IB Computer Science IA — Attendance System for the Marching Band

Jacob Samurin

January 11, 2023

Contents

1 Criterion A: Planning

1.1 Defining the Problem

The problem of Mr. Todd Fessler (my client) is that for our marching band class there is no good and efficient way of taking attendance. The way of taking attendance right now is that the "Drum Majors" who are the overall leaders in the marching band, go around and ask each row for their attendance, and it took a very long time to take attendance.

1.2 Rational for Proposed Solution

My solution will make it possible for the leaders of each row to take attendance then the Drum Majors won't have to go row by row. This will also skip the Drum Major step completely since my client will have direct accesses to the app. This will make it easier for everyone and faster, so we can have a longer rehearsal times.

1.3 Success Criterion

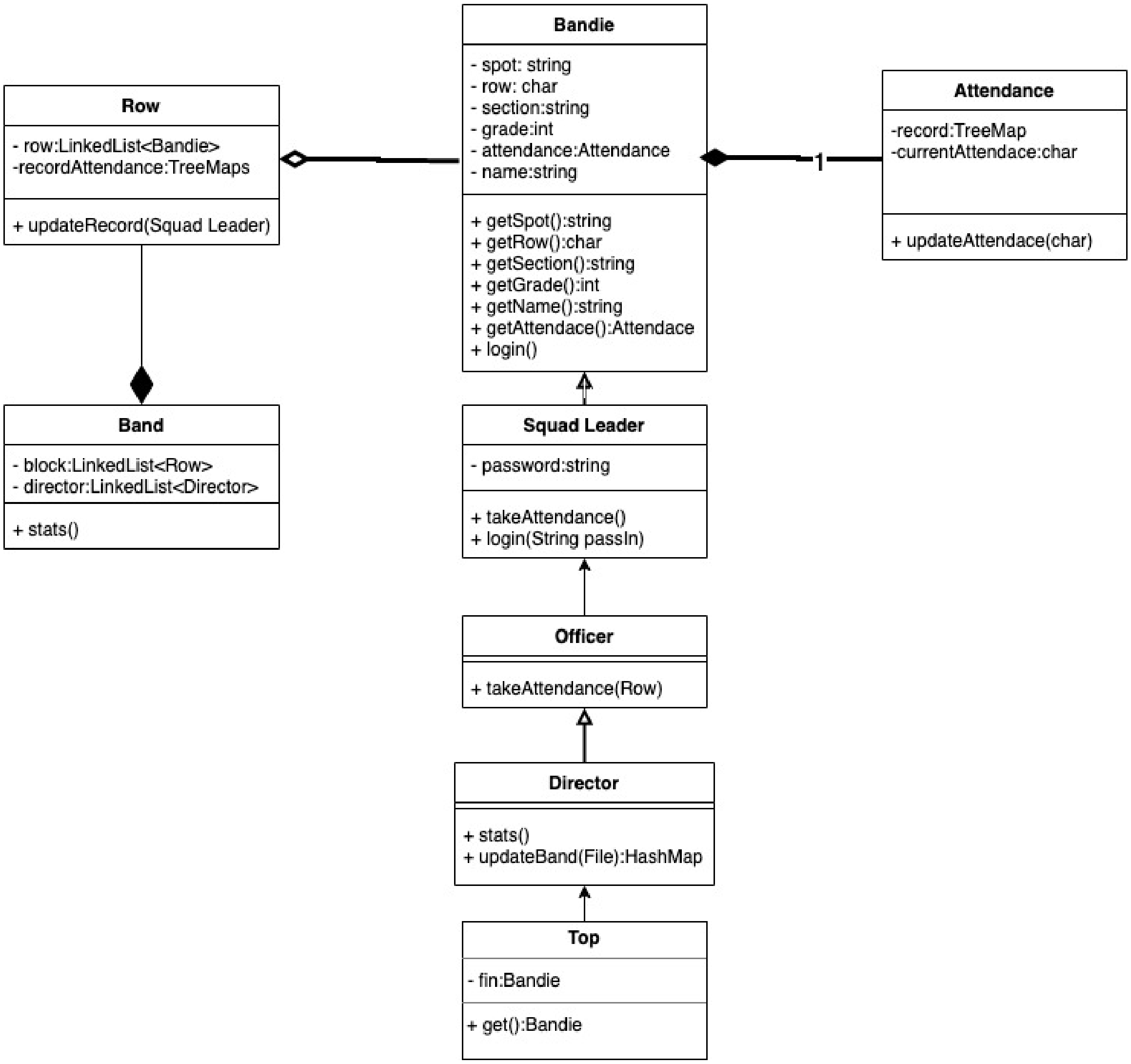
Different classes each level of leaders
A web app will be created add the different levels can edit different kinds of pages
Make it accessible from a phone
Make it accessible from an iPad
Make it accessible from a laptop/computer
Have the sever running the web app set up

2 Criterion B: Solution Overview

2.1 Sketch

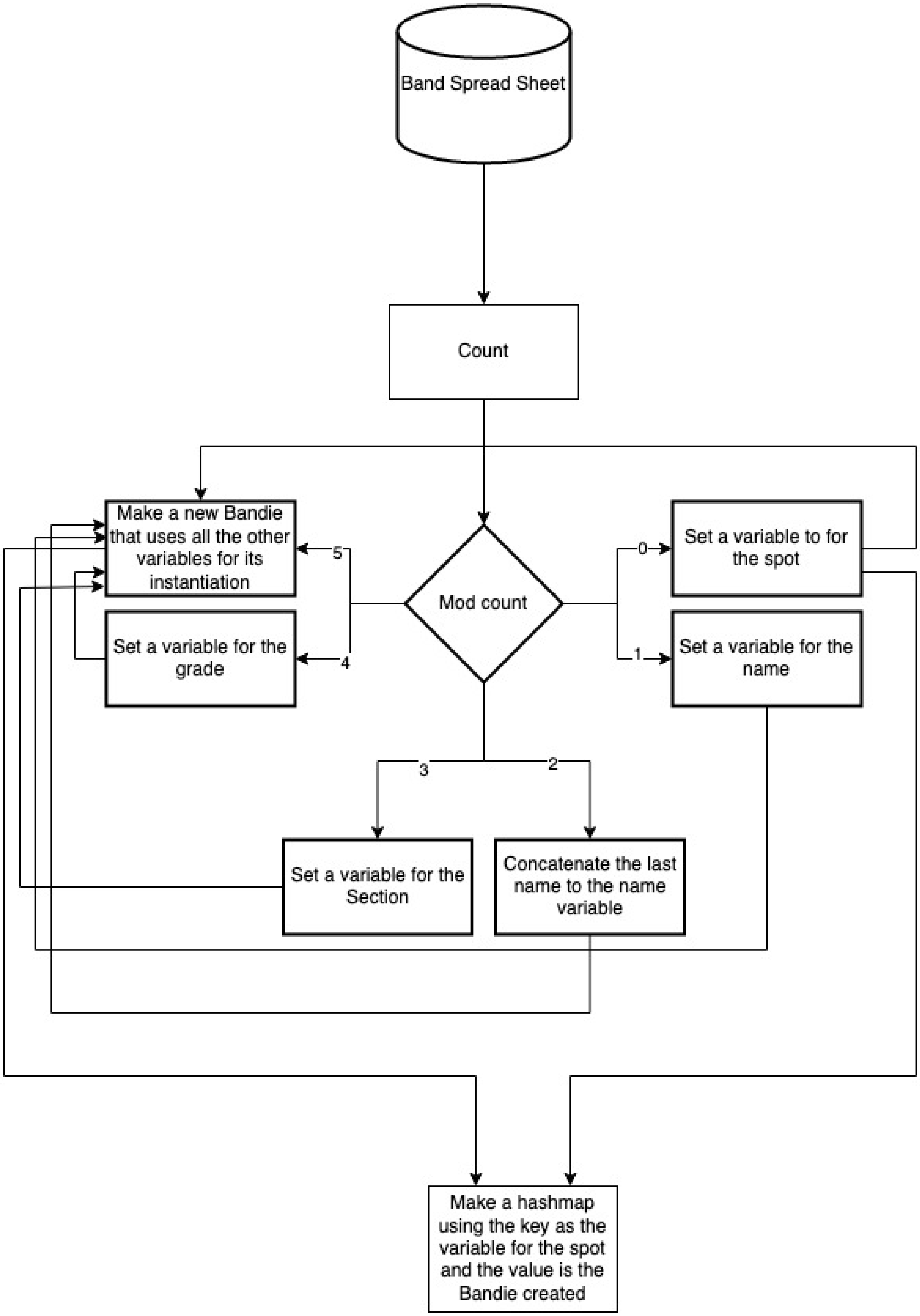
2.2 UML Diagram

This is the basic outline of the different classes and data types



2.3 Flowchart

This is the basic flow of the updateBand method


Band Spread Sheet
Count
Mod count
Make a new Bandie that uses all the other variables for its instantiation
5
Set a variable for the grade
4
0
Set a variable to for the spot
1
Set a variable for the name
3
2
Set a variable for the Section
Concatenate the last name to the name variable
Make a hashmap using the key as the variable for the spot and the value is the Bandie created

2.4 Pseudocode

For the pseudocode I will be writing out the updateBand method

```
item = read in a file for every section that stops in a comma
while there is a next item
if tem%6 equals 0
spot = item
if its k-row
row = first 2 letter
else
row = first letter
if temp%6 equals 1
name = item + " "
if temp%6 equals 2
name += item
if temp%6 equals 3
section = item
if temp%6 equals 4
grade = item
if temp%6 equals 5
new object t of type Top(spot, row, name, section, grade, item)
temp++
```

2.5 Development Plan

This is a plan for how to create the final product

- Create the Bandie class and the other levels of authorization all the way to Director/Admin
- Write the Row class with the hash table for the whole row's attendance and a linked list for the Bandies in the Row
- Create the Band class with the linked list for the directors and another linked list for the rows
- Create the Attendance class to store the past and current attendance for each Bandie
- Be able to upload a new CSV file to upload the structure of the band
- Store past attendance for each Bandie in a CSV file
- Make a web app that will ...
  1. Have a login screen where any level above Bandie has a password
  2. Have the different screens for Bandies, Squad Leaders, and Drum Majors.
  3. For Directors, they will also have a Statistics page with a tagline at the top(refer to the Sketch above)

2.6 Test Cases

Case	Outcome
Logging into a higher level account	• An incorrect password will send errors to the user. • A correct password will let the user through. • An invalid password(i.e. sending in the wrong data type) will send an error to the user
Uploading a spreed sheet file	• If the file is not a CSV file then will pass an error to the user ”Improper File type” • If the file is a CSV but is not properly formatted it will pass an error to the user ”Improper formatted” • If the file is a CSV and the first line is properly formatted it will upload the file

2.7 Record of Tasks

Task Number	Planned Action	Planned Outcome	Time Estimated (Minutes)	Target Completion Date	Criterion
1	Brainstorming with my client	An idea for the project	30	May 5, 2022	A
2	Interview	Get to know what my client wants	8.5	May 24, 2022	A
3	Write a draft proposal	proposed to teacher	60	May 26, 2022	A
4	Write a proposal	Re-proposed to teacher	75	Aug 22, 2022	A
5	make a Record of Tasks	Organize a timeline	15	Aug 23, 2022	B
6	Second Interview	Get a better Idea from my client	7	Sep 1, 2022	A
7	Create UML diagram	Get a UML diagram	90	Oct 20, 2022	B
8	Working on criterion B	Complete the outline of the criterion B	20	Oct 21, 2022	B
9	Third Interview	Get a better idea of the statistics section	6	Oct 26, 2022	A
10	Create drawing UI	To put my idea on to paper	45	Nov 5, 2022	B
11	Re-setup the document	Make it easier to work on the document	180	Nov 8, 2022	A/B
12	added a case to the Test Cases section	Work on a part needed for the finishing Criterion B	45	Nov 9, 2022	B
13	added a case to the test Cases section	work on a part needed for the finishing criterion b	15	Nov 9, 2022	B
14	fix the UML and finish Flowchart and Pseudocode	fixed UML and finished Flowchart and Pseudocode	90	Dec 19, 2022	B
15	writing the hash table section	started work on the writing part of criterion C	30	Jan 8, 2023	C

3 Criterion C: Development

3.1 Hash table (Kind Of)

I used a broken down hash table. A hash table is an array of linked list, I made my own class that used a linked list called Row and another class called Band which used an array of Rows to store each member of the band. I create an object called Row which is a linked list and using an array of Rows in my Band class I made a broken down hash table. This hash table is being used for the storage of all the members of the band. To reset the hash table I have made two methods one side is in the Band class and the Director class

3.2 Reading a File

I am reading an CSV file using scanner. CSV is a file formatted with data seporated with commas and new lines. I'm using Scanner to go through the CSV file and I can do it easily by setting the delimiter to a comma I can easlly go through the entire file and stop at every stop. A CSV file can be edited through a spreadsheet editor.

4 Criterion E: Evaluation

5 Appendix A: Interview

Interview conducted in person Interview 1 initial information gathering

Student - Basically my plan right now for the for the program is there's going to be different tiers. So, there's going to be a squad leader tier, a drum major tier, and then a director/admin tier. Squad leaders will be able to change attendance within the row, for Drum Major will be able to change attendance for the entire band and then for directors they can change the attendance for the entire band and on top of that there's some statistics like what rows haven't taken attendace yet, who turned into attendance for which row, and then maybe some row of the week stuff like most improved week to week, move most improved within the week, or best overall within the week or for the entire year. I'll be able to collect all that kind of data just from attendance. Client – That's awesome. So would it be like you're thinking, like you said squad leaders have control over their rows. Student - and then drum majors the entire band and then you guys can take this for the entire band and then you guys can do attendance for the entire band and then on top of that you can look at statistics and stuff Client - Is there any way once like for example of this squad leaders have submitted their attendance for the day is there a way for them to update it? Student - I mean I could definitely implement that. Client - or like a way for them to update it until the end of class. Somebody shows up that's the issue I think yeah if somebody shows up late on excused then they're marked absent for the whole day but we need to know if they were late that day or if they were asking because that changes Student –I can probably set up a window of time like every morning from like 7:30 to 8:30 maybe even earlier I can set it to like 7 like to whenever time the period ends nowadays. Talking about some basic structure

6 Appendix B: References