Criterion A: Planning

Client identification

My client is the school's coordinator of the International Baccalaureate section and my math teacher. During the summer break, she participates in creating a schedule for next year, and one of her responsibilities is assigning classes to classrooms.

Detailed scenario context

Until now, class assignments in my school were done manually. We have eleven classrooms in the school building, each with a different size and screen accessibility. The process was conducted by assigning the largest classes to the largest classrooms and then, if necessary, changing them according to teachers' preferences for screen availability, favourite classrooms, or the need to hold homeroom hours in homeroom classrooms.

Rationale for the product

Assigning classes to classrooms due to various aspects like earlier described teachers' preferences is both manual difficulty and inefficiency, but also very time-consuming. The process of random classrooms substitution and distinguishing whether it is suitable arrangement by correcting incompatibilities. That is why the program would relieve the coordinator from assigning classrooms to classes if needed. (See Appendix – Evidence 1 and 2 for details)

Idea

I aim to create a program, in which the user can add an earlier prepared schedule without assigned classrooms, edit teachers’ preferences and classroom features. Then, create a schedule with assigned classrooms to the classes that will handle the exceptional cases, like the homeroom hours in homeroom classrooms, and teachers’ preferences. In the end, my coordinator also seeks to find out what classrooms are empty and how many empty places are left. That is why the program displays a prepared schedule inside the program, where the user can to select the specific hour and check the building situation.

Technologies

For this task, I decided to choose PyQt as this environment provides a well-documented framework and has the benefits of a high-level language that allows for object-oriented programming. Python also allows for easy translation of ideas, performs calculations at high efficiency, and has strong libraries for Excel file handling (Pandas, openpyxl). While it is difficult to perform all these tasks in java efficiently.

Criteria for success

My client can:

Input an Excel file with a schedule (Criterion 1).

Add and remove teachers from the program’s memory (Criterion 2).

Edit classrooms’ screen availability, name, and size (Criterion 3).

Edit teachers’ screen availability preferences, favourite classrooms, and subjects held with their sizes (Criterion 4).

Check empty classes and empty places in the given classroom at a given hour (Criterion 5).

The program:

Complies with specific scenarios regarding teachers’ preferences and then assign classes by size (Criterion 6).

Returns an Excel file with classrooms’ numbers beside the subject (Criterion 7).

Link to client consultation

During the consultation, I sensed that it is important for the program to assign classrooms according to teachers’ preferences. Another aspect was an easy-to-use interface, as checking the situation in the building requires a straightforward preview, so a user interface is necessary. (See Appendix – Evidence 1 and 2 for details)

Extensive writing word count: 358