

Topic 1: Internship Application Tracker

A. Overview

When you enter the final years of your university life, one of the biggest issues that you'll find is internships. The longer you dive into the world of Computer Science, the more you are exposed to the importance of internships. People boast about it, cry about it, stress out about it, and most importantly, joke about it. An internship is a crucial moment in a Computer Science student's life, and sometimes CS students spend hours to days to weeks, constantly applying for internships that can (hopefully) get them a job (that's not in McDonalds) soon. Sometimes they even apply to tens if not hundreds of different companies just to get a small light on their resume. So your task is to help them simplify and organise their internship search journey with the Internship Application Tracker!

B. Problem Statement

Now back to being serious. Your task for this semester is to cook up a program that allows students to organise their internship applications. This system should allow students to input the info about their application, and then update the info as time goes on.

The student should also be able to view each application that they inputted into the system, as well as update the status of the application. For example, if the student gets an interview, he/she should be able to set its status as 'interview'. Finally, the student should also be able to delete applications that they input without messing up the information in the CSV file.


C. Basic Features (8 marks)

Features	Description	Suggestions
Adding Internship Applications (2 marks)	<p>Users should be able to add intern applications to their list. The intern applications must be added to an Excel file (CSV). The usage of a txt file for this system is not allowed.</p> <p>Users should be able to add the following details:</p> <ul style="list-style-type: none"> - Company Name - Internship Title - Application date - Interning Location (optional) - Contact Information (optional) - Notes (optional) - Internship Status ("Applied" by default) <p>The user must enter the company name and internship title, if not the system will not accept it. The application date should not be entered, but instead, the system must take the date of when the data has been entered into the system.</p> <p>For optional slots, the user can add the internship application even if it is null. If there is no data added in these optional slots, it must also be transferred as null. No -, no N/A, only null.</p> <p>A word limit should also be implemented.</p> <p>For Internship Status, it should not be entered. Instead, it should be inputted into the database as "Applied" by default.</p>	<p>Utilize Object-Oriented Programming concepts to ease the transfer of data from the user to the system.</p> <p>While this is feasible using CLI, I would suggest you implement a GUI, as it is not only an extra feature that will give you more marks (explained later in the document), but it will also ease the user to enter their details and ensure a smooth presentation.</p> <p>Be careful with the optional details, as those are the ones that will mess up the data in the Excel file later on.</p>

Displaying Internship Applications (1.5 marks)	<p>Users should be able to see the internship applications that they entered in a clear format. This display is also needed for the users to interact with the data later on for other features.</p> <p>The displayed data does not need to be displayed in tabular form. As long as the user can easily see the data and interact with it, then it is sufficient.</p> <p>The data displayed must not only be neat, but correct as well.</p>	<p>You might encounter problems with displaying null data, as File I/O concepts can be really tricky when dealing with null Excel columns, so be careful with those.</p> <p>I suggest displaying the data in a list version (something like JSON but a bit neater) instead of displaying it in tabular form, which might make it harder for the user to see.</p>
Updating / Changing details (2 mark)	<p>Users should be able to change any details that they have entered. For example, let's say the user misspelt "Petronas" as "Potrenas". The user should be able to go to the displayed data and edit it to its proper spelling.</p> <p>This is especially important with the notes section, as users might want to add/remove some notes that they wrote about the intern application.</p> <p>Make sure the users adhere to the same conditions as when they first entered the data into the system.</p>	<p>This is another reason why I suggest implementing a GUI. However, if you don't want to implement a GUI, I suggest just adding an extra section specifically to change data so that it's better for the eye to see.</p> <p>Also how you want to update the data in the CSV file is completely up to you. There are a lot of methods (removing the line and replacing it with a new line, editing the line etc), but I suggest choosing the one that makes the change look the smoothest.</p>
Removing intern applications (1.5 marks)	<p>Users should be able to delete the internship applications.</p> <p>The deletion must be smooth. It should also not disrupt or corrupt the rest of the data.</p>	<p>When you delete an entire line, you must make sure that line is still not there as it might disrupt the display process later on.</p>
Marking the	Users should be able to change	If you mastered the changing

status of internship applications (1 mark)	<p>the status of the internship applications.</p> <p>This is similar to changing/updating other details, but instead of the user typing it themselves, you have to give them options to select how they want to mark the status. They should be given the following options:</p> <ul style="list-style-type: none"> - Applied - Interview - Awaiting results - Offer Received - Rejected 	<p>details part, this should be easy. Its backend is the same thing, but instead of letting the user freely enter whatever, you give options for them to select instead.</p> <p>If you're using CLI, I suggest adding colour to your text to make it look nicer.</p>
--	---	--

D. Extra Features (max 4 marks)

Features	Description	Suggestions
Graphical User Interface (2 marks)	<p>Users should be able to interact with the system easily and freely.</p> <p>The GUI must be user-friendly, bugless, and most importantly, have a great design.</p> <p>1 mark will be given for user-friendliness, while another will be given if the design is easy to look at (instead of the default buttons and text fonts)</p>	<p>For beginners, I suggest utilizing JavaFX or Swing. There are a lot of tutorials out there relating to how to make projects with these libraries, and it's pretty straightforward.</p> <p>If you're an expert, however, I do recommend utilizing React with Java Spring Boot and MongoDB for the database. Me and my team did this last year, but honestly, if you want to have a stress-free first semester, don't. Here's the tutorial, though:</p> <p> Full Stack Development wi...</p>
Database	Instead of storing it in an Excel	Here you have 2 possibilities:

(2 marks)	<p>file, the system can have the option of storing it in a database.</p> <p>This makes the storing of data much more secure, and it opens the possibility of the system becoming online instead of just a local widget.</p>	<p>Either you use MySQL or MongoDB.</p> <p>I suggest the latter, but it's really up to you. I can see both databases being useful for this project, in their own way.</p>
<p>Login and Registration (account feature)</p> <p>(2 marks)</p>	<p>Users should be able to utilize their accounts in the program.</p> <p>In other words, multiple users can log on to the same program on the same device at different times.</p> <p>However, the system must find a way to make sure that only the current user's internship applications are displayed when they login to their account. User A must not be able to see user B's applications on their account.</p>	<p>For this feature, you will need to have 2 CSV files/databases. One for users and one for all the internship applications.</p> <p>To make sure it only displays the current user's applications, you must have an extra feature in every data that links that data to the current user.</p>
<p>Sorting and filtering</p> <p>(1 mark)</p>	<p>Users should be able to sort their internship applications by their status.</p>	<p>This part is a free mark. As long as you add some sort of sorting mechanism to ease the eyes of the user you should be fine.</p> <p>My suggestion is to let the user sort by the application's status.</p> <p>In other words, you make it so that it clumps "Rejected" at the bottom, "Offer Received" at the top, etc. It's up to your creativity.</p>
Search feature	Users should be able to search	If you're using CLI, just add

<p>(1 mark)</p>	<p>for specific internship applications based on Company Name, Job Title and Location.</p> <p>The search feature does not need to have a fuzzy search, but it cannot be case or space sensitive. If the user searches “universitymalaya”, University Malaya should appear, and vice versa.</p>	<p>another section to search for internship applications.</p> <p>If you’re using GUI, add a small bar at the header of your system so that it’s easier to access.</p>
<p>Any additional features (1 mark each)</p>	<p>This is completely up to the creativity of the student and the leniency of the marking lecturer/demonstrator.</p> <p>If the lecturer/demonstrator deems the feature to be good enough to count as an extra feature, then the student can be awarded 1 mark for each feature</p>	<p>I don’t suggest going down this route if you care about marks, because whether or not the feature counts as an extra feature, it’s up to your marking lecturer/demonstrator.</p> <p>However, I think being creative is the best part of any coding assignment, and if you want to develop a really good assignment, let your ideas flourish!</p>

E. Example output

Here are the example outputs:

1. This is the thing you will need to replicate in your program:
https://docs.google.com/spreadsheets/d/10DtWIMbXQcHNXu3aoQndMcMIsAviY-IWfP_1t8R5AYA/edit?gid=0#gid=0
2. This is an example of the tracker. You can navigate to other features by entering certain keys or buttons if you have a front-end:

```
run:
Company Name: Petronas
Internship Title: Data Analyst
Application Date: 19/11/2024
Interning Location: -
Contact Information: 017-9582568
Notes:
I hate life
Internship Status: Rejected

Company Name: Shell
Internship Title: Data Scientist
Application Date: 19/11/2024
Interning Location: Kuala Lumpur
Contact Information: -
Notes: -
Internship Status: Applied

Company Name: Riot Games
Internship Title: Game Developer
Application Date: 19/11/2024
Interning Location: Singapore
Contact Information: -
Notes:
This company gives high wage
Good Work Environment
Internship Status: Offer Received

Enter 0 to go to next page, Enter 1 to go to previous, Enter 2 to go to main menu
```

F. Tips and suggestions

1. Please use GitHub. It will make your life and that of your groupmates so much easier, as it will help you guys collaborate with each other. Well, unless your groupmates are free riders and you're soloing the project, then I guess GitHub isn't needed. But if you do have free-riders, do inform your tutor, as a lack of teamwork will affect your marks.
2. Be extra careful when inputting and changing data. File I/O is a very tilting part of FOP, and there's always something wrong with the inputted data. Be sure to always double-check these parts, as it can lead to various issues down the line. Use relative file paths as well. Don't pull file locations (i.e. from C:\Users\user...), because how are other people going to code as well?
3. Keep your code clean and easy to read. You're not the only one doing the assignment. If your groupmates can't understand the gibberish names you give to the objects and variables, how are you going to have a smooth collaboration? If possible, use comments as well, lecturers, demonstrators and fellow teammates will be reading your code, so make their life easier.

G. Contact me

If you have any questions, feel free to contact me on WhatsApp/Telegram at +60179582568 (Aman). This assignment is focused on the fundamentals of File I/O, as well as the basic concepts of Object-Oriented Programming. Enjoy and have fun, don't stress yourself too much.