

Team M
Requirements/Use Cases Document

Functional Requirements:

1. A user can search for an existing person by entering the first or last name.
 - a. Information to be provided by the user:
 - i. First name or Last name
 - b. The results will provide the user with the person or persons which share the name given. If no person is found by this name there will be an error returned
2. A user can search for the grandparents of an existing person
 - a. Information to specify:
 - i. pID of the known person
 - b. The results will provide the person who was searched for and their grandparents.
 - c. Limits: the user must know the pID of the person whose grandparents they want to find
3. A user can search for an existing person by their pID
 - a. Information to be provided:
 - i. pID
 - b. The result will be a person associated with that specific pID, or an error will shown that the pID does not belong to anyone
4. A user can add a new person to the existing tree in the app
 - a. Automatically pass (1) to check if the person is in the app or not
 - b. Information the user needs to specify:
 - i. Name
 - ii. Birthday
 - iii. City of birth/residence
 - iv. Any known family relationships (spouse, parents, children)
 - v. pID
 - c. The new person will then be added to the existing tree.
5. For any person, find people of a specified relationship

This has only been partially implemented. You can find both parents together through the `getParents` method and both grandparents through the `getGrandparents` method (see requirement 2). “Child of” and “Partner of” were not implemented due to bugs.

**Utilizing the relationship using “mother of”, “father of” or “child of” is the least erroneous way to navigate a specific relationship. This is because there can be a possibility of multiple grandparents and multiple grandchildren within any branch of the overall family tree.

- a. Parent is Mother(Full name) is mother of “c”, Father(Fullname) is father of “c”.
- b. Grandparent is searched through “mother of”.

- c. Child is Child(Full name) is child of “a”
 - d. Grandchild is searched through “child of”.
 - e. Cousin is searched through “child of”.
 - f. Partner is a person who does not have “child of” defaults as partner.
6. A user can provide two pIDs to determine if they are related (they will have a common ancestor somewhere “up” in their family tree)
- Not implemented**
- a. The user will provide the pID of each of the existing persons they want to search for a relationship of.
 - b. The system will search for the common ancestor and return them or return an error if not found.
 - c. Limits: only shows first related ancestor, does not continue to check for other ones
7. A user can search for an existing relationship using an rID
- a. User has to specify:
 - i. rID
 - b. The result will be the information for the two people of the relationship specified or if rID has no relationship associated with it, an error will be returned.

Nonfunctional Requirements:

1. Use Java v11 and IntelliJ for our program development. Save our work in the course GIT repository for your team: Comp330Fall2020TeamM
2. Use of Java Swing for creating an interface
3. Structure the system so that the user interface is separate from the logic and searching functions.
4. Have a well-structured functional decomposition of the app into separate parts. This decomposition should support separate development of key components by individual programmers.

Use Case Narratives:

Event: A user provides a file to read in

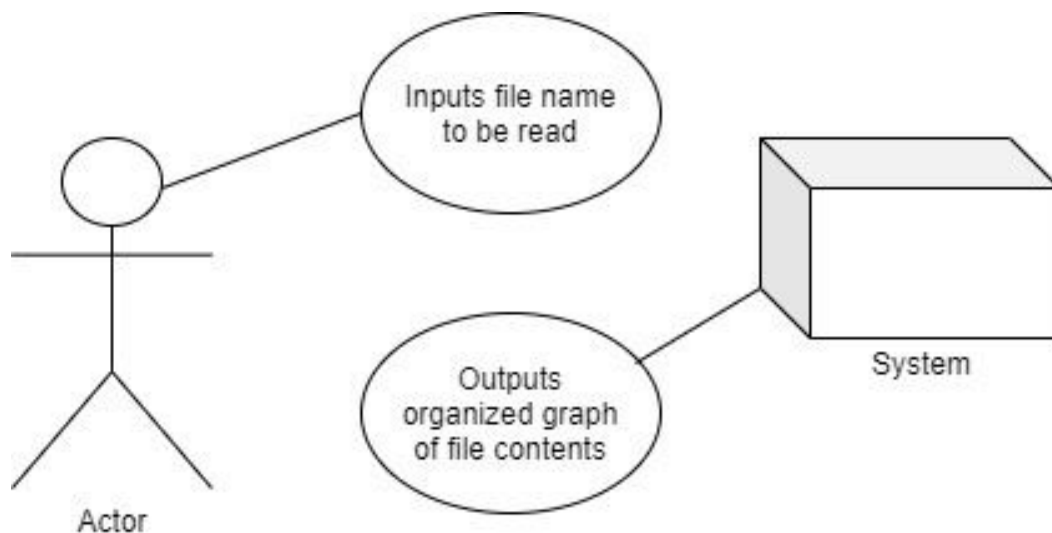
Actor: Person

Purpose: To read in a file from which to create a family tree from

Overview: The actor provides the system with a file, the system reads the file and provides the graph to the user. If the file is wrongly formatted, the system returns an error to the actor.

Precondition: The user must have a file to input.

Postcondition: The user can print a visual of a family tree.



Use Case Narrative:

User

1. The user uploads a file to be read into the app
4. The user receives the output of a graph

System

2. The system reads the file and separates the pieces of information into a graph data structure
3. The system outputs the graph to the user

Alternative Flow:

Line 2: File structure is incorrect and the system cannot process information. Print error. Return to step 1.

Event: A user can search for an existing person by entering the first or last name.

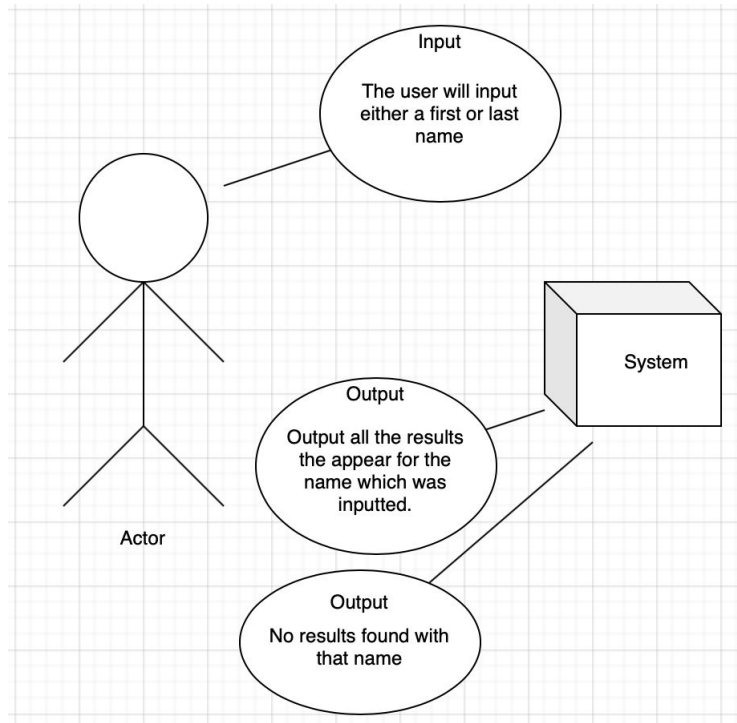
Actor: Person

Purpose: Search if a person exists in the tree by using first or last name

Overview: The actor provides either the first or last name of a person and submits it. The system will traverse the tree and find all the people who have that name, if none found, an error is returned. If found, the system outputs all people who have the name inputted.

Precondition: The user must provide either a first or last name.

Postcondition: The user will know all people in the tree who have the name they inputted.



Use Case Narrative:

User:

1. User inputs either a first or last name for person they are searching for
2. User submits the info into the search
6. The user receives the output displaying all results of the inputted name

System:

3. The system will traverse the tree searching for people with the name
4. Collects every instance of the inputted name
5. The system outputs all the people with the same name as given by actor

Alternative Flow:

Line 3: If the tree is traversed and people with the name cannot be located, the system will print an error. Return to step 1.

Event: A user can search for the grandparents of an existing person

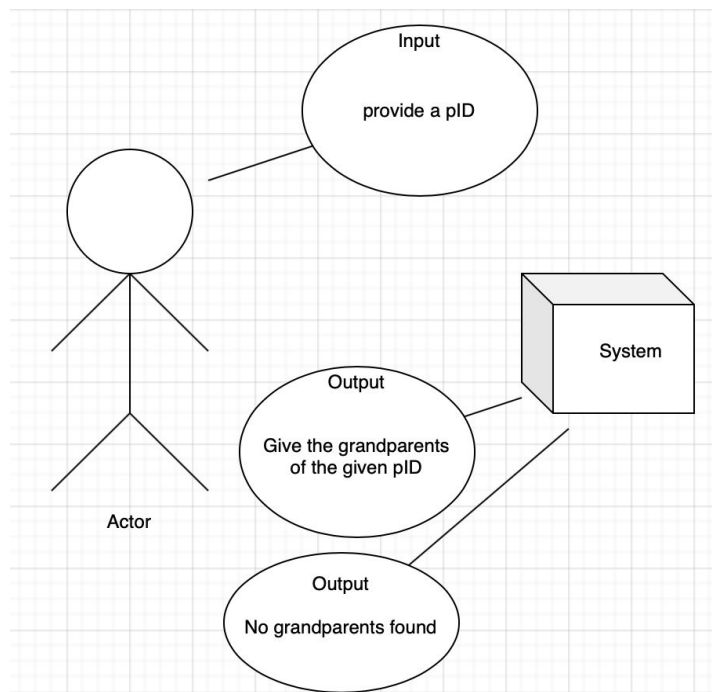
Actor: Person

Purpose: Search for grandparents of an existing person

Overview: The actor will input the pID of the person whose grandparents they want to find. Then from the dropdown, they will choose “grandparents.” The system will locate the person attached to the pID and use the getGrandparents method to get the grandparents. The system will then output the grandparents. If the system does not find a person for the pID then it will return an error. If the system does not find grandparents for pID then it will return an error.

Precondition: The user must have a pID and choose grandparents from selection

Postcondition: The user will know the grandparents of given pID



Use Case Narrative:

User:

1. User inputs a pID and selects grandparents from the dropdown
2. User submits the info into the search
6. The user receives the output displaying the grandparents of the pID

System:

3. The system will traverse the tree searching for person with given pID
4. When found, it will get the grandparents of the pID
5. The system outputs the grandparents of pID

Alternative Flow:

Line 4: If the tree is traversed and the pID cannot be located, the system will print an error.

Return to step 1.

Line 5: If the pID does not have grandparents, the system will return an error. Return to step 1.

Event: A user wants to determine if a person is known to the app using pID

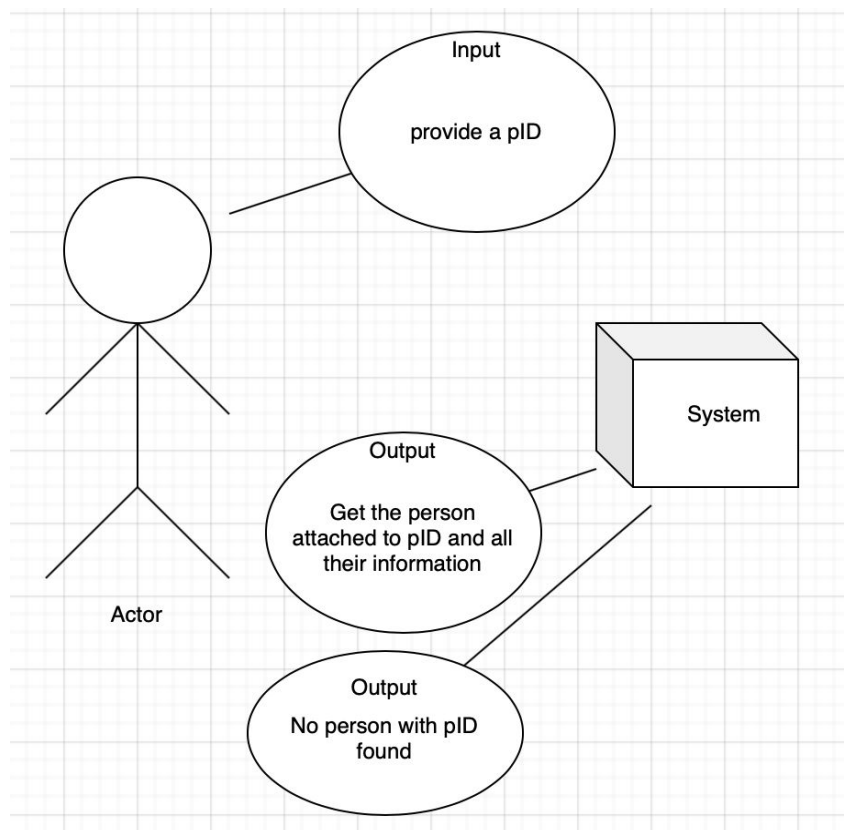
Actor: Person

Purpose: A user can check if a person is known in the family tree to the system

Overview: The actor provides the pID to search for a person and submits it. The system traverses the tree until it finds the person matching the pID and returns them along with related info. If no person found the system prints an error.

Precondition: The user must have the pID

Postcondition: The user will know if a person is or is not known to the app



Use Case Narrative:

User

1. The user provides the pID of the person
2. The user submits the entered information for the search
5. The user receives the person along with related information

System

3. The system searches through the tree of people until a person with a matching pId is found.
4. The system outputs the person and their information and relationships to the user

Alternative Flow:

Line 3: The system searches through the system and does not find a match. Prints a system error.
Return to step 1.

Event: A user can add a new person to the existing tree in the app

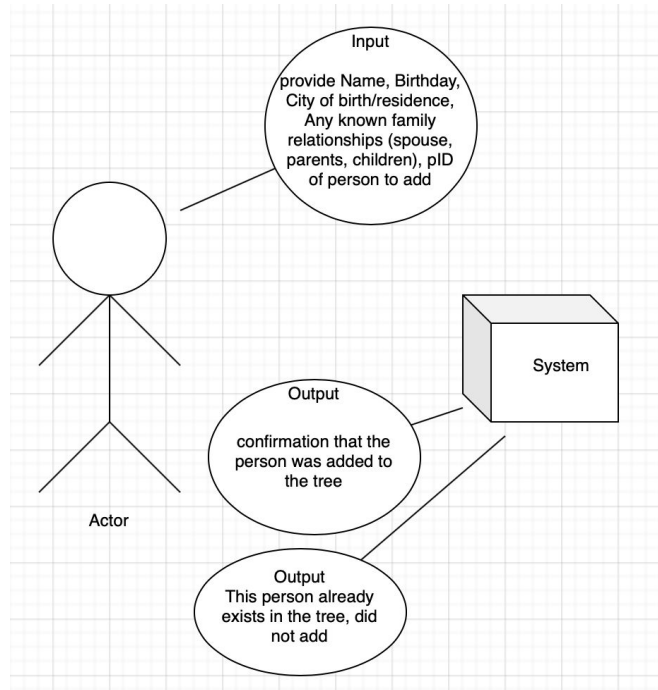
Actor: Person

Purpose: Add a person to the existing tree

Overview: The actor will input all the info they have on the person (ie. Name, Birthday, City of birth/residence, Any known family relationships (spouse, parents, children), pID). The system will check if the person is already part of the tree, if not then the system will add the new person to the tree. If already part of the tree it will return an error.

Precondition: The user must have all needed information to give the system

Postcondition: The user will have added a new person to existing tree



Use Case Narrative:

User:

1. User inputs all needed information
2. User submits the info to be added to the tree
6. The user receives the output confirming the addition of a new person

System:

3. The system will traverse the tree searching for person with given pID, to check if they already exist
4. If not found, the system will take the info provided by the user and add a new person to the tree
5. The system outputs confirmation that a new person is added

Alternative Flow:

Line 4: If the tree is traversed and a match for the new person is found, the system will return an error to the user. Return to step 1.

Event: A user can find people of a specified relationship for any person

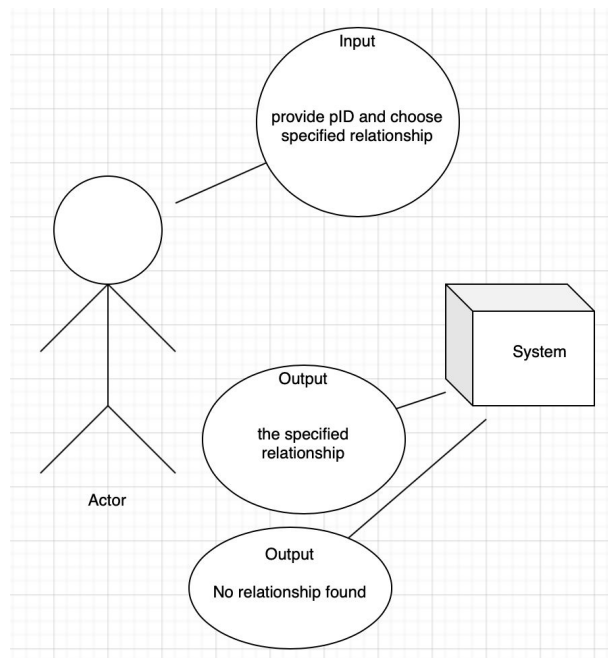
Actor: Person

Purpose: Find a specified relationship for any person in the tree

Overview: The actor will input the pID for the person and then select from a dropdown menu what person they would like to find (ie. Parents, Grandparents, Child or Partner). The system will take the pID and search for the person in the tree. If not found, return an error. If found, the system will call the method to find the specified relationship. If the specified relationship is not found, returns an error. If found, the system returns the information of the specified relationship.

Precondition: The user must have a pID to provide

Postcondition: The user will have the specified relationship they were searching for



Use Case Narrative:

User:

1. User inputs a pID and selects a relationship to search
2. User submits the info to be added to the tree
6. The user receives the info for the specified relationship

System:

3. The system will traverse the tree searching for person with given pID, to check if they exist
4. If found, the system will then use the appropriate method to find the specified relationship
5. The system outputs the info for the specified relationship

Alternative Flow:

Line 4: If the tree is traversed and no match for the relationship is found, the system will return an error to the user. Return to step 1.

Event: A user wants to determine if two existing people are related

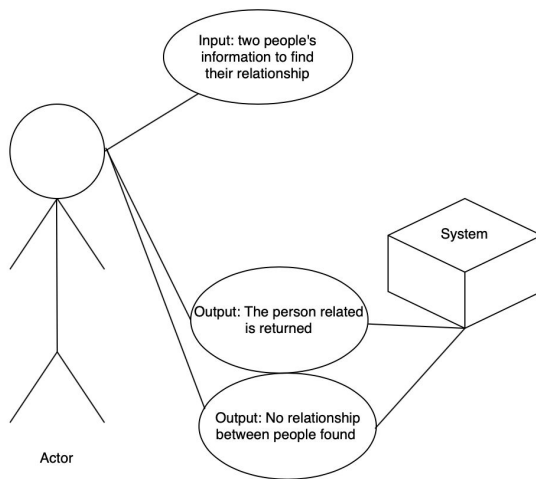
Actor: Person

Purpose: Determine if a relationship exists between two people.

Overview: The actor provides the needed info for two people and submits it. The system will traverse the tree to find both people, if one, both or none are found an error is returned. If found, the system will then traverse the nodes related to each until a relationship is found, if no relationship found an error is outputted. If a relationship is found, the system returns the relationship to the actor.

Precondition: The user must provide info on two existing people

Postcondition: The user will know whether two people are related or not.



Use Case Narrative:

User:

1. User inputs names of the two people they want to find a relationship for
2. User submits the info into the search
7. The user receives the output displaying the relationship

System:

3. The system will traverse the tree towards the root for each person
4. It will create a list for each person of related ancestors and compare each person added, checking if they are identical
5. If the system finds a person who overlaps for both, this person is outputted to the user as the member both are related to
6. The system outputs the relationship

Alternative Flow:

Line 3: If the tree is traversed and one or both people cannot be located, the system will print an error. Return to step 1.

Line 4: Once the whole tree is traversed and no relationship is found between the people the system will output an error that there is no relationship between the two people. Go to step 7

Event: A user can search for an existing relationship using an rID

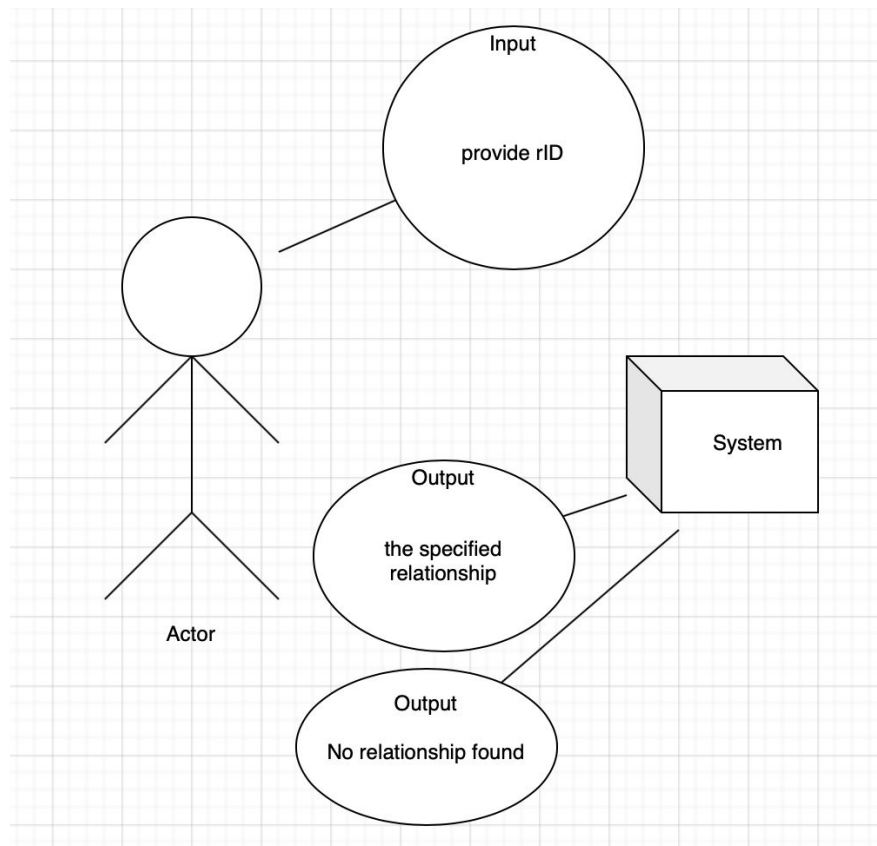
Actor: Person

Purpose: Find a specified relationship using a rID

Overview: The actor will input an rID. The system will search the tree to find the rID. If not found, an error is returned. If found, the system will output the two persons of the relationship.

Precondition: The user must have a rID to provide

Postcondition: The user will have the specified relationship



Use Case Narrative:

User:

1. User inputs a rID
2. User submits the info to be added to the tree
5. The user receives the info for the specified relationship

System:

3. The system will traverse the tree searching for the two people with that specific rID to check if they exist
4. If found, the system will output the relationship

Alternative Flow:

Line 4: If the tree is traversed and no match for the rID is found, the system will return an error to the user. Return to step 1.