Elderly Wearable Fitness Tracker

Embedded Software Engineering

Group_Assignment_Report

ENSE810

Maneesh Maleedu

18033725

1. Use Cases of the System:

-Tracking and Displaying Step Count Data: Users can use the application to monitor their step count, which may be useful for fitness goals or health tracking.

- Start and Stop Step Count Tracking: The system allows users to initiate and terminate step count tracking, providing control over data collection.

- Displaying Data on Hourly, Daily, and Weekly Step Counts: Users can view their step counts aggregated by different time intervals, allowing for trend analysis.

- Providing Navigation Between Different Data Display Tabs: The application offers a user-friendly interface with tabs to switch between hourly, daily, and weekly step count views.

- Updating Step Count Data in Real-Time: Real-time updates ensure that users see the most current step count data, enhancing the user experience.

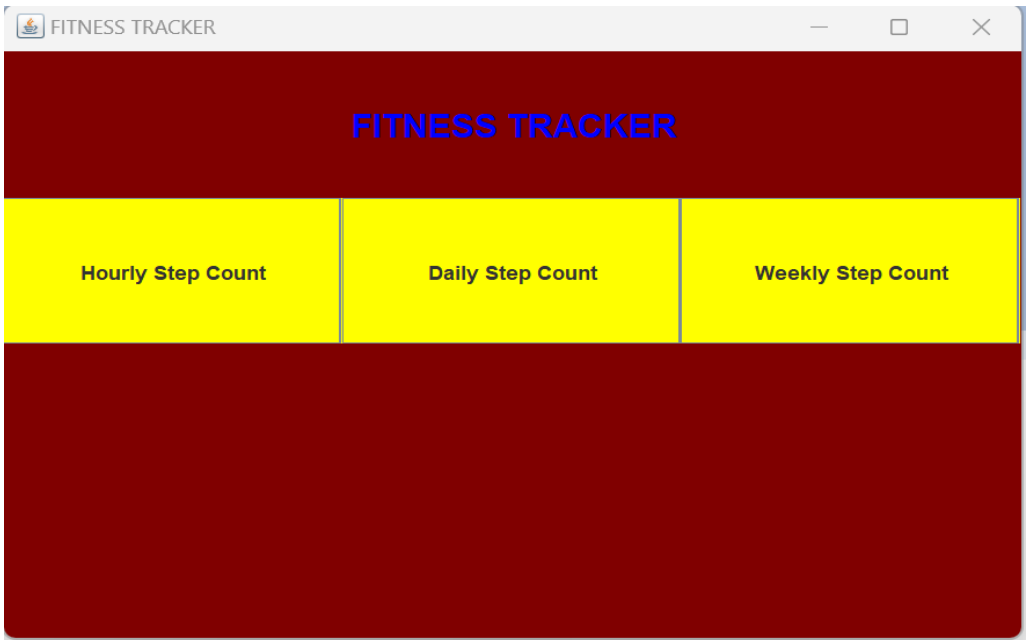
- Switching Between Home Page and Data Display Pages: Users have the flexibility to navigate between the home page and detailed step count data views, providing a seamless user journey.

2. Functional Requirements:

- **Step Count Tracking:** The system's core functionality is to accurately track and display step count data, which serves as the primary purpose of the application.

- **Data Display:** Users can access and visualize their step count data over different time frames, meeting the requirement for data presentation.

- **Navigation:** The application allows users to switch between different data display tabs, ensuring a user-friendly and intuitive interface.

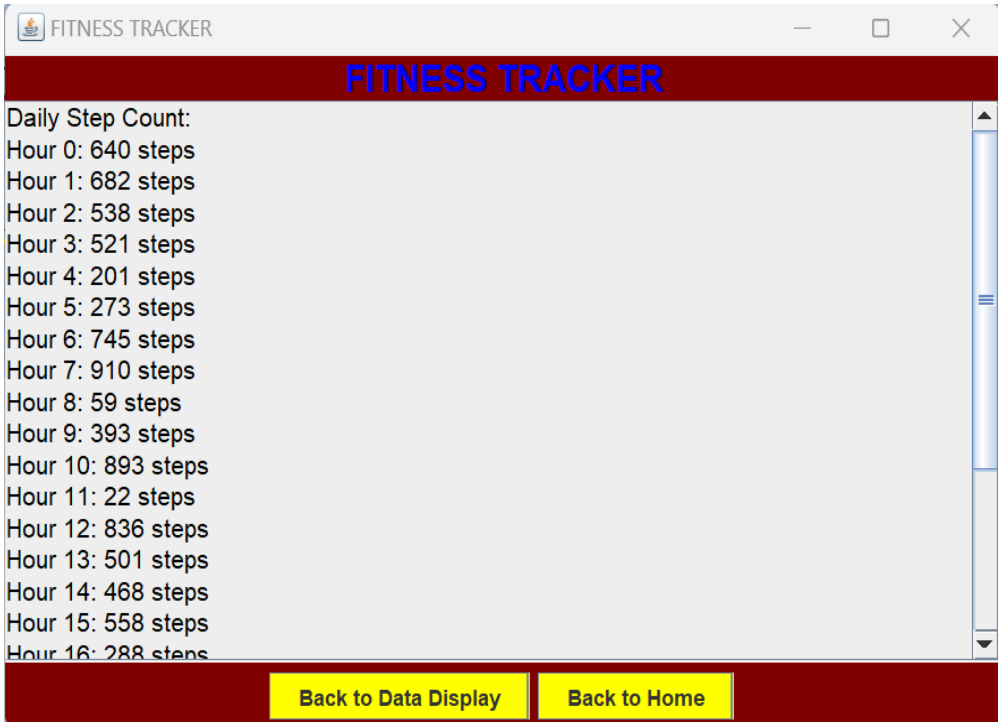


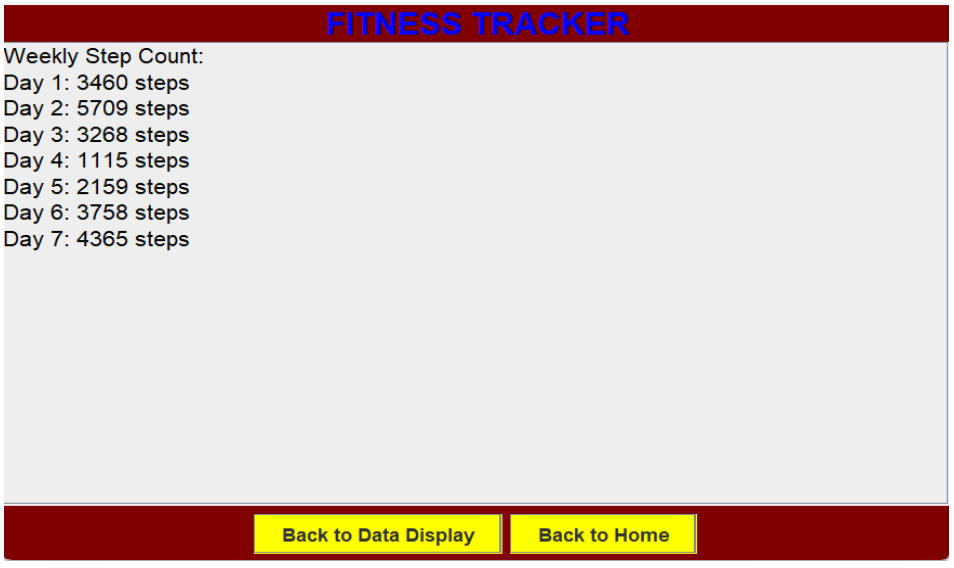
3. Non-Functional Requirements:

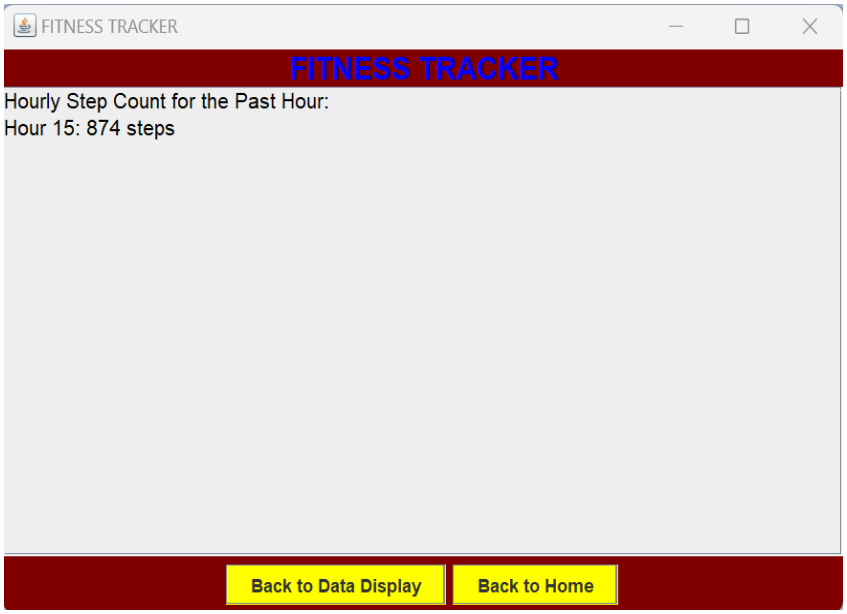
- **Real-Time Updates:** The system ensures that step count data is updated in real-time, guaranteeing that users always have access to the most recent information.

- **Usability:** The graphical user interface is designed to be user-friendly, with intuitive layouts, clear labels, and easy navigation.

- **Performance:** The application is designed to be responsive and efficient, providing a smooth user experience even when displaying large datasets. The pictures below represent the data display functionality of the fitness tracker.







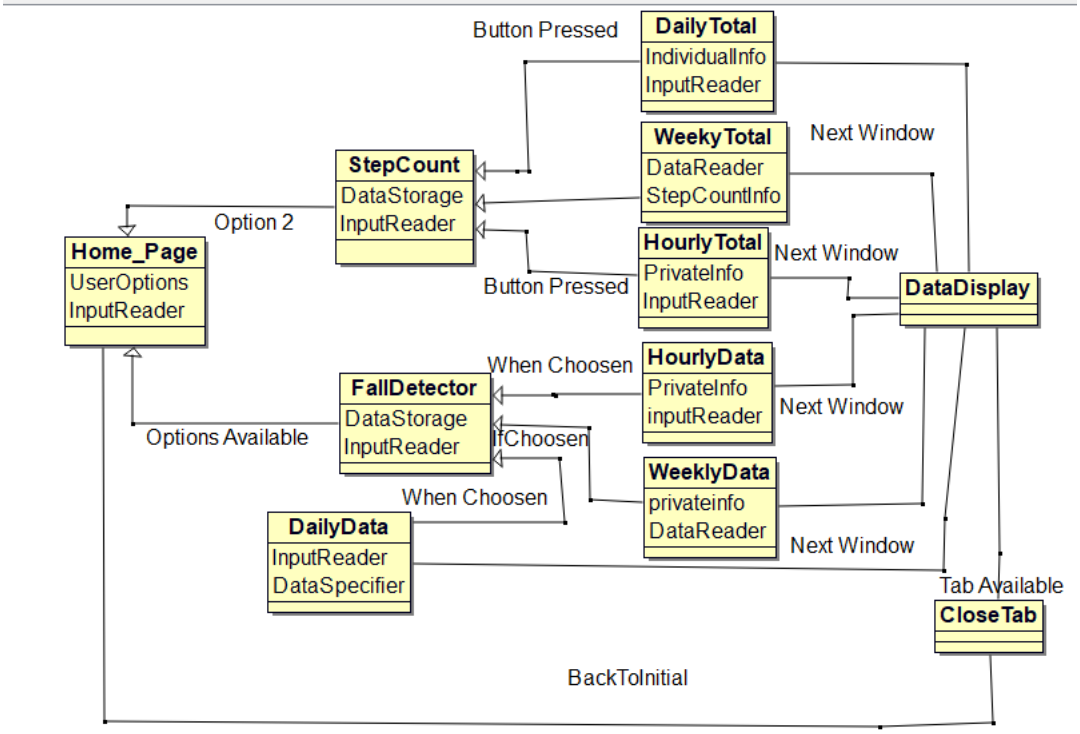
4. Architectural Design:

- **GUI Components:** The graphical user interface components are responsible for presenting information to the user. These components include elements like buttons, labels, and panels, all managed by the Tkinter library in Python.

- **Data Access Component:** This component interacts with the database, retrieving step count data and enabling its use in the application. In your case, an SQLite database is used with SQL queries to access and manipulate data.

- **Real-Time Update Component:** The real-time update component periodically refreshes the displayed step count data, enhancing the user experience with timely information.

- **Control Logic:** The control logic acts as the intermediary between the GUI components, data access component, and real-time update component, ensuring seamless communication and coordination among these parts.



5. Approach in Implementing the Coding:

- The Tkinter library, a widely used Python GUI framework, is employed to create the graphical user interface, simplifying the development of the application's front end.

- An SQLite database is selected for its ease of use and integration with Python, facilitating data storage and retrieval.

- Real-time updates are achieved using the `after` method in Tkinter, allowing for periodic data refresh.

- The code is structured into classes and methods, promoting modularity and maintainability. Some code snippets are provided below for reference.

```python
import tkinter as tk
import random

class FitnessTrackerGUI:
    def __init__(self, root):
        self.root = root
        self.root.title("FITNESS TRACKER")
        self.root.geometry("600x400")
        self.root.configure(bg="#800000")

        self.selected_tab = ""
        self.create_home_page()

    def create_home_page(self):
        self.home_panel = tk.Frame(self.root, bg="#800000")
        self.home_panel.grid(row=0, column=0, padx=10, pady=10, sticky="nsew")

        title_label = tk.Label(self.home_panel, text="FITNESS TRACKER", font=("Arial", 20, "bold"), fg="blue", bg="#800000")
        title_label.grid(row=0, column=0, columnspan=3, pady=10)

        tab_panel = tk.Frame(self.home_panel, bg="#800000")
        tab_panel.grid(row=1, column=0, columnspan=3)

        start_button = tk.Button(tab_panel, text="Start Step Count", bg="yellow", command=self.handle_start)
        stop_button = tk.Button(tab_panel, text="Stop Step Count", bg="yellow", command=self.handle_stop)
        display_button = tk.Button(tab_panel, text="Display Data", bg="yellow", command=self.switch_to_data_display)

        start_button.grid(row=0, column=0, padx=10)
        stop_button.grid(row=0, column=1, padx=10)
        display_button.grid(row=0, column=2, padx=10)

    def handle_start(self):
        print("Start Step Count")

    def handle_stop(self):
        print("Stop Step Count")

    def switch_to_data_display(self):
        self.selected_tab = ""
        self.create_data_display_page()

    def create_data_display_page(self):
        if hasattr(self, 'data_display_panel'):
            self.data_display_panel.destroy()
```

```python
def switch_to_hourly_step_count(self):
    self.selected_tab = "Hourly Step Count"
    self.create_step_count_page()

def switch_to_daily_step_count(self):
    self.selected_tab = "Daily Step Count"
    self.create_step_count_page()

def switch_to_weekly_step_count(self):
    self.selected_tab = "Weekly Step Count"
    self.create_step_count_page()

def create_step_count_page(self):
    if hasattr(self, 'step_count_panel'):
        self.step_count_panel.destroy()

    self.step_count_panel = tk.Frame(self.root, bg="#800000")
    self.step_count_panel.grid(row=0, column=0, padx=10, pady=10, sticky="nsew")

    title_label = tk.Label(self.step_count_panel, text="FITNESS TRACKER", font=("Arial", 20, "bold"), fg="blue", bg="#800000")
    title_label.grid(row=0, column=0, columnspan=3, pady=10)

    step_count_text = self.display_step_count()
    step_count_label = tk.Label(self.step_count_panel, text=step_count_text, font=("Arial", 14), fg="black", bg="#800000")
    step_count_label.grid(row=1, column=0, padx=10, pady=10)

    back_to_data_display_button = tk.Button(self.step_count_panel, text="Back to Data Display", bg="yellow", command=self.switch_to_data_displ
    back_to_home_button = tk.Button(self.step_count_panel, text="Back to Home", bg="yellow", command=self.create_home_page)

    back_to_data_display_button.grid(row=2, column=0, padx=10)
    back_to_home_button.grid(row=2, column=1, padx=10)

def display_step_count(self):
    result = f"{self.selected_tab}:\n"
    if self.selected_tab == "Hourly Step Count":
        result += self.display_hourly_step_count()
    elif self.selected_tab == "Daily Step Count":
        result += self.display_daily_step_count()
    elif self.selected_tab == "Weekly Step Count":
        result += self.display_weekly_step_count()
    return result
```

```
    DECLARE threshold FLOAT;
    SET threshold = (SELECT moving_average + (2 * standard_deviation) FROM (SELECT AVG(v_sum) AS
moving_average, STDDEV(v_sum) AS standard_deviation FROM StepCount ORDER BY `time` DESC
LIMIT 100) AS subquery);

    IF NEW.v_sum > threshold THEN
        INSERT INTO FallAlert (`time`, `acc_x`, `acc_y`, `acc_z`, `v_sum`)
        VALUES (NEW.`time`, NEW.`acc_x`, NEW.`acc_y`, NEW.`acc_z`, NEW.`v_sum`);
    END IF;
END
```

6. Collaboration in Concept Formulation, Design, Development, and Implementation:

- Concept Formulation: The initial concept of a Fitness Tracker GUI was shaped through discussions, taking into account user requirements and objectives.

- Design: Collaborative efforts were made to design the graphical user interface, user interactions, and the approach to accessing and presenting data.

- Development: The coding phase involved the implementation of the entire system, including GUI components, database integration, and real-time updates.

- Implementation: The culmination of the project was the deployment of the Fitness Tracker GUI, ready for users to interact with and benefit from.