Iram Jehan, Tara Le, Sophia Sklar, Ian Tompkins, Tien VoNguyen

COMP 330

Dr. Honig

7 December 2020

Use Case Model

1. Use Case Narrative

|  |  |
| --- | --- |
| Use case: | **Read in initial information from text file** |
| Primary Actor: | User |
| Purpose: | To build a genealogy tree with family members and relationships |
| Overview: | The user will input a text file into the system to be read. The system will read in this information to build a genealogy tree with all the listed members, their information, and relationships. The information is stored in a hashmap. |
| Type: | Essential |
| Preconditions: | File must contain family member names, relationships, and other required information |
| Postconditions: | Data structure created with all member information connected by relationships |
| Special requirements: | Text file must always be in the same format/order of information |

**Flow of Events**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. This use case begins when the user desires to build a genealogy tree. |  |
| 1. The user inputs a text file with family member’s first and last name, birth date and place, and possible death date and place, and relationships such as marriages and children. | 1. A Person hash map is created with all the family member’s names and their information. Another Relationship hash map is created to connect members by marriage or children. |

**Alternative Flow of Events**

Line 3: Improper formatting of text file causes error when importing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Use case: | **Add members to database** |
| Actor: | User |
| Purpose: | To add new members to the tree when new relationships are made or children are born |
| Overview: | The user will be able to add new members to the tree when a child is born or when there is a new marriage. The user will input a string of Person attributes separated by commas. The information will include the member’s person ID number, last name, first name, suffix, birthday, birth city, death day, death city, and relationship ID number. The relationship ID depends on if the added member is a new child or a new marriage. If any of these are not applicable (such as death information), the user can enter an empty space between commas. |
| Type: | Essential |
| Preconditions: | In order to add a member, the user must have already created an initial tree from importing a file |
| Postconditions: | New member will be in the tree connected by relationship to current member |
| Special requirements: | User must add relationship(s) in addition to adding member’s name and information |

**Flow of Events**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. This use case begins when the user wants to add members to the tree. |  |
| 1. The user will input all information for the new member including person ID number, last name, first name, suffix, birthday, birth city, death day, death city, and parent ID number.     4. After the person is made, the user will then be asked if the person is married. If they are, they will enter a String of comma separated Relationship attributes including the partnership ID, and the person ID of both partners. | 1. Data structure is updated with the new member’s information and relationships.   5. Data structure is updated with new marriage relationship. |

**Alternative Flow of Events**

Line 3: Added member already exists in database

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Use case: | **Search for member in tree** |
| Actor: | User |
| Purpose: | To find a member’s information in the tree |
| Overview: | The user can enter the person’s first name, last name, or their ID number to search for a person. The search will then return the member name, their birth date and city, and death date and city (if applicable). |
| Type: | Essential |
| Preconditions: | Genealogy tree must already have members and their information |
| Postconditions: | The user will see the member that they searched for and their birth and death information |
| Special requirements: | There must be a data structure already developed. Data structure must contain member that the user searches for. |

**Flow of Events**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. This use case begins when the user wants to search for a family member in the tree. |  |
| 1. The user can search for a member by inputting their first name, last name, or person ID. | 1. The search function will return the searched family member’s first and last name, their birth date and city, and possible death date and city. |

**Alternative Flow of Events**

Line 3: Searched member does not exist in tree. Please check spelling or search for another name.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Use case: | **Find parents** |
| Actor: | User |
| Purpose: | To find a person’s parents |
| Overview: | The user will type a person ID into the search. The program will then return the searched person’s name along with their parents. |
| Type: | Essential |
| Preconditions: | Genealogy tree must already have person and their relationship |
| Postconditions: | The user will see the person that they searched for and their parents |
| Special requirements: | There must be a data structure already developed. Data structure must contain the person that the user searches for. |

**Flow of Events**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. This use case begins when the user wants to search for the parents of a family member in the tree. |  |
| 1. The user can search for parents by inputting a person’s ID. | 1. The search function will return the searched family member name and their parents. |

**Alternative Flow of Events**

Line 3: Searched person does not exist in tree.

Line 3: Searched person does not have any parents listed in the tree.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Use case: | **Find children** |
| Actor: | User |
| Purpose: | To find a person’s children |
| Overview: | The user will type the relationship ID of the parents into the search to find their children. The search will then return the person’s name and any children they have. |
| Type: | Essential |
| Preconditions: | Genealogy tree must already be built with family members and their relationships |
| Postconditions: | The user will see the person that they searched for and any children that they have |
| Special requirements: | There must be a data structure already developed. Data structure must contain the person that the user searches for. |

**Flow of Events**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. This use case begins when the user wants to search for a family member’s children. |  |
| 1. The user can search for a family member’s children by inputting the relationship ID of the parents. | 1. The search function will return the names of the parents as well as their children. |

**Alternative Flow of Events**

Line 3: Searched relationship ID does not exist in the tree.

Line 3: Searched person does not have any children

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Use case: | **Find sibling relationships** |
| Actor: | User |
| Purpose: | To find a person’s siblings |
| Overview: | The user will type a person ID into the search to find their location in the tree. The search will then return the person and any siblings they have. |
| Type: | Essential |
| Preconditions: | Genealogy tree must already have the person and their relationship |
| Postconditions: | The user will see the person that they searched for and their siblings |
| Special requirements: | There must be a data structure already developed. Data structure must contain the person that the user searches for. |

**Flow of Events**

|  |  |
| --- | --- |
| **Actor Action** | **System Response** |
| 1. This use case begins when the user wants to search for siblings within the tree. |  |
| 1. The user can search for a sibling relationship by inputting a person ID. | 1. The search function will return the searched family member name and any members that are found to be their siblings. |

**Alternative Flow of Events**

Line 3: Searched person does not exist in tree.

Line 3: Family member does not have any known siblings

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2. Use Case Diagram

