Criterion A: Planning

The scenario

My client, Ms. Rýdza, was head of the organising team of an initiation course for junior IB students this year. Every year, DP2 students from our school organise this course for IB juniors, who get to know each other better thanks to numerous activities. I noticed a flaw in how the groups for these activities were created. Namely, juniors had to be divided into groups by hand, repeatedly. This task took considerable amounts of time. Due to its tediousness, in some cases, it led to the organisers accidentally duplicating or removing students' names when creating groups, resulting in confusion. I decided to meet Ms. Rýdza[1] to find out about how groups were created and propose an improvement of the process.

The group-creating process was the following: Before each activity started, new randomised groups had to be created. Sometimes, juniors who previously attended the same class had to be separated to encourage contact with new classmates; other times, as Ms. Rýdza pointed out, the groups were created so as to get as many boy-girl pairs as possible. The groups also varied in size, which was either given or had to be calculated from the required total number of groups.

Rationale for solution

I offered to prepare for Ms. Rýdza a bespoke application with a straightforward UI to simplify the group creating process, saving time for future organising teams. The currently available applications do not provide the required functionality (*i.e.* they only create random groups) or are in the works (e.g. the project Team Optimizer, which will likely be paid). **Therefore, developing this application is sensible.**

Implementation-wise, I am most experienced in Java. Java offers cross-platform support, which will extend the use of the product for future organising teams whose members work with various operating systems. Moreover, Java's JavaFX API will help me achieve the required simplicity of the user interface. Furthermore, the use of objects will result in more readable and maintainable code and thus a higher quality product for Ms. Rýdza. Therefore, I will opt for Java as a programming language

Word count: 347 words

[1] Initial consultation with Ms. Rýdza, outlined in Appendix 1

Success criteria

1. UI is straightforward for the user
2. Student data (name, gender, previously attended class) are protected from unauthorised access
3. Students' names contain diacritic and are displayed correctly
4. User can manage students (add and remove, or edit the associated fields)
5. User can create groups by specifying group size or number of groups to be created, and whether students should be dispersed based on previously attended class, gender, or both
6. User can set the title of the activity for which groups are being created
7. Groups are created non-deterministically
8. User can add notes and select a captain for each group after groups were created
9. User can export the activity name, created groups, notes and captains to a PDF file
10. Future users can download the application from an online repository