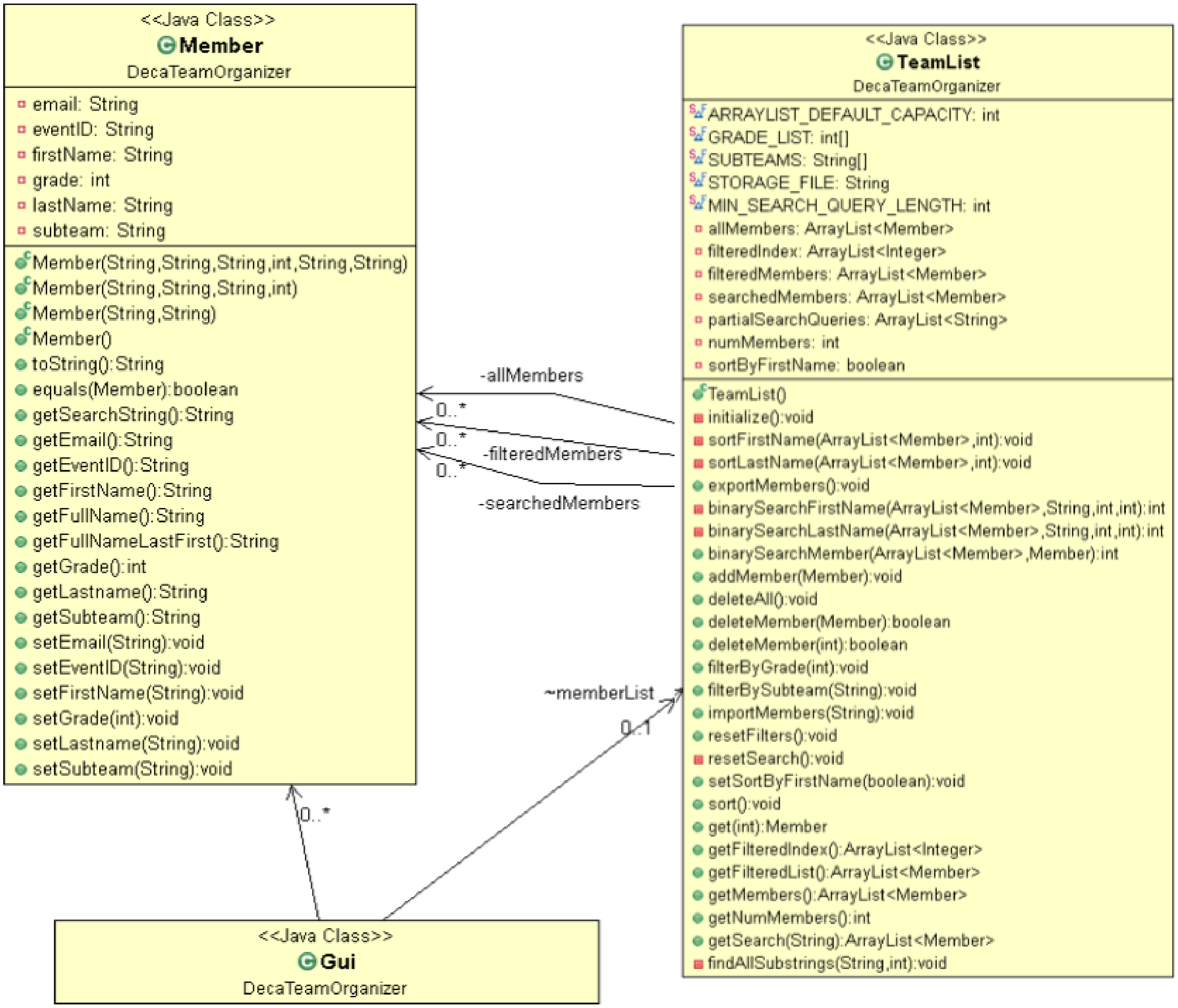
Criterion B: Design

Class Functionality

UML Diagram:


<<Java Class>>
Member
DecaTeamOrganizer
email: String
eventID: String
firstName: String
grade: int
lastName: String
subteam: String
Member(String,String,String,int,String,String)
Member(String,String,String,int)
Member(String,String)
Member()
toString():String
equals(Member):boolean
getSearchString():String
getEmail():String
getEventID():String
getFirstName():String
getFullName():String
getFullNameLastFirst():String
getGrade():int
getLastname():String
getSubteam():String
setEmail(String):void
setEventID(String):void
setFirstName(String):void
setGrade(int):void
setLastname(String):void
setSubteam(String):void
-allMembers
0..*
0..*
-filteredMembers
0..*
-searchedMembers
~memberList
0..1
0..*
<<Java Class>>
TeamList
DecaTeamOrganizer
ARRAYLIST_DEFAULT_CAPACITY: int
GRADE_LIST: int[]
SUBTEAMS: String[]
STORAGE_FILE: String
MIN_SEARCH_QUERY_LENGTH: int
allMembers: ArrayList<Member>
filteredIndex: ArrayList<Integer>
filteredMembers: ArrayList<Member>
searchedMembers: ArrayList<Member>
partialSearchQueries: ArrayList<String>
numMembers: int
sortByFirstName: boolean
TeamList()
initialize():void
sortFirstName(ArrayList<Member>,int):void
sortLastName(ArrayList<Member>,int):void
exportMembers():void
binarySearchFirstName(ArrayList<Member>,String,int,int):int
binarySearchLastName(ArrayList<Member>,String,int,int):int
binarySearchMember(ArrayList<Member>,Member):int
addMember(Member):void
deleteAll():void
deleteMember(Member):boolean
deleteMember(int):boolean
filterByGrade(int):void
filterBySubteam(String):void
importMembers(String):void
resetFilters():void
resetSearch():void
setSortByFirstName(boolean):void
sort():void
get(int):Member
getFilteredIndex():ArrayList<Integer>
getFilteredList():ArrayList<Member>
getMembers():ArrayList<Member>
getNumMembers():int
getSearch(String):ArrayList<Member>
findAllSubstrings(String,int):void
<<Java Class>>
Gui
DecaTeamOrganizer

Note that the Gui Class's instance variables and methods are not shown because the diagram would not fit the page. These can be viewed in the class documentation in Appendix 1.

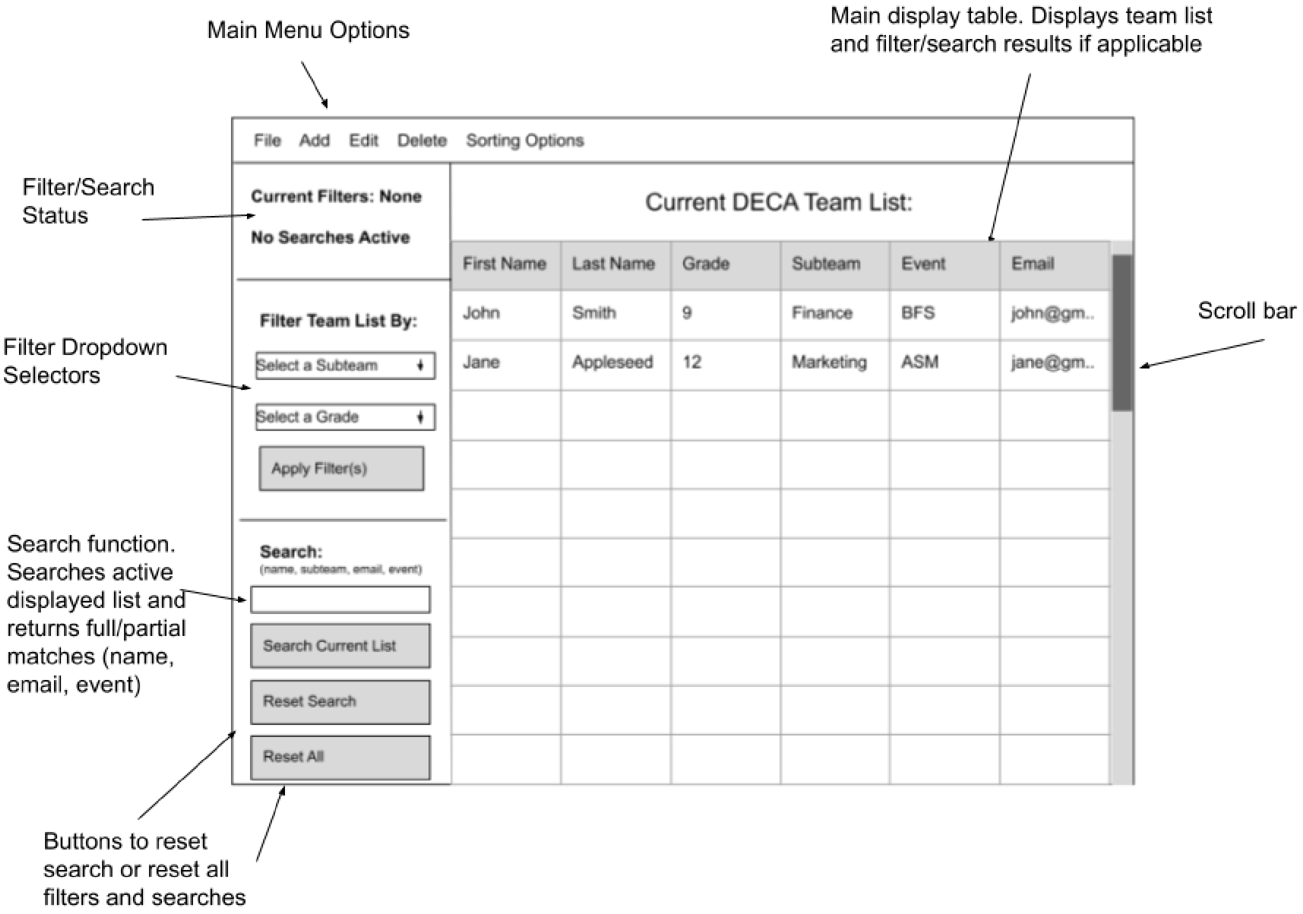
Member Class:
Stores data about each Member in the team. Data stored:

First Name	Last Name	Grade	Subteam	Event ID	Email

TeamList Class:
The TeamList Class stores ArrayLists of Member Objects that are in the DECA Team. It provides the functionality and algorithms for adding, editing, and deleting members, and also provides the filtered, sorted, searched team lists for the GUI.

Gui Class:
The Graphical User Interface for the product with a dependency on the TeamList and Member class to function. Displays members in tables, and graphical interface for user interaction.

GUI Design:



Main Menu Options

Main display table. Displays team list and filter/search results if applicable

Filter/Search Status

Current Filters: None

No Searches Active

Filter Dropdown Selectors

Search function. Searches active displayed list and returns full/partial matches (name, email, event)

Buttons to reset search or reset all filters and searches

Scroll bar

Current DECA Team List:

First Name	Last Name	Grade	Subteam	Event	Email
John	Smith	9	Finance	BFS	john@gm..
Jane	Appleseed	12	Marketing	ASM	jane@gm..

Figure 1: Mockup of the planned Graphical User Interface

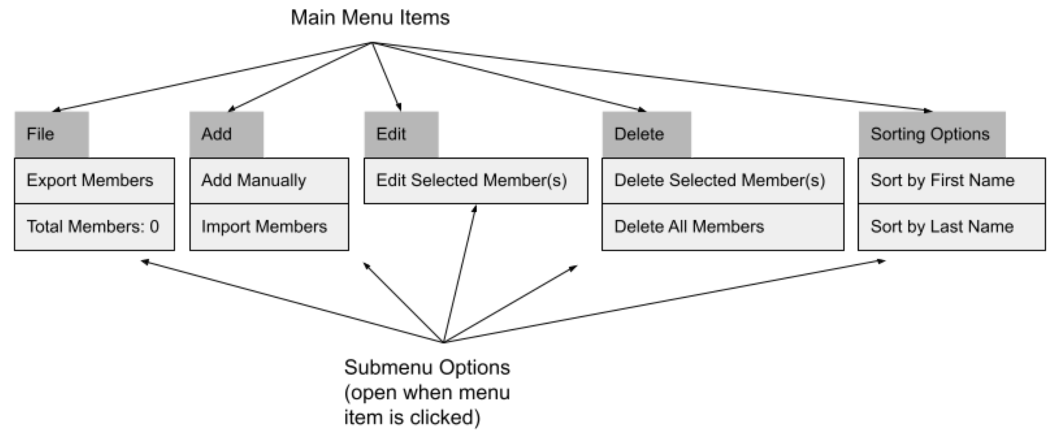
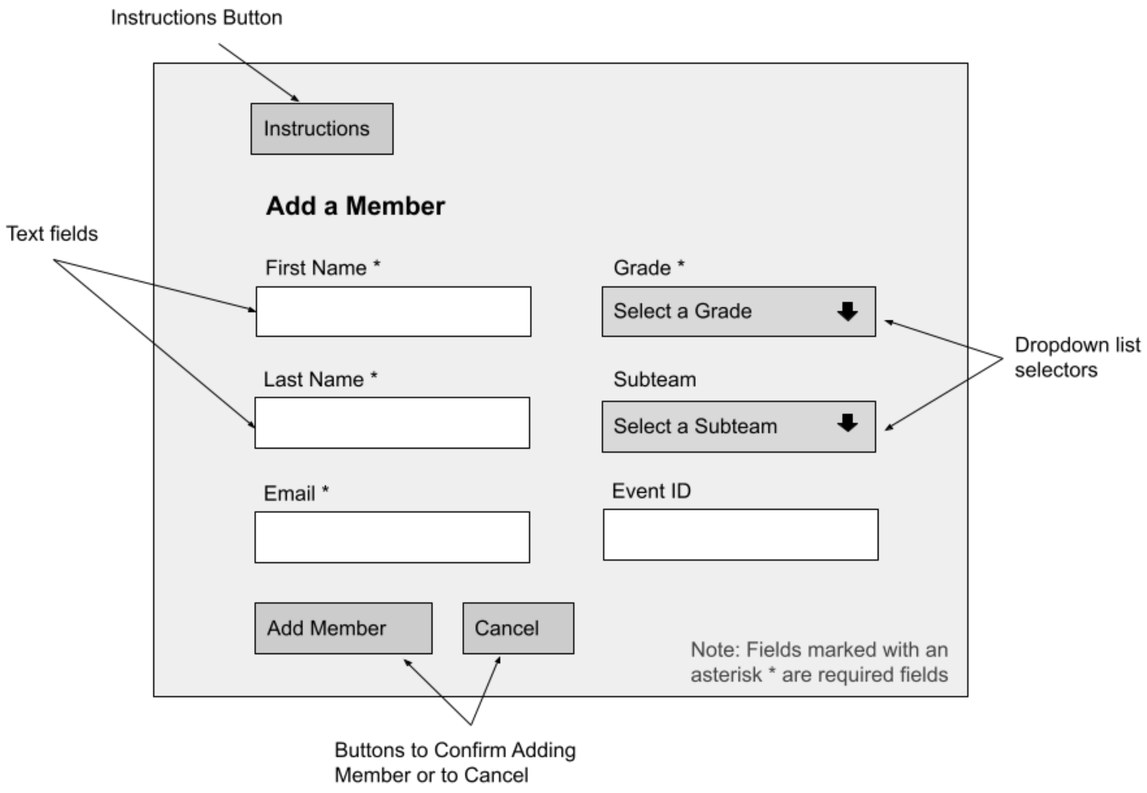


Figure: Main Menu Options Expanded from Figure 1



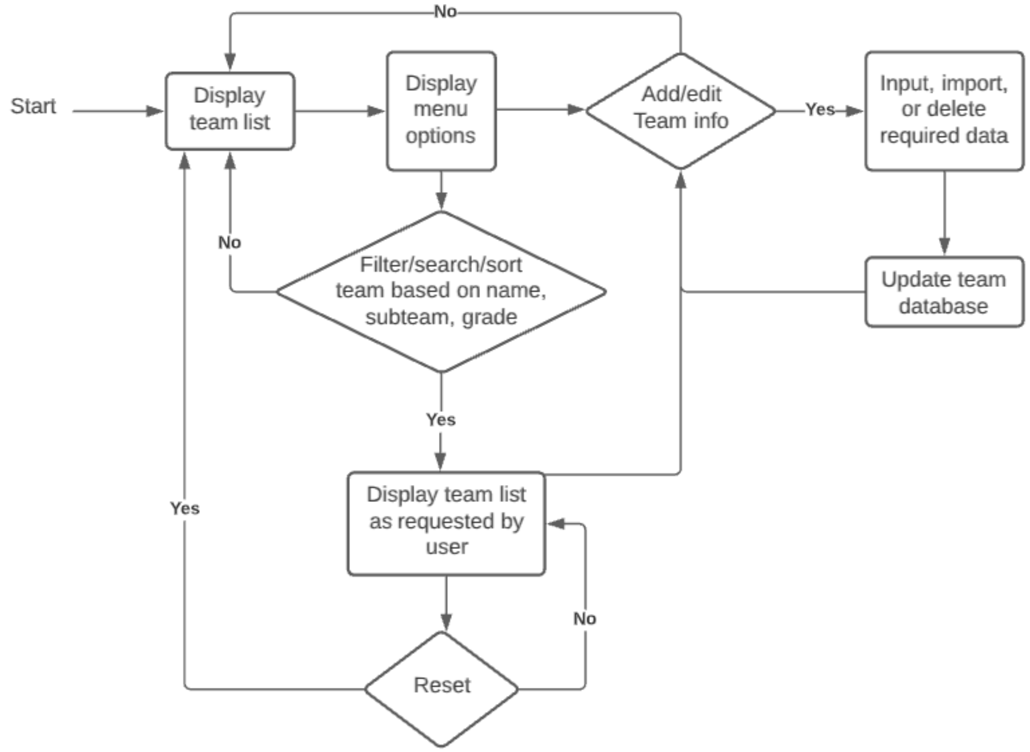
Note: Fields marked with an asterisk * are required fields

Figure 2: Mockup of the GUI for adding a member (GUI for editing a member will be similar, except that the fields will be pre-populated with the existing member’s information)

Input Data:

Input Team Data: Method 1	Extra Information:
Team data can be imported into the program by importing a pre-formatted CSV file	The program will read data from the file and add it into the team database for display
Input Team Data: Method 2	**Extra Information:**
Members can be manually added or edited using the panel in Figure 2.	- Grade and Subteam input is done through a dropdown (JComboBox) to prevent erroneous entries. - All text fields are trimmed of spaces and names have their first letter capitalized. - Event IDs are fully capitalized.

Program Process Plan:



Start
Display team list
Display menu options
Add/edit Team info
No
Yes
Input, import, or delete required data
Update team database
Filter/search/sort team based on name, subteam, grade
No
Yes
Display team list as requested by user
Yes
No
Reset

Figure 3: Basic Program Process Plan

Development Plan:

Date Range	Functions
1 Week	1. Create GUI framework and main menu options
2 Weeks	1. Input using file and outputting the team list in the program 2. Input manually using program GUI input 3. Edit or delete existing team data 4. Export of all team data back into an excel file
2 Weeks	1. Sorting, searching, and filtering algorithms 2. Display sorted, search, filtered team lists as required by user 3. Export email list

Testing Plan:

Test Scenario	Method and Success Condition
Add Members individually or by bulk	-Adding individual members works -Required fields must be filled out, but other fields (subteam, event ID) can be filled out late. -Importing a file of members works -Imported members can override existing list or be added to the existing list
Sorting works	-Sorting options: sort by either first name or last name -Members are auto sorted when added, edited, or deleted -Table columns can be rearranged by user
Filtering members by grade, subteam, or by grade and subteam	-Filter displays proper list of members -Filtering by grade or subteam should work individually or together (i.e. filtering for grade 9 and finance subteam returns only members in grade 9 and finance)
Search function works	-Search should retrieve both exact and partial matches -Results are sorted in order of relevance -Searches by names, email, event ID -Search function will only search within the displayed list
Status messages work	Status messages should show for searching or filtering
Reset buttons work	-Search button resets search, but not filters -Reset all button resets all filters and search
Editing members works	-Edit members individually allows editing of all fields -Editing by bulk only allows editing of subteam and event

	-Editing works from main list, filtered list or search results -After editing, all lists update and display properly
Deleting members works	-Delete individual members from any displayed list -Delete all members from the team -After deletion, all lists update and display properly
Exporting emails works	-Emails can be selected/exported from any displayed list
Data validation works	-If invalid data is entered, show error popup to user. Make sure program does not crash
Error messages work	-Error messages with instructions should appear if adding, editing, deleting, filtering, or searching members is not done properly
File export works	User can retrieve exported csv file of team info
Automatically saves data	Upon closing and reopening the program, any previous data should be automatically saved
Program is responsive	If the window is resized, the program should resize its contents to fit properly and still work properly

Word Count: 81 (class descriptions)