NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY POLITEHNICA OF BUCHAREST - ROMANIA

FACULTY OF AUTOMATIC CONTROL AND COMPUTER SCIENCE
SYSTEM ENGINEERING FIELD

FINAL REPORT – *My Carbon App*

1) Introduction

The carbon footprint refers to the total amount of greenhouse gases (mainly carbon dioxide) that are emitted into the atmosphere as a result of different human activities. The rising levels of CO2 and other greenhouse gases are among the most critical environmental concerns, significantly contributing to air pollution, climate change, and global warming.

With growing awareness of these issues and the urgent need to reduce emissions and promote an eco-lifestyle, the purpose of this project is to help individuals and households better understand their carbon footprint impact on the environment. By providing tools to measure, track, and manage personal CO_2 emissions from transportation, household energy consumption or dietary choices, this concept encourages informed decision-making for reducing environmental degradation.

2) Objectives of the project

The objective of *MyCarbonApp* is to provide users with a powerful tool to measure and manage their carbon footprint. With these goals in mind, the application was developed to focus on three core areas:

- ✓ Personal carbon footprint (from daily activities such as commuting, respiration or waste management contributions),
- ✓ Household carbon footprint (related to housing consumption details in terms of electricity, heating and gas, each factor)
- ✓ Dietary carbon footprint (emissions based on food choices / diets).

It allows users to input data or choose options regarding their usual lifestyle. By giving users insight into the CO_2 emissions generated by casual activities — such as transportation, household energy use, diet, and waste disposal — the app empowers them to take actionable steps toward reducing their environmental impact. Therefore, the goal of this piece of software is to promote sustainable living and global initiatives to decrease greenhouse gas emissions level and protect the environment for next generations.

3) Analysis / Documentation ("State of the Art")

As a calculator-based application, MyCarbonApp prioritizes simplicity in user interface design to ensure users can easily input and validate their data. Before developing our app, extensive research was conducted on existing carbon footprint calculators, including online tools, mobile applications, and enterprise solutions. The following section provides a comparative analysis of these solutions, their strengths, and their limitations.

- **Online Carbon Footprint Calculators (WWF Carbon Footprint Calculator, EPA Carbon Footprint Calculator, Carbon Footprint Ltd. Calculator):** they are highly accessible for all platforms, provide quick estimations based on user input and government / scientific data but lack real-time data integration. This means that calculations are based on predefined coefficients without adapting to user-specific conditions or personalized consumption patterns. Moreover, they don't have an offline functionality and require a strong network connection.
- **Mobile Applications (JouleBug, Sphera Eco):** these apps are portable, convenient for continuous tracking of carbon footprints and gamify carbon reduction, increasing user engagement. However, in most cases they rely only on one category, such as diet, transport, or home energy use, rather than on a holistic analysis. They don't offer detailed breakdowns or scientific justification for their estimations or emission factors.
- **Enterprise Solutions (SAP Sustainability Footprint Management):** use advanced data and AI analytics for environmental impact assessments but are too complex for personal use.

To address the shortcomings of existing tools, MyCarbonApp was developed as a hybrid solution that balances usability, accuracy, and feature comprehensiveness. Unlike many existing applications that depend on external API calls for updated emission factors, our approach incorporates:

- Real-time personalization: Users can input detailed transport, diet, waste and household consumption data, dynamically adjusting calculations based on individual metrics such as transport types (car, scooters, bikes, public transportation), average breath-related emissions, fuel consumption, waste disposal methods or residential emissions, making the app more tailored to the user's daily routine.
- Offline functionality: Unlike online tools, MyCarbonApp runs calculations locally without requiring internet access.
- User-friendly yet detailed insights: The app provides interactive elements while maintaining scientific accuracy in emissions calculations.
- Help Center – a page with detailed explanations about how the Carbon Footprint for each selected category is calculated, including all estimations and emission factors used.

The modular OOP design of MyCarbonApp allows for the seamless addition of future features such as renewable energy integration and advanced diet calculations.

We wanted to extend the traditional carbon footprint calculation models by introducing these additional features. The core logic for calculating emissions relies on well-established carbon emission factors (e.g., CO_2 per liter of fuel or per kWh of electricity consumed) which are derived from governmental and scientific studies that estimate emissions based on average consumption patterns per selected country. Key sources include:

- The Intergovernmental Panel on Climate Change (IPCC) for transport and energy emission values.
- The Food and Agriculture Organization (FAO) for dietary carbon impact factors.
- The United Nations Framework Convention on Climate Change (UNFCCC) for sustainability benchmarks.
- Lege5 (https://lege5.ro/Gratuit/ha4demjwgu3q/factorii-de-conversie-in-emisii-de-gaze-cu-efect-de-sera-co-2-echivalent-ghid-anexa-nr-4-la-ghid-factori-de-conversie?dp=gqytcnztga3dimi&utm_source=chatgpt.com) for Romania's emission factors measurements.

Each emission calculation follows established methodologies, ensuring that results align with real-world environmental impacts. The use of adjustable coefficients further refines accuracy by allowing updates based on new scientific findings. Some diet options are based on estimations of multiple available options for a specific field (for example: Chocolate, desserts, sweets for "Sweets" category).

4) Justification of the solution

The solution provided is justified by the increasing need to quantify individual environmental impact. By allowing users to input specific data like transportation modes and daily habits (e.g., kilometers traveled by using specific transport options), the application can offer a more accurate and personalized estimation of carbon emissions compared to generalized calculators available in other platforms.

MyCarbonApp was designed using C++ and the Qt framework, leveraging their cross-platform capabilities, high performance, and extensive GUI support. The justification for these choices is detailed below:

- ✓ Cross-platform compatibility:
  - Qt provides a unified API that allows the application to run on Windows, macOS, and Linux with minimal modifications.
  - The UI components are implemented using *QWidgets*, ensuring native integration with different operating systems.
  - *QStackedWidget* is used for seamless navigation between different carbon footprint categories (transport, home, diet, waste management).The

architecture using custom functions (e.g: *createIndividualPage, createHomePage, createDietPage*) for collecting user data or widgets / message boxes for creating a dynamic and interactive user interface is practical, as it allows real-time data input and feedback, making the application easy to use.

- ✓ Efficient performance and low resource consumption
    - *QMap* and *std::vector* are utilized for efficient data storage and retrieval, optimizing lookup operations for user inputs.
    - Memory management is optimized using smart pointers (std::unique_ptr, std::shared_ptr) to prevent memory leaks.
    - Lazy initialization is used for UI elements (e.g., *infoWindow* (Help Center) is only created when accessed) to minimize memory overhead.
- ✓ Modular and Scalable Design (thanks to OOP features – more details in the next section: “Implementation details”)
- ✓ User-Friendly Graphical Interface: The UI is built using *QStackedWidget*, ensuring only relevant sections are displayed, reducing memory overhead. Interactive elements include:
    - *QComboBox* for transport modes, fuel types, and dietary selections.
    - *QListWidget* for multi-selection inputs, such as transport options and waste disposal methods.
    - *QLineEdit* with validation for numerical inputs like distances traveled and food consumption quantities.
    - *QDebug* for tests and debugging using the console.
    - Input validation mechanisms prevent errors (e.g., non-numeric values) using *QValidator*. Warning: the current version only supports integer input values. Decimal numbers are not recognized (more details about limitations are available in the section: “Drawbacks”).
    - Dynamic UI updates: Elements like transport input fields are added or removed dynamically based on user selections.
- ✓ Dynamic Waste Management Tracking: Users can log waste disposal methods (plastic, paper, food waste) and calculate CO_2 impact accordingly. The system differentiates between landfill, recycling, and composting, applying appropriate emission coefficients. *QMap*-based storage ensures fast lookup of waste-related emission factors.
- ✓ Dietary Carbon Footprint Calculation Based on Consumption Data: Users input weekly food consumption amounts, instead of selecting broad dietary categories. The system applies per-kilogram CO_2 emission factors, making calculations more precise than traditional estimators. Food-related emission factors are stored locally for consistency and can be updated manually as needed.

Compared to other potential approaches, such as hardcoding static values or limiting transportation modes, this solution offers flexibility. By including various transport modes, fuel types, and scenarios (like household, advanced diet control, waste management or respiration rate), the application stands out as a more comprehensive tool. As a result, this solution combines accuracy, user-friendliness, and comprehensiveness.

Moreover, users can select their country (Romania, Ukraine, Portugal, Spain) from a *QComboBox*, adjusting household emission calculations based on country-specific energy grid emissions. This makes *MyCarbon* both practical and impactful in the context of modern carbon management tools.

5) Implementation details

Our application involves object-oriented programming principles and rules (such as behavioral patterns, methods and attributes of different classes and objects ready to manipulate, store or modify data, based on a specific scenario). The latest version features some minor changes in terms of UI, new waste section and a revamped diet page, responsiveness and performance optimizations. More details about our implementation and features are displayed below:

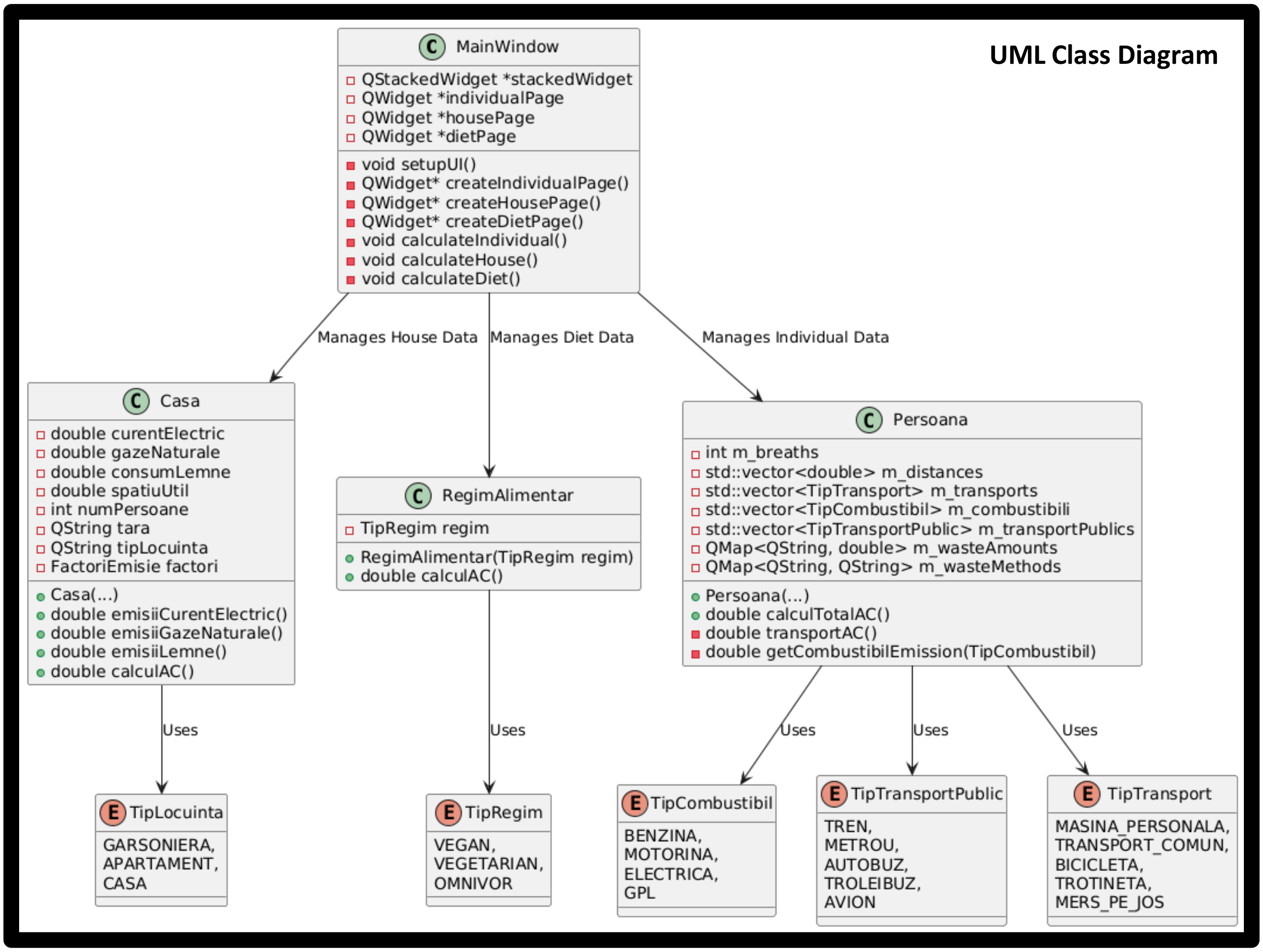
a) UML Diagrams (separate *.jpg* files):

- ✓ Class diagram → shows the key classes such as *MainWindow, IndividualData, Home, Diet,* and their relationships.
- ✓ Sequence diagram → depicts the sequence of actions when a user calculates their carbon footprint using different transport modes (such as: cars, busses, trains, metro, bicycle, scooters) or fuel consumption (petrol, diesel, LPG)

b) OOP elements utilized:

- ✓ Classes → *MainWindow, IndividualData, Home, Diet,*
- ✓ Objects → Instances of different transport (Car, Bike, Bus), fuel and diet types.
- ✓ Inheritance → *MainWindow* inherits from *QMainWindow,* gaining all base functionalities while adding specific UI and CO_2 calculation features.
- ✓ Encapsulation → *MainWindow* manages the UI and related functions and *QLineEdit, QComboBox, QListWidget* are private members, accessed only through dedicated methods.
- ✓ Polymorphism (Dynamic behavior is achieved using Qt slots and signals) → *void showIndividualPage(), void showHousePage(), void showDietPage()* – functions dynamically switch UI pages; *void calculateIndividual(), void calculateHouse(), void calculateDiet()* – different CO2 calculation methods based on selected categories; *void onTransportTypeChanged(QListWidgetItem *item)* – updates UI dynamically when a (public) transport type is changed (signal mechanism)
- ✓ Modular architecture → UI elements are separated from the calculation logic, making the system scalable and easy to maintain.

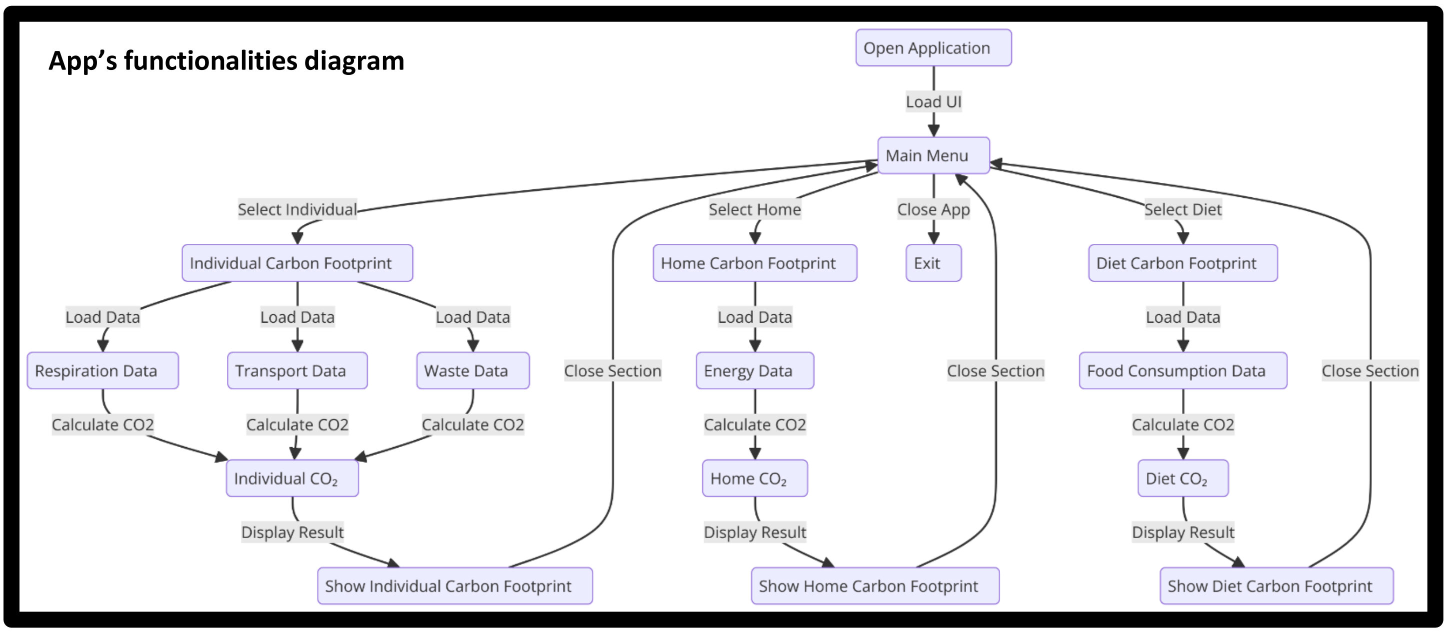
✓ Observer Pattern → the system notifies different modules (e.g., display of results) when the user input is processed and emissions (average Carbon footprint) are calculated; *void goBack()* – handles navigation without recreating UI components (signal-slot mechanism)



Each design / behavioral / back-end element or functionality has been carefully developed by the team members who are second-year students from the National University of Science and Technology "Politehnica" of Bucharest, Faculty of Automatic Control and Computer Science - System Engineering (Automatic Control) field. Obviously, each member had a crucial role during the development of this project:

✓ Radu-Andrei Cîrcioroabă → Data processing and calculation collaborator (responsible for developing mathematical models to estimate the carbon footprint efficiently, based on the three categories presented above: individual carbon footprint, diet/eating habits, and housing/residence) & layout designer.

- ✓ Andrei-Valentin Bălălău & Ionuț-Cristian Pleșeanu → Graphic User Interface contributors (responsible for implementing Qt graphic elements, using designs from our layout designer and features like adaptive video resolution, responsive design, and functional buttons) & integration developers (connecting visual elements with backend functionalities).
- ✓ Ștefăniță Lican & Eduard-Nicușor Șipanu → Core developers, monitoring overall performance and implementing core features in C++ (main functionalities of a simplified carbon measurement app, including object-oriented operations with classes and various linking methods).
- **!** **More details about our roadmap, role insights, project updates and future plans are available on our GitHub repository: https://github.com/andrewen11/MyCarbon**

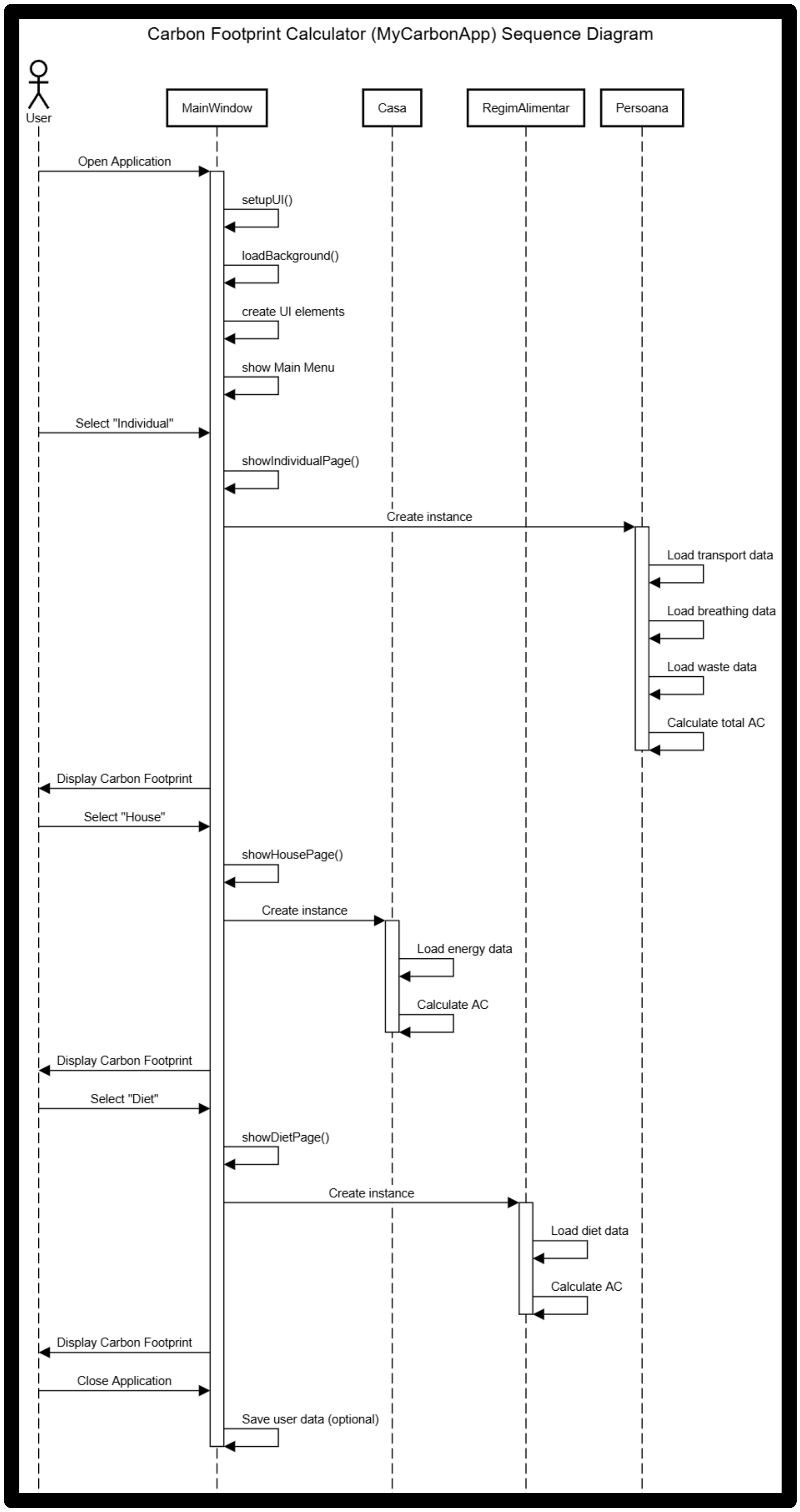


6) App's functionalities

1. The main menu presents options to calculate the carbon footprint based on different categories: *Individual (breathing, transport, waste), Home (energy usage based on electricity, gas, firewood), Diet (CO_2 impact based on food consumption)*. When the user selects a category, the interface updates dynamically to show relevant input fields for the selected category.
2. Users manually enter values like weekly food consumption, monthly transport distances, house size, or waste disposal habits. Inputs are restricted to integers, ensuring consistency in calculations. The system applies predefined CO_2 emission factors (saved in *carbon_classes.h*) to the user's inputs and the final results are scaled annually. Some of them are based on some sublinear calculation models and estimations of energy consumptions per different housing types or usable space.

3. Calculations occur in real-time, updating results without requiring external API calls. The UI updates instantly based on user selections (e.g., choosing a car changes fuel type options).
4. The total CO_2 footprint is displayed for the selected category. If needed, users can navigate back and adjust values to see how different choices affect emissions.
5. The app provides personalized insights based on the user's footprint (e.g., "Your carbon footprint is high! Consider reducing red meat and processed foods."). Users can compare their footprint with benchmark values to see how they rank environmentally (usually, an average Carbon footprint lower than 2000 kg CO2 emissions / year indicates a low footprint). The application ensures that data is stored only temporarily, focusing on privacy and fast processing.

More details on the exact methods used to calculate the total carbon footprint for each section can be found in the *'Help Center'* folder or by clicking the *Help* button in the main window.



7) Performance indicators and optimizations

There were several key performance indicators for the application taken into consideration, some of them being properly implemented in the final stage of the project's development. Most of them include:

- ✓ Accuracy → The application provides accurate carbon emission calculations based on the type of transport selected and distance traveled, with coefficients for each mode. Comparison of calculated results with established carbon footprint benchmarks and data confirms a deviation of less than **6-7%,** making the estimations highly reliable.
- ✓ Efficiency → Real-time calculation ensures a quick response (estimation) after user input. The average execution time for carbon footprint calculations remains below 1 s, ensuring near-instant results for user inputs. The app handles different types of data (transport, breathing rate) without technical glitches or bugs.
- ✓ Scalability → The system was extended to add even more types of transport and additional emission factors (e.g., energy consumption based on country selection) or new selection panels (e.g: (natural) garbage collection method / use), without affecting the general performance (in terms of speed).

Obviously, there is room for improvement in terms of speed and memory usage. The following potential updates can reduce memory usage by up to **30%** and improve UI responsiveness by at least **20%**:

Features	**Current implementation**	**Potential improvement / suggestion**
Calculation Efficiency	Processes entire dataset on each input change	Incremental recalculations **only** for modified inputs
UI Performance	UI elements dynamically created & stored in memory	Remove UI elements when deselected to free memory
Memory Usage (for result pop-up + recommendations)	*QPixmap* images may use excessive memory	Downscale images dynamically before loading
Transport/Waste Updates	Iterates through all selected items on every update	Precompute emission multipliers and update only when necessary
Emission Factors	Stored locally, requiring manual updates	Allow manual updates by using external file(s) (CSV)

8) Drawbacks:

1. Calculation issues: The application performs instantaneous calculations but does not store past data.
2. Input limitations: All input values must be integers; other numerical formats are not supported in this version.
3. UI resize issue: The Help Center window does not resize dynamically with the main window or match its full-screen size.
4. Taskbar behavior: When launching the app and pressing *"Start Calculator"*, the app's icon disappears from the taskbar. This is intentional, as the start menu window is hidden to keep only the main calculator window active. However, if the user switches to another application, there is a risk that *MyCarbonApp* may become invisible (as it runs as a background process). To reopen it, users need to check the *Task Manager*, look for *"CarbonFootprintCalculator.exe"* under background processes and select it.

9) Conclusions

"MyCarbonApp" offers a unique focused approach to monitoring and managing carbon footprints. The incorporation of both transport-related and physiological data sets it apart from other similar applications on the market.

It demonstrates a strong combination of user-friendliness, scientific accuracy, and comprehensive coverage of carbon-emitting activities. Its ability to offer personalized insights into daily lifestyle choices positions it as a practical tool for individuals who are conscious of their environmental impact and who can inspire meaningful behavioral changes, such as adopting more sustainable transport methods or reducing unnecessary trips.

Through ongoing updates and expansion, the app could grow into a key resource for users looking to reduce their carbon footprint and contribute to global sustainability efforts by including broader environmental factors like advanced home energy use or accurate emission estimations for other residential / industrial / commercial buildings.