# Attendance Pages Defect Report

This report documents the defects and test failures uncovered in the Sparta Engineering 76 final project, as a result of testing the features of the Trainee Attendance and Trainer Attendance pages associated with the POM for the website created during the Sparta Engineering 72 final project.

## 1. Null Pointer

**Page(s):** Trainee & Trainer

**Test Case(s):** SP2020-204, SP2020-205

This error is thrown during automation testing when attempting to check that either the number of days in a given week is correct, or that a week has been toggled. On the trainee and trainer attendance pages, the weeks documenting the attendance of the trainees are stored in the HTML in the following way:


Where each element with tag name 'tbody' represents a row in the table. Each tbody element contains a number of child elements with the tagname 'tr'. The number of days in a week can be identified by the number of these tr elements contained within each tbody element. Additionally, whether or not

a row is toggled can be determined from the first element in this list of tr elements, which contains the following attributes:

```
▼<tbody>
  ▶<tr style="text-align: center" class="accordion-toggle" id="accordion1" data-toggle="collapse"
  data-parent="#accordion1" href="#collapse14" aria-expanded="true">…</tr> == $0
  ▶<tr class="hide-table-padding">…</tr>
  ▶<tr class="hide-table-padding">…</tr>
  ▶<tr class="hide-table-padding">…</tr>
  ▶<tr class="hide-table-padding">…</tr>
  ▶<tr class="hide-table-padding">…</tr>
  ▶<tr class="hide-table-padding">…</tr>
  </tbody>
```

Where the attribute 'aria-expanded' represents the state of the row, "true" being toggled and "false" or null being not toggled. The exception arises from the following method in the abstract class 'Attendance':

```
public boolean isToggledOnWeek(int week){
    String isToggled = weeks.get(weeks.size() - week).findElements(rows).get(0).getAttribute(isExpanded);
    if (isToggled.equals("true")){
        return true;
    }
    return false;
}
```

Where 'weeks' is a list of tbody web elements, 'rows' is a list of tr web elements. The list of tr elements is created using a simple 'By' The program throws a Null Pointer exception when selenium is unable to correctly populate the list of tr web elements. This occurs when trying to perform an operation on the String 'isToggled', which is null if the list cannot be created. The same issue also occurs in the method used to count the number of days in the week for the same reason.

## 2. Index Out Of Bounds

**Page(s):** Trainee & Trainer

**Test Case(s):** SP2020-200 - SP2020-205

This error is thrown during automation testing when attempting to initially click on any of the weeks in the trainee or trainer attendance page. In order to initially populate the list of tbody web elements, the following annotation is used:

```
@FindBy(tagName = "tbody")
public List<WebElement> weeks;
```

This size of this list is determined by the number of rows in the weeks table. When operating correctly, the list will be populated with at least 12 tbody web elements. However, in some instances, selenium will incorrectly populate the list with only 2 elements. The exception arises when using the following method in either attendance trainee or trainer classes:

```
public void clickWeek(int week){
    WebElement weekRow = new WebDriverWait(webDriver,  timeOutInSeconds: 5)
            .until(ExpectedConditions.elementToBeClickable(weeks.get(weeks.size() - week)));
    if (week > 0 && week <= weeks.size()) {
        weekRow.click();
    }
}
```

In the case where the list is not correctly populated, when attempting to access an element which is greater than the number of elements in the lust, an IndexOutOfBounds exception is thrown. For example, when attempting to access index 8 (ie week = 8), the method will use weeks.size() – week, which equals -6, thereby throwing the exception.

## 3. Timeout

**Page(s):** Trainee & Trainer

**Test Case(s):** SP2020-200 - SP2020-205

This error can be thrown in any method that uses an explicit wait that asserts if an element is clickable. This is noted in the defect report because, in some cases, selenium will throw a timeout exception despite all elements on the current page being loaded and clickable.

## 4. Incorrect Number of Days in Week

**Page(s):** Trainee & Trainer

**Test Case(s):** SP2020-204, SP2020-205

This is a defect in the website. This defect was identified when counting the number of days in each week row for the trainee and trainer attendance pages. When the problems defined in Sections 1-3 do NOT occur, the program can correctly identify the number of days in a week and will assert this value against the expected value of 5. In certain cases there are more than 5 days in a week, this defect is a result of a feature where it is possible to add new days to a week.

## 5. Setup

**Page(s):** Trainee & Trainer

**Test Case(s):** SP2020-200, SP2020-206

This issue may occur throughout any of the test cases when attempting to establish the pre-requisites or when attempting to navigate to the correct page to initiate tests. At seemingly random points in each test, Selenium will not be able to correctly perform the intermediary steps required to navigate throughout the website. This can include: Not clicking an element that has been successfully found, not sending keys to an element that has been found, incorrectly identifying an element and navigating to an incorrect page.

## Conclusion

The majority of the issues in this report are the result of the way Selenium is interacting with the GUI, rather than actual defects in the GUI itself. There were a number of adjustments made to the POM and the methods within it in order to improve the reliability of the identification and selection of web elements. Including: Using different methods of finding elements (css, tagname, ID etc), making use of both implicit and explicit waits to allow the program time for all elements to load correctly or become clickable, running the tests on different machines and also running the tests with and without cucumber. However, these changes made no impact on the reliability of the program as a whole. In addition, there was no consistency in terms of which tests would fail in each run or how many tests would fail. Meaning that running the same test several times would produce highly unpredictable results, failing for different reasons each time depending on which of the documented defects occurred.

The primary defect in the GUI itself has been successfully identified in Section 4.