Garrett Meyer

Professor Rivas

Software Development I

9 December 2016

Final Write-up

When I first started the project I had high hopes for the schedule I originally planned on coding, but the idea I had for the schedule in the beginning of the project changed into a whole new idea and code that I did not think I would have done. I will walk you through how I once perceived my project as a simple walk in the park to my worst nightmare and how I turned that nightmare into a code that I would be so proud of.

In the beginning of the Project I wanted to focus on businesses that have schedules for all their employees. The reason why I wanted to create this schedule is because I worked at a Golf Club this past summer and the manager had to organize a weekly schedule for every branch of the business and it was very time consuming. After narrowing my idea to a schedule my motive of the program was to make a solution that can make the process of creating a schedule much quicker and efficient. My original idea was that I would incorporate “back-ups”, which would be in a “for loop”, in the schedule along with a notification center. The back-ups are in case an employee that was intended to work at a certain time can no longer make it. From personal experience having a back up is crucial just in case that “what if” does in fact occur.

I first created a handwritten schedule with six designated timeslots: 8am-10am, 10am-12pm, 12pm-2pm, 2pm-4pm, 4pm-6pm, and 6pm-8pm. Once, the layout was completed I needed employees to fill in those slots. The employee’s names came from

one of my favorite shows “The Office”, I highly recommend it, and I took 10 names: Jim, Dwight, Mike, Andy, Ryan, Stanley, Creed, Toby, Gabe, and Kevin. After getting all the names down I tried figuring out how many employees I needed for first priority and back-ups. Inserting the names into the slots was not hard, but distributing an even amount of shifts to 10 employees was. Only four employees got eight shifts, while the other six got seven shifts. When all that work was done, it was time to start coding.

During the implementation stage of the project there were several problems with the program and to get it to run. There were several problems with the coding that would cause my program to not be easy to follow, allow back-ups, the schedule itself, and the notification center.

At first I wanted to make the program easy to comprehend and easy to keep track of. I realized later on in the implantation stage that the code that I was having a difficult time with was not going to be easy to comprehend, thus forcing me to try and make it much easier to follow and not get lost in it. My final solution to making the schedule easier was to have the program work by filling in who is working at the designated time slot for each day of the week. In the code I inserted an “or” after the first employee’s name and I inserted a second employee’s name, which established that either one can work. By adding a “back-up” the Manager does not have to worry about finding a fill in, they can just look at the program and the system will show him who said they would be able to work. The possibility of back-ups makes scheduling so much easier for both parties involved.

When I began coding I made separate pages “schedule”, “employee”, and “timeslots”. “Schedule” is the main page where everything would come together to

display the actual schedule. “Employee” was a page with all the information on my employees that included their name, availability, and the code for how I was thinking of implementing it. “Timeslots” was the page for the timeslots of the schedule with the index of each timeslot and day. What went wrong for me was that the codes were all out of whack and some codes would not match with the code I wanted thus forcing me to start over and take a different route to completing the program.

After I got the wheels rolling with my new code I hit another obstacle and that was the way of implementing the back-ups. I wanted to create a “for-loop” code that would state that if the employee could not make that another one would, which would be known as the back up. But while I was trying to figure that I came to the conclusion that I would have to do for loops inside the array lists. The “for loops” were going to be a major problem because that would force me to delay the time of me finishing the whole program. At this point I had several things completed, but now I had to figure out another new way to get the back-ups into the schedule.

The last problem that occurred was the notification center. At this point I was done with all my schedule coding and my program was able to run swiftly. If I were to add a notification center it would be longer and much harder than the coding I had just completed. Thus, I came to the conclusion of removing the notification center just to keep my sanity and not ruining what I had already finished.

My coding was a very difficult process, but everything I needed to do revolved around array lists. First I had to get the borders for the schedule so I used “println” with a bunch of “----“. They were inserted three times, one before and after the main bar with the time and days of the week and the other below all the information that were plugged

in. I used an array list for each day of the week with the employee working at that time and time slots. The two array lists were linked through a for-loop that would put the two together and at the end of the loop I had it print all the information. Lastly, I inserted comments for the readers to understand what each function do and why I put it there, along with a couple of sentences at the top explaining my program for anyone who did not know what this code was for.

After all the problems and I finally knew how I wanted this program to look I started to organize it and began running it. After several tries I had finally got it to run smoothly and quickly and it projected my schedule, as I wanted it to look. My schedule program was a very long process, but at the end of it I feel like I learned a lot from it and happy that I accomplished this hard task.