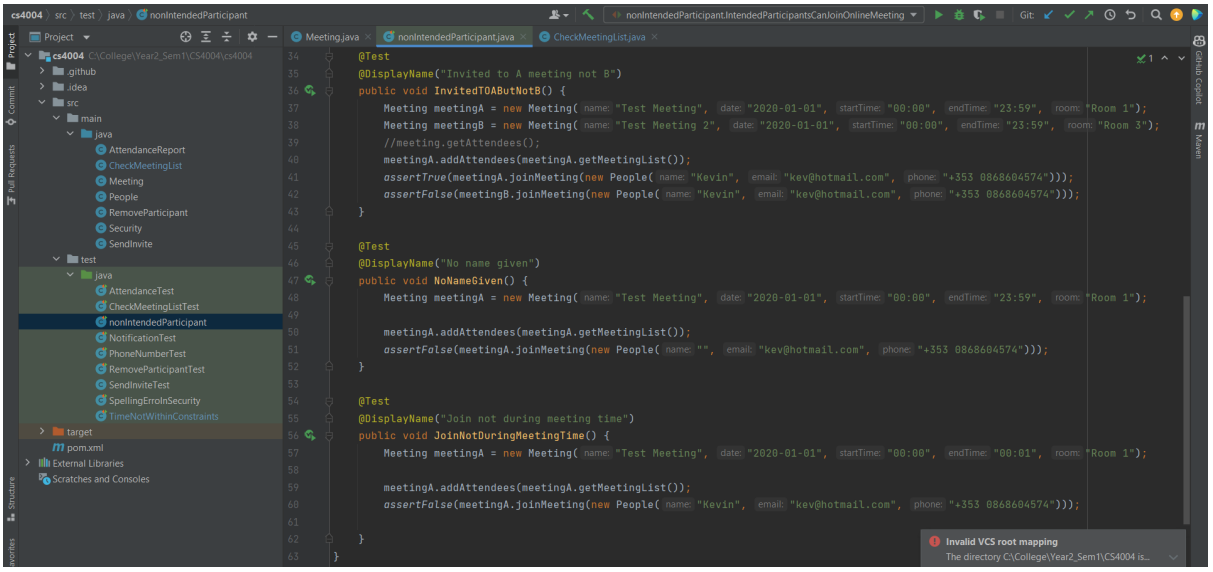
First test passing



Additional Coverage

```
public class nonIntendedParticipant {
    @Test
    @DisplayName("Meeting is private, invited participants have access to join")
    public void IntendedParticipantsCanJoinOnlineMeeting() {
        Meeting meeting = new Meeting( name: "Test Meeting", date: "2020-01-01", startTime: "00:00", endTime: "23:59", room: "Room 1");
        //meeting.getAttendees();
        meeting.addAttendees(meeting.getMeetingList());
        assertTrue(meeting.joinMeeting(new People( name: "Kevin", email: "kev@hotmail.com", phone: "+353 0868604574")));
    }

    @Test
    @DisplayName("Meeting is private, non invited participants cannot join")
    public void NonIntendedParticipantsCantJoinOnlineMeeting() {
        Meeting meeting = new Meeting( name: "Test Meeting", date: "2020-01-01", startTime: "00:00", endTime: "23:59", room: "Room 1");
        //meeting.getAttendees();
        meeting.addAttendees(meeting.getMeetingList());
        assertFalse(meeting.joinMeeting(new People( name: "Ronny", email: "rontheswan@hotmail.com", phone: "+353 0868604574")));
    }

    @Test
    @DisplayName("Name and email match but phone doesn't")
    public void NonIntendedParticipantCantJoinAsNumbersDontMatch() {
        Meeting meeting = new Meeting( name: "Test Meeting", date: "2020-01-01", startTime: "00:00", endTime: "23:59", room: "Room 1");
        //meeting.getAttendees();
        meeting.addAttendees(meeting.getMeetingList());
        assertFalse(meeting.joinMeeting(new People( name: "Kevin", email: "kev@hotmail.com", phone: "+353 0861234567")));
    }
```

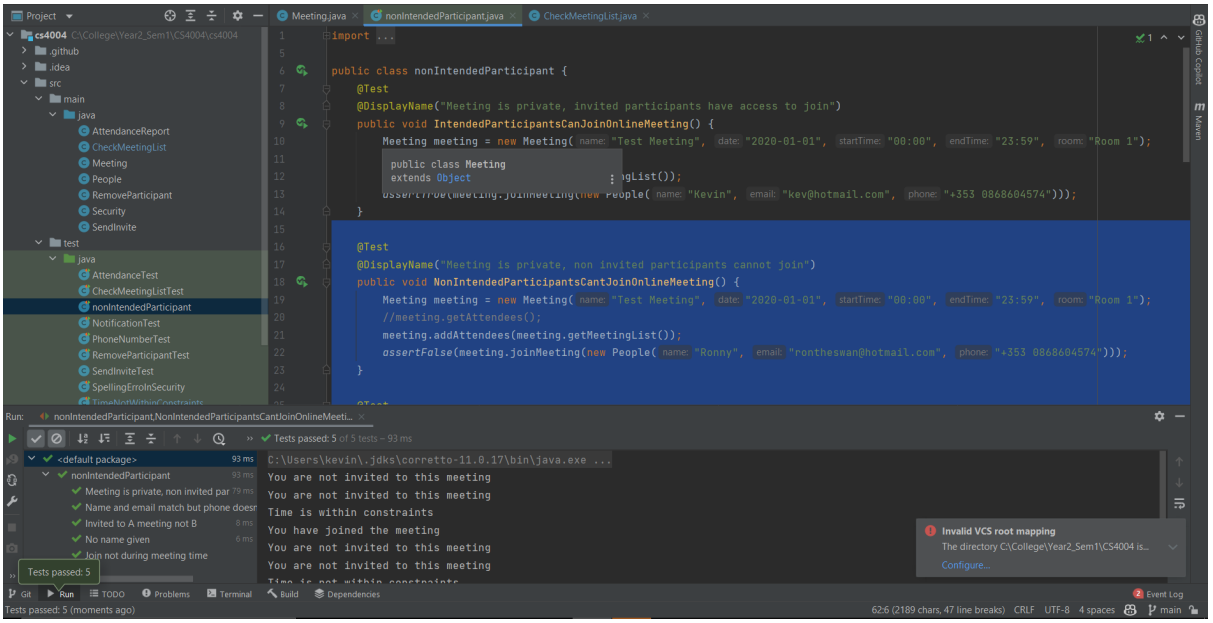
Invalid VCS root mapping
The directory C:\College\Year2_Sem1\CS4004 is...



Code Enhancement

```java
public boolean joinMeeting(People people) {
    boolean joined = false;
    int hour = LocalDateTime.now().getHour();
    int min = LocalDateTime.now().getMinute();
    String time = hour + ":" +  min;
    //check current time between start and end time
    if(checkAttendees(people) && checkTime(time)){
        System.out.println("You have joined the meeting");
        joined = true;
    } else {
        System.out.println("You are not invited to this meeting");
    }
    return joined;
}
```

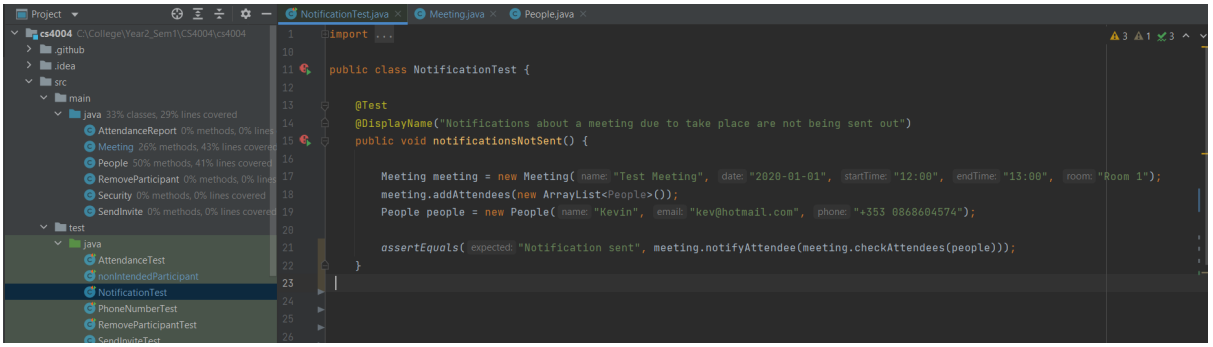
All test’s still Passing



4.3.3.1 Test for Notifications about a meeting due to take place are not being sent out

I created a test in which notifications first were being sent out and then failing to send. I had to build a method that notified the person if the person was within the list. I left the notify method empty as was not required from the test. One thing that became apparent was the need for simplicity. Sometimes you don't need to write out the entire code for a method. Just leaving an empty method can suffice until a further time when code is needed.

4.3.3.2 Code developed for test

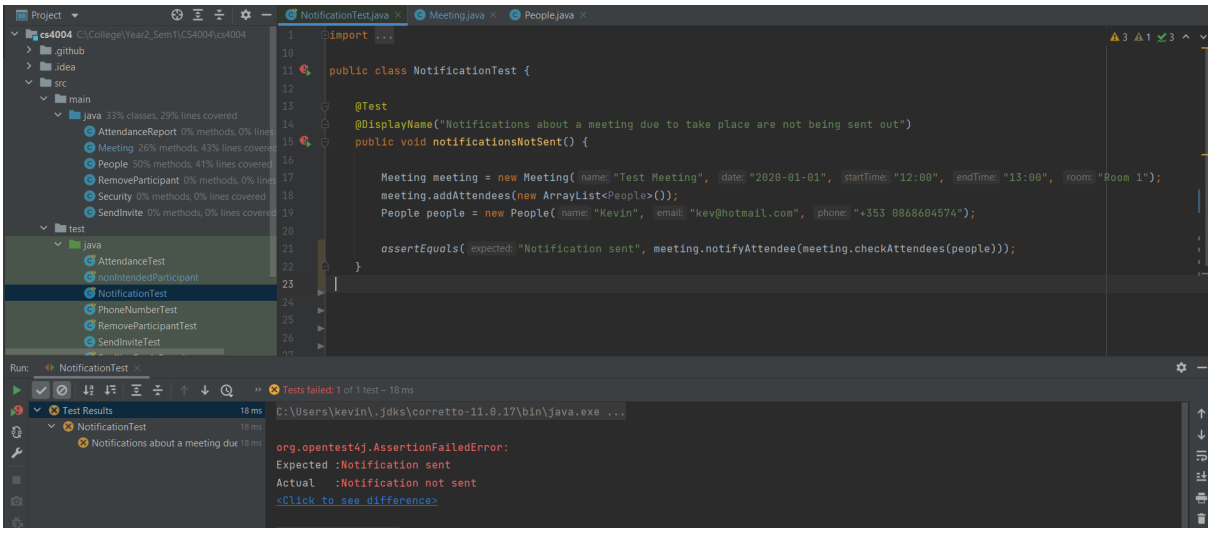


4.3.3.3 Code developed for program

```java
public String notifyAttendee(boolean val) {
    if ( val == true) {

        //method to send notification
        return "Notification sent";
    }
    return "Notification not sent";
}

3 related problems
public boolean checkAttendees( People people) {

    //if the meeting list contains the people
    boolean val;
    if (getMeetingList().contains(people)) {
        return true;
    } else {
        return false;
    }
}
```

4.3.3.4 Screenshot of success and failure

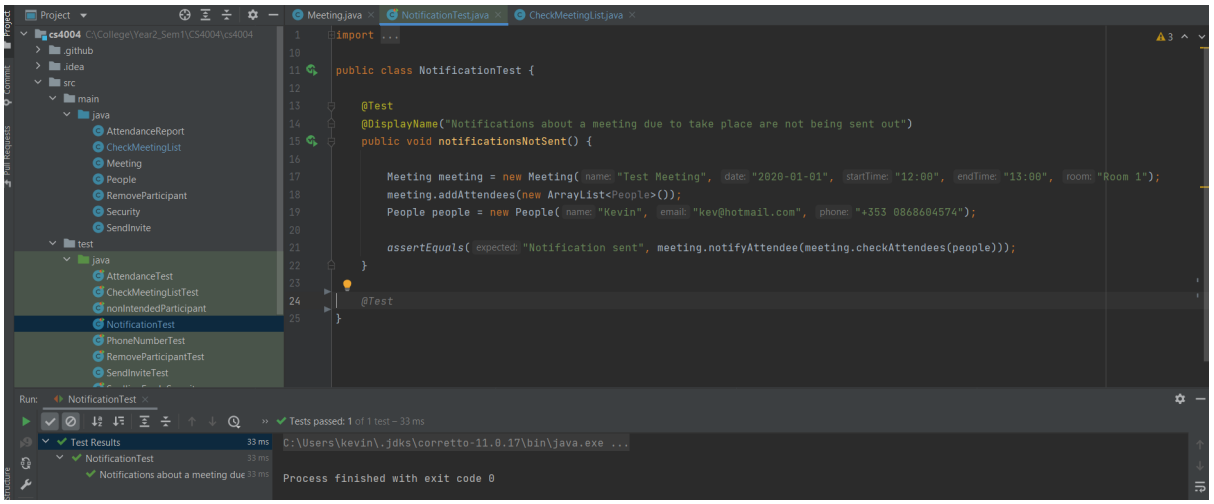
First test failing



Code Enhancement

```java
public boolean checkAttendees( People people) {

    //if the meeting list contains the people

    for(People p : meetingList) {
        if(p.getName().equals(people.getName()) && p.getEmail().equals(people.getEmail())
                && p.getPhone().equals(people.getPhone())) {
            return true;
        }
    }
    return false;
}

public String notifyAttendee(boolean val) {
    if ( val == true) {

        //method to send notification
        return "Notification sent";
    }
    return "Notification not sent";
}
```

First test passing



Additional Coverage

```
import ...

public class NotificationTest {

    @Test
    @DisplayName("Notifications about a meeting due to take place are not being sent out")
    public void notificationsNotSent() {

        Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
        meeting.addAttendees(new ArrayList<People>());
        People people = new People( name: "Kevin",  email: "kev@hotmail.com",  phone: "+353 0868604574");

        assertEquals( expected: "Notification sent", meeting.notifyAttendee(meeting.checkAttendees(people)));
    }

    @Test
    @DisplayName("Notifications about a meeting due to take place are being sent out")
    public void notificationsSent() {

        Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
        meeting.addAttendees(new ArrayList<People>());
        People people = new People( name: "Kevin",  email: "kev@hotmail.com",  phone: "+353 0868604574");

        assertEquals( expected: "Notification sent", meeting.notifyAttendee(meeting.checkAttendees(people)));
    }
```

```
C:\Users\kevin\.jdks\corretto-11.0.17\bin\java.exe ...

Process finished with exit code 0
```

All Tests Passing

```
import ...

public class NotificationTest {

    @Test
    @DisplayName("Notifications about a meeting due to take place are not being sent out")
    public void notificationsNotSent() {

        Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
        meeting.addAttendees(new ArrayList<People>());
        People people = new People( name: "Kevin",  email: "kev@hotmail.com",  phone: "+353 0868604574");

        assertEquals( expected: "Notification sent", meeting.notifyAttendee(meeting.checkAttendees(people)));
    }

    @Test
    @DisplayName("Notifications about a meeting due to take place are being sent out")
    public void notificationsSent() {

        Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
        meeting.addAttendees(new ArrayList<People>());
```

```
C:\Users\kevin\.jdks\corretto-11.0.17\bin\java.exe ...

Process finished with exit code 0
```

4.3.4.1 Test for Time for Meeting not within Constraint Sets

I wished to create a set of parameters to limit the scope of the meetings. It can be used to place constraints on the

possible meeting dates. It sets a preferred start and end date. This in theory would be used by the company to synchronise schedules with meeting attendees. For example, with the move to online a worker may only be available in person from Monday to Wednesday within the week. I wrote the initial test with a start and end date. After I got the tests to pass, I realised I shouldn't be able to create a date for the meeting in the past. Obviously this would be nonsensical. I created a new constraint only allowing meet times after the current date. This was integration testing as I was still testing on the meeting unit.

4.3.4.2 code developed for test

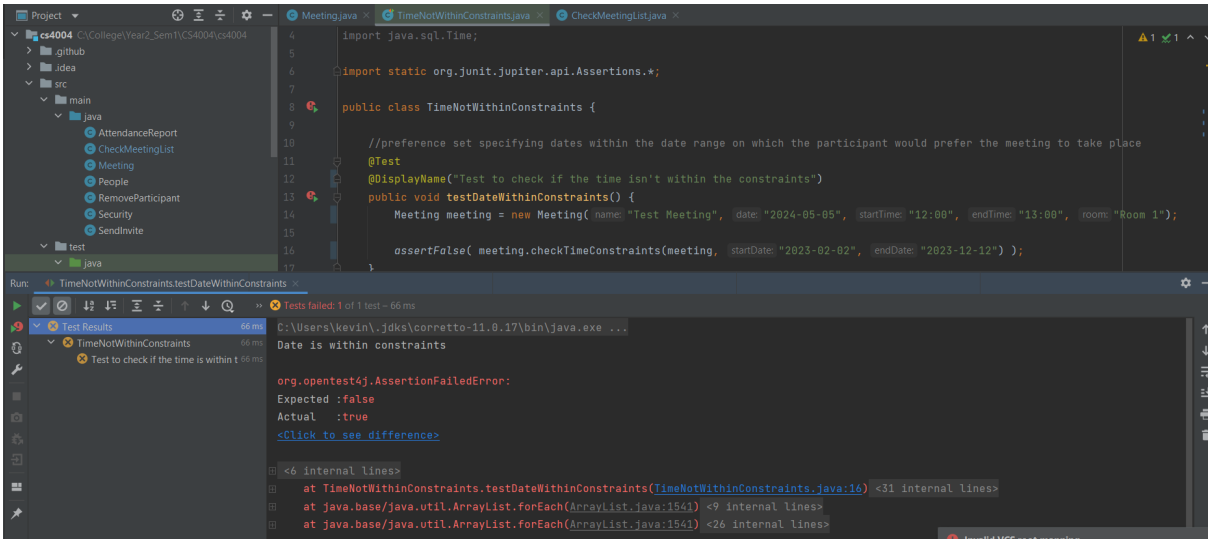
```
public class TimeNotWithinConstraints {

    //preference set specifying dates within the date range on which the participant would prefer the meeting to take place
    @Test
    @DisplayName("Test to check if the time isn't within the constraints")
    public void testDateWithinConstraints() {
        Meeting meeting = new Meeting( name: "Test Meeting",  date: "2024-05-05",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");

        assertFalse( meeting.checkTimeConstraints(meeting,  startDate: "2023-02-02",  endDate: "2023-12-12") );
    }
```

4.3.4.3 Code developed for program

```
public boolean checkTimeConstraints(Meeting meeting, String startDate, String endDate) {

    LocalDate date = LocalDate.parse(meeting.getDate());
    LocalDate strDate = LocalDate.parse(startDate);
    LocalDate edDate = LocalDate.parse(endDate);
    if (date.isAfter(strDate)) {
        System.out.println("Date is within constraints");
        return true;
    } else {
        System.out.println("Date is not within constraints");
        return false;
    }
}
```

4.3.4.4 Screenshot of success and failure

First test failing



Code Refactoring

```
public boolean checkTimeConstraints(Meeting meeting, String startDate, String endDate) {

    LocalDate date = LocalDate.parse(meeting.getDate());
    LocalDate strDate = LocalDate.parse(startDate);
    LocalDate edDate = LocalDate.parse(endDate);
    if (date.isAfter(strDate) && date.isBefore(edDate)) {
        System.out.println("Date is within constraints");
        return true;
    } else {
        System.out.println("Date is not within constraints");
        return false;
    }
}
```

First test passing

```java
@Test
@DisplayName("Participants of meetings are receiving No attendance , regardless if they actually attend or not.")
public void testAttendance() {


    Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
    People john = new People ( name: "John",  email: "john111@gmail.com",  phone: "+3530876351423");
    People paudie = new People ( name: "Paudie",  email: "paudthelaud@gmail.com",  phone: "+3530866651423");
    meeting.addAttendee(john);
    meeting.addAttendee(paudie);
    meeting.joinMeeting(john);

    assertEquals( expected: "Attended", meeting.checkAttendance(john));
    assertEquals( expected: "No attendance", meeting.checkAttendance(paudie));

}
}
```

```
C:\Users\olan\.jdks\corretto-11.0.17\bin\java.exe ...
You have joined the meeting

Process finished with exit code 0
```

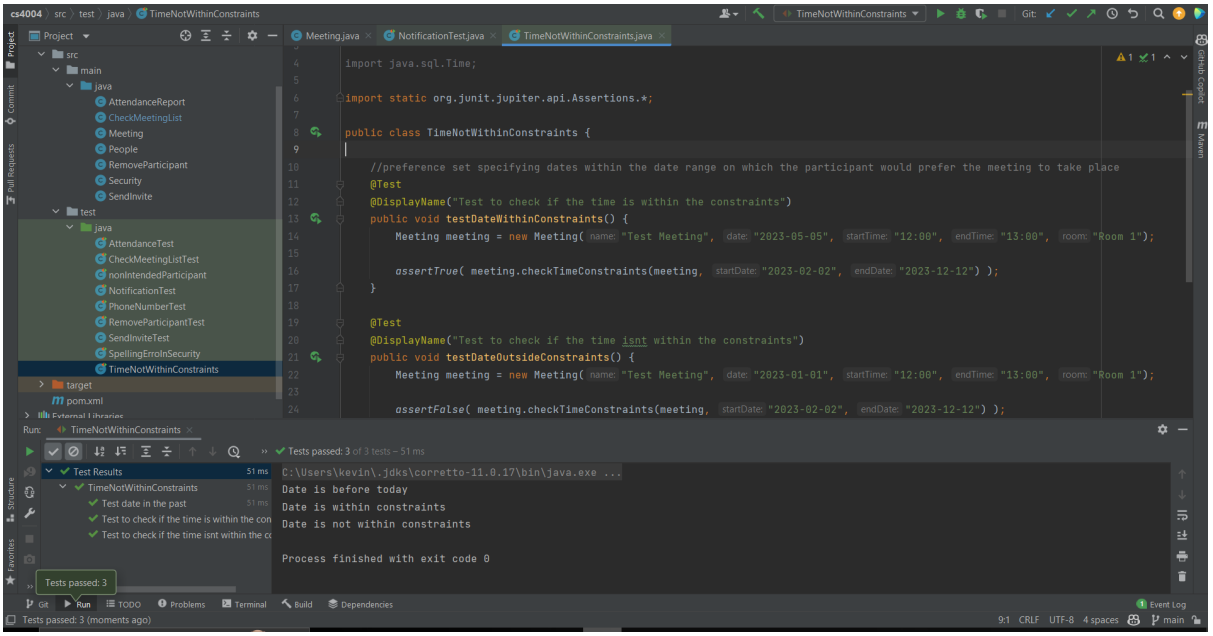
```java
import org.junit.jupiter.api.DisplayName;
import org.junit.jupiter.api.Test;

import java.sql.Time;

import static org.junit.jupiter.api.Assertions.*;

public class TimeNotWithinConstraints {

    //preference set specifying dates within the date range on which the participant would prefer the meeting to take place
    @Test
    @DisplayName("Test to check if the time isn't within the constraints")
    public void testDateWithinConstraints() {
        Meeting meeting = new Meeting( name: "Test Meeting",  date: "2024-05-05",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");

        assertFalse( meeting.checkTimeConstraints(meeting,  startDate: "2023-02-02",  endDate: "2023-12-12") );
    }
//...
}
```

```
C:\Users\kevin\.jdks\corretto-11.0.17\bin\java.exe ...
Date is not within constraints

Process finished with exit code 0
```

Code Enhancement

```java
public boolean checkTimeConstraints(Meeting meeting, String startDate, String endDate) {
    LocalDate now = LocalDate.now();
    LocalDate date = LocalDate.parse(meeting.getDate());
    LocalDate strDate = LocalDate.parse(startDate);
    LocalDate edDate = LocalDate.parse(endDate);
    if(date.isBefore(now)) {
        System.out.println("Date is before today");
        return false;
    }else if (date.isAfter(strDate) && date.isBefore(edDate)) {
        System.out.println("Date is within constraints");
        return true;
    } else {
        System.out.println("Date is not within constraints");
        return false;
    }
}
```

Test pass



4.3.5.1 Tests for Participants of meetings are receiving "No attendance" , regardless if they actually attend or not.

This test used integration testing in combining the join method used for non intended participants can join a private meeting. I found this test difficult to write but eventually when I got it out I realised you couldn’t really make any enhancements to the code. I added in lot’s of coverage to test every possible outcome

4.3.5.2 code developed for test

```java
@Test
@DisplayName("Participants of meetings are receiving \"Full attendance\" , regardless if they actually attend or not.")
public void testAttendance() {


    Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
    People john = new People ( name: "John",  email: "john111@gmail.com",  phone: "+353087635142 3");
    People paudie = new People ( name: "Paudie",  email: "paudthelaud@gmail.com",  phone: "+353086665142 3");
    meeting.addAttendee(john);
    meeting.addAttendee(paudie);
    meeting.joinMeeting(john);

    assertEquals( expected: "Attended", meeting.checkAttendance(john));
    assertEquals( expected: "No attendance", meeting.checkAttendance(paudie));

}
```

4.3.5.3 code developed for program

```java
8 usages   Kevin +2 *
public boolean joinMeeting(People people) {
    boolean joined = false;
    if(checkAttendees(people)) {
        System.out.println("You have joined the meeting");

        joined = true;
    } else {
        System.out.println("You are not invited to this meeting");
    }
    return joined;
}

2 usages   Kevin +1 *
public String checkAttendance(People people) {
    String attendance = "";
    if(attendees.contains(people)) {
        attendance = "Attended";
    } else {
        attendance = "No attendance";
    }
    return attendance;
}
```

4.3.5.4 screenshots of success and failure

First test failing

```
Olan *
@Test
@DisplayName("Participants of meetings are receiving No attendance , regardless if they actually attend or not.")
public void testAttendance() {


    Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
    People john = new People ( name: "John",  email: "john111@gmail.com",  phone: "+3530876351423");
    People paudie = new People ( name: "Paudie",  email: "paudthelaud@gmail.com",  phone: "+353086651423");
    meeting.addAttendee(john);
    meeting.addAttendee(paudie);
    meeting.joinMeeting(john);

    assertEquals( expected: "Attended", meeting.checkAttendance(john));
    assertEquals( expected: "No attendance", meeting.checkAttendance(paudie));

}
}
```

```
Tests failed: 1 of 1 test – 34 ms
34 ms
Full attendance", regardl 34 ms
C:\Users\olan\.jdks\corretto-11.0.17\bin\java.exe ...
You have joined the meeting

org.opentest4j.AssertionFailedError:
Expected :Attended
Actual   :No attendance
<Click to see difference>
```

Code refactoring

```
8 usages   Kevin +2 *
public boolean joinMeeting(People people) {
    boolean joined = false;
    if(checkAttendees(people)) {
        System.out.println("You have joined the meeting");
        attendees.add(people);
        joined = true;
    } else {
        System.out.println("You are not invited to this meeting");
    }
    return joined;
}

2 usages   Kevin +1 *
public String checkAttendance(People people) {
    String attendance = "";
    if(attendees.contains(people)) {
        attendance = "Attended";
    } else {
        attendance = "No attendance";
    }
    return attendance;
}
```

First test passing

```
Olan *
@Test
@DisplayName("Participants of meetings are receiving No attendance , regardless if they actually attend or not.")
public void testAttendance() {


    Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
    People john = new People ( name: "John",  email: "john111@gmail.com",  phone: "+3530876351423");
    People paudie = new People ( name: "Paudie",  email: "paudthelaud@gmail.com",  phone: "+3530866651423");
    meeting.addAttendee(john);
    meeting.addAttendee(paudie);
    meeting.joinMeeting(john);

    assertEquals( expected: "Attended", meeting.checkAttendance(john));
    assertEquals( expected: "No attendance", meeting.checkAttendance(paudie));

}
}
```

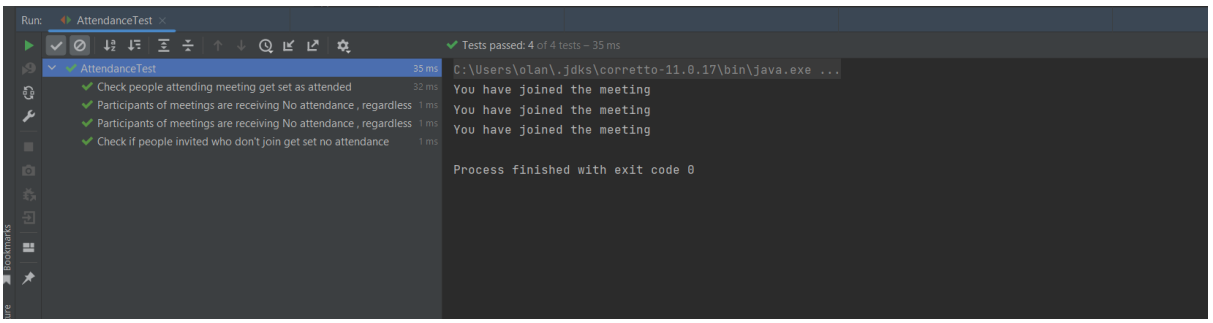
```
AttendanceTest.testAttendance
Tests passed: 1 of 1 test – 36 ms
AttendanceTest                                                    36 ms
  Participants of meetings are receiving No attendance , regardless 36 ms
C:\Users\olan\.jdks\corretto-11.0.17\bin\java.exe ...
You have joined the meeting

Process finished with exit code 0
```

Adding more coverage

```java
new *
@Test
@DisplayName("Participants of meetings are receiving No attendance , regardless if they actually attend or not.")
public void checkMeetingStatusOfPeopleNotInvitedToMeeting() {
    Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
    People sean = new People( name: "Sean",  email: "sean@gmail.com",  phone: "+3530876351423");
    People cian = new People( name: "Cian",  email: "Cian@gmail.com",  phone: "+3530876351423");
    assertEquals( expected: "No attendance", meeting.checkAttendance(sean));
    assertEquals( expected: "No attendance", meeting.checkAttendance(cian));
}

new *
@Test
@DisplayName("Check people attending meeting get set as attended")
public void checkPeopleAttendingMeetingGetSetAsAttended() {
    Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
    People john = new People( name: "John",  email: "john@gmail.com",  phone: "+3530876351423");
    People paul = new People( name: "Paul",  email: "paul@gmail.com",  phone: "+3530876351423");
    meeting.addAttendee(john);
    meeting.joinMeeting(john);
    meeting.addAttendee(paul);
    meeting.joinMeeting(paul);
    assertEquals( expected: "Attended", meeting.checkAttendance(john));
    assertEquals( expected: "Attended", meeting.checkAttendance(paul));
```

```java
new *
@Test
@DisplayName("Check if people invited who don't join get set no attendance")
public void checkIfPeopleInvitedWhoDontJoinGetSetNoAttendance() {
    Meeting meeting = new Meeting( name: "Test Meeting",  date: "2020-01-01",  startTime: "12:00",  endTime: "13:00",  room: "Room 1");
    People Ellie = new People( name: "Ellie",  email: "ellie@gmail.com",  phone: "+3530876351423");
    People Andrew = new People( name: "Andrew",  email: "andrew@gmail.com",  phone: "+3530876351423");
    meeting.addAttendee(Ellie);
    meeting.addAttendee(Andrew);
    assertEquals( expected: "No attendance", meeting.checkAttendance(Ellie));
    assertEquals( expected: "No attendance", meeting.checkAttendance(Andrew));

}
```

All test’s passing



4.3.6.1 Test for Can't create an account, phone number not valid

Firstly, I knew I wanted to have an isValid method to check if the phone number is valid by how many characters it contained. As I was testing, I found lot’s of enhancements such as if it’s an ‘08’ number and if a number is inputted with spaces, the spaces are deleted. If I was to do this test again, I would look at including different types of international numbers. I used regression testing and implemented lot’s of enhancements to make sure the test wouldn’t fail again (for Irish numbers)

4.3.6.2 code developed for test

```
@Test
@DisplayName("Can't create an account, phone number appearing not valid")
public void phoneNumberNotValid() {
    var phoneNumber = new People( name: "Kevin",  email: "kev@gmail.com",  phone: "+3530879043214");
    assertEquals( expected: true, phoneNumber.isValid(phoneNumber));
```

4.3.6.3 code developed for program

```java
public class People {
    2 usages
    private String name;
    2 usages
    private String email;
    2 usages
    private String phone;

    public People(String name, String email, String phone) {
        this.name = name;
        this.email = email;
        this.phone = phone;
    }

    public String getName() {
        return name;
    }

    public String getEmail() {
        return email;
    }

    2 usages
    public String getPhone() {
        return phone;
    }
```

```java
    public boolean isValid(People phoneNumber) {
        boolean expected = false;
        if (phone.length() == 13 && phone.startsWith("+353"))  {
            expected = true;
        }
        return expected;
    }
}
```

4.3.1.4 Screenshots of success and failure

First test failing:

```
Olan *
@Test
@DisplayName("Can't create an account, phone number appearing not valid")
public void phoneNumberNotValid() {
    var phoneNumber = new People( name: "Kevin",  email: "kev@gmail.com",  phone: "+3530879043214");
    assertEquals( expected: true, phoneNumber.isValid(phoneNumber));
```

```
PhoneNumberTest.phoneNumberNotValid
Tests failed: 1 of 1 test – 39 ms
PhoneNumberTest                                   39 ms
  Can't create an account, phone number appearing not valid   39 ms
C:\Users\olan\.jdks\corretto-11.0.17\bin\java.exe ...

org.opentest4j.AssertionFailedError:
Expected :true
Actual   :false
<Click to see difference>
```

Code refactoring + code enhancement for format of number:

```
9 usages   Olan *
public boolean isValid(People phoneNumber) {
    boolean expected = false;
    if (phone.length() == 14 && phone.startsWith("+353"))  {
        expected = true;
    }
    return expected;
}
}
```

Test passing

```
52  @Test
53  @DisplayName("Can't create an account, phone number not valid")
54  public void phoneNumberShouldBeValid() {
55
56      var phone = new People( name: "Kevin",  email: "kevin@gmail.com",  phone: "+3530868604574");
57      var phone2 = new People( name: "Olan",  email: "Olan@gmail.com",  phone: "086805743");
58        var phone3 = new People( name: "Millie",  email: "millie@gmail,com",  phone: "+354 0868604574");
59
60      assertTrue(phone.isPhoneNumberValid());
61      assertFalse(phone2.isPhoneNumberValid());
62      assertFalse(phone3.isPhoneNumberValid());
63
64
```

```
ctions.yml
s classes, 28% lines covered
s4004-meeting
ing 0% methods, 0% lines covered
le 60% methods, 80% lines covered
rity 100% methods, 100% lines covered
neNumberShouldBeValid
Tests passed: 1 of 1 test – 14 ms
14 ms  C:\Users\olan\.jdks\corretto-11.0.17\bin\java.exe ...
```

Code enhancement for position of 08 in a phone number

```
}
if (phone.length() == 14 && phone.startsWith("+353") && phone.charAt(4) == '0' && phone.charAt(5) == '8') {
    expected = true;
}
return expected;
```

Test and Test passing

```
@Test
@DisplayName("Test to see if phonne number that doesn't contain an 08- after +353")
public void phoneNumberNotAn08() {
    var phoneNumber1 = new People( name: "Brian",  email: "brian@gmail.com",  phone: "+3530800000000");
    var phoneNumber2 = new People( name: "Pat",  email: "pat@gmail.com",  phone: "+3530901234567");
    var phoneNumber3 = new People( name: "Martin",  email: "martin@gmail.com",  phone: "+3531895674312");
    var phoneNumber4 = new People( name: "John",  email: "join@gmail.com",  phone: "+353076543210");
    assertTrue(phoneNumber1.isValid(phoneNumber1));
    assertFalse(phoneNumber2.isValid(phoneNumber1));
    assertFalse(phoneNumber3.isValid(phoneNumber1));
    assertFalse(phoneNumber4.isValid(phoneNumber1));
}
PhoneNumberTest.phoneNumberNotAn08
Tests passed: 1 of 1 test – 27 ms
PhoneNumberTest 27 ms
Test to see if phonne number that doesn't contain an 08- after +3 27 ms
C:\Users\olan\.jdks\corretto-11.0.17\bin\java.exe ...
Process finished with exit code 0
```

Coverage for start of phone number

```
@Test
@DisplayName("Test to see if there is no + at start of number")
public void phoneNumberNoPlus() {
    var phoneNumber_1 = new People( name: "Brian",  email: "brian@gmail.com",  phone: "35308686745671");
    var phoneNumber_2 = new People( name: "Pat",  email: "pat@gmail.com",  phone: "-3530879058321");
    assertFalse(phoneNumber_1.isValid(phoneNumber_1));
PhoneNumberTest.phoneNumberNoPlus
Tests passed: 1 of 1 test – 36 ms
PhoneNumberTest 36 ms
Test to see if there is no + at start of number 36 ms
C:\Users\olan\.jdks\corretto-11.0.17\bin\java.exe ...
Process finished with exit code 0
```

Code Enhancement for spaces inputted in phone number

```java
10 usages  Olan *
public boolean isValid(People phoneNumber) {
    boolean expected = false;
    if (phone.length() > 14) {
        phoneNumber.setPhone(phoneNumber.getPhone().replace( target: " ",  replacement: ""));
        System.out.println(phoneNumber.getPhone());
    }
    if (phone.length() == 14 && phone.startsWith("+353") && phone.charAt(4) == '0' && phone.charAt(5) == '8') {
        expected = true;
    }
    return expected;
}
```

Test and test passing

```java
@Test
@DisplayName("Can't create an account, phone number appearing not valid")
public void phoneNumberInputtedWithSpaces() {
    var phoneNumber8 = new People( name: "Ben",  email: "ben@gmail.com",  phone: "+353 0868604574");
    var phoneNumber9 = new People( name: "Paul",  email: "paul@gmail.com",  phone: "+353   086860457");
    var phoneNumber10 = new People( name: "John",  email: "john@gmail.com",  phone: "+  3 5 3  08 6   87 54 3 2 1");
    var phoneNumber11 = new People( name: "Cian",  email: "Cian@gmail.com",  phone: "-353089234  1243");
    assertEquals( expected: true, phoneNumber8.isValid(phoneNumber8));
    assertEquals( expected: false, phoneNumber9.isValid(phoneNumber9));
    assertTrue(phoneNumber8.isValid(phoneNumber8));
    assertFalse(phoneNumber11.isValid(phoneNumber11));
}
}
```

```
PhoneNumberTest.phoneNumberInputtedWithSpaces
Tests passed: 1 of 1 test – 41 ms
PhoneNumberTest                                   41 ms
  Can't create an account, phone number appearing not valid   41 ms

C:\Users\olan\.jdks\corretto-11.0.17\bin\java.exe ...
+3530868604574
+353086860457
-3530892341243

Process finished with exit code 0
```

Screenshot that all test’s still pass

```
Tests passed: 4 of 4 tests – 73 ms
PhoneNumberTest                                   73 ms
  Can't create an account, phone number appearing not valid   70 ms
  Test to see if phonne number that doesn't contain an 08- after +3   1 ms
  Can't create an account, phone number appearing not valid   1 ms
  Test to see if there is no + at start of number   1 ms

C:\Users\olan\.jdks\corretto-11.0.17\bin\java.exe ...
---- IntelliJ IDEA coverage runner ----
sampling ...
include patterns:
exclude patterns:
+3530868604574
+353086860457
-3530892341243
Class transformation time: 0.2980196s for 913 classes or 3.2641796276013145E-4s per class
```

4.3.7.1 Test for Participants not being removed from meeting list

Participants need to be able to be removed from the meeting list on request. The test I developed returns true if the participant is successfully removed.

Code developed for program

```
Meeting.java   RemoveParticipant.java   RemoveParticipantTest.java

    @org.junit.jupiter.api.Test
    void removeParticipant() {

        Meeting a = new Meeting( name: "new",  date: "01/01/2001",  startTime: "10:00",
                 endTime: "11:00", room: "A");
        RemoveParticipant test = new RemoveParticipant(a);
        People Jerry = new People( name: "Jerry",  email: "a",  number: "a");
        ArrayList<People> b = new ArrayList<~>();
        b.add(Jerry);
        a.addAttendees(b);

        assertTrue(test.removeParticipant(Jerry),  message: "Remove Participant");
    }
}
```

```
    private Meeting meeting;
    1 usage
    public RemoveParticipant(Meeting meeting) { this.meeting = meeting; }

    1 usage
    public boolean removeParticipant(People person){

        for(int i = 10; i < meeting.getMeetingList().size(); i ++){
            if(meeting.getMeetingList().get(i) == person){
                meeting.getMeetingList().remove(i);
                return true;
            }
        }
        return false;
    }
```

First Test Fail

```java
class RemoveParticipantTest {

    @Test
    void removeParticipant() {
        Meeting a = new Meeting( name: "new",  date: "01/01/2001",  startTime: "10:00",
                endTime: "11:00", room: "A");
        RemoveParticipant test = new RemoveParticipant(a);
        People Jerry = new People( name: "Jerry",  email: "a",  number: "a");
        ArrayList<People> b = new ArrayList<~>();
        b.add(Jerry);
        a.addAttendees(b);

        assertTrue(test.removeParticipant(Jerry),  message: "Remove Participant");
    }
}
```

```
Tests failed: 1 of 1 test – 40 ms

RemoveParticipantTest > removeParticipant() FAILED
    org.opentest4j.AssertionFailedError at RemoveParticipantTest.java:21
1 test completed, 1 failed
> Task :test FAILED
FAILURE: Build failed with an exception.
* What went wrong:
```

Code Refactoring and Test Passing

Code refactored to remove participant from meeting list

```java
1 usage
public RemoveParticipant(Meeting meeting) { this.meeting = meeting; }

1 usage
public boolean removeParticipant(People person){

    for(int i = 0; i < meeting.getMeetingList().size(); i ++){
        if(meeting.getMeetingList().get(i) == person){
            meeting.getMeetingList().remove(i);
            return true;
        }
    }
    return false;
}
```

```java
class RemoveParticipantTest {

    @org.junit.jupiter.api.Test
    void removeParticipant() {

        Meeting a = new Meeting( name: "new",  date: "01/01/2001",  startTime: "10:00",
                endTime: "11:00", room: "A");
        RemoveParticipant test = new RemoveParticipant(a);
        People Jerry = new People( name: "Jerry",  email: "a",  number: "a");
        ArrayList<People> b = new ArrayList<~>();
        b.add(Jerry);
        a.addAttendees(b);

        assertTrue(test.removeParticipant(Jerry),  message: "Remove Participant");
    }
```

Tests passed: 1 of 1 test – 40 ms

```
"C:\Program Files\Java\jdk-17\bin\java.exe" ...

Process finished with exit code 0
```

4.3.8.1 Test for Meeting invite sent multiple times to participants

Test compares meeting list to expected meeting list to ensure that test are sent only once to each participant

Code developed

```java
@Test
void sendInvite() {
    Meeting meeting = new Meeting( name: "New",  date: "13/11/2022",  startTime: "10:00",
            endTime: "11:00",  room: "A1");
    CheckMeetingList a = new CheckMeetingList(meeting);
    People Jimmy = new People( name: "Jimmy",  email: "jimmy@gmail.com",  number: "0821921");
    People Timmy = new People( name: "Timmy",  email: "timmy@gmail.com",  number: "0828329");
    ArrayList<People> temp = new ArrayList<>();
    temp.add(Jimmy);
    temp.add(Timmy);
    meeting.addAttendees(temp);
    String[] email = {"jimmy@gmail.com", "timmy@gmail.com","kev@hotmail.com",
            "olan@hotmail.com","kev@hotmail.com"};
    assertArrayEquals(a.checkMeetingList(), email);
```

```java
1 usage
public CheckMeetingList(Meeting meeting){
    this.meeting = meeting;
}

1 usage
public String[] checkMeetingList(){
    String[] temp = new String[meeting.getMeetingList().size()];
    for(int i = 0; i < meeting.getMeetingList().size() * 2; i++){
        temp[i] = meeting.getMeetingList().get(i).getEmail();
        temp[i + 1] = meeting.getMeetingList().get(i).getEmail();
    }
    return temp;
}
```

Code Refactoring and Test passing

Code refactored to ensure only one invite is sent to each participant

```java
1 usage
public CheckMeetingList(Meeting meeting){
    this.meeting = meeting;
}

1 usage
public String[] checkMeetingList(){
    String[] temp = new String[meeting.getMeetingList().size()];
    for(int i = 0; i < meeting.getMeetingList().size(); i++){
        temp[i] = meeting.getMeetingList().get(i).getEmail();
    }
    return temp;
}
```

Meeting.java | CheckMeetingList.java | CheckMeetingListTest.java | RemoveParticipant.java | RemoveParticipantTest.java

```java
    void sendInvite() {
        Meeting meeting = new Meeting( name: "New",  date: "13/11/2022",  startTime: "10:00",
                endTime: "11:00",  room: "A1");
        CheckMeetingList a = new CheckMeetingList(meeting);
        People Jimmy = new People( name: "Jimmy",  email: "jimmy@gmail.com",  number: "0821921");
        People Timmy = new People( name: "Timmy",  email: "timmy@gmail.com",  number: "0828329");
        ArrayList<People> temp = new ArrayList<>();
        temp.add(Jimmy);
        temp.add(Timmy);
        meeting.addAttendees(temp);
        String[] email = {"jimmy@gmail.com", "timmy@gmail.com","kev@hotmail.com",
                "olan@hotmail.com","kev@hotmail.com"};
        assertArrayEquals(a.checkMeetingList(), email);
    }
}
```

Tests passed: 1 of 1 test – 42 ms

```
"C:\Program Files\Java\jdk-17\bin\java.exe" ...

Process finished with exit code 0
```

5.0 GitHub Actions

5.1: What is github Actions CI/CD?

Firstly, you may wonder what CI and CD stand for. CI stands for continuous integration and CD stands for continuous delivery. Github Actions is one of many CI/CDplatforms such as Jenkins, Jetbrains which allows you to automate your build, test and deployment pipeline. We set up our platform to build and test everytime someone pushes code to our repository on github. Github provides a virtual machine from linux, windows and mac to run a workflow from and we ran ours on linux. A workflow is an automated process that you can configure which will run one or more jobs and is defined by a .yml file. For github actions to work, a directory in the repository must be defined by ***'.github/workflows/yourname.yml'*** .

A job is a set of steps in a workflow that is either a shell script command that will be executed or an action that will be run. Steps are executed in order and are also dependent on each other. We configured our project with maven so set up a workflow in github itself with a template called "Java with maven". This enabled us to run our tests with continuous integration and would notify us for example if we were working on a new test and started editing code that is used for another test, if that old test would fail with these new code changes. Our pom file looked as so:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>

    <groupId>org.example</groupId>
    <artifactId>CS4004</artifactId>
    <version>1.0-SNAPSHOT</version>

    <properties>
        <maven.compiler.source>11</maven.compiler.source>
        <maven.compiler.target>11</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <dependencies>
        <dependency>
            <groupId>org.junit.jupiter</groupId>
            <artifactId>junit-jupiter</artifactId>
            <version>5.9.0</version>
        </dependency>

        <dependency>
            <groupId>org.mockito</groupId>
            <artifactId>mockito-inline</artifactId>
            <version>4.8.1</version>
        </dependency>
        <dependency>
            <groupId>org.mockito</groupId>
```

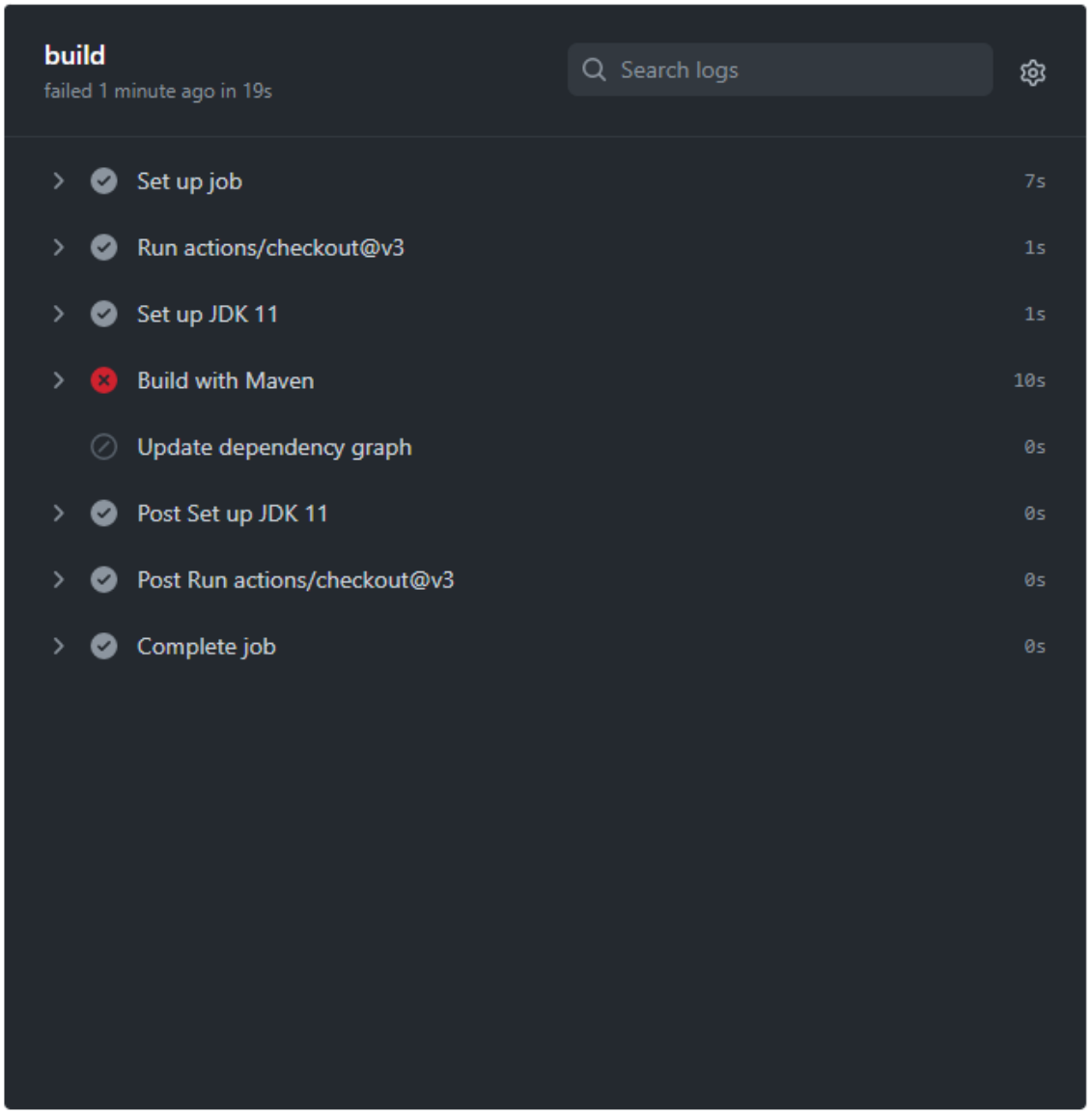
```xml
            <groupId>org.mockito</groupId>
            <artifactId>mockito-junit-jupiter</artifactId>
            <version>4.8.1</version>
            <scope>test</scope>
        </dependency>
        <dependency>
            <groupId>org.mockito</groupId>
            <artifactId>mockito-junit-jupiter</artifactId>
            <version>4.8.1</version>
            <scope>compile</scope>
        </dependency>
    </dependencies>

</project>
```

And our .yml file (ours was called CI.yml) looked as such:

```yaml
# This is a basic workflow to help you get started with Actions

name: CI/CD with Github Actions

# This workflow will build a Java project with Maven, and cache/restore any dependencies to improve the workflow execution time
# For more information see: https://docs.github.com/en/actions/automating-builds-and-tests/building-and-testing-java-with-maven

# This workflow uses actions that are not certified by GitHub.
# They are provided by a third-party and are governed by
# separate terms of service, privacy policy, and support
# documentation.


on:
  push:
    branches: [ "main" ]

jobs:
  build:

    runs-on: ubuntu-latest

    steps:
    - uses: actions/checkout@v3
    - name: Set up JDK 11
      uses: actions/setup-java@v3
      with:
        java-version: '11'
        distribution: 'temurin'
```

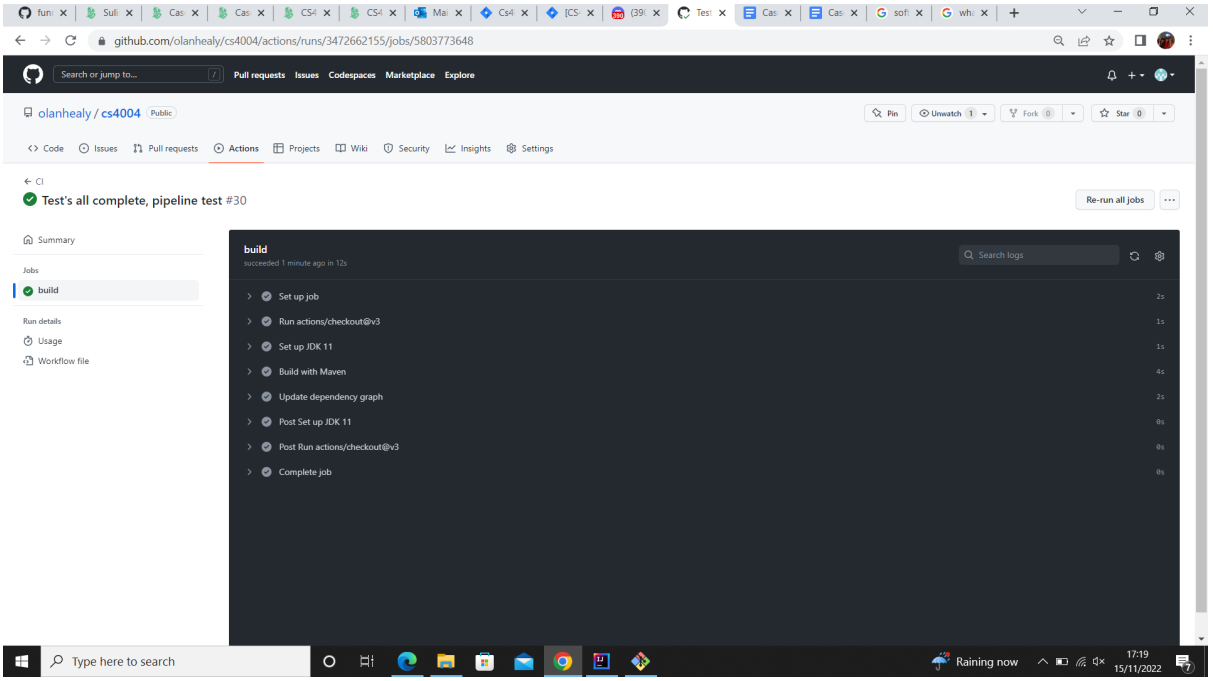
```yaml
    - name: Build with Maven
      run: mvn -B package --file pom.xml

# Optional: Uploads the full dependency graph to GitHub to improve the quality of Dependabot alerts this repository can receive
    - name: Update dependency graph
      uses: advanced-security/maven-dependency-submission-action@571e99aab1055c2e71a1e2309b9691de18d6b7d6
```

Build with maven part is a key component of this .yml file as it builds and runs all of the test’s in our program. For example, we pushed code which we knew test’s would fail and this is the output we see on github actions:


build
failed 1 minute ago in 19s
Search logs
Set up job 7s
Run actions/checkout@v3 1s
Set up JDK 11 1s
Build with Maven 10s
Update dependency graph 0s
Post Set up JDK 11 0s
Post Run actions/checkout@v3 0s
Complete job 0s

```
72  Error:  Failed to execute goal org.apache.maven.plugins:maven-surefire-
    plugin:2.12.4:test (default-test) on project CS4004: There are test failures.
73  Error:
74  Error:  Please refer to /home/runner/work/cs4004/cs4004/target/surefire-reports for
    the individual test results.
75  Error:  -> [Help 1]
76  Error:
77  Error:  To see the full stack trace of the errors, re-run Maven with the -e switch.
78  Error:  Re-run Maven using the -X switch to enable full debug logging.
79  Error:
80  Error:  For more information about the errors and possible solutions, please read
    the following articles:
81  Error:  [Help 1]
    http://cwiki.apache.org/confluence/display/MAVEN/MojoFailureException
82  Error: Process completed with exit code 1.
```

Our last commit was with all test;s fully written and so we did and pipeline test and could easily see that all our test's still passed:



5.1 Reflection

Overall, we (Olan and Kevin) found using github actions very useful. We could have used Jenkins but it is more complicated

then github actions. As we were uploading our code to github anyways, it was the obvious choice to use continuous integration with github actions. External research is very important and trying to improve ourselves as aspiring software developers. We now have an idea on how pipelining works and will benefit us tremendously for future projects

6.0 Conclusion

Olan Healy (21318204): 33.3(recurring)

- Bug report 3.3.1, 3.3.5, 3.3.6
- Test’s 4.3.1.1, 4.3.5.1, 4.3.6.1
- GitHub actions

Kevin Collins (21344256): 33.3(recurring)

- Bug report 3.3.2, 3.3.3, 3.3.4
- Test’s 4.3.1.2, 4.3.1.3, 4.3.1.4
- Github actions

Sean Capilis (213423342): 33.3(recurring)

- Bug report 3.3.7, 3.3.8
- Test’s 4.3.7.1, 4.3.8.1