**Project Milestone 1**

**Name:** Hardee Rameshchandra Garala **Std. ID**: B00869195

**Faculty**: Prof. Mike McAllister **Date**: 4th Nov 2021

**Objective**

To analyse the project problem in depth so it becomes easy to decide the data structure and algorithm to be used and plan the project effectively.

There are various details that we get from the user for the implementation of this program. We need database to store these details. Some details in this problem are mandatory marked with an asterisk (\*) while others are not mandatory. To create the family tree we take PersonIdentity class and make nodes of type PersonIdentity and using these nodes, the family tree can be created. Firstly we will add the person to the database and the tree by creating addPerson method. Later we pass this added person as an argument in different methods to store data like reference, notes, and different attributes for that individual.

Family tree database will include information about individuals which are as below:

1. Name\*
2. Date of birth
3. Location of birth
4. Date of death
5. Location of death
6. Gender
7. Occupation
8. References to source material
9. Notes to individual

To create the database related to photos, videos and any other media related to a person first we store the media and establish a relation from person table to media table with all the details provided through calling different methods like recordMediaAttributes, peopleInMedia and tagMedia. We create Media Archive database which contains below details:

* Filename\*
* Date of picture taken
* Location

1. Location name
2. City name
3. Province
4. Country name

* Tags
* All individuals

Once media details are stored we can implement different method to get this data and process it to get a desired output. We can create a tree depending on the relation between all the people added. We also record child, partnering and dissolution that becomes helpful while finding relation between 2 individual and getting the names of people of X generations related to a person Y. The relations can be as below:

* Parent/Child relations
* Partnering ceremony relations – marriage
* Partnering dissolutions – divorce

**Basic algorithm:**

Through the database and data structures created, we can obtain the required results by following below algorithms roughly for respective questions.

To find relation between X and Y:

1. Find number of ancestors X have from the designed tree.
2. Find number of ancestors Y have from the designed tree.
3. Take Number of ancestors of X – Number of ancestors of Y

* If this number is greater than 0: It means that X has more ancestors, so move to the parent that is the difference of Number of ancestors of X and Number of ancestors of Y away from X. Then move to the next parent one step at a time for both X and Y and compare parents. At some point if there is a common ancestor that will be found and we can say if there is a relation and calculate degree of cousinship and degree of removal.
* If number is less than 0: It means that Y has more ancestors, so perform the vice versa steps of above.
* If number is 0: It means that both are at same level in a tree and there is no need for any node to move up to its parent and we can start comparing parents right away and move up till we find the common node.

1. For all the above 3 conditions if there is no common ancestor, then it can be said that X and Y are not related.

To show references and notes of person X:

1. Firstly we need to check for person X in database.
2. Write a query to fetch requested details
3. Return the requested data

To find ancestors or descendants of person X for Z generations:

1. Find X in tree.
2. Move to the top side of the tree for Z times from X and store name in each visited nodes to find the ancestors.
3. Similarly to find the descendants of the

To list pictures of a given set of people in particular time range.

1. For the set of people gain all the pictures.
2. Since date may be partial that is only year might be present. Hence, need to sort. Check years of all these pictures.
3. Arrange chronologically
4. Take first year
5. Starting from January take all months chronologically
6. if for same month date present take dates in order
7. For those partial date for which only month present, take it after the complete date containing part of that number of the day of month is complete.
8. For the partial date containing only year take it after completing all the months of that year.

More algorithms might be needed to solve other problems as encountered once start working on the project. For now we can say that these will suffice to achieve the required result.