SWDD- [obtain number from Conduct of Engineering Document Numbering [SharePoint site](https://coe.lanl.gov/APs/DocNum/SitePages/Home.aspx)]

PyFamilyTree group

**PyFamilyTree**

#### Software Design Document

**Name (s):**

- Artie Dana

- Muyiwa Oguntimilehin

- Ilami Erasmus

- Daniel Schmidt

Date:

**TABLE OF CONTENTS**

***1.0*** ***INTRODUCTION 3***

***1.1*** ***Purpose 3***

***1.2*** ***Scope 3***

***1.3*** ***Overview 3***

***1.4*** ***Reference Material 3***

***2.0*** ***SYSTEM OVERVIEW 3***

***3.0*** ***SYSTEM ARCHITECTURE 3***

***3.1*** ***Architectural Design 3***

***3.2*** ***Decomposition Description 4***

***3.3*** ***Design Rationale 4***

***4.0*** ***DATA DESIGN 4***

***4.1*** ***Data Description 4***

***4.2*** ***Database Schema 5***

***5.0*** ***COMPONENT DESIGN 6***

***6.0*** ***HUMAN INTERFACE DESIGN 6***

***6.1*** ***Overview of User Interface 6***

***6.2*** ***Screen Images 6-11***

***6.3*** ***Screen Objects and Actions 12***

***7.0*** ***REQUIREMENTS MATRIX 12-15***

***8.0*** ***USER MANUAL 15-20***

***9.0*** ***APPENDICES 21***

***9.1*** ***Use Cases 21-24***

***9.2*** ***User Stories 24-30***

***9.3*** ***Sequence Diagrams 30-32***

***9.4*** ***Activity Diagrams 32-38***

## INTRODUCTION

## Purpose

## Scope

The PyFamilyTree app is a family tree application that allows users to create and maintain family trees, add members, and keep track of genealogical information. The goals of this project are to provide a user-friendly and secure application that can handle high traffic and large amounts of data. The objectives of the project are to develop an app that is scalable, maintainable, and easily updatable.

## Overview

This document provides an overview of the PyFamilyTree app and its organization. It outlines the system architecture, data design, component design, and human interface design. The document also includes a requirements matrix to trace system components and data structures ot the requirements outlined in the Software Requirements Specification (SRS document)

## Reference Material

The reference material used for this document includes the SRS document, software development best practices, and documentation provided by the associated third-party software libraries.

## SYSTEM OVERVIEW

The PyFamilyTree app is a web-based family tree application that allows users to create and maintain family trees, add members, and keep track of genealogical information. Users can view their family tree, edit family member information, and add new family members. The app is designed to be user-friendly, scalable, and secure.

## SYSTEM ARCHITECTURE

## Architectural Design

The PyFamilyTree app is designed using a 3-tier architecture consisting of a presentation layer, application layer, and database layer. The presentation layer is responsible for the user interface and handles all user input and output. The application layer contains the business logic of the application and handles data processing. The database layer is responsible for storing and retrieving data.

## Decomposition Description

The PyFamilyTree app is decomposed into three main subsystems: the user interface subsystem, the business logic subsystem, and the data storage subsystem. The user interface subsystem handles all user input and output and communicates with the business logic subsystem to process data. The business logic subsystem contains all the application’s business logic and communicates with the data storage subsystem to store and retrieve data.

## Design Rationale

The 3-tier architecture was chosen to separate the application into distinct layers to increase modularity and scalability. This architecture also enables the use of caching and load balancing to handle high traffic and large amounts of data. The usage of a database layer allows for easy storage and retrieval, and the use of the business logic layer enables easy maintenance and updates.

## IMPROVEMENT

## Enhancement

## 5.0 USER MANUAL

## 6.0 TEST CASES

## 6.0.1 ID and Name

## 6.0.2 Related use case numbers(s), sequence diagram name(s)

## 6.0.3 Description of test cases

##### 

## 6.0.4 Input data description

## 6.0.5 Pre- and post- condition

## 

## 6.0.6 Testing procedure

## 6.0.7 Expected result vs. actual result

### 6.1 TEST DATA

### 6.2 TRACEABILITY ANALYSIS

### 

### 6.0.10 DEPLOYMENT PROCEDURE AND CORRESPONDING DIAGRAM

**6.11 PRIORITIZED USE CASES**

**6.12 FUNCTIONALITY IN SRS**

##### 

##### 

##### 

##### 