GUI Documentation

User Documentation
Introduction
Welcome to the Registration Estimate Calculator, a user-friendly application designed to forecast the final number of registrations for events based on current data and historical trends. This tool is intended for event organizers and marketers to help inform strategic decisions regarding additional advertising needs.

Getting Started

- **Installation:** Before using the Registration Estimate Calculator, ensure that Python 3.10 is installed on your computer, as the application is developed with this version. You can download Python 3.10 from the official Python website. Once Python is installed, you can launch the GUI by executing the provided script file.

User Interface
The interface of the Registration Estimate Calculator includes:

1. An entry field for the Current Number of Registrations: Enter the most recent count of registrations for your event.
2. An entry field for the Days Left in Registration Period: Input the number of days remaining until the registration deadline.
3. A Calculate button: Click this to compute the estimated final registrations.
4. A Clear button: Use this to reset all fields and calculations.
5. Display areas for the Estimated Final Registrations, Upper Bound Estimate, and Lower Bound Estimate: These areas display the calculated results.

Using the Calculator

1. Enter the current number of registrations in the designated field.
2. Input the number of days left until the end of the registration period in the corresponding field.
3. Click the Calculate button to generate the estimated final registration count, along with upper and lower bounds that account for variability.
4. Review the calculated estimates displayed on the screen.
5. To perform a new calculation, click the Clear button and repeat the steps.

Troubleshooting
If you encounter issues while using the application, such as error messages or unresponsive buttons, first ensure that all inputs are numeric and valid. If the problem persists, restart the application and try again.

Technical Documentation
Overview
This document provides a comprehensive overview of the technical aspects of the Registration Estimate Calculator application. It is intended to guide future developers through the application's functionalities, architectural design, and codebase for maintenance or further development.
Running the Application
Requirements
Ensure Python 3.10 is installed on your system to run the application without compatibility issues.
Libraries and Modules

- The application uses the latest versions of the following Python modules as of its development:
- Tkinter for the GUI interface.
- matplotlib for plotting (if applicable in extensions).
- Other standard libraries such as math and datetime for calculations and date manipulations.
- **Execution**

Run the script with Python. Ensure the data file path is correctly specified.
Architecture Overview
The application follows a modular architecture, centred around a GUI built with the TKinter library. The main components include input fields for user data, calculation logic for forecasting registrations, and display labels for showing the results.
Components

- **Input Fields:** Capture user inputs for the current number of registrations and days left in the registration period.
- **Buttons:** Include "Calculate" for initiating the forecast calculation and "Clear" for resetting the input fields and results.
- **Display Labels:** Used to show the estimated final registrations, and the upper and lower bound estimates.

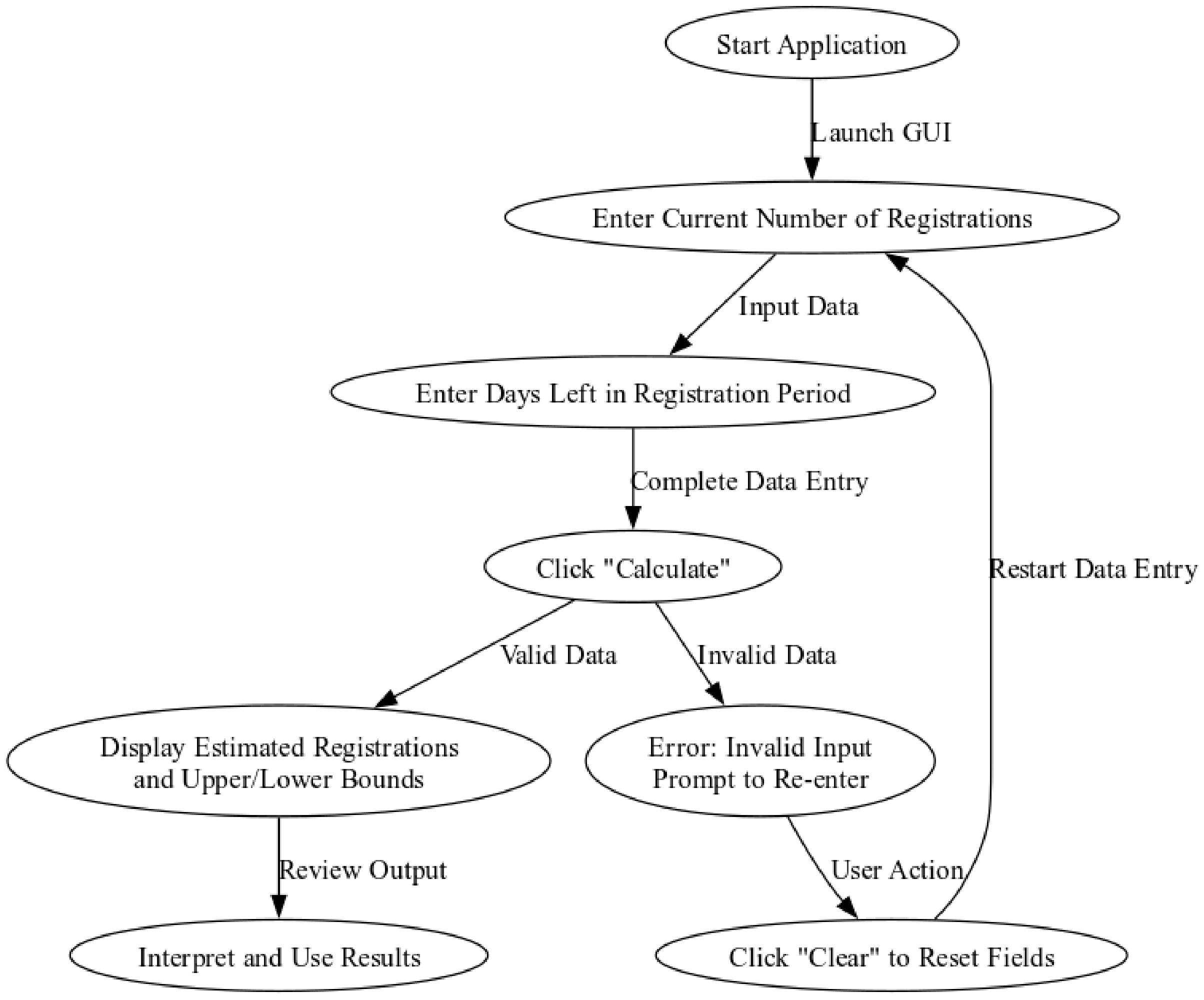
Gradient Calculation Logic

- The core logic involves calculating the estimated increase in registrations using a pre-calculated average gradient and its error. The formulae used are:
- Estimated Final Registrations = Current Registrations + (Average Gradient * Days Left)
- Upper Bound Estimate = Current Registrations + ((Average Gradient + Gradient Error) * Days Left)
- Lower Bound Estimate = Current Registrations + ((Average Gradient - Gradient Error) * Days Left)
- Error Handling
- The application includes error handling mechanisms for invalid inputs (non-numeric), displaying error messages through messagebox. showerror() and allowing users to correct their inputs.

Interface Design
The GUI layout is designed for simplicity and ease of use, with clear labels for each field and button. The design ensures that users can easily navigate the application and understand how to input data and interpret the results.
Code Organisation
The application's code is organised into functions corresponding to each major operation, such as data input, calculation of estimates, and updating the GUI with results. Comments and docstrings are used throughout the code to explain the purpose and usage of each function and variable.

Testing
Basic functional testing has been conducted to ensure the application performs as expected under various input scenarios. Future developers are encouraged to extend these tests, especially when adding new features or modifying existing functionalities.
Version Control
The application's source code is managed using Teams File space for version control.

Diagram showing Interaction Flow


Start Application
Launch GUI
Enter Current Number of Registrations
Input Data
Enter Days Left in Registration Period
Complete Data Entry
Click "Calculate"
Valid Data
Invalid Data
Display Estimated Registrations and Upper/Lower Bounds
Error: Invalid Input Prompt to Re-enter
Review Output
User Action
Interpret and Use Results
Click "Clear" to Reset Fields
Restart Data Entry