```
// Define ClubCategory enum
enum ClubCategory { Arts, Technical, Social, General, Sports }

// Define Member structure
struct Member {
    string name
    string student_id
    vector<string> clubs
}

// Define Club structure
struct Club {
    ClubCategory category
    vector<string> members
}

// Define unordered map for member and club information
unordered_map<string, Member> member_hash_table
unordered_map<string, Club> club_hash_table

// Function to add a new member
add_member(name, student_id, clubs):
    Create new Member object
    Set member attributes (name, student_id, clubs)
    Add member to member_hash_table
    Iterate over clubs:
        Add member's name to the corresponding club in club_hash_table

// Function to append a new member to file
add_member_in_file(name, student_id, clubs):
    Open ClubMembersDetails.csv in append mode
    Write member details (name, student_id, clubs) to the file
    Close the file

// Function to convert ClubCategory to string
str_category(category):
    Switch on category:
        Return corresponding string representation

// Function to add a new club
add_new_club(club_name, category):
    Open ClubCategoriesDetails.csv in append mode
    Write club details (club_name, category) to the file
    Close the file
```

```
// Function to search members by club name
search_by_club_name(club_name):
    Check if club_name exists in club_hash_table:
        Print all members of the club

// Function to search members by name
search_by_member_name():
    Input member name
    Search member in member_hash_table:
        If found, print member details

// Function to search members by ID
search_by_id():
    Input member ID
    Search member in member_hash_table:
        If found, print member details

// Function to search members by club category
search_by_club_category():
    Input category
    Iterate over club_hash_table:
        If club category matches input category:
            Search members of that club and print them

// Function to view all members by club and category
view_all_members_by_club_and_category():
    Iterate over each club category:
        Print category
        Iterate over clubs in that category:
            Search members of the club and print them

// Function to remove member by ID
remove_member_by_id(student_id):
    Open ClubMembersDetails.csv
    Open a new file for writing
    Iterate over lines in the file:
        If the member ID matches the given ID, skip that line
        Otherwise, write the line to the new file
    Close both files
    Delete the original file and rename the new file

// Function to read member details from file
read_ClubMembersDetails(filename):
```

```
    Open ClubMembersDetails.csv
    Read each line from the file:
        Parse member details and clubs
        Add member to member_hash_table
    Close the file

// Function to read club categories from file
read_ClubCategories(filename):
    Open ClubCategoriesDetails.csv
    Read each line from the file:
        Parse club name and category
        Add club to club_hash_table
    Close the file

// Main function
main():
    Initialize data structures
    Read member and club category details from files
    Display menu options and prompt user for input
    Perform corresponding actions based on user input until the user chooses to exit
```