ABSTRACT

Creating an age calculator using Tkinter involves designing a graphical user interface (GUI) where users can input their birthdate, triggering a calculation to determine their current age. Tkinter, Python's standard GUI library, facilitates this by allowing us to build a window (`Tk()`) and populate it with various widgets like labels, entry fields, and buttons. The main components include a label prompting users to enter their birthdate, an entry widget for them to input the date, and a button to initiate the calculation. Upon clicking the button, a function reads the birthdate input, computes the age based on the current date using Python's `datetime` module, and updates a label with the calculated age. Proper layout management using Tkinter's grid or pack system ensures these elements are arranged logically within the window. Finally, running `mainloop()` starts the Tkinter event loop, allowing the GUI to be displayed and interacted with by the user, effectively creating a simple yet functional age calculator application.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

An age calculator built using Tkinter leverages Python's GUI capabilities to create an interactive tool for determining a person's age based on their birthdate. Tkinter provides a straightforward way to design a user-friendly interface with widgets like labels, entry fields, and buttons. The calculator prompts users to input their birthdate, typically through a text entry field. Upon submission, a calculation function processes this input, computing the age by comparing the birthdate against the current date obtained from Python's datetime module. The resulting age is then displayed prominently on the interface, usually beneath the input fields or as a popup message. Tkinter's grid or pack layout managers help organize these components within the application window, ensuring clarity and ease of use. This project not only demonstrates the integration of user input and computational logic but also showcases the versatility of Tkinter in developing practical applications that engage users through intuitive graphical interfaces.

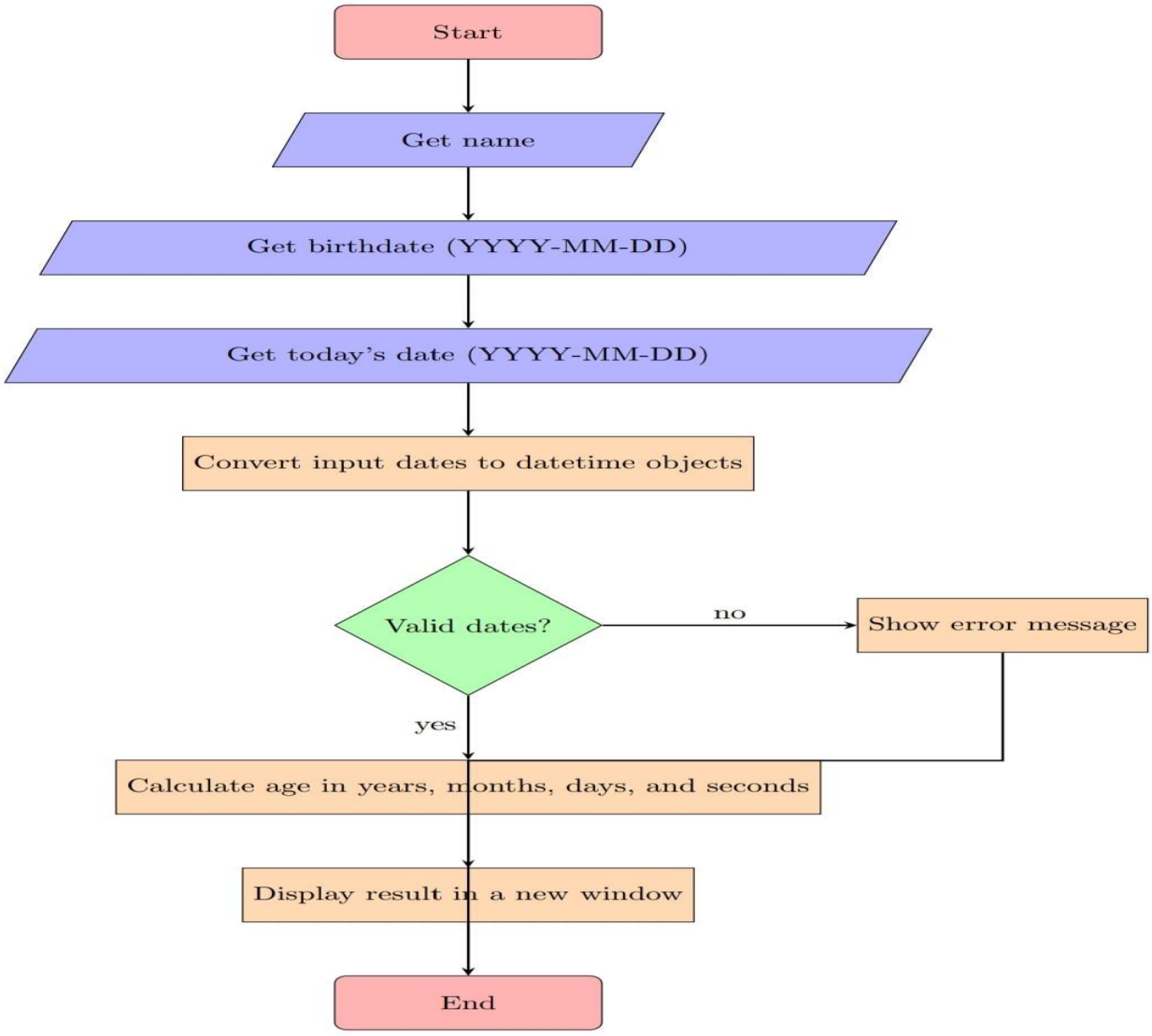
CHAPTER -2

DESIGN

The GUI application is designed to calculate a person's age based on their birthdate and today's date. The main window (`root`) is titled "Age Calculator" with a sky blue background (`#87CEEB`). It includes several tkinter widgets arranged vertically: a `Label` prompting for the user's name, followed by an `Entry` widget for name input; a `Label` for entering the birthdate in the format "YYYY-MM-DD" with an associated `Entry` widget for user input; and another `Label` and `Entry` pair for today's date input. Beneath these inputs, a `Button` labeled "Calculate Age" triggers the `calculate_age()` function upon clicking. This function retrieves inputs, parses them into datetime objects, calculates the age in years, months, days, and seconds, and displays the result in a new `Toplevel` window (`result_window`). If the date format is incorrect, an error message is shown using `messagebox.showerror()`. The result window displays the calculated age details formatted neatly with labels for each unit of time (years, months, days, seconds), ensuring clarity and user-friendly presentation.

CHAPTER 3

FLOWCHART



CHAPTER 4
Module Description

Module Name: AgeCalculatorTkinter

Description: The AgeCalculatorTkinter module is a graphical user interface (GUI) application designed to calculate a person's age based on their birthdate and today's date. It leverages the tkinter library for creating a user-friendly interface with interactive elements.Users are prompted to input their name, birthdate (in the format YYYY-MM-DD), and today's date through respective text entry fields. A "Calculate Age" button triggers the calculation process, where the entered dates are parsed into datetime objects for accurate age computation. The calculated age in years, months, days, and seconds is then displayed in a separate result window (Toplevel) formatted for clarity and readability. The module emphasizes intuitive design, precise date handling, and effective communication of age calculation results through a visually appealing GUI.

Key Features:

- **Graphical User Interface (GUI)**: Utilizes tkinter to create windows, labels, entries, buttons, and message boxes.
- **Date Parsing and Calculation**: Converts user-input dates into datetime objects for accurate age computation using Python's datetime module.
- **Age Presentation**: Displays calculated age details (years, months, days, seconds) in a dedicated result window with structured formatting.
- **Error Management**: Notifies users of invalid date formats through error messages, ensuring data integrity and user guidance.

Usage:

1. Launch the module within a Python environment.
2. Enter your name, birthdate, and today's date in the specified format (YYYY-MM-DD).
3. Click "Calculate Age" to initiate the age calculation process.
4. View the calculated age details presented in a separate window designed for clarity and ease of understanding.

Dependencies:

- Python 3.x
- tkinter library (standard with Python)
- datetime module (standard with Python)

CHAPTER 5

CODE

PYTHON PROGRAM

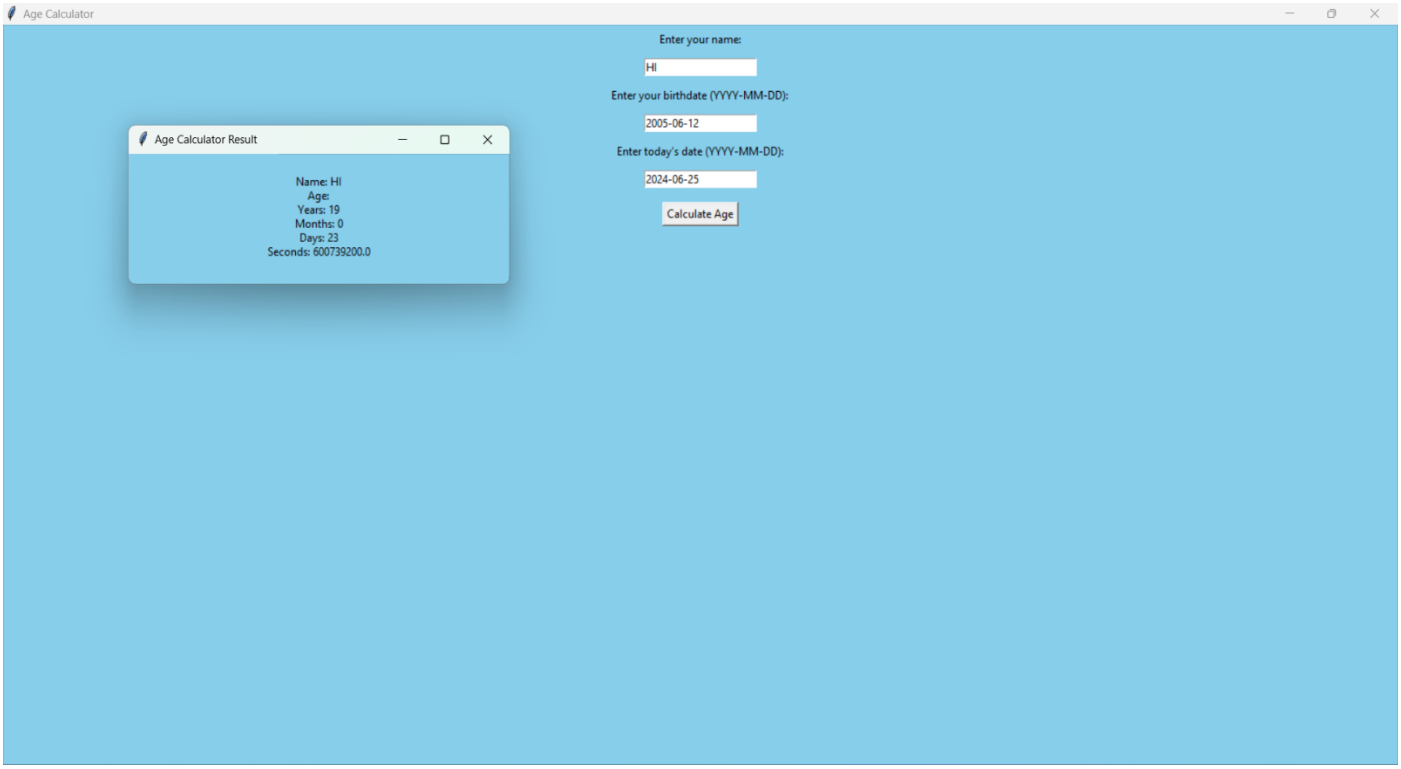
```
import tkinter as tk
from tkinter import messagebox
from datetime import datetime
def calculate_age():
    name = name_entry.get()
    birth_date_str = birth_date_entry.get()
    today_date_str = today_date_entry.get()
    try:
        birth_date = datetime.strptime(birth_date_str, '%Y-%m-%d')
        today_date = datetime.strptime(today_date_str, '%Y-%m-%d')
        age = today_date - birth_date
        age_years = age.days // 365
        age_months = (age.days % 365) // 30
        age_days = age.days % 30
        age_seconds = age.total_seconds()
        result_text = (
            f"Name: {name}\n"
            f"Age:\n"
```

```
 f"Years: {age_years}\n"
            f"Months: {age_months}\n"
            f"Days: {age_days}\n"
            f"Seconds: {age_seconds}"
        )
        result_window = tk.Toplevel(root)
        result_window.title("Age Calculator Result")
        # Set sky blue background color
        result_window.configure(bg='#87CEEB')
        result_label = tk.Label(result_window, text=result_text, padx=20, pady=20,
bg='#87CEEB')
        result_label.pack()
    except ValueError:
        messagebox.showerror("Error", "Please enter valid dates in the format YYYY-MM-DD")
# Create main window
root = tk.Tk()
root.title("Age Calculator")
# Set sky blue background color
root.configure(bg='#87CEEB')
# Create and place widgets
name_label = tk.Label(root, text="Enter your name:", bg='#87CEEB')
```

```
name_label.pack(pady=5)

name_entry = tk.Entry(root, width=20)

name_entry.pack(pady=5)

birth_date_label = tk.Label(root, text="Enter your birthdate (YYYY-MM-DD):",
bg='#87CEEB')

birth_date_label.pack(pady=5)

birth_date_entry = tk.Entry(root, width=20)

birth_date_entry.pack(pady=5)

today_date_label = tk.Label(root, text="Enter today's date (YYYY-MM-DD):", bg='#87CEEB')

today_date_label.pack(pady=5)

today_date_entry = tk.Entry(root, width=20)

today_date_entry.pack(pady=5)

calculate_button = tk.Button(root, text="Calculate Age", command=calculate_age)

calculate_button.pack(pady=10)

root.mainloop()
```

CHAPTER 6

IMPLEMENTATION



 Imports and Function Definition:

- tkinter, messagebox: These are imported for GUI creation and displaying error messages.
- datetime: Used for handling dates and calculating age.
- calculate_age(): This function retrieves inputs (name, birthdate, today's date), calculates the age, and displays the result.

 calculate_age() Function:

- It retrieves the values from name_entry, birth_date_entry, and today_date_entry.
- Attempts to convert the date strings (birth_date_str and today_date_str) into datetime objects.
- Calculates the age by subtracting birth_date from today_date.
- Converts the age into years, months, days, and seconds.
- Constructs a string (result_text) displaying the name and calculated age details.
- Creates a new Toplevel window (result_window) to display the result in a formatted manner.

 Main Application (root) Setup:

- Creates the main window (root) for the application titled "Age Calculator".
- Sets a light blue background (#87CEEB) for aesthetic purposes.

 Widgets:

- Labels (name_label, birth_date_label, today_date_label): These prompt users to enter their name, birthdate, and today's date.
- Entry widgets (name_entry, birth_date_entry, today_date_entry): These allow users to input their name and dates.
- Button (calculate_button): When clicked, it triggers the calculate_age() function to calculate and display the age.

CHAPTER 7

CONCULSION

The age calculator program developed using Tkinter exemplifies a practical application of Python's GUI capabilities, specifically tailored to calculate and display a person's age based on user-provided birthdate and today's date inputs. This program is structured around facilitating user interaction through intuitive graphical elements: entry fields for inputting data, a button to trigger the age calculation, and a separate window to neatly present the computed results.

Functionally, the program begins by prompting users to enter their name, birthdate, and today's date in the format `YYYY-MM-DD`. Error handling mechanisms ensure that the dates are correctly formatted, providing clear feedback in case of input discrepancies. Upon clicking the "Calculate Age" button, the program utilizes Python's `datetime` module to compute the difference between the birthdate and current date, accurately determining the user's age in years, months, days, and seconds.

CHAPTER 8

REFERENCE

1. Source code
2. https://www.w3schools.com/
3. https://www.youtube.com/