Criterion A: Design

1. Problem Description

Year after year, DECA is the largest club at my school, usually with over 80 members a year. DECA is a business competition and members of the team are partitioned into different subteams with different events.

As a club executive in the DECA club, I help record and organize the members of our team. However, with the existing team management software (Google Spreadsheets/Excel), it is inefficient for club executives like myself to sort or filter for groups of members and display the data about each team member in a visually-appealing manner. For example, it is very difficult to filter members by their subteam or export the email addresses of specific members (i.e. grade 9s or people in the Finance subteam) without having multiple spreadsheets for different subsets of members.

Ultimately, the existing team management system used by club executives is not as effective or efficient as it could be. To improve our efficiency, I decided to write a team management software application designed specifically for organizing members in a DECA club. I will be both the developer and the client for this project.

Word Count: 188

2. Proposed Solution

The goal of my program is to reduce time spent by club's executives organizing members and viewing relevant information about the different members/subdivisions of our team.

The application will implement a team tracking and management system with features specific for DECA that will help make managing DECA teams more efficient and effective. For example, there will be functions to filter by grade or by subteam, and then the option to export selected emails from filtered lists or team search results (which is important when we need to send emails to specific subsets of the team). The program must also be able to import team signup data from a preformatted spreadsheet so we can easily import members from our Google form signup sheets (which can be exported to a spreadsheet file).

I decided to use OOP in Java because the complexity and nature of this program tends well to OOP principles (such as creating different Member objects to hold data about members, and a TeamList object to partition the functions of the team lists). Using this modular style of programming will also make it easier to extend the product in the future.

This solution with its DECA-specific management features will ultimately streamline the process of adding members, editing their information, and overall managing our team.

Word Count: 215

3. Functionality and Success Criteria

General Application Functionality:

- User-friendly Graphical User Interface (GUI)
- Error boxes will inform the user of errors
- Data cleansing and validation of inputs
- Automatically save and reopen data when the program is closed and re-opened

Inputting and Editing Members:

- Add team members by importing a CSV file or through manual input
- Ability to input a member's full name, grade, DECA event ID, DECA subteam, email
  - **Grade**: only grades 9-12 are allowed
  - **Subteams**: Finance, Marketing, Business Administration, Principles, Writtens, Hospitality & Tourism, None (when a new member registers for the team, they are not immediately placed into a subteam)
- Edit or delete individual members
- Bulk edit selected members (only subteams or event IDs can be bulk edited)
- Bulk delete selected members

Sorting, Filtering, and Searching:

- Sort members in alphabetical order by first name or last name
- Filter members by subteam or grade, or by both
- Search filtered list or full team list by name, event ID, or email
  - Search should return partial matches sorted by relevance

Total Word Count: 403