Cierra Danielson
Natsnet Tekie
Group 2

General Program Documentation

12/06/2024

Program Overview

The GPA Tracker is a Python application that helps users manage academic courses and calculate their Grade Point Average (GPA). It features a graphical user interface (GUI) built with tkinter and includes functionalities for adding courses, assigning grades, and viewing GPA results.

File Descriptions

1. **main.py**
    - **Purpose**: Serves as the entry point for the application. It initializes the main window and provides navigation to manage courses and view GPA.
    - **Key Features**:
        - Creates the main application window with buttons for managing courses and viewing GPA.
        - Sets up the program's icon, title, and layout.
    - **Dependencies**:
        - tkinter: For GUI rendering.
        - gui_utils: For handling course management and GPA results logic.
        - functions: For core functionalities like adding courses and GPA calculation.
2. **gui_utils.py**
    - **Purpose**: Contains helper functions and windows for managing courses and displaying GPA results.
    - **Key Features**:
        - **Course Management**:
            - Add new courses with credit values.
            - Assign grades to existing courses.
            - Display all courses in a table format using a Treeview widget.
        - **GPA Results**:
            - Display the calculated GPA based on entered grades and course credits.
            - Show a detailed breakdown of courses, grades, and credits in a table.
    - **Dependencies**:
        - tkinter: For GUI components.
        - functions: For business logic like course management and grade calculations.
3. **functions.py**
    - **Purpose**: Implements core business logic for handling courses, grades, and GPA calculations.
    - **Key Features**:
        - **Gradebook Management**:
            - Add courses with unique names and credit values.
            - Assign grades to courses.
        - **GPA Calculation**:

        - Calculates GPA based on weighted grade points and credits.
    - **Dependencies**:
        - None (self-contained).

Program Flow

1. The user launches the application (main.py).
2. The main window provides options to:
    - Manage courses (open_course_management in gui_utils.py).
    - View GPA results (open_gpa_results in gui_utils.py).
3. Courses and grades are stored in a gradebook dictionary (in functions.py), and GPA is calculated dynamically based on this data.