Code Snippets and Algorithms (Pseudocode & Trace Tables)

Saving Attendance

Pseudocode for the onDateChanged() function of the Date Spinner

```
// PREVIOUS_DATE (String), DATE_SPINNER (JSpinner), MIN_DATE (Starting Date), DATE (Today's Date) are already
existing variables
// getDate function turns DateSpinner date into string
// getTime function gets integer value of a Date in String format or DateSpinner time
// saveAttendance(Date) saves the attendance for that date to the attendance records
// loadAttendance(Date) loads the attendance for that date from the attendance records

SELECTED_DATE = DATE_SPINNER.getDate()
if (DATE_SPINNER.getTime() < MIN_DATE.getTime() or DATE_SPINNER.getTime() > DATE.getTime())
        if (DATE_SPINNER.getTime() < MIN_DATE.getTime())
                DATE_SPINNER.setValue(MIN_DATE)
        else
                DATE_SPINNER.setValue(DATE)
        end if
        SELECTED_DATE = DATE_SPINNER.getDate()
        saveAttendance(SELECTED_DATE)
end if
if (not PREVIOUS_DATE = SELECTED_DATE)
        saveAttendance(PREVIOUS_DATE)
        loadAttendance(SELECTED_DATE)
end if
PREVIOUS_DATE = SELECTED_DATE
```

Trace Tables for the onDateChanged() function of the Date Spinner

(Dates are in Days away from min date)

Selected Date in between min date and date:

Line in Code	Prev. Date	Select. Date	Date Spinner	Min. Date	Date	Attendance changes
1	4	5	5	0	7	none
9	4	5	5	0	7	none
13	4	5	5	0	7	Saved(4) and loaded(5)
14	5	5	5	0	7	none

Selected Date below min date:

Line in Code	Prev. Date	Select. Date	Date Spinner	Min. Date	Date	Attendance changes
1	0	-1	-1	0	7	none
4	0	-1	0	0	7	none
9	0	0	0	0	7	Saved(0)
14	0	0	0	0	7	none

Selected Date above date:

Line in Code	Prev. Date	Select. Date	Date Spinner	Min. Date	Date	Attendance changes
1	7	8	8	0	7	none
6	7	8	7	0	7	none
9	7	7	7	0	7	Saved(7)
14	7	7	7	0	7	none

Finding Analysis Values (Grades and Attendance Percentage)

// DATA[GRADE] (List of Students in class), GRADES_RECORDS[GRADE] (ArrayList), ANALYSIS_GRADES_RECORDS are already existing variables

```
loop I from 0 to DATA[GRADE].size()
        AVERAGE_GRADE = 0
        GRADES_NUMBER = 0
        loop J from 0 to GRADES_RECORDS[GRADE}.size()
                if (DATA[GRADE][I][0] = GRADES_RECORDS[GRADE][J][0])
                        GRADES_NUMER = GRADES_NUMBER + 1
                        AVERAGE_GRADE = AVERAGE_GRADE + GRADES_RECORDS[GRADE][J][3]
                end if
        end loop
        TOTAL = 0
        If (not GRADES_NUMBER = 0)
                TOTAL = AVERAGE_GRADE / GRADES_NUMBER
        ANALYSIS_GRADES_RECORDS[I] = TOTAL
end loop
```

Pseudocode for the getAttendance() function of the Analysis Panel

// TOTAL_DAYS (DATE - MIN_DATE), DATA[GRADE] (List of Students in class), ATTENDANCE_RECORDS[GRADE] (ArrayList), ANALYSIS_ATTENDANCE_RECORDS are already existing variables

```
loop I from 0 to DATA[GRADE].size()
        ATTENDED_DAYS = 0
        loop J from 0 to ATTENDANCE_RECORDS[GRADE}.size()
                if (ATTENDANCE_RECORDS[GRADE][J][I] = true)
                        ATTENDED_DAYS = ATTENDED_DAYS + 1
                end if
        end loop
        TOTAL = 0
        if (not TOTAL_DAYS = 0)
                TOTAL = 100 * ATTENDED_DAYS / TOTAL_DAYS
        ANALYSIS_ATTENDANCE_RECORDS[I] = TOTAL
end loop
```