DESIGN DOCUMENT

for

Family Tree Generator

Yunus Tufan Bozkurt - 20140602003

Umut Kanpalta - 20150602036

Berke Parıldar - 20190602031

Ozan Yücel – 20190602043

Table of Contents

1. Introduction

1.1 Purpose

This document presents a detailed description of the project's design, giving the functional structure, algorithms, and program functions. This document could be considered a guideline for any developer who may wish to develop this project or modify it by any way.

1.2 Scope

This document details the architecture and modules that were discussed in the software requirement document. Some diagrams are available to visualize the architecture.

2. Program Functions and Design

2.1 Language and External Tools

Java will be programming language for this project. Swing will be used as GUI library.

2.2 Program Capabilities

User can edit family trees, delete family trees, and export or import a family tree via using File I/O. Program also should be able to merge specific two trees (if there is a node in common), find relation between two person, and search family trees or family members.

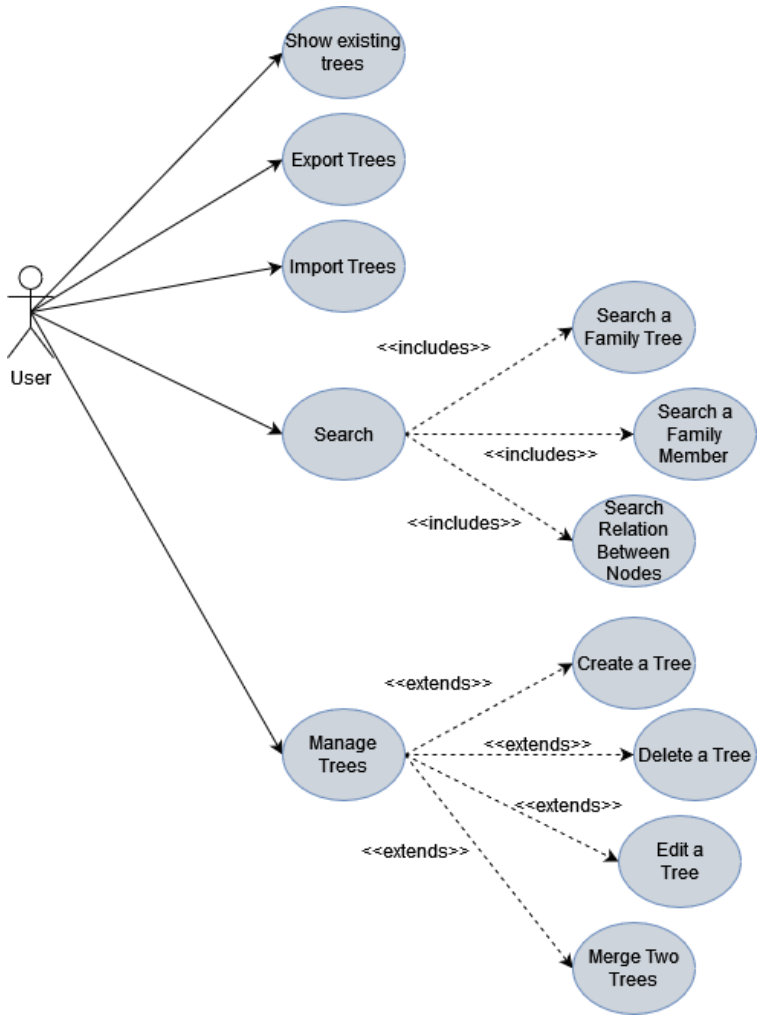
2.3 Component Decomposition

The application will be consisting of five classes. These classes are Person, Tree, Relation, TreeModification, and Main classes. The person class will be used to set the information about a person, such as name, and will contain the parent and children status, which later will be stored in the tree class. The tree class will have information about the trees, for there will be multiple trees available in the program, ready for user access depending on preference. It will also store the user input for the person. Relation class will be the one setting the relation status between people when the conditions are met.

The main class is for the execution of our software. For the creation of a new tree, user input for the person's information will be asked. This information will be categorized by the person class, and later will be stored in the tree class. The relation class will have functions for the relation status between relatives and will set the relation status when asked. The TreeModification class will be called by UI elements such as add, delete, or edit. This class will also contain a method to merge two existing trees.

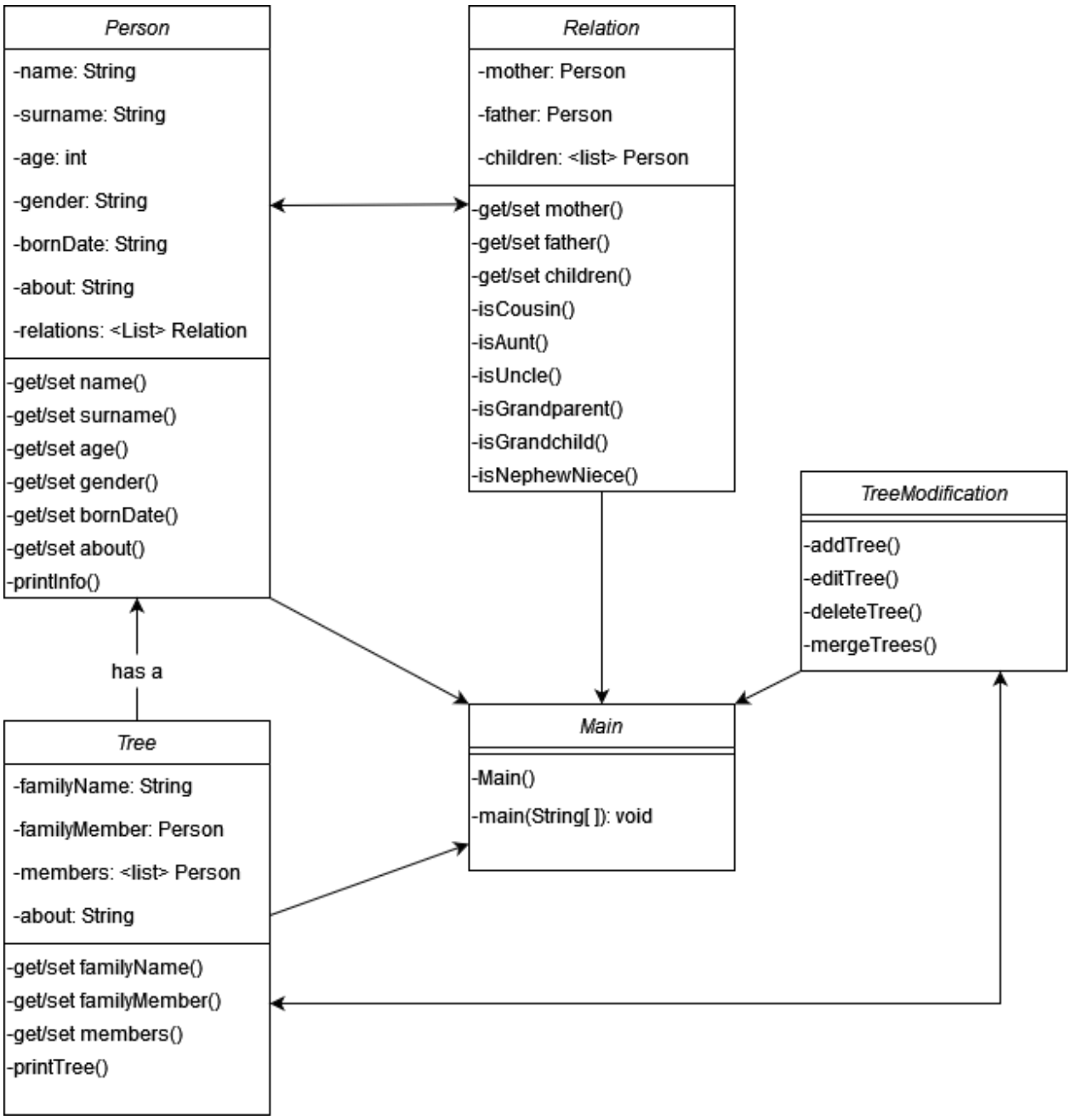
3. Visual Explanations

3.1 Use Case Diagram



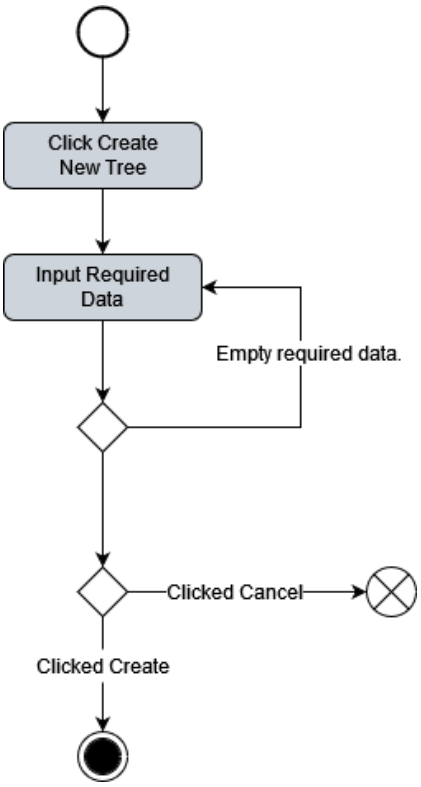
Our system provides five services to the users. These services are as follows: show, import, export, search and manage the trees.

3.2 Class Diagram



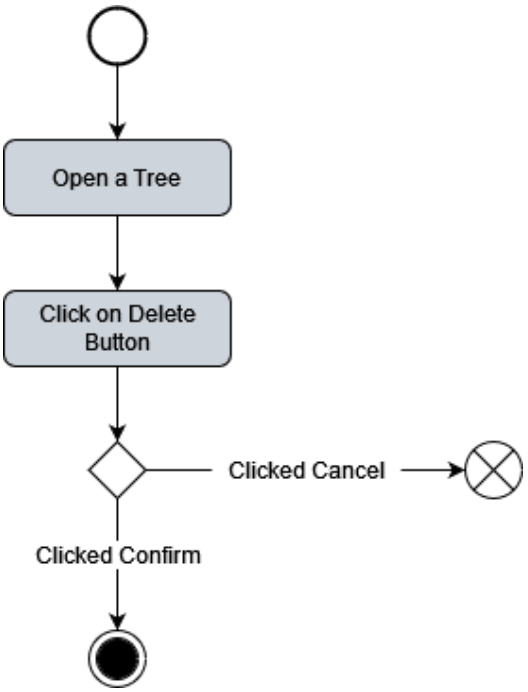
3.3 Activity Diagrams

3.3.1 Create Family Tree Activity Diagram



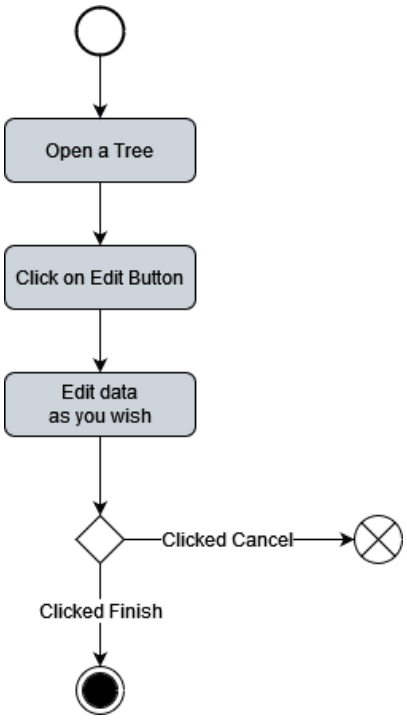
Users can create a family tree from scratch and fill it with the required information.

3.3.2 Delete Family Tree Activity Diagram



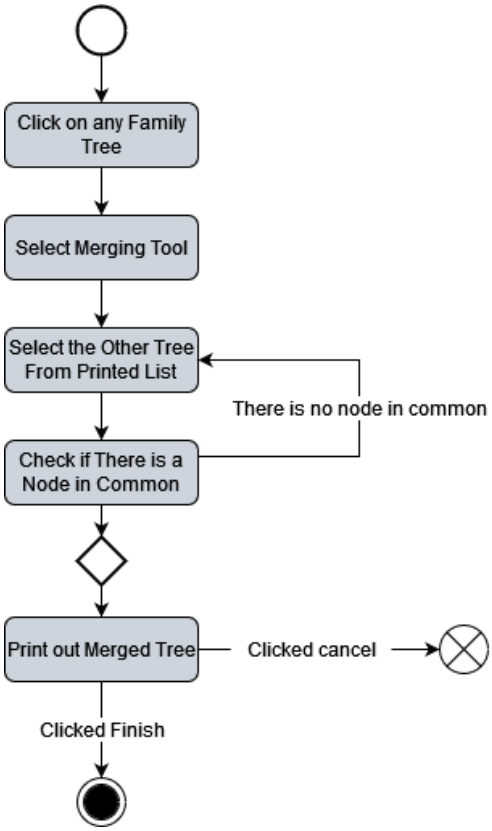
Users can delete an existing family tree.

3.3.3 Edit Family Tree Activity Diagram



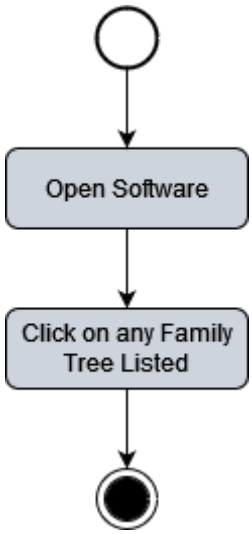
Users can make changes to the family tree and the data it contains.

3.3.4 Merge Family Trees Activity Diagram



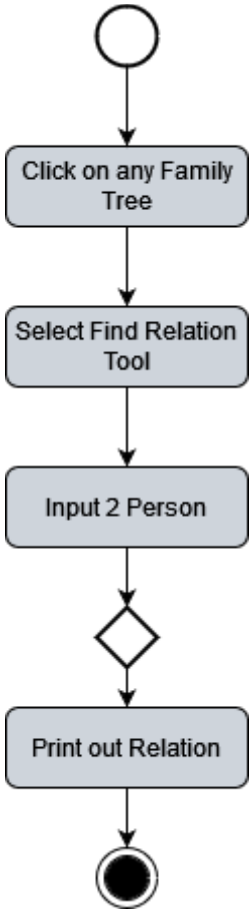
Users can merge two existing family trees if the necessary conditions are met.

3.3.5 Open Family Tree Activity Diagram



Users can work with multiple trees. Users can open any family tree that software lists.

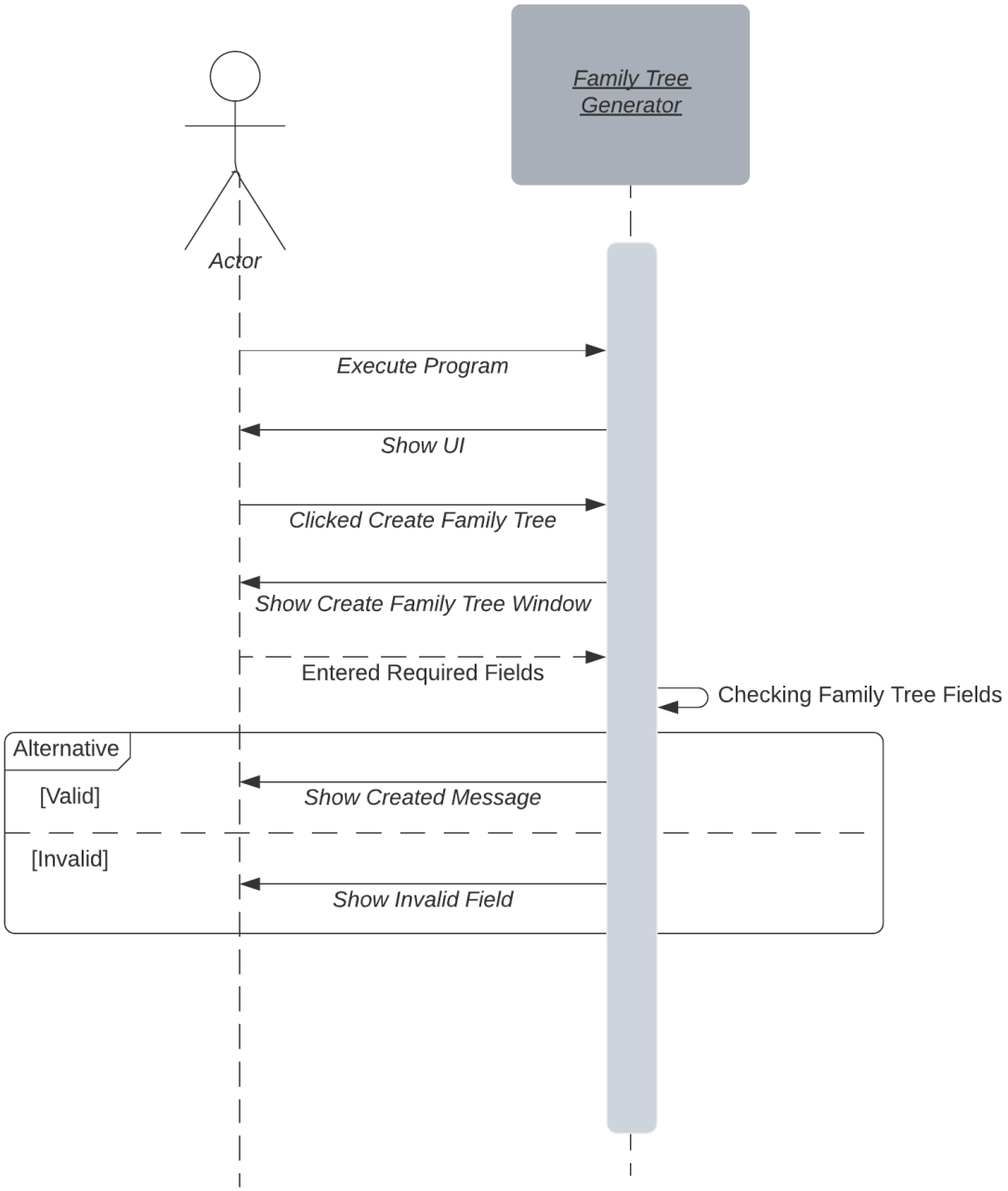
3.3.6 Relation Finding Activity Diagram



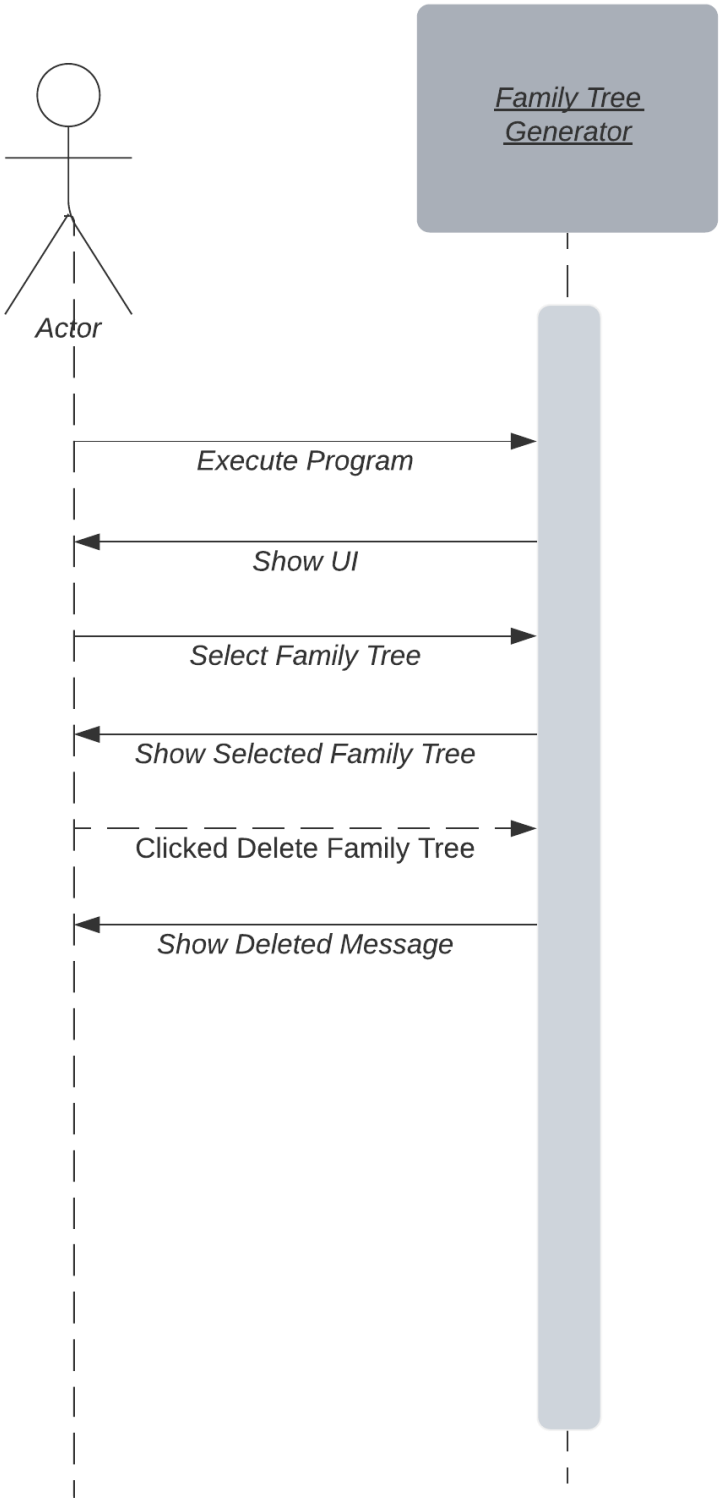
Users can ask for the relationship status of two selected family members.

3.4 Sequence Diagrams

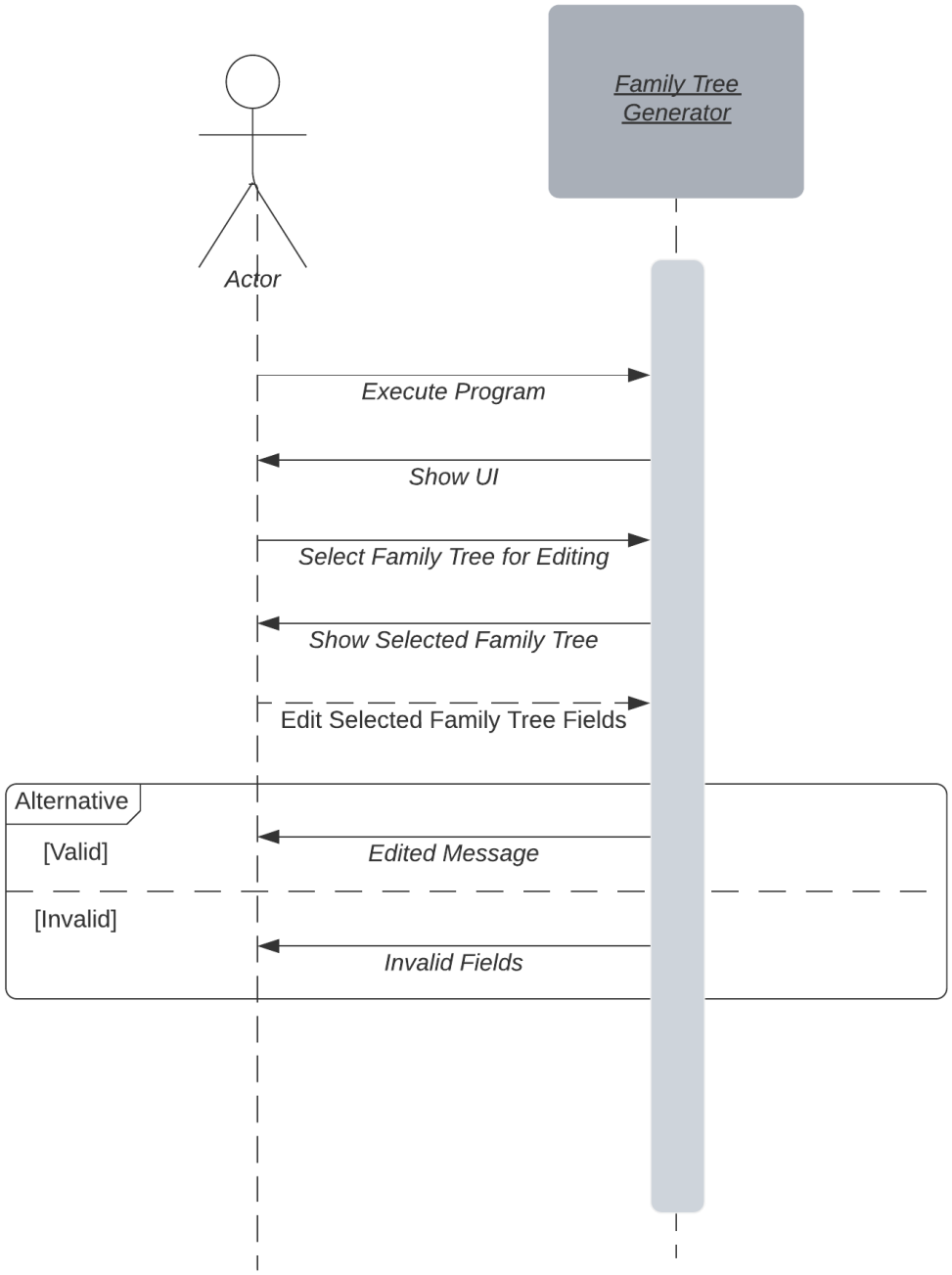
3.4.1 Create Family Tree Sequence Diagram



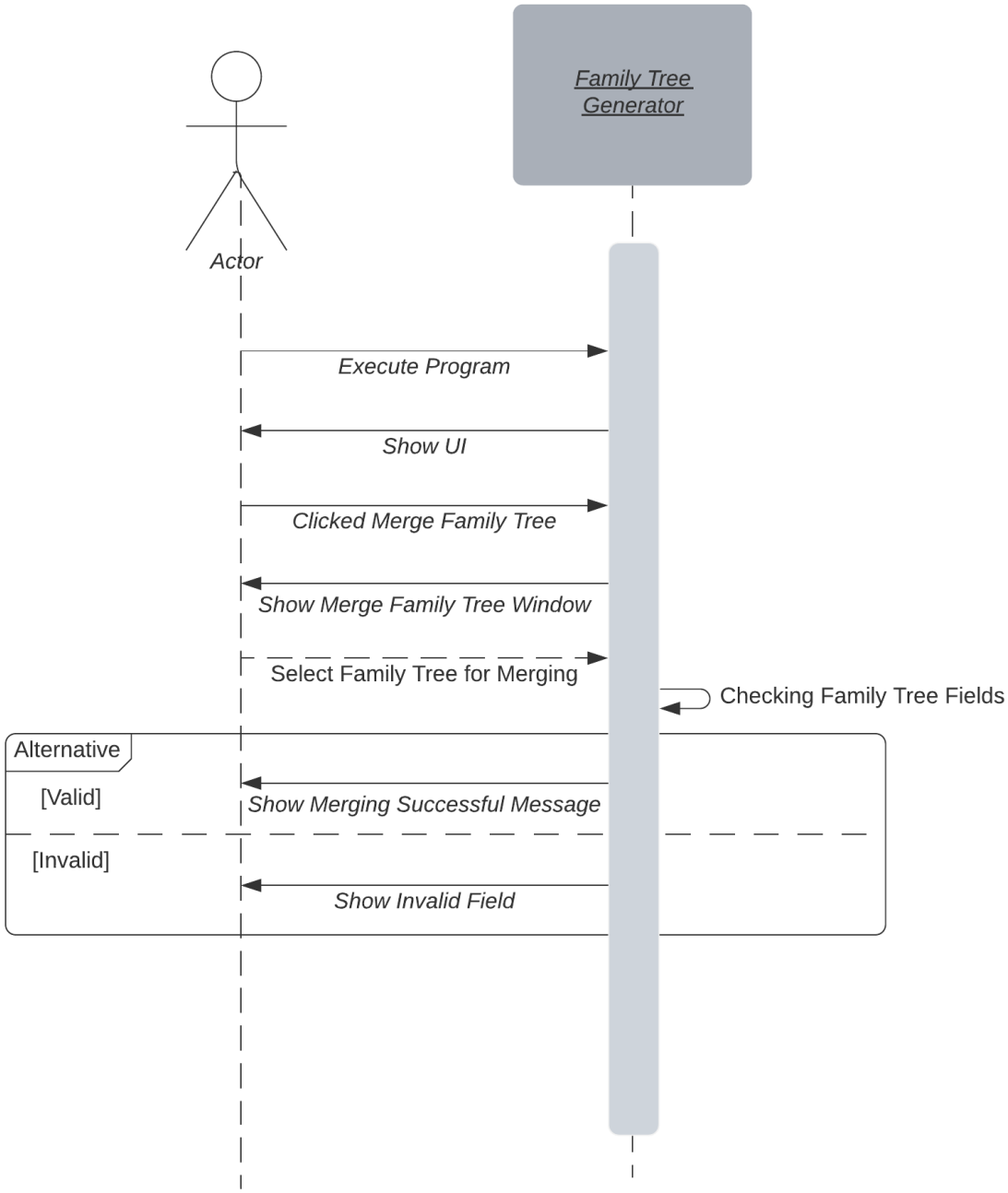
3.4.2 Delete Family Tree Sequence Diagram



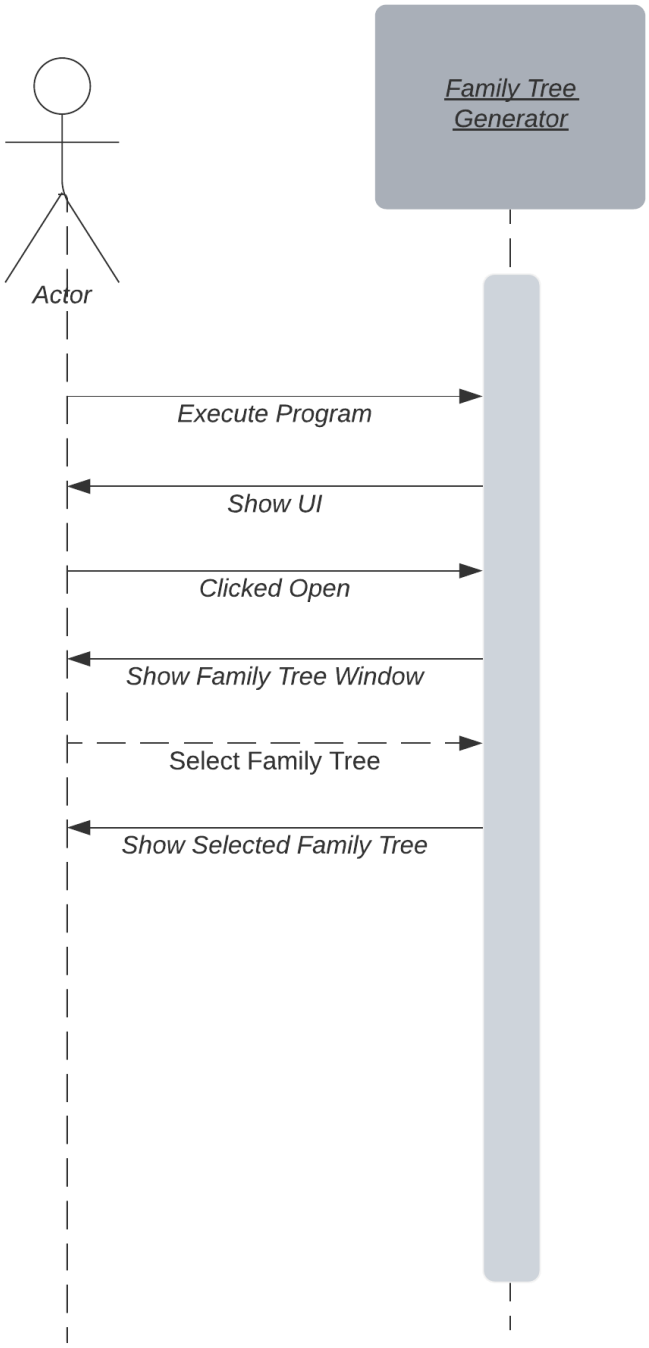
3.4.3 Edit Family Tree Sequence Diagram



3.4.4 Merge Family Tree Sequence Diagram



3.4.5 Open Family Tree Sequence Diagram



3.4.6 Relation Family Tree Sequence Diagram