

Criterion E: Evaluation

Meeting success criteria according to the developer testing:

Success criterion	Status
UI is straightforward for the user	Met. Consultation of preliminary design and of the final product with the client confirms this.
Data are protected from unauthorised access	Partially met. Correct credentials are required to gain access to the app. All data are encrypted using Caesar cypher. Due to the encryption method's considerable weaknesses, this criterion is considered not fully met.
User can manage students	Met. A custom screen was developed for this purpose.
User can set the title of the activity	Met.
User can create groups based on a number of input criteria	Met. User can create groups based on group size or number of groups, gender and previously attended class.
User can add notes and select a captain for each group	Met.
User can export the created groups to a PDF file	Not met. Currently, only text files are generated.
The product is accessible from an online repository	Met. The product is publicly available on my private repository.

Client feedback

The client ascertained that except creating PDF files, all success criteria were met. The client is satisfied with the final product:¹

Client's opinions	
The user liked:	The level of security.
	That student data are shown in a table. Editing data and sorting table columns is easy.
	The attention to detail when adding new students.
	The way activity name and notes can be added.
	That it suffices to click on a student's name in a group to declare them as the group's captain.
The user did not like:	That columns in the table containing data about students were not sorted by surname.
	That there was no possibility of direct input of a number when selecting the number of groups/group size.
	The unclear caption "Filter gender" on the screen for creating groups.

¹ A more extensive account is available in Appendix 2

	That there is no alert about completion and location of the file after created groups were exported.
	That text files were used for storing created groups instead of PDF files.

Future improvement

Besides addressing the issues discussed in the previous table, the following could be implemented:

Online database

Currently, the program works with an offline text file, which limits the use on multiple computers despite the app's cross-platform support. For example, if one organiser amends the file (e.g. a student is removed from the student list), the file has to be shared manually with other organisers who might want to use the application on their computer. To resolve this, the application could store data in an online database on a webserver.

Stronger encryption

Hosting data online would require significantly stronger means of encryption than the currently-implemented Caesar cypher, such as the Advanced Encryption Standard (AES).

PDF file generation

Currently, a text file is used for exporting created groups. This, however, provides limited means for formatting. Therefore, properly-formatted PDF documents could be created to improve the readability of output. However, using a library such as iText would entail extra cost.

Cloud storage

Currently, all generated text files are stored locally. The use-case however requires sharing of the generated files. Therefore, the created files could be pushed directly to a cloud folder accessible for every organiser from their mobile device. This could be achieved for example using the Google Drive API.

Emails

At present, the app helps the organisers divide junior IB students into groups. The juniors are not able to check to which group they belong. An email could be sent to each junior stating which group they belong to and possibly some other details (e.g. notes). This could be easily achieved by requiring the students' email addresses when adding them to the student list and working with Gmail API.

Profiling

Currently, each logged-in user has all the privileges – to add and delete students and to edit student data as well as to create groups. However, it might be necessary to modify the privileges for certain users. User profiling could be done thanks to the use of credentials – a different username-password combination would result in different privileges assigned to a user.

Word count: 649 words