**SE 302**

Principles of Software Engineering

**Software**

**Design**

**Document**

for

**Family Tree Builder**

**Team 8 - Project Members**

Ozan ŞAHİN (20190602036)

Ege ALTIOK (20190602002)

Barış ÖZDİL (20170602023)

Efe KAYLAK (20180602019)

**Course Lecturer and Client**

Asst. Prof. Kaya OĞUZ

**Table of Contents**

**I. PREFACE3**

I.I. Document Version3

I.II. Change History3

**II. INTRODUCTION 4**

II.I. Purpose of the Document4

II.II. Scope of the Document4

**III. ATTRIBUTES OF DESIGN ENTITIES5**

III.I. TreeOperations5

III.II. Member5

III.III. Birthday6

**IV. INTERFACE DESCRIPTION7**

**V. DECOMPOSITION DESCRIPTION8**

V.I. Activity Diagram8

V.II. Class Diagram8

V.III. Use Case Diagram9

1. **PREFACE**
   1. **Document Version**

* Document Version: 1.1
  1. **Change History**
* **v1.1**
  + Fixed document reference issues
* **v1.0**
  + Initial version

1. **INTRODUCTION**
   1. **Purpose of the Document**

* This Software Design Document briefly explains the structure of the components of the Family Tree Project.
* This document will provide a design which will satisfy functional and non-functional requirements that are stated in the Software Requirements Document.
  1. **Scope of the Document**

This document shall contain the general definition and features of the project, behaviour of the system, user’s interaction with the system.

1. **ATTRIBUTES OF DESIGN ENTITIES**
   1. **TreeOperations**

**Variables**

• **tree\_content** stores the tree content of the tree

• **active\_tree** is a GUI component that resembles a tree-like view

**Methods**

• **create\_tree()** gets input from the user and creates a new tree with the input name.

• **add\_member()** gets input from user and creates new member with that information.

• **check\_relation()** requests user to select two members then checks what relation they have.

• **import\_tree()** imports raw tree data and constructs the tree.

• **export\_tree()** exports raw tree data to a user selected destination.

• **merge\_trees()** requests user to select a new tree to merge with his/her currently selected family tree.

• **save\_as\_image()** creates an image file of the tree at the user selected destination.

• **update\_info\_tab()** updates the selected member tab with the information of the member object that user has clicked on.

• **add\_filter()** gets input from the user and adds that input as a new filter in the filter list.

• **remove\_filter()** removes the selected filter from the filter list.

* 1. **Member**

**Variables**

* **name** is the name of the family member
* **surname** is the surname of the family member
* **age** is the age of the family member
* **birthday** is the Birthday object of the family member
* **member\_layer\_level** is the member’s vertical position relative to its root member node (0 -> grandparent, 1 -> parent, 2 -> child)

**Methods**

• **check\_level()** checks and returns the value of member\_layer\_level attribute.

* 1. **Birthday**

**Variables**

* **day** is the day of the birthday
* **month** is the month of the birthday
* **year** is the year of the birthday

**Methods**

* **date\_to\_string()** converts the attributes of the Birthday object to string format.

1. **INTERFACE DESCRIPTION**



* **Tree Operations Field** allows user to use the operations in the TreeOperation class.
* **Filters Field** allows users to find the people that they are looking for by adding and removing filters.
* **Family Tree Contents** is the field where the family tree contents are displayed. Users can select a person from the family tree.
* **Selected Person Info** allows users to see the selected person's information in this field.

1. **DECOMPOSITION DESCRIPTION**
   1. Diagram

      Description automatically generated**Activity Diagram**
   2. Diagram

      Description automatically generated**Class Diagram**
   3. **Use Case Diagram**

Chart, diagram

Description automatically generated