CSC Match - Initial Problem Description

Every social group today has a way to connect with like-minded individuals (e.g. FarmersOnly.com). You've been tasked with understanding and designing an application that can track Computer Science students and their interests. This app will be used to find other students with similar interests for study sessions.

You are to create class diagram(s) in StarUML, showing

- All of the Classes
- All of the Attributes
- All of the Operations
- All of the Relationships (inheritance, realization, association, dependency)

Functional Requirements

Below, you will find the requirements for the product. Not all of these requirements will be evident in the high-level design! In your design, you are capturing the major concepts, attributes and relationships.

1. The user should be presented with a menu of the following activities:
    a. Load the Members
    b. Save the Members
    c. List All Members
    d. Add a Member
    e. Remove a Member
    f. List Member
    g. Add an Interest to a Member
    h. Quit
2. Each Member includes a name, and year (1-5).
    a. A freshman is a 1, sophomore 2, etc. A CS student with a degree is a 5.
3. Each Interest includes a topic (e.g. "Assembly Language", "Java", "Digital Design", "Cyber Security", "Web Design", "C++", etc.) and an interest level, from 0-10 (least to most).
    a. There is no predefined list of Topics, the user can enter anything they want.
4. Each Member may have any number of Interests (including none).
5. When adding a new Member, make sure their name is unique.
6. When saving the Members, ask the user for a filename. If the file cannot be written, tell the user and ask for another name.
    a. Its ok to overwrite an existing file. No warning is necessary.
7. When loading the Members, ask the user for a filename. If the file cannot be read, tell the user and ask for another name.
8. When adding an Interest to a Member, if the Interest already exists, replace the Interest with the new one.

9. When Listing All the Members, list only their names in the order they were added to the application.
10. When Listing a Member, list his/her name, year and list of interests, from highest interest, to lowest interest. Also, list the name of his/her top five "Matches".
    a. The name and order of top matches may change as Members gain additional Interests, or their interest level's in certain topics change.
11. A Member is matched to another by computing a "Compatibility Score", using the rules listed below. Note that A's interest in B may not be the same as B's interest in A.
    a. If a topic matches, the score is increased by (my_interest x their_interest). A compatible interest!
    b. For any topics they have that I don't have, increase the score by (their_interest/2), rounded down. The other person is interesting in general.
    c. Example: What is the score for A compared to B?

Member A		Member B	
Topic 1	2	Topic 1	5
Topic 2	3	Topic3	3
Topic 3	5	Topic 4	7
		Topic 5	1
		Topic 6	9

A's interest in B Score = (2 x 5) for Topic 1 match + (5 x 3) for Topic 3 match + 7/2 for Topic 4 + 1/2 for Topic 5 + 9/2 for Topic 6 = 10 + 15 + 3 + 0 + 4 = 32

    d. Example: What is the score for B compared to A?

B's interest in A Score = (2 x 5) for Topic 1 match + (5 x 3) for Topic 3 match + 3/2 for Topic 2 = 10 + 15 + 1 = 26

12. When quitting, if the user has made any changes since they last saved, warn the user. If they still insist on leaving, allow it. Otherwise, cancel the "Quit" so the user can go back and save.
13. Any requirement which is unstated is left to the designer to invent. Document this.