Domestic Student Relocation Emissions Calculator

Ivan Ivanov

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Bachelor of Science
of the
University of Aberdeen.

Department of Computing Science

2024

Declaration

No portion of the work contained in this document has been submitted in support of an application for a degree or qualification of this or any other university or other institution of learning. All verbatim extracts have been distinguished by quotation marks, and all sources of information have been specifically acknowledged.

Signed: *Ivan Ivanov*

Date: April 2024

Abstract

The growing urgency of addressing climate change necessitates that organizations, including universities, address their environmental impact. However, existing emissions measurement tools often fail to capture the unique needs of universities, especially regarding student relocation travel emissions.

This dissertation presents the development, testing, and evaluation of the innovative StudentGreenTravel software, designed to be used by the University of Aberdeen to accurately calculate its CO2 emissions resulting from student relocation. The tool employs groundbreaking methodology for calculating carbon dioxide emissions based on anonymized data (postcodes), chosen methods of transport, and travelled distance, while offering a convenient user interface and features such as customization of emission factors and invalid data handling.

Beyond its technical contributions, this endeavor fosters a deeper understanding of the importance of achieving Net Zero and mitigating environmental impact.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Dr. Kamaran Fathulla, for his unwavering support and guidance throughout this project. His advice was instrumental in achieving the outcome of this project. I am particularly grateful for his patience and encouragement as I navigated the challenges of this endeavour.

I am deeply thankful to Roederer Rose Lyne, the Net Zero and Emissions Manager at the University of Aberdeen, for generously sharing her expertise and resources. Her provision of data, tools and insights were critical to the project's success. Additionally, her invaluable and constant feedback greatly enhanced the functionality and usability of the software.

Finally, I would like to thank the University of Aberdeen Computing Science Department for providing me with the knowledge and skills required to undertake this project. The department's dedication to excellence and growth, together with the opportunity to learn from industry experts throughout my degree, are invaluable assets which I will always cherish.

Contents

Acronyms

CO_2 Carbon Dioxide. 10, 20

ACUPCC American College and University Presidents' Climate Commitment. 12

API Application Programming Interface. 31, 34, 35, 37, 56

AQ Additional Question. 48, 51

CA Certification Authority. 45

CEPI Confederation of European Paper Industries. 16

CF Carbon Footprint. 11, 12, 15–17, 19

CO2e Carbon Dioxide Equivalent. 15, 16

DISRTECT Domestic and International Student Relocation Travel Emissions Calculator Tool. 13, 14, 19, 46, 48–50

EMR Estates Management Record. 13

GDPR General Data Protection Regulation. 29

GHG Greenhouse Gas. 10, 12, 15–18, 20

GHGs Greenhouse Gases. 15–17

GUI Graphical User Interface. 14, 33, 39, 53

HEI Higher Education Institution. 11, 28

HEIs Higher Education Institutions. 13, 15, 17

IDE Integrated Development Environment. 33

IEA International Energy Agency. 20

IT Information Technology. 48

NSPL National Statistics Postcode Lookup. 34

OS Operating System. 56

PBCCD Public Bodies Climate Change Duties. 20

PC Personal Computer. 44

SDGs Sustainable Development Goals. 11, 20

SQL Structured Query Language. 33, 34

SUS System Usability Scale. 48–51

UCCCfS Universities and Colleges Climate Commitment for Scotland. 20

UI User Interface. 29, 31

UK United Kingdom. 19, 20, 22, 26, 27, 30, 34, 35, 37, 40, 52

UN United Nations. 19

US United States. 19, 20

Chapter 1

Introduction

1.1 Overview

The current project focused on both measuring and substantially improving the precision of the emissions gauged from domestic student travel at the University of Aberdeen. This initiative stands as a pivotal endeavour in the context of global climate change. In recent years, climate change has emerged as a central focus of many nations and organizations around the world. This greater attention stems from the frequent incidence of different natural disasters such as severe storms, heatwaves, droughts, and rising sea levels. These phenomena are widely believed to be related to the ongoing processes of urbanization and industrialization in modern society (Sekoai and Yoro, 2016). Greenhouse gases, particularly Carbon Dioxide (CO_2), are foremost among the factors driving climate change. This gas is primarily emitted through the combustion of fossil fuels, which is mainly interconnected with human activities (Thiruvenkatachari et al., 2009).

Many papers on this topic have been published that talk about the different impacts which businesses and organizations have on the environment through their activities. However, only recently the question of the impact on the environment of universities and their activities has been raised. First, Shields (2019) demonstrated that the movement of international students contributes to greenhouse gas emissions, with globally mobile students emitting above the average traveler. Furthermore, he pointed that the combined carbon footprint of internationally mobile students from air travel alone rivals the total emissions of a small nation.

Universities, operating as institutions dedicated to education, research, and community outreach, hold a significant responsibility in generating knowledge, embedding sustainability into educational and research initiatives, and advocating for environmental concerns within society (Güereca et al., 2013; Valls-Val and Bovea, 2021). Furthermore, their activities generate direct (Scope 1) and indirect (Scope 2 & 3) emissions as per the Greenhouse Gas (GHG) protocol (Ranganathan and Corbier, 2011). These three scopes help delineate direct and indirect emission sources of an organization, as shown in Figure 1.1.

Scope 1 or direct emissions occur from sources that are owned or controlled by the organization. For example, emissions from chemical production in owned or controlled process equipment. Scope 2 emissions are from the generation of purchased electricity which is consumed by the institution. Scope 3, which is considered optional in regards to measuring and reporting, is related to indirect emissions that occur as a consequence of the organization's activities, but the sources of these emissions are not owned or controlled by the organization itself (Ranganathan and Corbier, 2011).

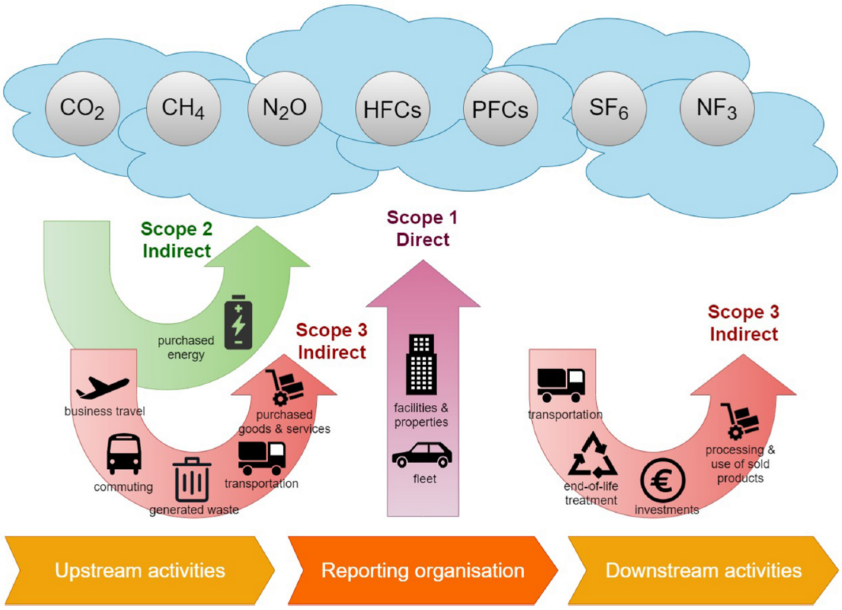

Figure 1.1: Scopes of Emissions (WRI and WBCSD (2013))

In the university context, Scope 3 is related to construction, transport, consumption of materials (water, paper, electrical and electronic devices, laboratory chemicals, etc.) and waste generation (non-hazardous, hazardous, and electrical and electronic equipment, etc.) (Townsend and Barrett, 2015). Transport emissions are considered a significant portion of total Scope 3 emissions across numerous studies. For instance, in De Montfort University these emissions account for 29% of the total carbon footprint (Ozawa-Meida et al., 2013). However, because travel-related emissions have been considered optional and, until recently, solely comprised student and staff commuting between their term-time address and the Higher Education Institution (HEI) (Davies and Dunk, 2015), this omission highlights a substantial gap in the literature regarding the sustainability and environmental impact of higher education.

Research on higher education and sustainable development has extensively explored the advantages and value of international student mobility and education like cultural, personal, and career outcomes (Roy et al., 2019) but has often neglected to consider their environmental impact (Cortese, 2003). Furthermore, the oversight of the broader social and economic contexts within which universities operate often results in a failure to recognize the limitations in addressing sustainability issues, thereby aggravating the critical gap related to understanding the link between international student mobility and its contribution to global climate change (Shields, 2019).

The recent surge in awareness of this void can be attributed primarily to various international agreements and contracts that aim to significantly reduce carbon emissions by mid-century. Such agreements are the 2015 Agenda 2030 for Sustainable Development (Hanemann, 2015) that outlined 17 Sustainable Development Goals (SDGs) to be achieved by 2030, and the 2019 European Green Deal (Commission et al., 2019), which included 50 specific actions to combat climate change. They in turn prompted the development of different guidelines and recommendations oriented towards calculating Carbon Footprint (CF) in universities.

The American College and University Presidents' Climate Commitment (ACUPCC) (Dautremont-Smith et al., 2009) provides institutions with recommendations for reporting, planing for climate neutrality and creating programs for sustainability on campus. Cool Campus (Simpson, 2009) offers guide in action planning and tips for calculating the CF as well as mitigating measures. The Second Nature Platform (Nature, 2020) presents a series of recommendations for selecting the emission sources, calculating GHG emissions and analysing their evolution or identifying mitigation strategies. These are just a few examples of the current guidance documents available. As a consequence of these developments, tracking and measuring different emissions related to various activities and productions became a standardized approach that facilitates planning and decision-making (Valls-Val and Bovea, 2021).

1.2 Problem Identification and Motivation

In order to effectively measure and mitigate their carbon footprint, universities rely on emission calculators or trackers (Figure 1.2), sophisticated tools designed to capture various inputs such as electricity consumption, water usage, staff and student travel distances via diverse transportation modes, waste management data, and more. These calculators serve as invaluable instruments for quantifying emissions and informing sustainability strategies (Birnik, 2013).

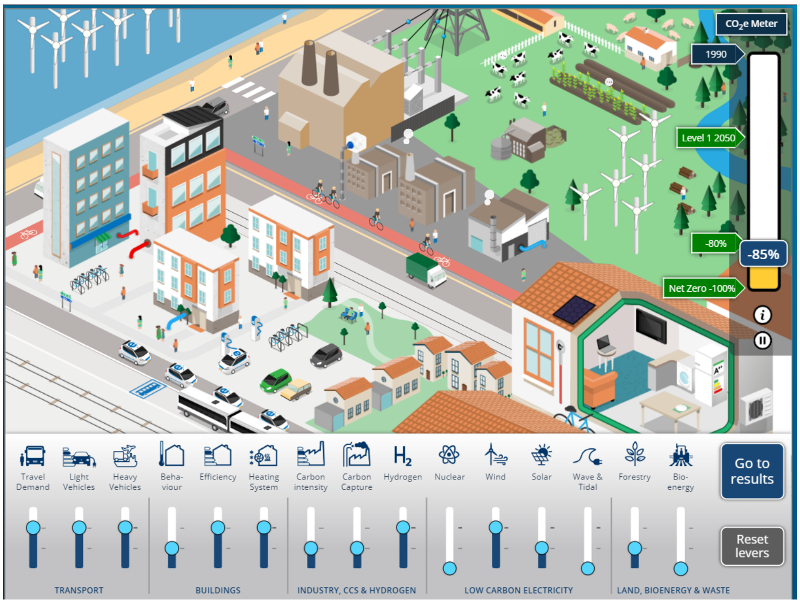

Figure 1.2: UK Government – My 2050 Carbon Calculator (2020)

However, existing emission calculators often fall short in accounting for emission sources within institutional settings such as Scope 3 emissions (Valls-Val and Bovea, 2021). This deficiency has spurred some universities to take proactive steps by developing their own calculators to better align with their specific operational contexts. Notable examples of such initiatives include the CO2UNV, SIMAP 2020, and the Domestic and International Student Relocation Travel Emissions Calculator Tool. While these tools provide better flexibility and suitability, they often require further refinement to accurately capture all emission sources, particularly those associated with student travel (Schaltegger and Csutora, 2012; Williams et al., 2012).

Additionally, universities face two major persistent obstacles in accurately quantifying student travel emissions:

1. **Confidentiality constraints**: Information such as students' home addresses and transportation methods can be considered personally identifiable information, while information on frequency and means of travel can reveal personal habits or financial situations. Universities have a responsibility to protect student privacy, and collecting detailed data might raise concerns about potential misuse (Earp and Payton, 2001).

2. **Data availability and reliability constraints**: An uncertainty assessment of input data for such a measurement, performed at the De Montfort University, showed 75% uncertainty for international students air travel data (Ozawa-Meida et al., 2013). As a result, it was concluded that this type of emissions is frequently underestimated or overlooked in sustainability assessments.

Addressing these challenges and enhancing the accuracy of emission calculations for student travel is crucial for universities to develop effective strategies for reducing their carbon footprint and advancing sustainability goals (Koester et al., 2006).

Such challenges and limitations are present in the current emissions calculator tool utilized at the University of Aberdeen as well. Specifically, the tool calculates travel distance of a student from their home country's capital to the university without considering additional factors such as home address and layover locations. While this can be considered somewhat acceptable for international students, this approach greatly affects the accuracy of calculations for domestic students. Consequently, the tool produces highly inaccurate and possibly overshooting or underestimating data, undermining the reliability of emission assessments and the development of targeted mitigation strategies.

An illustration of the effect of such constraints are the findings from the 2013/2014 Estates Management Record (EMR) return (HESA, 2014), where only 27 HEIs reported on all available Scope 3 sources. There, emissions reported by two HEIs appeared flawed, while the emissions from the remaining 25 institutions accounted for 71% of the total reported emissions. This disparity clearly illustrates the significance of Scope 3 sources, emphasizing how narrowly set boundaries can substantially underestimate emissions, ultimately providing a misleading picture of an organization's carbon footprint (Matthews et al., 2008).

1.3 Objectives

To target the above-described limitations of the Domestic and International Student Relocation Travel Emissions Calculator Tool (DISRTECT), a new travel emissions calculator software will be developed that will include:

- **More accurate travel distance and emission calculations**
  - The main goal of the project is to improve travel distance measurements and emission calculations for domestic students by considering students' home addresses, mode of transport, distances to closest transport hub, distances to Aberdeen transport hubs and layover location.

- **Options for modifying travel methodology assumptions**
  - This is a key goal as it allows accounting for different travel assumptions, such as the proportion of students travelling by various means of transport to Aberdeen and the number of trips they perform on a yearly basis.
- **User-friendly interface**
  - Another major goal of the project is to transfer the current and new functionality of the tool into a new Graphical User Interface (GUI) that will enhance user experience.
- **Visualizations - interactive graphs and travel routes**
  - A secondary goal which will further enhance user experience and make the acquired data easier to read and analyze.

1.4 Contributions

This dissertation greatly extends the groundwork laid by two key contributors at the University of Aberdeen: Estrid Jonsson, a former travel emissions intern, and Roederer Rose Lyne, the Net Zero and Emissions Manager. While Estrid Jonsson is not directly involved in the current project, Roederer Rose Lyne serves as the project's client.

Estrid Jonsson played a key role in developing the initial methodology for calculating student travel emissions, while Roederer Rose Lyne supervised the implementation of the Domestic and International Student Relocation Travel Emissions Calculator Tool (DISRTECT). These developments form the foundation of the current project. Additionally, their tool includes various calculation methodologies for both national and international students, emission factors for different modes of transport, and statistics for emissions per country. This functionality enabled the university to gain a deeper understanding of its Scope 3 emissions and this way assess its performance in comparison to other institutions.

The current project enhances previous calculation methods by providing more detailed insights into the travel patterns of home students, including factors such as home addresses, travel routes, and modes of transport. Additionally, the project offers greater data granularity by providing council-level data on travel distances and emissions, and introduces a new Graphical User Interface (GUI) to improve user experience.

Chapter 2

Background and Related Work

This chapter discuses previous work in the field, including established standards and methodologies for carbon footprint measurement in universities. It explores the challenges which are often encountered, along with recurring patterns and trends in the obtained results. Additionally, the chapter provides an in-depth examination of existing solutions, highlighting their strengths and shortcomings. It concludes with an overview of the increasing international student mobility and its environmental impacts.

2.1 Carbon Footprint Measurement in Universities

This section delves into established methodologies for measuring carbon footprints in universities, along with the challenges encountered. Additionally, it explores the recurring patterns and trends observed in the results obtained from previous studies.

2.1.1 Standards and Methodologies

Higher Education Institutions (HEIs) play an important role in promoting sustainability and should be an example of a sustainable organization (Klein-Banai and Theis, 2013). According to Valls-Val and Bovea (2021), calculating, tracking, and reporting the Carbon Footprint (CF) is often considered a fundamental step for HEIs aiming to transition towards sustainability. The CF, as defined by ISO 14064–1 (2018), represents the total amount of greenhouse gases (GHGs) produced by the activities of an organization, measured in units of carbon dioxide equivalent (CO2e). This measurement encompasses various pollutants and is typically quantified using the GHG emissions formula. With this formula, the emissions from a specific source (E_s) are obtained from the product of activity data (AD_s) and the respective GHG emission factor (EF_s):

$$E_s = AD_s \times EF_s$$

Once the GHG emissions from all specific sources are obtained, they are added up to quantify the total carbon footprint in CO2e of the organization.

Numerous international standards and frameworks have been developed to calculate the CF of organizations, including HEIs. Notable examples are the GHG Protocol (2011), ISO 14064-1 (2018), and PAS 2050 (2008), with the most renowned being the GHG Protocol. In their literature review of 35 studies, Valls-Val and Bovea (2021) found that 54% of the reviewed institutions utilize this protocol, which further cements its reputation as the *de facto* standard for carbon accounting (Ascui and Lovell, 2012).

The purpose of such frameworks is to provide guidelines for identifying emission sources, collecting activity data, and calculating emissions. However, despite their availability, there is a lack of standardization in CF measurement methodologies (Durojaye et al., 2020) as many of them consider different emission sources and emission factors. Additionally, emission factors applied for each source are quite variable and country-dependent in respect to a given university.

Furthermore, a preliminary step for calculating the GHGs of an organization, is the preparation of a greenhouse gas emissions inventory. A GHG emissions inventory is similar to a life cycle assessment that illustrates the environmental impact of activities that generate greenhouse gasses (Eggleston and Tanabe, 2006). These activities can span from resource extraction to waste management and are commonly targeted in sustainability initiatives (Klein-Banai and Theis, 2013). Essentially, these inventories serve as reference points (Bailey and LaPoint, 2016). Nonetheless, there is lack of standardization in this matter as well (Arioli et al., 2020). This is due to the fact that universities are located in different areas, vary in size, number of buildings, number of students, some of them have their own power plants, while others do not, and these are just some of the factors that must be considered when creating an emissions inventory.

2.1.2 Challenges in Measuring Emissions

Due to this lack of uniformity in methodologies, many universities partly deviate from the established schemes and apply individual allocations which better suit their needs and contexts (Helmers et al., 2021). These adjustments present a difficulty in obtaining comparable results and lead to differences in scope, boundaries, emission sources, emission factors used, which are often study-specific. For instance, Dias and Arroja (2012) outline the differences in estimations for office paper between ISO14040, PAS 2050 and Confederation of European Paper Industries (CEPI) frameworks at 4.64g, 4.74g and 4.29g CO2e per A4 sheet respectively.

A significant challenge related to emission measurements is the accurate quantification of Scope 3. Ozawa-Meida et al. (2013) found that despite a compelling case for quantifying these emissions, up to 80% of a carbon footprint can be attributed to unreported indirect emissions. This gap is further supported by the findings of Vásquez et al. (2015), who evaluated Scope 3 emissions to be around 68% of total CF at the Curicó Campus of Talca University, and Gómez et al. (2016) stating that these emissions comprise up to 80% in Castilla-La Mancha University. One reason behind this exclusion is that guidance and organizations themselves often favour the emission sources for which data is readily available through documents and bills (Scope 1 and Scope 2) (Robinson et al., 2018). This is especially helpful in reducing the difficulty and costs of these calculations for universities (Valls-Val and Bovea, 2021).

On the other hand, indirect emissions, specifically from commuting and international travel, are considered challenging to acquire. As the travel data is often collected from surveys and questionnaires, it inevitably involves a degree of uncertainty due to its dependence on the credibility of the answers and the level of involvement in participants. That is why different assumptions need to be made (Letete et al., 2011). Most often metrics such as the number of trips students make per year, number of students on exchange programmes and shares of method of transport are assumed (Gaffron and Niemeier, 2015). These assumptions introduce even more disparities and thus lead to either overestimating or underestimating CF calculations and highlight the need for a common framework. This necessity is further supported by the growing student mobility, a topic elaborated upon in section 2.3.

2.1.3 Result Regularities and Patterns

Examining the CF measurements across various HEIs reveals intriguing patterns and regularities among the challenges of achieving consistency and comparability. Strikingly, studies consistently demonstrate significant disparities in carbon footprint results, emphasizing the inherent difficulty in carbon footprint analysis. While attempts to normalize results offer different perspective, disparities persist, highlighting the earlier-mentioned methodological challenges. Scope 3 emissions, related to student and staff travel, are the main focus as they represent a substantial source of greenhouse gas emissions in HEIs (Townsend and Barrett, 2015). Despite the increased awareness, studies such as those by Bailey and LaPoint (2016) and Townsend and Barrett (2015) emphasize incomplete data regarding this type of emissions, particularly related to commuting, study abroad, and air travel.

On the other hand, certain papers indicate that even though results are not as accurate as some may desire, they have a positive impact on the general awareness of emission sources and promote the idea that increased awareness can lead to better management of these emissions (Whitmarsh et al., 2011). Furthermore, some argue that the clear increase in total emissions may be a results of the recently started measurement of Scope 3 emissions (Brand and Preston, 2010).

Overall, these patterns underscore the necessity for ongoing research and collaboration to address methodological challenges and enhance the reliability and comparability of carbon footprint assessments in the higher education sector, especially in anticipation of the projected increase in international student enrollment (Davies and Dunk, 2015).

2.2 Existing Emission Calculators and Tools

This section delves into a detailed discussion of existing emission calculators, highlighting their respective advantages and disadvantages.

2.2.1 CO2UNV

CO2UNV (2022) is a tool specifically designed to calculate the carbon footprint of universities, which was developed by the Department of Mechanical Engineering and Construction at the Universitat Jaume I. It is based on the GHG Protocol (2011) and allows selection of emission sources in universities for all three scopes that are considered in the protocol. More specifically, the tool incorporates all Scope 1 and Scope 2 sources identified in a comprehensive analysis of 33 academic papers from various international universities, each detailing their methodologies for calculating carbon footprints. However, only the Scope 3 sources, which were considered in more than 40% of the studies, are covered by the tool. CO2UNV incorporates default emission factors that are sourced from official or governmental Spanish origins, but allows for user modifications and inclusions of new emission sources. An important aspect is the capability to compare carbon footprints from various years, particularly valuable for monitoring progress in reducing carbon footprint over time. Moreover, the results are presented both in table and graphical format which facilitates analysis and improves user experience.

Despite all of these advantages, the tool cannot serve a good purpose for measuring student travel emissions due to the fact that it only can calculate GHGs from commuting and business travel, as shown in Figure 2.1. Additionally, authors do not specify if the presented categories can be extended and if the tool is capable of creating whole new categories of emission sources. Another disadvantage, is the fact that the tool is currently a prototype in the form of an Excel file which does not enhance the user experience as required by the current project, so it cannot be considered as a viable option for addressing the specific problem described in this paper.

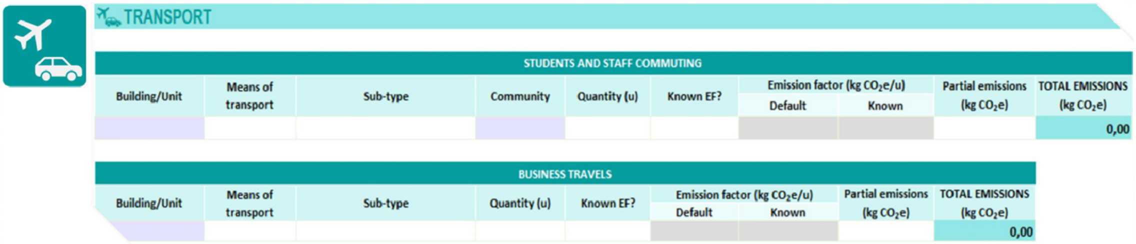

Figure 2.1: Transport Emissions included in CO2UNV tool (Valls-Val and Bovea, 2022)

2.2.2 SIMAP 2020

SIMAP is a carbon and nitrogen-accounting platform that can track, analyze, and improve campus-wide sustainability (Andrews and Leach, 2024). It was officially launched in 2017 but started as a master's thesis at the University of New Hampshire, USA in 2001. It collects, analyzes, and presents data on the emissions of greenhouse gases and losses of reactive nitrogen, attributed to institutional activities. Comparable to its Spanish counterpart CO2UNV, it utilizes the GHG Protocol (2011). However, unlike the CO2UNV it considers almost all known Scope 3 pollutants such as emissions from commuting of staff, faculty and students, business travel, study abroad air travel, student travel to/from permanent residence, etc. In addition, customization options for emission factors are available for all sources, albeit accessible only through payment for one of the higher tiers offered by the platform.

To calculate the carbon footprint of an educational institution, SIMAP requires descriptive information about the university's campus and data on what resources are consumed from the outlined scopes of the GHG protocol. The emission and loss factors are built into the system, but can be customized to suit the specific factors of the campus in question. The results can be filtered and presented as total footprint, or by scope, category, source, etc. There are numerous graph options and selection of year range, as shown in Figure 2.2. Notably, there is an option to export the greenhouse gas emissions inventory, the emission factors and the results (UNH Sustainability Institute, 2018).

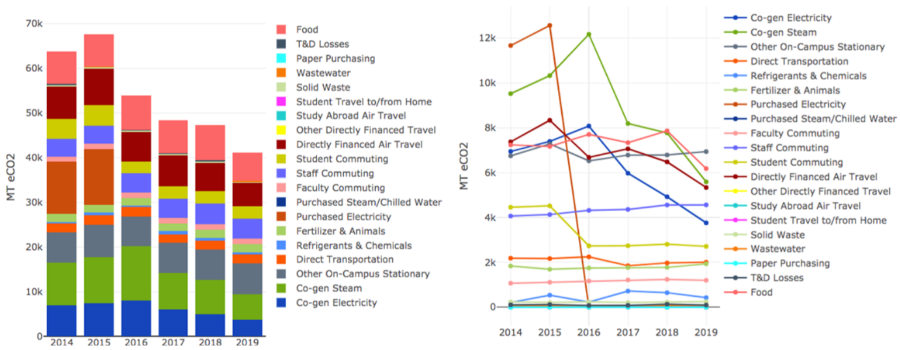

Figure 2.2: SIMAP Graphs (2024)

On the other hand, a disadvantage is that in order to utilize the complete Scope 3 emissions reporting functionality, the university must be subscribed to the Tier 2 plan of the system. Additionally, while the system provides customization of emission factors, it does not specify whether these factors can be tailored to align with European metrics, given that they are currently derived from US governmental sources. These drawbacks determine this tool as unsuitable for addressing the specific problem outlined in this paper.

2.2.3 Domestic and International Student Relocation Travel Emissions Calculator Tool

Following the introduction of its Aberdeen 2040 commitment to achieve net-zero carbon emissions by 2040, the Estates Department at the University of Aberdeen proceeded to develop its proprietary emissions calculation tool. The Domestic and International Student Relocation Travel Emissions Calculator Tool (DISRTECT), which is the currently available tool at the university, is solely concentrated on Scope 3 emissions from student travel between their home address and the university. This is attributed to the significant absence of these emissions from the institution's emission profile.

Currently, the tool relies on domicile and postcode data of students, collected by the Estates Department, to conduct the calculations. Regarding international students, it includes a list of United Nations (UN) member states along with countries provided by students from Registry. For UK students, Scotland is considered as one region, while the rest of the UK as another due to their separate tuition categories and the different travel methods that students from these regions utilize. The calculation methodology bases its computations on two trips per academic year, when the year starts and ends, and any potential travel by students outside of these journeys are considered the responsibility of the student. The instrument uses the capital city of the student's country of residence as the starting point location for each student trip. This choice is made for accuracy purposes, aiming to prevent over-counting of emissions. However, although this method is applied to both domestic and international students, it fails to depict the actual travel patterns, distances and means of transport of the domestic students (Jonsson, 2023).

The above-described limitations infer that the calculation of total emissions each year requires a level of assumptions and has some degree of errors. This is present in the two previously reviewed tools, but here, the reliability of the data is significantly affected compared to them. Moreover, the current state of the instrument does not allow for any assumptions in regards to the final leg of the journey - the trip between Aberdeen's bus/rail station or airport and the university. This is a key gap as these trips are also contributing to the overall carbon footprint (CF) of the students. In addition to these limitations, the calculator is currently housed within an Excel spreadsheet rather than having its own graphical user interface. This hinders its potential to become a nationally or internationally recognized standard in travel emissions calculation and to provide a more user-friendly experience. These are the main limitations of the tool that require addressing, as described earlier. Overcoming these limitations would present improved opportunities for identifying travel patterns and devising measures to reduce the carbon footprint of the university.

2.3 Rising Student Mobility and Its Environmental Implications

The mobility of international students has experienced significant growth over the past few decades, as highlighted by Kim and Zhang (2022). According to UNESCO data, the global student mobility rose from 2 to 6.36 million from 2000 to 2020. This surge in international travel among students coincides with a broader concern regarding transportation emissions, which account for approximately one-fifth of global carbon dioxide (CO_2) emissions (World Resource Institute, 2016). Anticipating the transport demand to escalate due to factors such as population growth, rising incomes, and increased accessibility to transportation modes like cars, trains, and flights, it becomes imperative to monitor and measure travel emissions (Ritchie, 2020).

The International Energy Agency (IEA) forecasts that the global transport (measured in passenger-kilometers) is expected to double, car ownership to increase by 60%, and demand for passenger aviation to triple by 2070 (IEA, 2020, p. 153-154). Combined, these factors would result in a large increase in transport emissions, considerable part of which could be attributed to domestic and international travel by students.

In the context of Scottish universities like the University of Aberdeen, there exists a legal obligation under the Universities and Colleges Climate Commitment for Scotland (UCCCfS) and the Public Bodies Climate Change Duties (PBCCD) to report annually on emissions through the PBCCD platform in November. However, emission tracking serves purposes beyond legal compliance. In the second chapter of their book "Carbon Footprint Analysis" Franchetti and Apul (2012) delineate four main categories of benefits an organization can derive from GHG minimization efforts: environmental, economic, corporate image and personal and social benefits. All of these benefits could be traced back to the Sustainable Development Goals (SDGs) which most universities use as a framework to achieve sustainability (Serafini et al., 2022).

English-speaking countries, particularly the US, the UK, Australia, and Canada, are of particular interest due to their status as leading destinations for international student recruitment (Choudaha and Chang, 2012). The United Kingdom is considered second to the US with an international student share of 11% of the global cohort of academically mobile students (Walker, 2014). Universities such as the University of Aberdeen, falling within this category, indirectly contribute to Scotland's overall emissions through student travel. To mitigate potential emissions growth with relation to increased student numbers and travel, universities must spread awareness among their communities, plan strategically, and promote sustainable methods of transport. However, achieving these objectives necessitates an accurate assessment of the current situation, which is done through gathering precise data.

Chapter 3

Design and System Architecture

This chapter provides and in-depth exploration of the requirements, design decisions, and system architecture of the StudentGreenTravel software. It offers insight into the design process, explaining the methods, considerations, challenges and limitations that shaped this process. Additionally, it provides an extensive description of the key methods and components of the system architecture.

3.1 Requirements

This section outlines the scenarios that illustrate potential user interactions with the software, followed by the functional and non-functional requirements derived from these scenarios.

3.1.1 Scenarios

The following scenarios are used to decide the functionality of the system:

1. A university staff member enters student data and travel assumptions to calculate travel distances and accumulated emissions.
2. A university staff member enters custom emission factors to be used in calculations.
3. A university staff member enters student data to find invalid data points.

3.1.2 Functional Requirements

These requirements are intended for both the user interface and the backend functions:

1. **Ability to input student postcodes dataset.**
   - The system should allow users to input a dataset with student postcodes in an Excel spreadsheets format (.xlsx).
   - The system should restrict input to this specific format to prevent from errors or incorrect values.
2. **Ability to input custom emission factors dataset.**
   - The system should allow users to input a custom dataset with emission factors for different methods of transport in an Excel spreadsheets format (.xlsx).
   - The system should have built-in (default) emission factors which the user could use in case there are no custom ones to input.
3. **Option for modifying number of trips that students make per year.**

- The system should have a pre-defined value for the number of study-related trips a student makes per year (e.g., 1).
- The user should be able to modify this value through an interface component.

4. **Options for modifying travel assumptions for the middle leg of the journey.**
   - The user should be able to modify the percentages related to the proportion of students travelling by each method of transport in relation to Scotland and Rest of the UK.
   - The system should implement validation rules and limitations to restrict invalid inputs (e.g., values exceeding or not reaching a certain limit).

5. **Options for modifying travel assumptions for the final leg of the journey.**
   - The user should be able to modify the percentages related to the proportion of students travelling by each method of transport in relation to all countries within the UK separately.
   - The user should be able to modify the percentages related to the proportion of students travelling by each method of transport in relation to the students residing in the same city as the university.
   - The system should implement validation rules and limitations to restrict invalid inputs (e.g., values exceeding or not reaching a certain limit).

6. **Calculation of travel distances and emissions based on postcodes (home addresses), mode of transport, number of trips per year, etc.**
   - The system should be able to calculate the travelled distance via each transport method.
   - The system should be able to calculate the corresponding emissions from these travels.
   - The system should be able to calculate the emissions per student in each country (the total emissions for the country divided by the number of students from the same country).
   - The software should be able to calculate these values both on country and council level using the travel assumptions and datasets.

7. **Generation of visualizations such as heatmaps, tables, etc. and travel routes.**
   - The software should display the results of the calculations in a structured and coherent manner, utilizing graphical representations such as heatmaps, tables, and charts.
   - The software should visualize possible travel routes.

8. **Identification of invalid data**
   - The software should be able to identify invalid data (e.g., a postcode missing a symbol).
   - The system should provide a list with the location of the invalid data points in the input dataset and the data point itself to the user.

9. **User-friendly interface for easy interaction.**
    - The system should provide a multitude of guides and tips to the user.
    - The software interface should be intuitive, meaning, it should be easy to understand and navigate, even for users with limited prior experience and technical expertise. This includes consistent layout, simple components, meaningful labels and predictable behaviour.

3.1.3 Non-Functional Requirements

The scenarios also lead to the introduction of the following non-functional requirements for the system:

- **Performance**: The software should be able to handle calculations efficiently, even with a large amount of data.
- **Usability**: The interface should be intuitive and easy to navigate for users with varying level of technical knowledge.
- **Reliability**: The software should produce consistent and accurate results.
- **Security**: The software should not store any sensitive data before, during or after the calculations are performed and should use anonymised data.
- **Scalability**: The performance of the software should not degrade significantly with a large amount of data.
- **Maintainability**: The software codebase should be well-structured, and documented to facilitate future updates and maintenance.

3.2 Design Methodology

The current section describes the strategies, methodologies, and tools employed in designing the application, including any associated challenges and limitations.

3.2.1 Planning

To plan and monitor the design process, a combination of Notion and Microsoft Excel was utilized. These tools offer a plethora of useful functions such as various templates and organizational methods, which proved crucial for synthesizing information. Additionally, they provide cloud storage which was vital for safeguarding against unexpected technical issues or data loss. With these tools, the initial schedule depicted in Figure 3.1 was devised. While this schedule was later adjusted to accommodate new additions, the general structure and major tasks remained the same.

Each task outlined in the Gantt chart was transferred to a list format in Notion, allowing for easy tracking as "To Do" or "Done" via check boxes. Moreover, a Notion diary, documenting daily tasks and progress, consisting of detailed descriptions of what and how was achieved, was produced. Furthermore, a dedicated notebook, with the information discussed with both my supervisor and the client, was maintained. This systematic approach allowed for better organization and seamless monitoring of progress, even in situations requiring alternation between tasks.

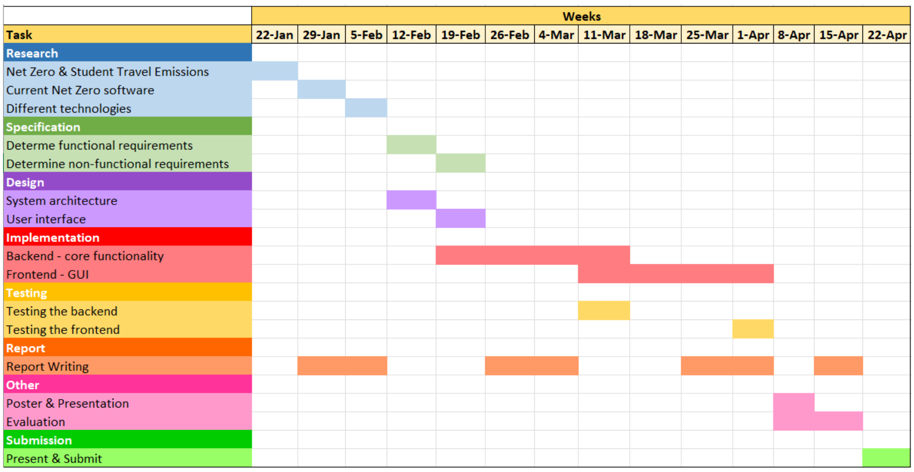

Figure 3.1: Gantt Chart of Project Plan

3.2.2 Process

Each of the eight major tasks, outlined in the project plan, excluding submission, were strategically combined with one or more other tasks to optimize time management and productivity. The rationale behind this strategy involved writing a part of the report after completing each task and thus minimizing the need for significant shifts of focus.

The initial phase of the project (late January - mid February) involved the researching, specifying and designing of the system along with regular report writing. A combination of scenario-based and user-centered methodology was chosen, emphasizing frequent interaction and feedback from the client (Abras et al., 2004). This methodology extended to the back end functionality as well. The way the distances and emissions were calculated, the emission factors, the preprocessing of the data, all of them were aligned with the intended user's preferences and desires.

Initially, the design process began with the gathering of user requirements (listed in sections 3.1.2 and 3.1.3) and later wireframes and prototypes were utilized to alter and further improve the software. The user requirements were compiled in Notion, while the wireframes were crafted using Miro. The prototypes were developed using PyQt6 and Python and they enabled the client to test them and provide immediate feedback, which facilitated refinement of the system. This iterative process allowed for increased usability, user satisfaction and engagement, and most importantly identifying and mitigating issues early in the development process.

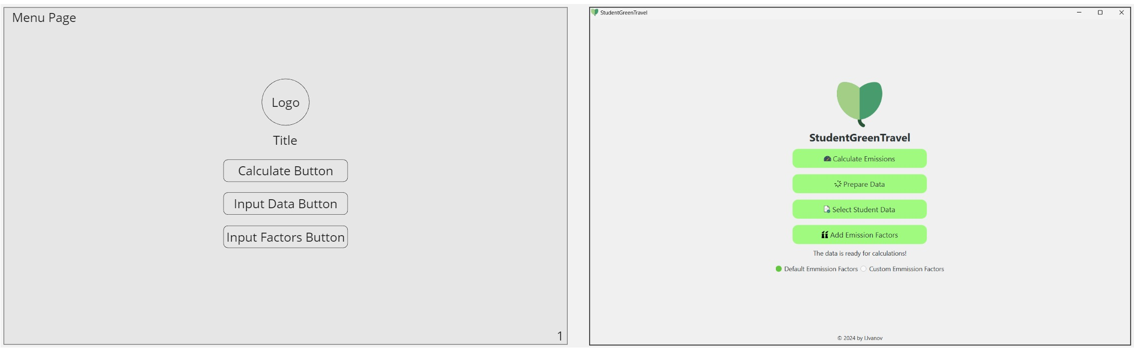

Figure 3.2: Menu Page Wireframe and Final Design

Through the system design phase, I accumulated experience in working with the PyQt6 application framework to create functional and visually-appealing interfaces. Its layouts and components turned out to be core elements of the user interface. My prior knowledge in CSS proved beneficial when styling the different components. Notably, I learnt that designing a complex and useful software requires time, effort and dedication, which became evident even at the beginning when the wireframes were designed.

These wireframes (shown in Figure 3.2 and 3.3 together with final design) were the result from the meetings with the intended user and the gathered intel from them. They represent the initial design choices and page layouts in the StudentGreenTravel software. The subsequent iterations incorporated client feedback and maintained the general design principles, while introducing further enhancements such as the tailored travel assumptions for Aberdeen and the addition of helpful tips and guides on every step of the calculation process.

3.2.3 Challenges

Throughout the various stages of the project, several challenges emerged, which significantly influenced the design decisions. For instance, acquiring and adapting certain datasets proved to be time-consuming, primarily due to the initial plan for an entirely offline system, as per the client's request. As a result, extensive filtering of large datasets was necessary to extract only relevant information (e.g., public airports only). While using offline datasets would have allowed users to operate the software in any circumstances, the challenge of obtaining some data as an offline resource led to the requirement for Internet access for certain functionalities.

Balancing research, design, and development with report writing required careful time management, especially managing the initial urge to prioritize development over foundational research and design.

Finally, adhering to the initial schedule was a challenge, due to the fact that throughout the design and implementation process, new functionalities and requirements were introduced which required more design and development time than initially anticipated. Such additions were the invalid data handling, and the display of distances and emissions by council area.

Nonetheless, with experimentation and testing of different approaches, libraries, and principles, these challenges were overcome, and a final design process procedure was established.

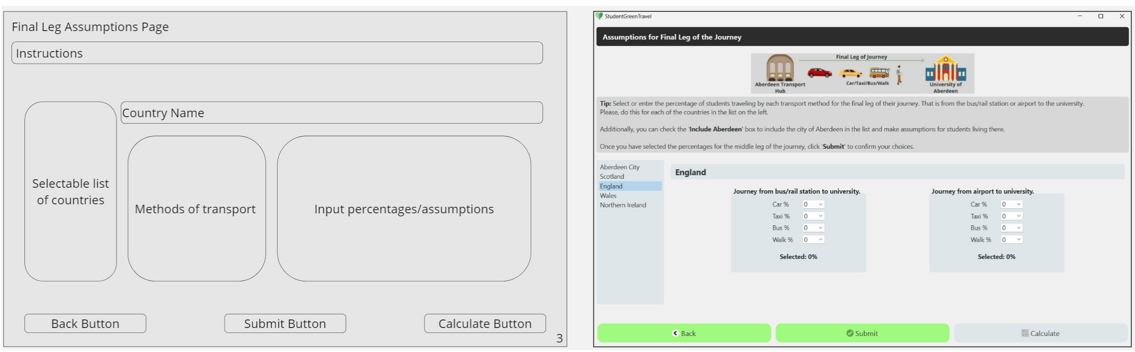

Figure 3.3: Final Leg Assumptions Page Wireframe and Final Design

3.2.4 Limitations

As with every task, there were some limitations. To ensure completion of the project within the set timeframe, the interface design focused on utilizing well-established PyQt components and layouts. However, this allowed for easier implementation and future improvements while maintaining intended functionality. Another limitation is related to the scalability of the project. This initial design focused on calculations related to UK students and if other student bases are requested, the system would need further development. User testing was also limited due to the busy schedules of the client and the limited number of users involved.

3.3 Design Decisions

This section explores the design choices made for the StudentGreenTravel application. These choices were guided by the goal of creating a user-friendly and effective tool for calculating travel emissions associated with student travel.

3.3.1 Student Groups

The StudentGreenTravel software is intended to calculate the travel emissions for students from the United Kingdom. These emissions are dependent on a number of factors. Starting location of the trip from home to Aberdeen and the methods of transport used are among the main factors. Moreover, all students, attending the considered university, contribute to its environmental impact. That is why the decision to handle Aberdeen, Scottish and non-Scottish students separately was made. The rationale behind this decision involved the fact that Aberdeen-based students are a great portion of the students enrolled in the University of Aberdeen. However, they do not travel from outside of the city as the rest of the students, but from their home address to university. Therefore, it was decided that they would be considered as a separate group.

The Scottish, or home students, were observed that they typically choose to travel from home to Aberdeen via bus, car, or train, while the non-Scottish students typically prefer plane, car, or train. These observations were made with the help of the data that the Domestic and International Student Relocation Emissions Calculator Tool utilized, which serves as a basis for this project. Thus, it was reached the conclusion that these would be the other student groups to consider. It is important to note that the non-Scottish students are considered all English, Welsh, and Northern Irish nationals.

3.3.2 Journeys

A journey is the travel from the student's home address to the University of Aberdeen and is the key element that emission calculations would be based on. To calculate emissions accurately, there would need to be an accurate measurement of the distance travelled through the journey and the methods of transport would need to be known. This stems from the fact that in the different stages of each travel, people utilize different transport modes. For instance, people might use a taxi to get to the airport, then board a plane, and finally get to their final destination by bus. These different vehicles produce different amounts of emissions. Thus, it was reached the conclusion that the different student groups might chose different modes of transport, which best suit their context and needs. Therefore, it was decided that the travel assumptions for these groups would be entered by the user. Additionally, each journey would be divided into three distinct stages, called legs, and there would be different methods of transport available for each leg. The three stages of each journey are:

Initial Leg

This refers to the trip from the student's residence to the nearest transportation hub. This transportation hub could be an airport or a bus/railway station, depending on the method of transport that the student would use in the next leg of the journey. Usually, for each part of the journey, travel assumptions, related to the proportion of students travelling by different modes of transport, would have to be entered but here due to the client's request, it was decided that the following assumptions would be incorporated as default:

- 40% of students travel by car.
- 40% of students travel by taxi.
- 20% of students travel by bus.

Middle Leg

This segment of the journey would incorporate several considerations. First is the limited availability of direct flights from most UK cities to Aberdeen, which prompted the decision to introduce a layover location for non-Scottish students travelling by air. London Gatwick airport was chosen as layover location, which was also used in the previous tool and proved sufficient due to it offering direct flights to Aberdeen. After arriving at this location, the group of students in question would fly to Aberdeen and thus their middle leg would be from the local airport through the layover airport to Aberdeen airport.

For the rest of the students, who would choose other transportation options, it was decided that the middle leg would be a trip from the local station to Aberdeen station, regardless of them choosing to travel by bus or train, as illustrated in Figure 3.4.

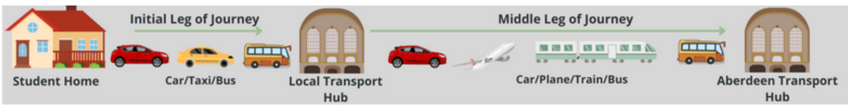

Figure 3.4: Visualization of the Middle Leg of the Journey

It is important to note that in the student data that the university utilized in the previous tool, and which is the base for the current decisions, there is an observed difference in preferences for transport methods between Scottish and non-Scottish students. Therefore, it was decided that transport methods available for the two student groups would be as follows:

- For Scottish students: bus, car, and train.
- For non-Scottish students: plane, car, and train.

The assumptions on number of students using each of the vehicles would be entered manually by the user.

It is worth mentioning that students who choose to travel by car were decided to be handled separately as their journey would not consist of legs. They are assumed to travel directly from their home address to the university.

Final Leg

This is the final stage of the journey which consists of travelling from the Aberdeen transport hub to the University of Aberdeen, as shown in Figure 3.5. Considering the available transport methods within Aberdeen, it was decided that during this stage the students would have car, taxi, bus or walking as transport options, regardless of them starting from the station or the airport. However, due to the previously described design decisions, Scottish students would start the final leg from the bus/railway station, while non-Scottish students would be divided into ones that travelled by air and ones that travelled by land in the previous leg. This stems from the fact that they are starting from the airport and station accordingly and the distances between these two locations and the university are different.

Importantly, the students residing in Aberdeen would also be considered as part of this leg. The only difference being that their final leg would be their only one and would be starting from their home address instead of a transport hub.

Again, the assumptions on number of students using each of the vehicles would be entered manually by the user.

Figure 3.5: Visualization of the Finale Leg of the Journey

3.3.3 Distances

As mentioned previously, the calculation of emissions requires the accurate measurement of the travelled distance via the designated vehicle. To achieve this, I decided that the distances traveled by students in each group using various modes of transportation on different legs of the journey would be added together to calculate the total distance traveled by that vehicle for the group. To be able to do this, I would need to capture the distance that each student travelled in each of the legs. For this, I would need to know the start and final location for each leg and be able to compute the distance between them. After a detailed research, I reached the conclusion that the easiest and most accurate way to do this is through the geographical coordinates of these locations, which would be an input to a Python library that would calculate the distance between the locations through their coordinates. Following this line of thought, I obtained data on UK postcodes, airports, railway stations and bus stations all along with their geographical coordinates. These datasets would allow me to perform the described computations for all legs of the journey.

3.3.4 Number of Journeys

As the design process of the system progressed, it became clear that the amount of travel that students perform is also a key factor. Due to the fact that students often travel back home or to other locations throughout the academic year, it was important to distinguish the study-related trips to the other trips. The study-related trips are the ones that students perform so that they can access the HEI and then leave at the end of the academic year. This led to the decision that as a minimum 2 trips are performed by each group for education purposes. Furthermore, it was observed the number of trips that the system would consider, would in turn affect the total distances and emissions as well. Therefore, this would be an essential component to incorporate.

3.3.5 Emissions

Once the total distances, vehicle types, and number of journeys were clear, the mechanism and tools for calculating the emissions would have to be devised. To achieve this, the methodology outlined in section 2.1.1 was employed. It consisted of multiplying the total distances for each vehicle and student group by the corresponding emission factor. Thus, the decision to obtain and incorporate emission factors into the system was made.

3.3.6 Invalid Data

From my experience working with human-gathered data in other courses, I knew that there is always room for error. Whether it is misspelling, wrong input, capitalization of symbols or something else, there is always something imperfect present. With this in mind, I concluded that the software would need to incorporate mechanisms for discovering and handling such invalid data points and inform the user about them.

3.3.7 Data Security

The student data collected and used by the University of Aberdeen for the journeys and emission calculations, as involved in the previous tool, is completely anonymized. This is due to General Data Protection Regulation (GDPR) policies being in place at the university. As a result, I would not need to implement mechanisms for encrypting or hashing the data when utilizing it as a foundation for this application. For this reason, the application would be tailored to operate exclusively with postcodes, complying with this established methodology.

3.3.8 User-Friendly Interface

Considering one of the main objectives of this project is to make the new emissions calculator more user-friendly, understanding user behaviour is essential, as highlighted by Kusano et al. (2013). To achieve this, several meetings with the client were conducted and the previously outlined scenarios, which describe how a user accomplishes tasks (Rosson and Carroll, 2000), were devised. With their help it was easier to identify the core functions of the UI. Then, it was proceeded to developing wireframes based on these initial user needs. They are used to capture the type and layout information of visual components but ignore their high-fidelity visual details (Chen et al., 2020).

The user interface design process was iterative and followed a combination of scenario-based and user-centered approach. Throughout this process, I had to come up with a way to distinguish the different functionalities of the system and incorporate useful guides through every step of the process. This was also requested by the intended user, so it was decided that the system would consist of different “pages”, each of which would be responsible for a different functionality. Furthermore, to prevent the user from inputting invalid values or such exceeding certain threshold, I decided that it would be best to make these pages dependent on one another and incorporate mechanisms for informing the user whether the input is correct or not. What this means is that each page would be accessible only if the input in the previous one is correct. This would further enhance the user experience and comply with the client’s requirements.

Drawing inspiration from online tools such as the "Carbon Footprint Calculator for Travel" of Sustainable Travel International and carbonfootprint.com's calculator, along with SIMAP tool, the software's design aims to align with industry standards and fulfill client expectations.

3.4 System Architecture

The system architecture follows the three-layered application architecture and consists of a Data, Back End and Front End layers. This approach separates the concerns of presentation, processing, and data storage, which makes the system more maintainable. Each of these layers interacts with and relies on the others, as depicted in Figure 3.6.

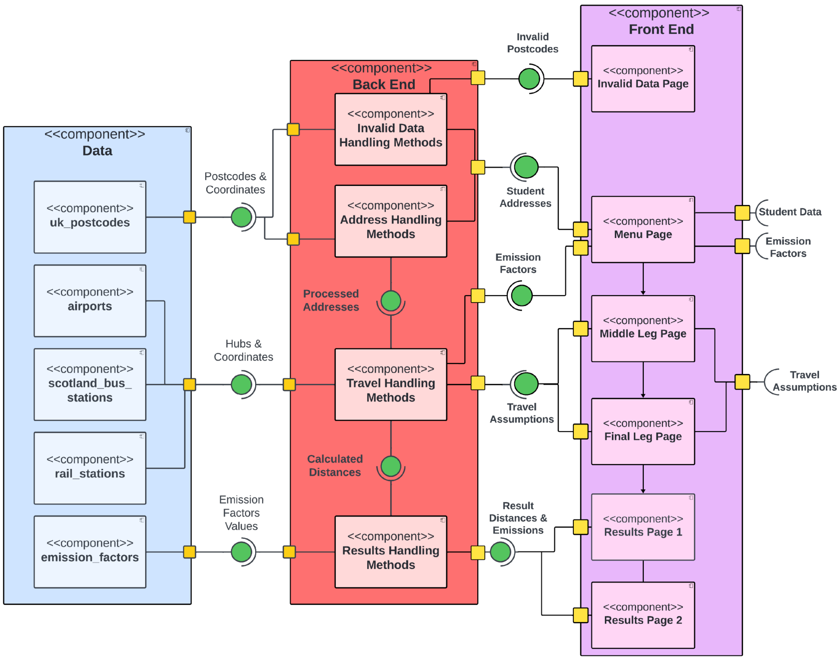

Figure 3.6: UML Diagram of System Architecture

3.4.1 Data

The Data layer consists of several data components, each of which is responsible for different data that the system utilizes. The uk_postcodes holds UK postcodes along with their longitude and latitude. The airports, scotland_bus_stations and rail_stations components hold the names of these transport hubs along with their geographical coordinates. These four components are essential for the distance calculations as described previously. Additionally, the Data module holds the emission_factors component where these values for different vehicles reside. These are used in the calculation of emissions and are built into the system instead of residing in a separate file.

3.4.2 Back End

The Back End of the system is related to the background functionality of the system. It consists of four distinct components:

Address Handling

This component deals with the described separation of student addresses. It is linked with the uk_postcodes Data element and performs checks on each user-entered postcode to find its country of origin.

Travel Handling

Travel handling is the calculation of distances between the start and final locations of each leg of the journey. It employs the uk_postcodes, airports, scotland_bus_stations and rail_stations Data components and the user-entered assumptions to find the corresponding distances.

Result Handling

This element of the Back End handles the computed distances from the previous component. It derives the total distances for each vehicle utilized by students from different countries. Furthermore, it computes the emissions associated with these vehicles and formats all of this data in preparation for visualising it.

Invalid Data Handling

The current component collects invalid data points from the user-entered dataset. It achieves this through the utilization of the uk_postcodes data and an online Application Programming Interface (API). The output of this component is a list with data points that are either invalid or erroneous.

3.4.3 Front End

The Front End consists of six distinct components (pages), each incorporating its own set of elements adapted for various user inputs and interactions. The menu page is related to the address handling and invalid data handling and employs the mechanisms that these Back End components contain. The middle leg page holds the User Interface (UI) components responsible for gathering the user input for this stage of the journey, while the final leg page holds the ones for the last stage of the journey. The two result pages are responsible for presenting the obtained results in an appropriate format and with visualizations to the user. Finally, the invalid page presents the invalid data points to the user.

Chapter 4

Implementation

This chapter describes the development process of the software, outlining the utilized tools and technologies. Furthermore, it provides a comprehensive overview of the core functionalities inherent in each of the system components.

4.1 Development Process

The StudentGreenTravel project involved developing a software tool that calculates student travel emissions based on their home addresses. It followed a sequential workflow, with each of the stages - data gathering, software backend development and software frontend development, building upon the previous one.

During the implementation phase of the project, the well-known methodology of Agile development approach was followed. It relies on continuous and iterative development with constant feedback from the stakeholders (Nerur and Balijepally, 2007). This methodology was aligned with the user-centered design approach.

To manage the development process, the Notion application was utilized to track the two-week sprints, tasks scheduled within the sprints and tasks completed during the sprints. At the end of each period, the progress of the project, measured in completed tasks and implemented features, was reported to the client and immediate feedback was gathered. This feedback helped meeting all of the requirements.

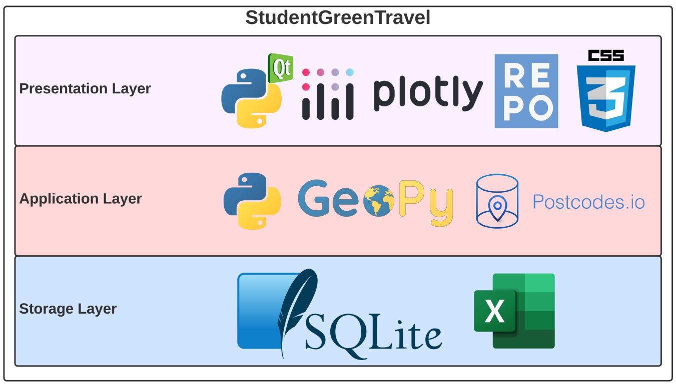

Figure 4.1: System Layers

Throughout the development process, the system evolved from a basic Python program utilizing simple data files in CSV format and printing resutls in the terminal, through an application with basic interface containing only base functionalities and having no styling, to finally reaching to a sophisticated software with three distinct layers: Storage layer (Data), Application layer (Back End) and Presentation layer (GUI or Front End), as illustrated in Figure 4.1. This division was designed to shield the user from the technicalities that the underlying layers (Storage and Application) present. Users interact solely with the Presentation layer, which as described earlier is connected to the other two.

4.2 Version Control and IDE

For the implementation of the system an Integrated Development Environment (IDE) and a Version Control system were used. The chosen IDE was Microsoft VS Code and the Version Control System was Git through GitHub Desktop application. These ensured the seamless development and safeguard against unexpected technical issues. Furthermore, a branching workflow was implemented. This means that each new feature of the software was developed in a new branch, while the main codebase of the product was kept on a separate one. This ensured code organization, facilitated experimentation, and mitigated the risk of losing progress or getting crucial files or components corrupted.

4.3 Development Technologies

4.3.1 Programming Languages

Python served as the main programming language for the project, with version 3.10.6 being utilized. The choice of Python stemmed from its flexibility and various libraries which ease development and data management (Saabith et al., 2019). This programming language was employed for developing all of the back end and front end functionalities.

Apart from base functionalities, the software incorporates styled GUI components and database queries. For styling, CSS was utilized in combination with PyQt components. The database queries were executed via SQL.

4.3.2 Libraries

The project utilized various libraries for different purposes. Beginning with the data components of the application, the `Numpy` and `Pandas` libraries were instrumental in reading and manipulating datasets, providing numerical calculation functionalities and dataframe creations. Additionally, the `itertools` library proved invaluable in dividing and arranging different data structures such as lists and dictionaries.

Two crucial libraries for the system proved to be GeoPy and PyQt6. GeoPy provided the opportunity to calculate distances for the various legs of journeys. It utilizes the WGS-84 ellipsoid model to calculate geodesic distance, which represents the shortest distance on the surface of an ellipsoidal model of the Earth (Karney, 2013). This model is globally recognized as the most accurate (Boucher and Altamimi, 2001).

PyQt6, was used to create the user interface. The reason to chose PyQt6 instead of alternatives like Tkinter is related to the fact that PyQt6 is more suitable and has more opportunities for data display, which is crucial for the project. Furthermore, the standard layouts and components eased development and allowed the integration of the SVG Repo's icons that enhanced the aesthetic appeal of the system.

Another crucial library is plotly in conjunction with PyQt6's QWebEngineView component, which enabled the production of interactive and publication-quality graphs depicting the resutls.

4.3.3 APIs

The "Postcodes.io" API played a vital role in the backup validation of user-provided postcodes and the extraction of necessary data such as the geographical coordinates and council area of every postcode. Python's requests library was employed to obtain responses from the API and use them accordingly.

4.3.4 Packaging Tool

The pyinstaller tool was utilized to bundle all the necessary modules and files into a standalone executable, which ensured that the software can be distributed and run on other computers without requiring a Python interpreter or any other dependencies. The reason to chose this tool stemmed from my previous experience with packaging code and applications with it.

4.4 Data and Database

This layer of the application consists of a database, holding more than 1.7 million UK postcodes along with their coordinates, sourced from the National Statistics Postcode Lookup (NSPL) which relies on Ordnance Survey data (2024). The NSPL dataset encompasses various statistics for UK postcodes but has been filtered down to the essential columns, namely postcode, longitude, and latitude.

The database utilizes the well-known and lightweight SQLite system. It is organized in a single table with designated columns for the postcodes, longitudes and latitudes. Each postcode, serving as the primary key, is stored as a string with all whitespace characters removed, while the longitude and latitude are real numbers. Additionally, the database was "vacuumed" with the corresponding Structured Query Language (SQL) query. This process streamlined the database structure, resulting in a more compact size.

The reasons to chose SQLite as a main database are several. Namely, it does not require a separate server process, which not only aligned with the initial requirement of making a fully-offline system, but it also translated to lower resource usage. Next, its simpler architecture allowed for easy creation of a database and faster data access and manipulation. These factors make it very suitable for local storage on devices as is the case with this application.

Apart from the database, there are three separate CSV files with transportation hubs and their coordinates. There is one with the bus stations in Scotland, containing data collected from Google Maps. A railway stations set with data from the Trainline EU's repository and an airports set, generated using the Python airportsdata library, which was filtered to contain UK's public airports only.

The reason to leave this data in CSV files is that these sets do not contain that many data points and creating a dedicated database would not introduce any performance or resource benefits. Moreover, this way, the data related to transportation hubs (airports, bus and train stations) is more accessible and easily modifiable, which is crucial for the project's future development. The data within these files was accessed through the Pandas' read_csv() method and then transformed into a dataframe.

Finally, the system incorporates default emission factors directly instead of relying on a separate data file. They contain values for each method of transport used in the calculations and can be overwritten with custom ones, if the user decides. These emission factors were derived from the Greenhouse Gas Reporting: Conversion Factors 2023 published on gov.uk.

4.5 Back End

The Back End is based on Python and all of the external libraries, methods and APIs are Python-based or Python-compatible. Sources for them are provided within the code. The different functionalities and classes are accommodated in separate files to facilitate code granularity and maintainability. These are described in detail in the next sections.

4.5.1 Address Handling

The address handling part is contained within the `preprocess_data.py` file and relies on `math`, `pandas` and `itertools` libraries. First, the data files related to the transportation hubs are read and a dictionary for each hub, containing its name as key and a tuple with the corresponding coordinates as value, is created. Additionally, the user input is cleared of white spaces. Then, the `determine_postcode()` method determines the country of each user-entered postcode as it checks it against lists containing the first two letters of each United Kingdom (UK) country postcode, as shown in Figure 4.2.

Figure 4.2: Code Snippet of `determine_postcode()`

There is also an additional check performed in the `find_country()` method through the "Postcodes.io" API. This interface provides detailed data on a single or a bulk of 100 postcodes. Its purpose is to identify the countries of the postcodes that could not be identified during the first check.

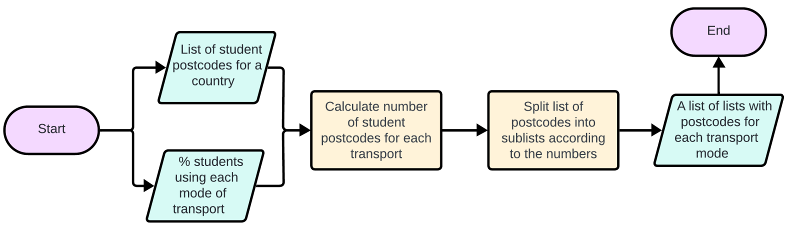

Figure 4.3: Flowchart of `divide_scotland()` and `divide_uk()` operation

Finally, the `divide_scotland()` and `divide_uk()` functions were developed, which handle Scottish and non-Scottish postcodes separately as of Chapter 1 section 1.3 and Chapter 3 section 3.3.1. These methods operate similarly by taking the output of the `determine_postcode()` method. For each postcode in the user-entered dataset, `determine_postcode()` rules out the country associated with this postcode and places it in the corresponding country list. Then, the divide methods use these lists together with the travel assumptions made for Scotland and the rest of the UK (relevant to Chapter 3 section 3.3.2). Thus, the number of students for each available transport method is calculated. Finally, these functions produce a list of lists containing the postcodes for each transport method and return them, as shown in Figure 4.3.

4.5.2 Travel Handling

A crucial component of the system is the `Travel` class, which accommodates all methods for distance calculations for the initial and middle leg of each student's journey. It stores student postcodes and receives the following parameters:

- bus stops in Scotland along with their coordinates.
- railway stations in the UK with their coordinates.
- airports in the UK with their coordinates.
- UK postcodes along with their coordinates.

Figure 4.4: Code Snippet of `air_travel()` function

There is a `calculate_distance()` function responsible for calculating the distance between a single point and a set of other points using GeoPy library. This distance is calculated in kilometers and considers the shortest distance on the surface of the Earth, similar to real-world travel routes.

A `closest_hub()` function considers a set of UK postcodes along with their coordinates and a set of transport hubs and their coordinates. By relying on the `calculate_distance()`, it finds the distance between the given postcode and all the transport hubs, corresponding to the selected transport. Then, it selects the closest transport hub and returns its name along with the distance to it.

The `find_coordinates()` method relies on the `requests` library and the "Postcodes.io" API to obtain the geographical coordinates of a postcode that is not found in the database.

The `air_travel()` function utilizes the airports data and the user-entered student addresses, as shown in Figure 4.4. It begins with initializing the result dictionary and the list of invalid data points. Subsequently, it connects to the UK postcodes database to obtain the geographical coordinates of each student postcode.

The main for loop goes through these postcodes. Within this loop, conditional statements handle the calculations of appropriate middle leg distances. The first conditional handles the students travelling from outside of London. For these students, the middle leg involves two stages: travel from the local airport to the layover (London Gatwick), and then from the layover to Aberdeen airport.

Conversely, the other conditional is related to the London-based students. For them, the function calculates the distance between the local airport and Aberdeen airport directly.

Finally, the postcode is set as the key of the result dictionary and the name of the closest airport, the distance to it and the travel distance for the middle leg are set as the values. If none of these conditions are satisfied, the postcode is added to the list of invalid data points.

Figure 4.5: Code Snippet of `land_travel()` main loop

Operating similarly to the previous method, `land_travel()` is used for both bus and train travel. This is due to the fact that the bus and railway station in Aberdeen are situated in the same location, and unfortunately, a method for considering train routes could not be identified. Hence, this function calculates both bus and train travel distances through the established method with the only difference being the conditional which checks if the closest station is Aberdeen, as shown in Figure 4.5.

Distinct from the other travel methods, `car_travel()` computes distances by utilizing student addresses and the postcodes database. Instead of looking for closest transport hubs and the distances to them, it directly finds the distance between the student's home address (postcode) and the University of Aberdeen, simulating a real-world car journey, as depicted in Figure 4.6.

Figure 4.6: Code Snippet of `car_travel()` function

Notably, all of these functions, take into account invalid postcodes, which are stored in a separate list and later used for the invalid data page. They utilize the `find_coordinates()` method to capture as many postcodes as possible during the calculations.

In terms of calculating travel distances for the final leg of students' journeys, there are two distinct functions: `assign_scotland()` and `assign_uk()`.

The `assign_scotland()` function handles Scottish students. It assigns them to the mode of transport they use to travel from Aberdeen station to the University of Aberdeen. As previously stated, Scottish students typically use buses and trains for their middle leg transportation. Consequently, their final leg distance is the distance between the Union Square Station and the University of Aberdeen (around 3 km).

On the other hand, the `assign_uk()` function handles students from England, Wales and Northern Ireland. Similarly, it assigns them to the modes of transport and calculates their final leg distance according to their starting position for this leg - Aberdeen airport or station.

Both of these functions rely on the `fleg_assumptions()` method to calculate the total distances travelled by students for each mode of transport. These distances depend on the starting point for the final leg and the user-entered percentages for the amount of students travelling by each method. The flow of operation of this function is illustrated in Figure 4.7.

Additionally, the `aberdeen.py` file handles the Aberdeen-based students separately, as

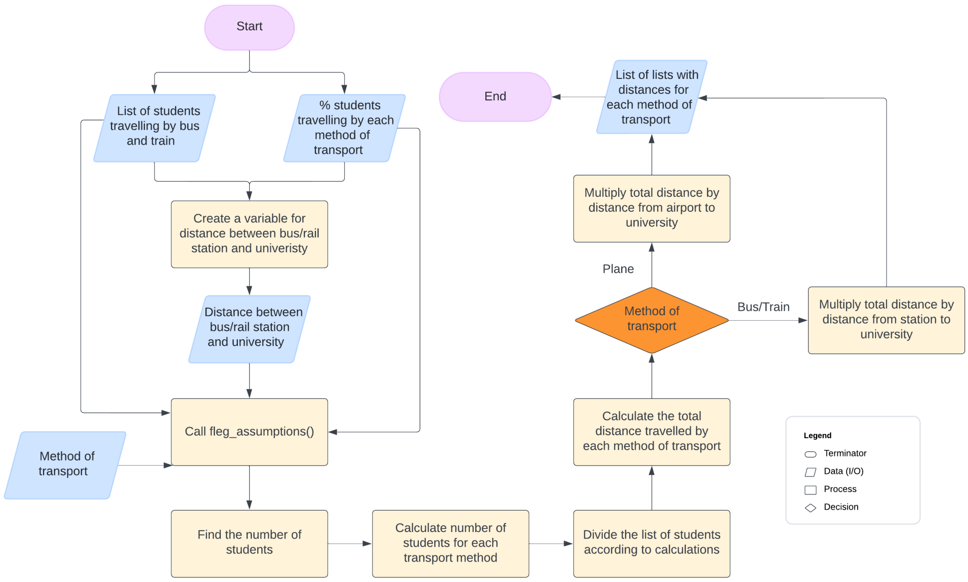

Figure 4.7: `assign_scotland()` and `assign_uk()` operation

outlined in the design decisions. Within this file reside the methods that are responsible for calculating the distance from each postcode to the university through the GeoPy library.

4.5.3 Result Handling

The resutls of the distance calculations are managed by the `main.py` file, along with some of the methods within the `utils.py` file. They obtain the corresponding emissions for these distances and format the travelled distances and emissions appropriately for display to the user.

4.5.4 Invalid Data Handling

The invalid data points are collected throughout the execution of the address and travel methods. These methods have incorporated invalid data collection lists, which are used to cross-reference such data points with the input dataset and present them to the user. The presentation, performed by the Front End, includes displaying the location of each invalid data point within the dataset. This approach enables users to easily identify and address any erroneous or incomplete data entries.

4.6 Front End

The Front End consists of six distinct `QWidgets`, each incorporating its own set of components, adapted for various user inputs and interactions. Each of them is accommodated in a separate class and file. The `Calculator` class assigns the other classes to a `QStackedLayout` to achieve the effect of pagination. Furthermore, this class holds the common for all pages variables and methods. From now the `QWidgets` would be referred as pages.

The user journey begins with the initial menu page, where they input a student dataset, consisting of postcodes, and have the option to upload custom emission factors. Although the Graphical User Interface (GUI) is based on PyQt6, for these purposes Tkinter's file dialog functionality was utilized due to easier implementation. Subsequently, the initial address handling is performed

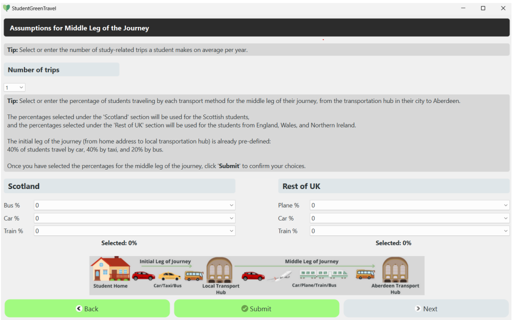

Figure 4.8: Middle Leg page

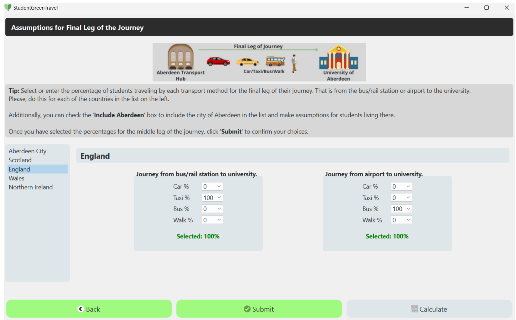

Figure 4.9: Final Leg page

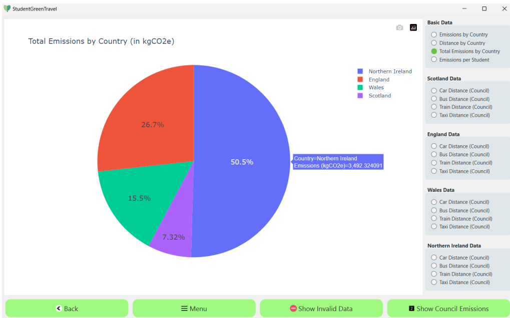

Figure 4.10: Results page

by the Back End functions. Upon completion of address handling, the user is informed that the data is ready for calculations. Thus, the next page becomes accessible, and the user can proceed with entering the number of trips that students perform per year and the travel assumptions for the middle leg of the journey. These inputs are entered through `QComboBox` components.

Upon accurate input of the travel assumptions, the user can proceed to the next page and input the assumptions for the final leg of the journey. Here, the data provided from the university allowed me to implement the option for the user to enter travel assumptions for each of the countries within the UK and Aberdeen, instead of only for Scotland and the rest of the United Kingdom.

Once these assumptions are submitted to the system, the user can view the results. They are accommodated in two separate pages which display base data, such as total distances and emissions for countries, and council area data for distances and emissions. The council data details distances and emissions accumulated by students in various councils of the countries in the United Kingdom. This functionality is achieved through the utilization of `QWebEngineView` component in combination with the `plotly` graphs library. The layouts of the above-described pages are shown in Figures 4.8, 4.9 and 4.10.

Finally, the user can access the invalid data page, which shows the list of data points and their location in the initial dataset, as depicted in Figure 4.11. This page contains a simple `QList-Widget` along with `QButtons` for showing and clearing the invalid list.

4.6.1 Styling of Components

The PyQt6 library components utilize the so-called "style sheets" to customize their appearance. These sheets hold CSS code, which enables the setting of different attributes such as color, font, size, padding, and more. In the current project, this approach has been adopted, with most of the style sheets for the components housed within the separate `style_sheets.py` file. This approach was implemented to enhance code readability and separation of concerns.

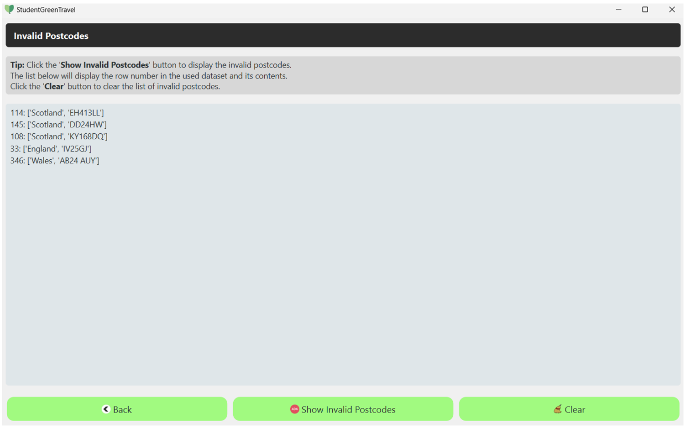

Figure 4.11: Invalid Data page

4.6.2 Calculation Results Display

To facilitate data analysis and enhance user experience, the results of the calculations are displayed in graphical formats. Initially, the plan was to present the base data via heatmaps and pie charts, while more detailed council data was to be presented using bar charts. However, due to England's numerous council areas, the results were almost unreadable this way, so it was decided to display this country's findings in a table format.

Additionally, the distances and emissions related to Scotland include those for Aberdeen, aligning with the client's request and considering Aberdeen as a city in Scotland where student emissions contribute to Scotland's overall data. However, separate distances and emissions specific to Aberdeen can be seen as a distinct bar within the Scottish bar charts related to council areas.

4.7 Packaging

Due to the nature of the project, packaging the application was required. This was achieved with the `pyinstaller` tool, as mentioned earlier.

First, I ran the command shown in Figure 4.12 in my terminal, which generated both a standalone executable and a .spec file. This file contains configurations for the packaging process, including instructions on what components such as data files, icons and pictures to include.

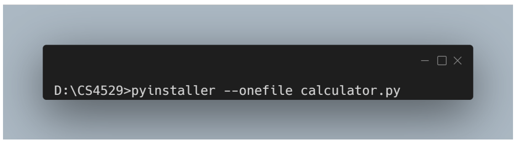

Figure 4.12: Command for Packing as "onefile"

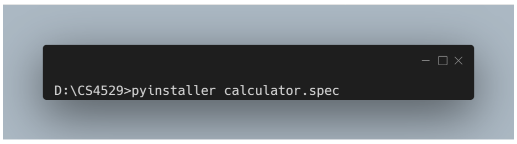

Figure 4.13: Command for Packaging through .spec file

Later, whenever a new feature was added or changes were made to the application, the .spec file was utilized to re-package the application. The terminal command is shown in Figure 4.13. This approach ensured that the standalone executable remained up-to-date with the latest changes and included all necessary components for proper operation.

4.8 Summary

In summary, considering all of the described implementations, the application fulfills the scenarios and meets most of the requirements that were identified, apart from the visualization of travel routes due to time limitations. However, in order to meet the requirements related to the accuracy and performance of the system, testing and evaluation had to be performed. These are discussed in the following chapter.

Chapter 5

Testing and Evaluation

This chapter delves into the comprehensive testing of the software, including the Back End, system-wide, and comparative testing. It outlines the methodologies employed and discusses encountered issues and limitations. Additionally, it goes into detail about the evaluation process, its resutls and limitations.

5.1 Testing

5.1.1 Back End Testing

Methodology and Results

This phase of testing included assessing the functionality of the Back End to ensure it operates as intended. To accomplish this, the Python `unittest` framework was employed, facilitating the testing of individual units of code, specifically, separate functions, to verify their expected output.

The unit tests are accommodated in a separate Python file called `test_code.py`. It consists of four classes that test the methods of other files. Each class consists of a `setUp()` method, as per the `unittest` principles, along with methods for assessing the Back End functions. These methods utilize test inputs, composed of random postcodes and values, to cover various scenarios both with valid and invalid data.

```python
def test_land_travel_train(self):
    # Test the land_travel method with train travel
    # Random Scottish postcode, generated using https://www.doogal.co.uk/PostcodeGenerator
    train_postcode = ['G42 9BJ']
    aberdeen_rail_station = (57.14372498503623, -2.0983252205325833)

    rail_stations, invalid = self.travel.land_travel(stations_dict, train_postcode, aberdeen_rail_station)
    # List comprehension to hold only the values of the rail_stations dictionary
    rail_result = [value for key, values in rail_stations.items() for value in values]
    # From Google Maps, the closest rail station to the postcode is Mount Florida and is around 300 metres
    # From Google Maps, the distance between Mount Florida and Aberdeen Rail Station is around 197 km
    self.assertEqual(rail_result[0], 'Mount Florida Railway Station')
    # Using math.isclose() and assertTrue() to check the distances
    self.assertTrue(math.isclose(rail_result[1], 0.3, rel_tol=0.1))
    self.assertTrue(math.isclose(rail_result[2], 197, rel_tol=1))
    self.assertEqual(invalid, [])
```

Figure 5.1: Code Snippet of testing the `land_travel()` method

In total, 13 tests were executed and all of them confirmed that the Back End functions as expected. Notably, the only file related to the background operations that has not been tested is `final_leg.py`. This stems from the fact that the methods within this file are almost identical to the ones in `preprocess_data.py` and testing them was considered a redundancy. Additionally, testing on emission calculation was not performed, due to its dependency on distances.

The rationale behind this decision was that correct distances, when multiplied by emission factors, would yield accurate emission results.

Testing procedures employed the `assertEqual()`, `assertTrue()`, and `assertAlmostEqual()` methods from the `unittest` framework in combination with some functionalities from Python's `math` module, as shown in Figure 5.1.

To run the tests, the `unittest.main()` command was utilized at the end of the file. This command automatically executes the tests and provides either `'FAILED (failures = number)'`, along with information on which test failed or `'OK'` if all tests are passed. In the current case, after careful preparation of tests and extraction of information on expected distances and postcodes from different sources (listed in source code), I obtained the `'OK'` response for all of my tests, as depicted in Figure 5.2. Therefore, it can be concluded that the Back End of the system works as expected.

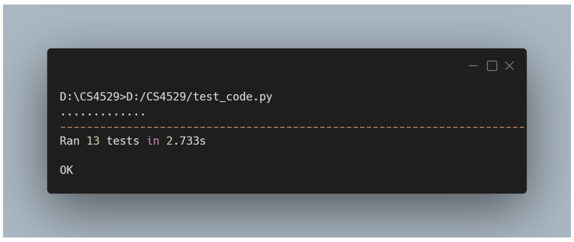

Figure 5.2: Results from unit tests

Known Issues

Although the system functions as expected, there are some known issues. One of them involves occasional discrepancies between the distances measured by StudentGreenTravel and those obtained from Google Maps, for example. However, these discrepancies are not deemed too significant during Back End testing and play testing.

Additionally, there is an identified issue with the bus stations dataset, specifically concerning the absence of primary bus stations in certain Scottish cities. This flaw occasionally impacts distance calculations for this mode of transport.

5.1.2 System Testing

Methodology and Results

The comprehensive testing of the entire system followed a play testing methodology, carried out across various development stages and utilizing different tools. This approach aimed to assess the functionality of the Data, Back End and Front End components in scenarios with valid and invalid data or inputs. By employing this method, early detection and resolution of usability issues and edge cases were facilitated.

First, the system was rigorously tested on my laptop, operating on Windows 11. Testing included running the application both from VS Code and as a packaged application, checking every PyQt6 component, along with its underlying logic. The output of the calculations and error or success messages were verified as well. Subsequently, I decided to test the packaged application on my Windows 10 PC which lacks Python dependencies. These tests yielded successful results and did not show any unexpected behaviour or issues, both with small (400 addresses) and large (13 000 addresses) datasets.

Further testing involved distributing the application to 10 evaluation participants and a test on a virtual machine environment. For this purpose, I utilized the VMware Workstation Pro 17 software with a virtual machine based on Windows 10, as illustrated in Figure 5.3. Once again, the system exhibited expected behaviour, receiving positive feedback from the participants.

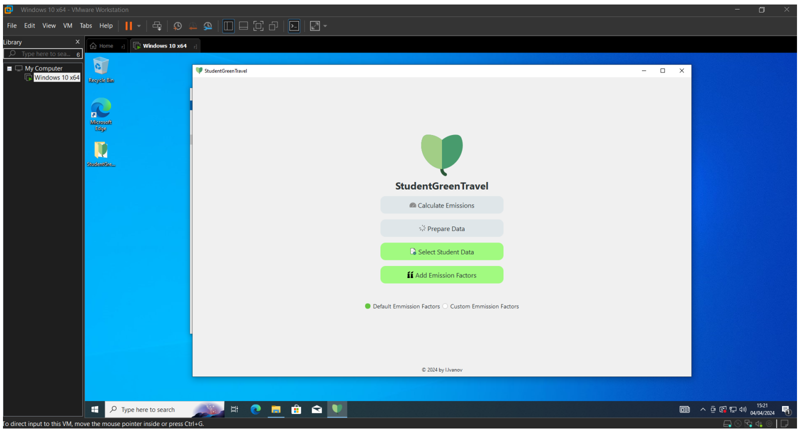

Figure 5.3: System Testing with VMware Workstation Pro

Limitations

The primary limitation identified in the system is related to the packaged version's compatibility, which is limited to Windows operating systems only. This arises from the packaging process, which was based on `pyinstaller` and executed on Windows. However, this constraint can be addressed in future versions of the system.

Known Issues

Although the system performed as expected during testing, there is an issue which was discovered by some of the evaluation participants. This prompted the further testing on a virtual machine without any of the Python or other dependencies that my personal machine has. The issue is related to packaging and the fact that I did not sign the application, neither the application was signed by a Certification Authority (CA) due to time and cost constraints. Consequently, Windows Defender sometimes detects StudentGreenTravel as potentially malicious software, which is not the case, due to it being created by an 'Unknown Publisher', as shown in Figures 5.4 and 5.5. However, this issue does not affect any of the software's functionalities.

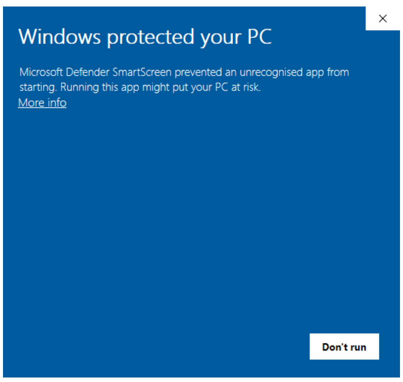

Figure 5.4: Unknown Publisher Warning

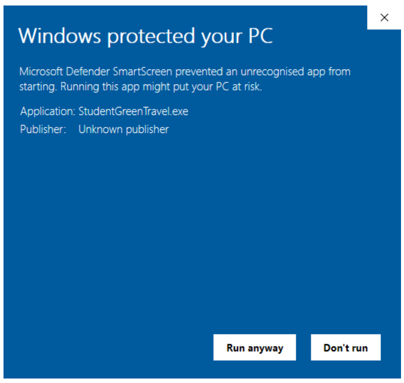

Figure 5.5: Unknown Publisher Warning

5.1.3 Comparative Testing

Methodology

Following the testing of both the Back End and the entire system, a comparative analysis was conducted between the old calculation tool and the new software to assess which one is more accurate. To facilitate this assessment, the following hypothesis was formulated:

- **Comparative Null Hypothesis**: There is no significant difference in the distance measurement accuracies of both systems.
- **Comparative Alternative Hypothesis**: There is a significant difference in the distance measurement accuracy of one of the systems.

To test this hypothesis, test data samples were generated, using the doogal.co.uk website, to be utilized in StudentGreenTravel. These samples comprised 10 files, each containing single postcodes for Aberdeen, Scotland, England, Wales, and Northern Ireland, totaling 50 postcodes. Then, the software was run using each of these test samples, and the total travel distances, encompassing all journey legs, were recorded for each postcode. Consistent travel assumptions were applied throughout the tests: Scottish students travel only by bus, while non-Scottish students split their travel equally between plane and train for the middle leg. The initial and final legs followed similar assumptions, with Scottish students using cars and others using taxis. This ensured that distances would be measured similarly for all postcodes.

Next, travel distances for the same postcodes were obtained from the previous tool. However, due to the way the DISRTECT functions and calculates distances, all distances were identical for all postcodes within a given country. This limitation is a result of the tool's reliance on distances solely from the capital of the country, irrespective of the student's actual location (postcode). Additionally, the tool lacks data on Aberdeen-based students, so the data for them was equivalent to that of Scotland.

Finally, the data from both tools underwent a paired samples T-tests with Equal Variances for each pair of data within each country, and resutls were derived.

Results

All resulting p-values were found to be lower than the set threshold of 0.05, as shown in Table 5.1, leading to the rejection of the Null Hypothesis. Consequently, the alternative hypothesis, indicating a significant difference in the distance measurement accuracy of one of the systems, was accepted. This conclusion was further supported by comparisons to Google Maps measurements, affirming that StudentGreenTravel performs more accurate distance calculations and consequently more precise emission calculations.

Location	Postcode	SGT	DISRTECT	P(T $\leq t$) two-tail
Scotland	EH48 2HD	183.7	156.02	0.000321639262577676
	G31 4RU	334.8	156.02	
	FK12 5JN	177.3	156.02	
	DD9 6RE	135.5	156.02	
	HS2 0QF	400.9	156.02	
	KW1 5BS	322.1	156.02	
	PA23 8HH	267.5	156.02	
	ML5 2HJ	275.1	156.02	
	ZE1 0XU	347.5	156.02	
	TD1 3SG	351.9	156.02	
Wales	CF64 4JL	926	641.24	1.25828E-12
	LL21 0TE	1033.7	641.24	
	NP25 5RN	893.1	641.24	
	SY2 6GL	1046.1	641.24	
	SA11 4AU	996	641.24	
	CF5 1BG	928.8	641.24	
	SY22 6TE	1037.5	641.24	
	NP15 2HP	898.6	641.24	
	LL16 5UF	1019.3	641.24	
	CF39 8JG	937.9	641.24	
England	EC1V 3PT	669.5	648.11	6.22156E-05
	AL5 4TD	780.9	648.11	
	BS39 4LD	886.8	648.11	
	YO23 7EA	1044.7	648.11	
	CV31 2BN	866.7	648.11	
	DH1 5AF	1165.6	648.11	
	GU51 5RX	769.5	648.11	
	BB7 9NL	1084	648.11	
	IP3 9ER	833.1	648.11	
	LE3 9BS	919.5	648.11	
Northern Ireland	BT61 8BR	1302.6	377.58	4.88999E-26
	BT23 8XQ	1250.1	377.58	
	BT32 4QN	1291.1	377.58	
	BT60 1EP	1292.8	377.58	
	BT36 5LN	1306.1	377.58	
	BT92 7BZ	1335.5	377.58	
	BT19 1JY	1246.4	377.58	
	BT63 5WQ	1282.3	377.58	
	BT10 0ER	1246.5	377.58	
	BT62 3SR	1292.5	377.58	
Aberdeen	AB21 9RJ	5.3	156.02	3.12322E-10
	AB55 5DS	66.4	156.02	
	AB16 7XX	2.3	156.02	
	AB16 5UF	2.9	156.02	
	AB44 1NX	60.4	156.02	
	AB55 4ES	69.2	156.02	
	AB41 8RD	22	156.02	
	AB31 6LU	33.9	156.02	
	AB10 1JG	2.2	156.02	
	AB56 1FA	76.6	156.02	

Table 5.1: Comparative Testing Results

5.2 Evaluation

5.2.1 Methodology

The evaluation employed the A/B testing methodology, where participants were presented with the two versions of the software and tasked with completing identical activities in both to determine which one performs better (Quin et al., 2024). These tasks were related to the scenarios outlined in section 3.1.1. The rationale behind utilizing this method is related to a core objective of this project - developing a more user-friendly system than the available one. To be able to determine if the new system is better, there had to be a baseline for comparison. This is the main purpose of A/B testing, which is why it was selected as a main evaluation methodology among others. It was complemented with two questionnaires that participants, included in the evaluation, had to answer.

The core aim of this evaluation was to determine whether the StudentGreenTravel software is more user-friendly than the Domestic and International Student Relocation Travel Emissions Calculator Tool (DISRTECT). Consequently, two hypotheses were developed:

- **Null Hypothesis 1**: There is no significant difference in the usability of both systems.
- **Alternative Hypothesis 1**: There is a significant difference in the usability of one of the systems.

This hypothesis is going to be tested through the System Usability Scale (SUS) scores, while the following ones - through the Additional Question (AQ) Scores, described later:

- **Null Hypothesis 2**: There is no significant difference in the specific functionalities of both systems.
- **Alternative Hypothesis 2**: There is a significant difference in the specific functionalities of one of the systems.

Ten students from different disciplines and universities volunteered for the evaluation. The sample consisted of six females and four males with diverse level of technical competence. The reason to chose students for the evaluation is that they provided a neutral starting point as they are unlikely to have used such software, allowing for more objective assessment of usability. Selecting participants with more work and life experience, especially ones working in the Information Technology (IT) or Net Zero field, might have resulted in somewhat biased results.

In terms of instruments, the evaluation utilized the StudentGreenTravel software, the demo version of the Domestic and International Student Relocation Emissions Calculator Tool, and questionnaires, based on the System Usability Scale (SUS). Additionally, the surveys contained eight additional questions, designed to enhance usability assessment. Notably, the demo version of the second application does not limit any of its functionalities, but it utilizes non-confidential data for its calculations, unlike the full version, which utilizes actual student addresses.

The SUS, renowned for its simplicity and flexibility, provided a standardized procedure for determining the usability of a system. Additionally, its resultant single score on a scale is easily understood by a wide range of people (Bangor et al., 2008). Due to these factors, this strategy was chosen for the current evaluation.

The SUS consists of ten standardized statements (or questions), each of which had to be rated with a number between 1 (strongly disagree) to 5 (strongly agree) in the current evaluation. Once the results from all participants were obtained, the SUS score from each participant was calculated, according to a standard procedure: for odd-numbered questions (1, 3, 5, 7, 9), 1 was subtracted from the user's rating and for even-numbered questions (2, 4, 6, 8, 10), the user's rating was subtracted from 5. Then, the values for all questions for each participant were summed and multiplied by 2.5 to obtain the final score. This score was then compared against the SUS score table, shown in Table 5.2, to determine the system's performance. Finally, both systems, examined in the A/B testing, were compared to derive conclusions.

SUS Score	Grade	Adjective Rating
80.4 - 100	A	Excellent
68.1 - 80.3	B	Good
68	C	Okay
51 - 67.9	D	Poor
0 - 50.9	F	Awful

Table 5.2: SUS Score Table

The previously-mentioned additional questions were designed to measure the usability of some of the specific features of both systems. These features included providing clear guidance, showing error messages and good formatting of calculation results. The questions had to be answered according to the SUS guidance with numbers from 1 to 5 and were as follows:

- I think the system displays the results in a nice format.
- I think it would be easy to analyze the results that the system displays.
- I think the application provided clear and useful guidance.
- I feel the application would be easy to remember how to use after a period of time.
- I think the application provided clear error messages.
- I found the application pleasant to use.

Additionally, at the end of the second survey, the participants were asked which of the two systems they generally found more user-friendly and effective for calculating emissions.

5.2.2 Results

SUS Scores

The results of the evaluation showed a clear difference in the usability of the systems with StudentGreenTravel (SUS-B) achieving an average SUS score of 90.5, while the DISRTECT (SUS-A) only reaching 54.75. These scores indicate grade A, or Excellent, performance for StudentGreenTravel, while the latter system received a D, or Poor, as per the SUS score table.

The resutls of SUS-B showed a very well-distributed values with the lowest score being 72.5 and the highest being 100 as illustrated in Figure 5.6.

Interestingly, it was observed that participants with lower technical proficiency tended to measure the usability of the system slightly lower in comparison with their moderate and highly

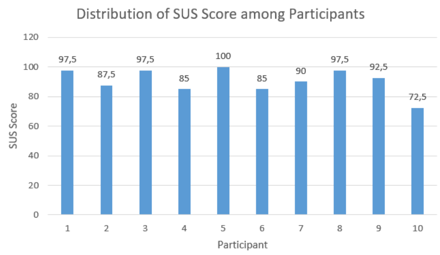

Figure 5.6: Distribution of SUS score for SUS-B

SUS Scores															
ID	Consent	Gender	TC	SUS1-B	SUS2-B	SUS3-B	SUS4-B	SUS5-B	SUS6-B	SUS7-B	SUS8-B	SUS9-B	SUS10-B	TOT-B	SUS-B
1	Consent	Female	High	3	4	4	4	4	4	4	4	4	4	39	97,5
2	Consent	Male	High	1	4	3	4	4	4	3	4	4	4	35	87,5
3	Consent	Female	High	4	4	4	3	4	4	4	4	4	4	39	97,5
4	Consent	Female	High	4	4	4	3	4	0	4	3	4	4	34	85
5	Consent	Female	High	4	4	4	4	4	4	4	4	4	4	40	100
6	Consent	Female	Low	3	4	3	3	4	4	3	4	3	3	34	85
7	Consent	Male	Medium	4	4	4	3	4	4	3	2	4	4	36	90
8	Consent	Female	Medium	4	4	4	4	4	4	4	4	4	3	39	97,5
9	Consent	Male	High	3	3	4	4	3	4	4	4	4	4	37	92,5
10	Consent	Male	Low	2	3	3	4	3	2	3	3	3	3	29	72,5

Figure 5.7: Table of SUS results for SUS-B

skilled peers, as shown in Figure 5.7. Additionally, the female students tended to rate the usability of the system higher than their male counterparts.

On the other hand, scores for DISRTECT did not show such a good distribution, as depicted in Figure 5.8. The highest score achieved for this system was 97.5, while the lowest was 17.5.

In this system, it was observed that participants with lower and moderate technical skill tended to assess it lower than their colleagues with higher skill. Notably, there are high-skilled participants which also assess the system lower. An example of this being the lowest score produced by a participant with a high technical expertise. Moreover, the female participants tended to measure the system usability lower than their male peers. These results are depicted in Figure 5.9.

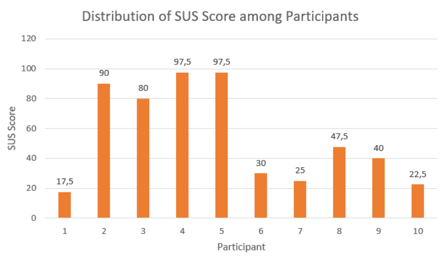

Figure 5.8: Distribution of SUS score for SUS-A

SUS Scores															
ID	Consent	Gender	TC	SUS1-A	SUS2-A	SUS3-A	SUS4-A	SUS5-A	SUS6-A	SUS7-A	SUS8-A	SUS9-A	SUS10-A	TOT-A	SUS-A
1	Consent	Female	High	0	0	0	3	1	2	0	0	0	1	7	17,5
2	Consent	Male	High	1	4	4	4	4	4	3	4	4	4	36	90
3	Consent	Female	High	3	2	3	3	4	4	4	2	4	3	32	80
4	Consent	Female	High	4	4	4	4	4	4	4	3	4	4	39	97,5
5	Consent	Female	Low	4	4	4	4	4	4	4	4	3	4	39	97,5
6	Consent	Female	High	1	1	0	3	1	1	1	0	1	3	12	30
7	Consent	Male	Medium	0	0	0	1	1	1	2	2	1	2	10	25
8	Consent	Female	Medium	1	3	2	3	2	2	1	2	1	2	19	47,5
9	Consent	Male	High	2	1	1	3	2	2	1	2	1	1	16	40
10	Consent	Male	Low	0	0	0	1	1	2	1	2	1	1	9	22,5

Figure 5.9: Table of SUS results for SUS-A

Additional Questions (AQs)

Before discussing the results from these questions, it is important to outline how they were obtained. The procedure followed was straightforward and SUS-inspired: scores from all participants were averaged for each question to yield final scores. Then, they were compared against a scale where 5 represented "Excellent" performance and 1 represented "Awful" performance, as depicted in Table 5.3.

The additional questions showed a better results than the SUS for the Domestic and International Student Relocation Travel Emissions Calculator Tool (DISRTECT), especially in the result formatting (score: 3.4) and analysis department (score: 3.9). Notably, these aspects are also strong points for StudentGreenTravel with scores of 4.7 and 4.5 respectively. However, participants noted shortcomings in the DISRTECT application, reporting a lack of clear and useful guidance, as well as not encountering error messages, with scores of 2 and 1.3 respectively. Furthermore, this system was not found very pleasant for the users with the average score for the corresponding question being 2.8. In these departments StudentGreenTravel excelled with scores of 4.6 and 4.7.

AQ Score	Grade	Adjective Rating
4 - 5	A	Excellent
3 - 3.99	B	Good
2 - 2.99	C	Okay
1 - 1.99	D	Poor
0 - 0.99	F	Awful

Table 5.3: AQ Score Table

5.2.3 Hypothesis Testing

Although the results of the A/B testing showed very convincing results, in order to be able to accept or reject the Null Hypotheses and thus conclude which is the more user-friendly system, a hypothesis testing had to be performed. This testing consisted of performing a paired samples T-tests with the SUS scores and AQ scores of both systems, utilizing the Data Analysis functionality of Microsoft Excel. As a threshold a p-value of 0.05, which is standard, was chosen. This meant that if the p-value of a test is lower than this threshold, there is a significant difference in usability and the Null Hypothesis is rejected.

The SUS T-test showed a p-value of 0.003716 Therefore, the Null Hypothesis 1, stating that there is no significant difference in the usability of both systems was rejected. The Alternative Hypothesis 1 stating that there is a significant difference in the usability of one of the systems was accepted. Therefore, it can be concluded that StudentGreenTravel is the more user-friendly system due to its overall better results.

The AQ T-test achieved a p-value of 0.000602 and thus rejected Null Hypothesis 2 and accepted Alternative Hypothesis 2. Therefore, it can be concluded that there is a significant difference in the specific functionalities of one of the systems, namely StudentGreenTravel.

Additionally, ANOVA tests were performed for both systems to examine the findings from section 5.2.2 that the technical skills of participants influenced their usability assessments. For both systems, the F scores were higher than the F critical scores, which meant that the technical skills of the participants influenced the way they assessed the usability of both systems.

5.2.4 Limitations

Due to the limited timeframe, there are some limitations to this evaluation. First, was the limited sample size of 10 participants, which although produced valuable results, could be expanded to produce more concrete results. Additionally, while students provided a valuable perspective for initial evaluation, future studies could involve more diverse participants to gain insights into usability for more experienced users.

5.3 Interpretation of Results

To summarize, the findings indicate that StudentGreenTravel operates as expected, and provide convincing proof that it is more usable and more accurate than its predecessor, as substantiated by the ANOVA and T-tests, and the comparative analysis results. Additionally, the system-wide testing shows that the packaged software is compatible with different Windows operating systems.

Chapter 6

Discussion

This chapter outlines the positive outcomes of the current project, along with the mistakes made and lessons learned that have enriched the project and the development process.

6.1 Positive Outcomes

Overall, the development process went well, and the software incorporates the objectives outlined earlier, albeit with the exception of the visualization of travel routes for students. Despite this limitation, the software remains fully operational, fulfilling the client's requests and expectations. With features such as uploading custom emission factors and the ability to make assumptions separately for countries within United Kingdom (UK), the software has the potential to expand beyond the University of Aberdeen's use. It contains all crucial components and functionalities, achieving its goals of enhanced usability and accuracy in comparison with its predecessor. This success is attributed to the Agile development approach and the workflow I followed, which proved crucial in the communication with the client and distribution of workload.

Taking detailed notes and sticking to a strict schedule were instrumental in producing this application and writing the report. Additionally, the availability of detailed descriptions of every step of the research and development, facilitated switching between tasks when necessary and being able to find the needed information when writing. Starting the report early allowed me to organize accumulated information without rushing, ensuring a high-quality outcome. Moreover, early planning enabled better reflection on existing chapters and improved preparation for upcoming sections.

Furthermore, testing and evaluation confirmed the good performance of StudentGreenTravel, providing valuable user feedback that praised its intuitive design and user-friendliness. This was possible owing to the PyQt6 library, which in my mind turned out to be a key component and a very good implementation decision. This stems from its various components and wealth of resources available for developing the user interface.

6.2 Learning from Experience

Despite the positive outcomes of the project, I encountered several obstacles and learned valuable lessons along the way.

My initial mistake was developing a rudimentary version of the software which operated on the terminal and utilized CSV files. In my mind, this was a good decision because I wanted to ensure accurate distance calculations first and worry about the rest of the project's components later. This approach proved time-consuming as I had to rewrite key portions of the code later on when I already had a designated postcodes database and had to make queries to it. Additionally, I realized the importance of validating queries to ensure correct data output. If I were to start over, I would design a database and make sure that I make the right queries to it first, and then create the methods for calculating distances.

Another mistake was focusing on the Back End functionality without writing a single line of code for the GUI. Again, this cost me time and made me rewrite whole methods and sections of code to make them compatible with the PyQt6 library. Now, I believe that a better approach would be to develop them simultaneously to achieve a more integrated and efficient workflow.

Although the following mistake has not caused me much trouble, I believe it is important to mention it. Occasionally, I caught myself committing substantial changes to my repositories due to getting carried away in coding. I acknowledge that this is not a good practice, especially when needing to revert to an older commit or do a reset. Although I made this mistake a few times, I am committed to avoiding it in future projects.

Finally, I realized that I should have tested the Back End during the development process instead of at the end. This would have saved me a lot of time and would have allowed me to identify the small errors I came across during testing a lot earlier and solve them.

To summarize, although my project managed to meet its goals, the mistakes I made could have cost me a lot. Small setbacks such as not being able to revert to an older commit or having to redo some coding could have easily affected the functionality or accuracy of the system. Moreover, they could have resulted in not completing the software in time. Therefore, I believe these are very important lessons that will undoubtedly benefit my career moving forward.

Chapter 7

Conclusion and Future Work

This chapter summarizes the project's accomplishments and limitations, detailing their personal impact and broader implications. It also proposes areas for potential improvements in the future and brings closure to the project's journey as a whole.

7.1 Project Summary

This project's objective was to develop a student travel emissions calculation software with a user-friendly interface and more accurate calculation methods in comparison with its predecessor, the Domestic and International Student Relocation Travel Emissions Calculator Tool. Additionally, the new software aimed to incorporate better customizability and flexibility. All of these aspects were captured by the project requirements, and almost all of them were addressed by the StudentGreenTravel software.

The customizability aspect, represented by functional requirements 1 and 2, was fulfilled through features such as student data and emission factors input capabilities.

Flexibility, as outlined in functional requirements 3, 4 and 5, was achieved primarily through the integration of PyQt6 components with travel handling Back End functions. These developments also contributed to enhancing the system's accuracy (functional requirement 6) through comprehensive methods of the Back End linked with various Data components. These features also ensured the compliance with the non-functional requirements related to performance, reliability, and scalability.

The user-friendliness of StudentGreenTravel was outlined by functional requirements 7, 8 and 9 and they were mostly met by the utilization of plotly, implementing invalid data handling capabilities of the Back End and designing intuitive components using PyQt6, along with detailed tips and guides within the software. These aspects also addressed the usability non-functional requirement.

Maintainability was ensured by the separation of concerns approach, related to implementing different features in separate files with well-documented code managed by a version control system.

Regarding security, while the collected data for this software is anonymized, the system was developed to only consider postcodes, without storing or utilizing any other sensitive or confidential data. This was achieved by the utilization of the GeoPy library, and handling addresses and travel data securely in the Back End. This way the non-functional requirement of security was met.

The only unmet requirement was the one related to the visualization of the travel routes of students from different countries due to time constraints. This feature could be considered for future iterations of StudentGreenTravel. However, given that it is a single sub-requirement, the project can be deemed an overall success, especially considering the client's satisfaction with the delivered product.

7.2 Personal Reflection

The StudentGreenTravel project put me in a position that I have dreamt about since the start of my degree - to step into the role of a software engineer. I can confidently say that it has been a pleasure and an incredibly enriching experience from start to finish. Despite encountering obstacles and having to handle all of them on my own, I firmly believe that such challenges are always beneficial in the long term. They serve as valuable experiences that can future-proof my career. However, these developments also instilled in me a newfound appreciation for teamwork. Throughout my degree, I had many opportunities to work in teams on different projects and always thought that if given enough time, I will be able to excel individually and produce superior results. This is not the case anymore. I have gained a deeper appreciation for teamwork and realize the importance of having reliable teammates, especially during challenging situations. While my supervisor was always supportive during the project, having a diverse team around opens up the potential for more innovative solutions to complex problems.

This is just one of the lessons I learned during this journey. I have come to understand that careful preparation, thoughtful consideration, unwavering dedication, and motivation are crucial keys to success in any endeavor. Although these principles may sound straightforward, this project underscored their importance in achieving meaningful results.

Furthermore, this venture has enabled me to acquire new skills and further develop the ones I already possessed. I am now somewhat proficient in working with PyQt6 and plotly, along with calculating distances between points based only on their coordinates. Additionally, I have developed my Python and databases knowledge even further. Invaluable skills which extend beyond the bounds of this project.

What fills me with pride is that I worked on a project which is more than a dissertation. I am glad I had the opportunity to work with a real client to build a tool that is going to ease their work. This is precisely the type of impactful work that motivates me to pursue a career as a software engineer. Additionally, the realization that I am leaving a lasting trace at the University of Aberdeen and the project holds potential for broader utilization beyond the university setting is one of my greatest achievements to date.

7.3 Future Work

While the StudentGreenTravel project met almost all of its requirements, there are areas where the application can be further enhanced and expanded. Here are some potential improvements and additions:

- The addition of visualizations of the routes chosen by students for their journeys. This could be implemented as a generalization for students from a certain country, or individual visualizations for each student. However, this may require additional data collection on student travel patterns.

- The software could be made cross-OS compatible. Currently, it does not function on Mac or Linux and the application could benefit from having dedicated versions for these operating systems. This could involve developing dedicated versions for these operating systems using new packaging methods, potentially opening doors for commercial use.
- Despite the tool providing detailed results on both country and council level, it might be a good addition to produce even more results. For instance, displaying council data on flights or offering even more graphical visualizations. This may require creating a new result page designs and developing new Back End methods.
- The software may benefit from larger datasets on transportation hubs and methods to improve the accuracy of distance calculations. For instance, there is a lack of data on Scottish bus stations, which could be addressed by utilizing an API.
- To prepare the product for a broader use beyond the University of Aberdeen, methods for calculating international travel distance could be implemented. However, this may require a major re-design and the utilization of an API for data retrieval.
- To achieve the last two suggestions for future work, the Google Maps API could be implemented to provide the needed data and distance measures. However, this requires additional funding as the API involves a fee for usage.

In addition to these software-based enhancements, there is potential for the project to delve into a more research-oriented direction, providing a solid foundation for data analysis and research on sustainable student travel practices. This could involve collaboration with researchers and institutions interested in this area, leveraging the software's capabilities for broader research purposes.

7.4 Conclusion

In conclusion, the project achieved the objectives of usability and accuracy, meeting the client's needs and expectations effectively. Personally, it has been a significant learning experience that has enhanced my Python and database management skills while introducing new ones - distances and emissions calculation, utilization of plotly and PyQt6.

Moreover, the project deepened my understanding of the critical importance of achieving Net Zero and the broader impact of emissions on our planet. It highlighted the necessity of mitigating these emissions and the role of technology in addressing environmental challenges.

The lessons that this endeavour provided me with through challenges and mistakes are the added value to the overall experience, which I was hoping for at the start. Therefore, I can conclude that this project has not only enriched my personal growth but also has the potential for a broader impact, contributing positively to environmental awareness and sustainable practices. Furthermore, it can serve as a foundation for addressing the well-known challenges of achieving reliable and comparable emissions calculations within the higher education sector.

Appendix A

User Manual

A.1 Student Journeys

A journey represents a student's travel from their home address to the University of Aberdeen. Emissions are calculated based on travel distance and chosen transportation methods.

A.1.1 Journey Stages (Legs)

Each journey consists of three stages called "legs":

1. **Initial Leg**: Travel from home to the nearest transportation hub (airport or station). The built-in assumptions for transport mode are:
   - 40% of students travel by car.
   - 40% of students travel by taxi.
   - 20% of students travel by bus.
2. **Middle Leg**: Travel between transportation hubs:
   - Non-Scottish and Non-London Students (Air Travel): Local airport →Layover airport (London Gatwick) →Aberdeen airport.
   - Other Students (Bus/Train): Local station →Aberdeen station.
   - Scottish students can choose Bus, Car, or Train.
   - Non-Scottish students can choose Plane, Car, or Train.
   - User inputs assumptions for the percentage of students using each option.
3. **Final Leg**: Travel from the Aberdeen transport hub to the university:
   - Transport options: Car, Taxi, Bus, Walking.
   - Starting location depends on previous leg:
     - Scottish Students: Start from bus/railway station.
     - Non-Scottish Students: Start from airport (air travel) or station (land travel).
   - Aberdeen residents start from their home address (no initial and middle leg).

A.1.2 Important Note

Students choosing car travel have a single leg directly from home to the university.

A.2 Input Data Format

StudentGreenTravel operates with .xlsx files for student data. Additionally, users have the option to upload custom emission factors .xlsx files, although they are not mandatory for the system to function properly as there are built-in ones. These files should be formatted in the specified manner to ensure proper functioning of the system.

A.2.1 Student Data

The student data must be in .xlsx format and include at least one column with student postcodes, positioned in column B of the spreadsheet as depicted in Figure A.1. It is not necessary to name these columns or adhere to any specific naming conventions.

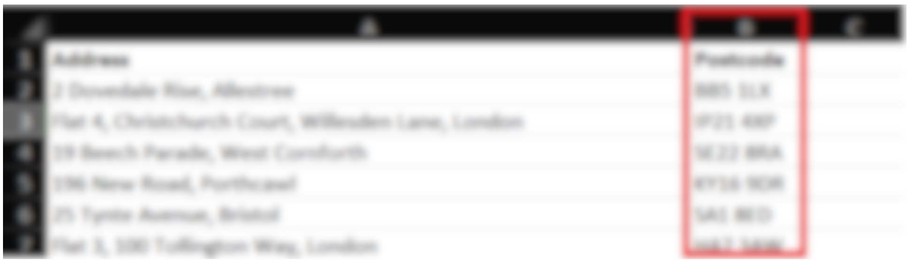

Figure A.1: Student Data Format

A.2.2 Emission Factors

If you choose to upload custom emission factors to the system, they should follow the format shown in Figure A.2. The transport method names should be in the A column of the spreadsheet and must use non-capital letters. The numerical values of the emission factors must be placed in the B column of the spreadsheet. The first row should have "Method" in cell A1 and "Factor" in cell B1 to designate the column headings.

	A	B
1	**Method**	**Factor**
2	car	0.18264
3	rail	0.035463
4	bus	0.118363
5	coach	0.027181
6	taxi	0.148615
7	ferry	0.02555
8	plane	0.03350

Figure A.2: Emission Factors Format

Emission Calculation

The emission factors, whether built-in or custom, are the values multiplied by the distance travelled using a specific mode of transport. This multiplication yields the corresponding emissions amount, measured in kg CO2e (carbon dioxide equivalent).

A.3 Performing Calculations

To perform travel distance and emission calculations, open the StudentGreenTravel application by double clicking on the icon depicted in Figure A.3 and follow the steps below.

Figure A.3: StudentGreenTravel Icon

A.3.1 Uploading Student Data (Mandatory)

Once the application is running, you will be greeted with the initial menu. Click the "Select Student Data" button, depicted in Figure A.4, to open a file dialog and select your .xlsx student data file.

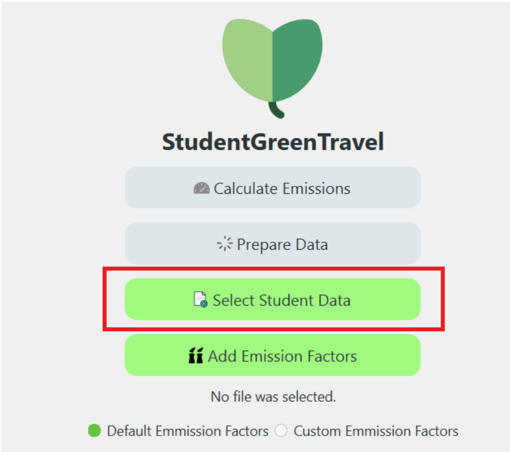

Figure A.4: "Select Student Data" button

Once, you have selected a student data file, you should see the confirmation stating "**Dataset:** Name_of_file.xlsx" and the "Prepare Data" button should become accessible, as shown in figure A.5.

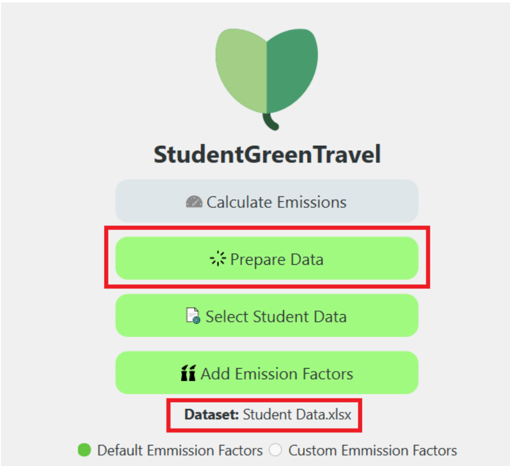

Figure A.5: Dataset Confirmation

A.3.2 Initial Data Processing

Once you have uploaded a data file successfully, you should prepare the data for calculations. This is done by clicking the “Prepare Data” button from Figure A.5. Once the data is ready for calculation, you will get the "The data is ready for calculations!" confirmation and the "Calculate Emissions" button will become accessible, as shown in Figure A.6.

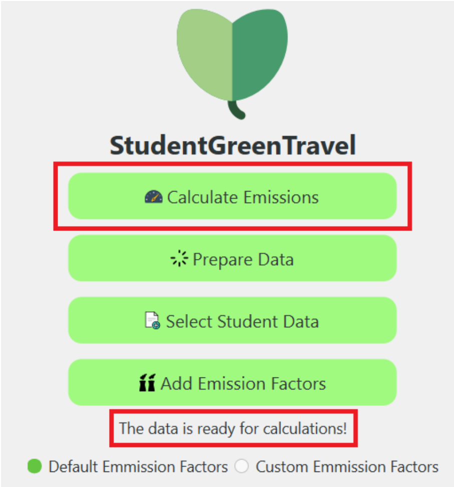

Figure A.6: Ready Data Confirmation

If you do not want to upload custom emission factors, press the "Calculate Emissions" button to proceed and go to section A.3.4. Otherwise, follow the instructions in the following section to upload custom emission factors.

A.3.3 Uploading Custom Emission Factors (Optional)

If you would like to upload custom emission factors, press the "Add Emission Factors" button, depicted in Figure A.7, and select an .xlsx file with emission factors from your computer. Once, you complete this, you should see the indicator for emission factors switched to "Custom Emission Factors", as shown in Figure A.8.

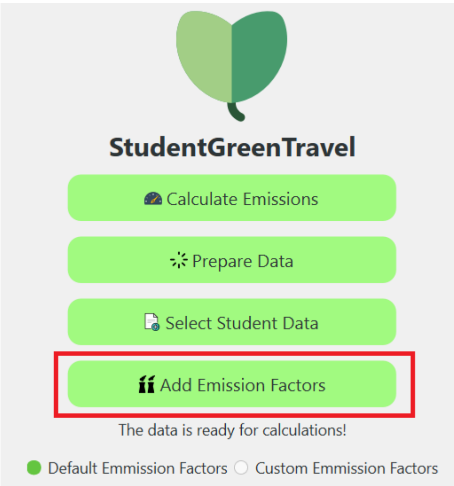

Figure A.7: Student Data Format

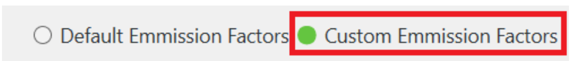

Figure A.8: Emission Factors Format

If you would like to go back to using the built-in emission factors, simply press the "Default

Emission Factors" option. To upload new custom emission factors, start the procedure again.

A.3.4 Entering Number of Trips and Middle Leg Assumptions

NOTE: The assumptions for the Initial Leg of the journey are pre-set in the system, so you only have to enter the ones for the Middle and Final legs.

After clicking the "Calculate Emissions" button, you will be greeted by the next page of the application, related to the middle leg of the journey. First, please enter the number of study-related trips students make per year in the designated place, as depicted in Figure A.9. Then, proceed to entering the travel assumptions for "Scotland" and "Rest of UK". These assumptions apply for students from Scotland and England, Wales and Northern Ireland respectively.

NOTE: Please, check the "Tip" sections in the application for further information, if necessary.

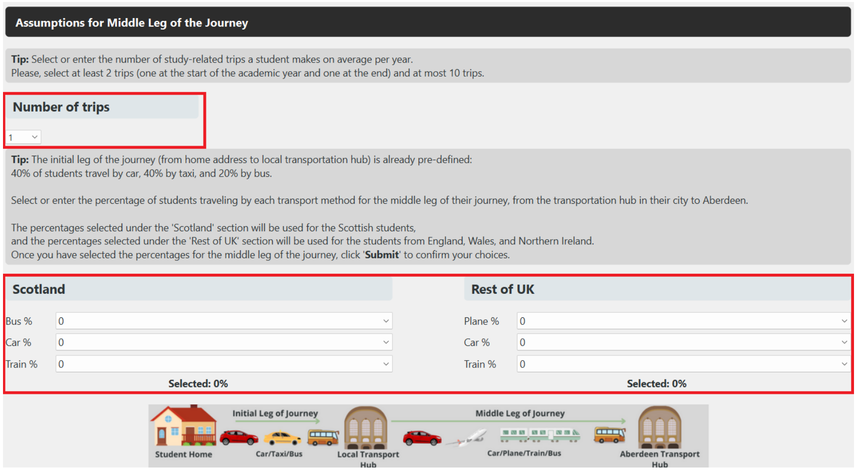

Figure A.9: Middle Leg Page

Submitting Assumptions

To submit the entered assumptions, click the “Submit” button. If percentages for both groups sum up to 100%, you should get the confirmation depicted in Figure A.10 and the “Next" button should became accessible, as shown in Figure A.11. Please, note that there are "Selected: number%" indicators below the "Scotland" and "Rest of UK" boxes that indicate what percentages you have selected. They become green when the percentages sum up to 100 and red when they exceed this value.

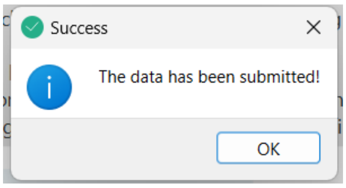

Figure A.10: Submit Success

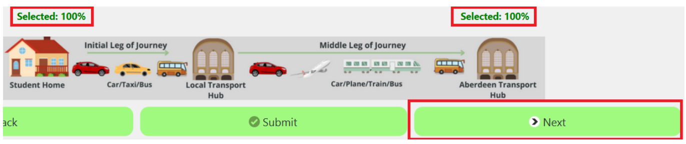

Figure A.11: Next Button Accessible

Wrong Assumption Percentages

If the percentages do not add up to 100, either one or both of the "Selected: number%" indicators will become red, and if you try to submit the assumptions this way, you will get the warning in Figure A.12.

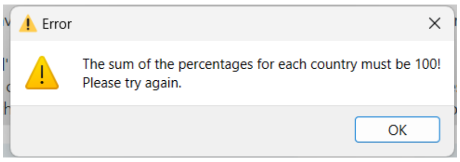

Figure A.12: Wrong Percentages Warning

If you get the confirmation in Figure A.10, please click the "Next" button to proceed to the next page.

A.3.5 Entering Final Leg Assumptions

Upon entering the middle leg assumptions successfully, you will be greeted with the Final Leg Assumptions page. To the left of the page, there is a list with the countries within UK and Aberdeen, as depicted in Figure A.13.

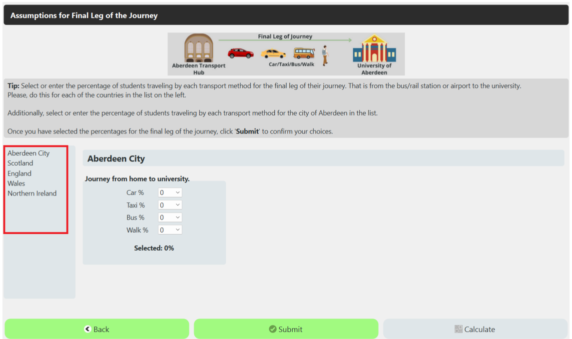

Figure A.13: List of Countries

From this list you can select the countries one by one and enter the final leg assumptions for each country and Aberdeen respectively. Note, that Aberdeen and Scotland have only input boxes for assumptions for the journey from home to university and from station to university respectively. On the other hand, the rest of the countries have input boxes for assumptions for both the journey from airport to university and from station to university, as shown in Figure A.14.

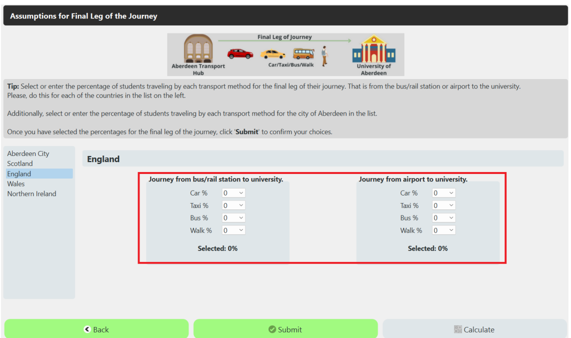

Figure A.14: Input Boxes for Final Leg Assumptions

Submitting Assumptions

Once you have entered final leg assumptions for all countries and Aberdeen, and all of them add up to 100%, follow the procedure similar to the previous page to submit the data to the system. If all assumptions are correct, you will get the familiar confirmation, but this time the "Calculate" button, shown in Figure A.15, will become accessible.

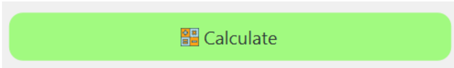

Figure A.15: Calculate Button

NOTE: Please, check the "Tip" sections in the application for further information, if necessary.

Calculation of Distances and Emissions

To start the calculation process, please click the "Calculate" button. Thus, you will start the calculation procedure and the following window with a loading bar will appear:

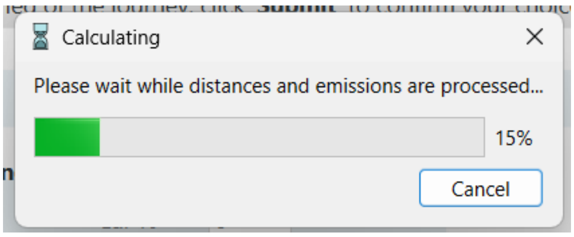

Figure A.16: Calculation Loading Bar

Once the calculations are done, you will automatically be taken to the results page, depicted in Figure A.17.

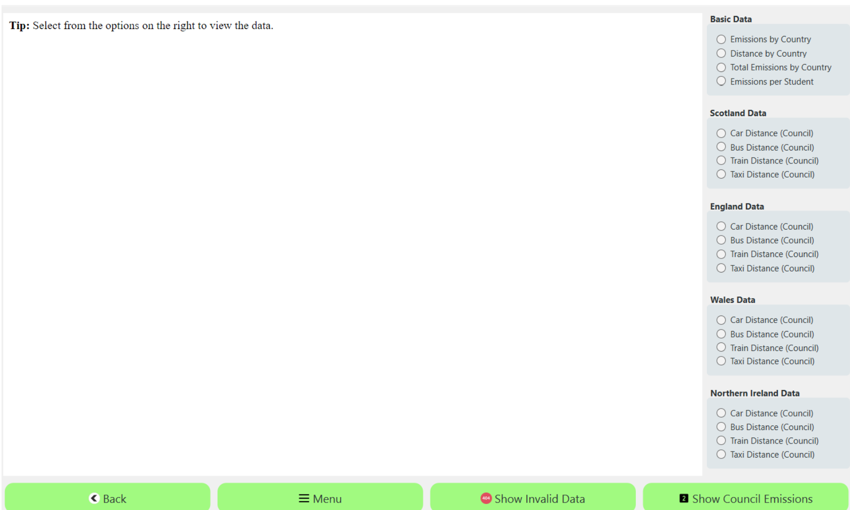

Figure A.17: Results Page

A.3.6 Checking Results

To check the "Base Data" results or the country-specific ones, select any of the options from the boxes to the right, shown in Figure A.18.

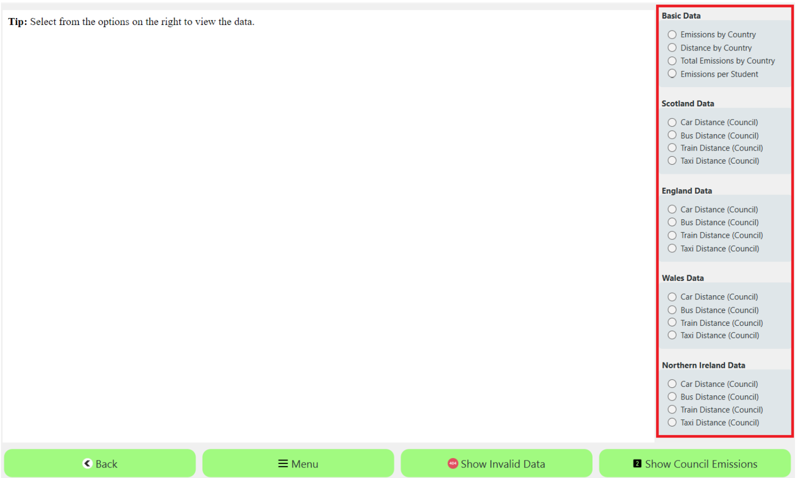

Figure A.18: Result Boxes

Once you click an option from any of the boxes, you will be greeted with the corresponding graphical representation of results, as depicted in Figure A.19.

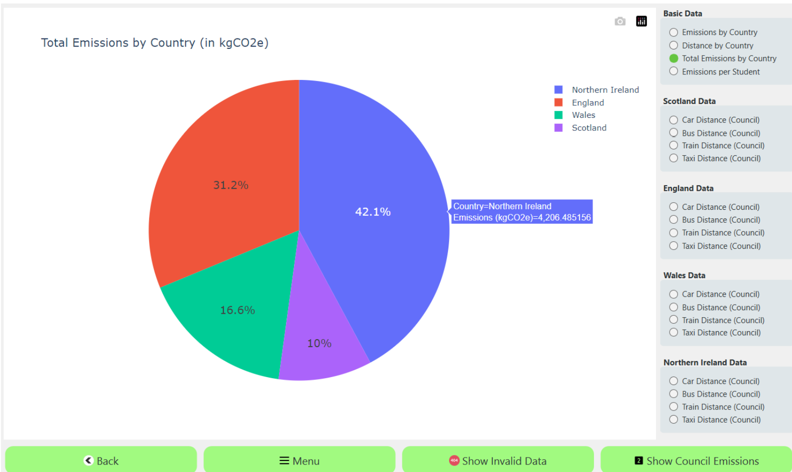

Figure A.19: Graphical Representation of Results

Initially, the page shows the boxes for distance calculations for each country's councils. To see the emissions-specific resutls for the councils of each country, press the "Show Council Emissions" button at the bottom of the page, shown in Figure A.20, and follow the familiar procedure to view the visualizations.

Figure A.20: "Show Council Emissions" button

If you desire, you can go back to the initial menu page and start new calculations by pressing the "Menu" button at the bottom of either of the resutls pages.

A.4 Checking Invalid Data

To check the invalid data points in the input dataset, you should follow the above-described procedure for calculating distances and emissions. However, when the results page is displayed, instead of clicking the options in the boxes to the right, click the "Show Invalid Data" button at the bottom of the result page, depicted in Figure A.21. This button will take you to the invalid data page.

Figure A.21: "Show Invalid Data" button

In the invalid data page, click the "Show Invalid Postcodes" button at the bottom-center of this page, to generate the list of invalid data points, along with their location in the input file. This button is shown in Figure A.22.

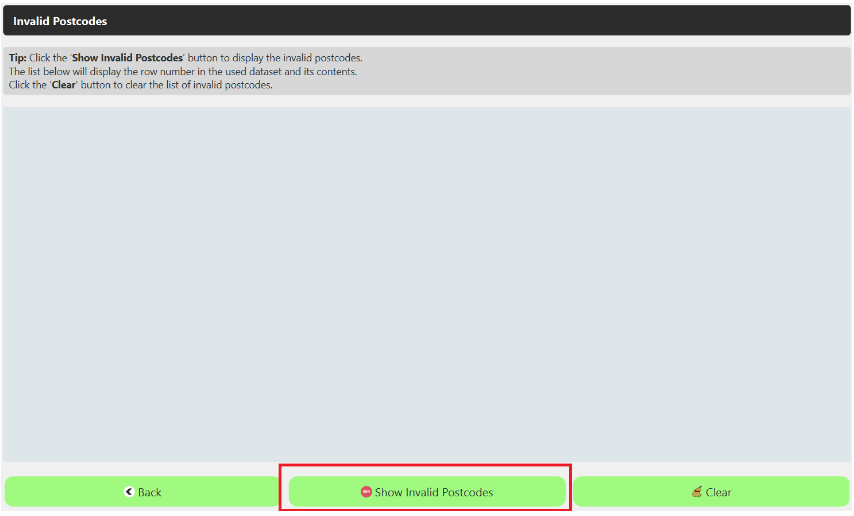

Figure A.22: "Show Invalid Postcodes" button

This action will generate a list with invalid data points, similar to the one shown in Figure A.23.

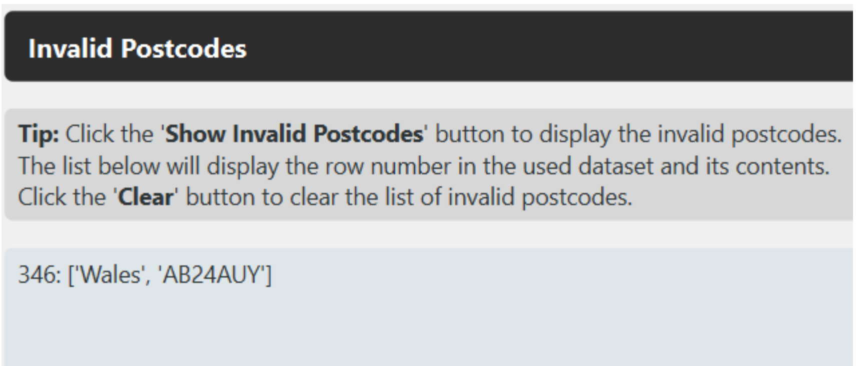

Figure A.23: List with Invalid Data Points

To clear the list, press the "Clear" button and to go back to the results page, click the "Back" button.

Appendix B

Maintenance Manual

This maintenance manual serves as a guide to understanding, installing, and modifying StudentGreenTravel. It provides information for both developers using the unpackaged (code) version and users choosing the convenient packaged version of the software.

B.1 Installation and Execution

B.1.1 Packaged Version

1. Unzip the "StudentGreenTravel.zip" file in a directory of your choice.

2. Start the StudentGreenTravel application.

B.1.2 Code Version

1. Install Python.

2. Install Pip.

3. Start Command Line Prompt.

4. Navigate to the directory where you have stored the application folder.

5. Run the "`pip install -r requirements.txt`" command to install the system dependencies.

6. Run the "`calculator.py`" command to execute the application .

B.2 System Description

B.2.1 Hardware and Software Dependencies

The operation of each version of the application is dependent upon meeting the following requirements:

Packaged Version

- Operating System: Windows 10 or Windows 11

- File compression software: 7-Zip or WinRAR

- Internet connection

Code Version

- Python 3.10 or newer
- geopy 2.4.1
- numpy 1.26.4
- pandas 2.1.4
- plotly 5.19.0
- PyQt6 6.6.1 and PyQt6_sip 13.6.0
- requests 2.31.0
- openpyxl 3.1.2
- Internet connection

B.2.2 Organization of System Files and Directories

Packaged Version

- `_internal` directory: Within this directory lie the `data`, `pictures`, and `icons`, housing their respective files that can be edited as needed, alongside all the application's components and packages.
- StudentGreenTravel.exe: the application .exe file

Code Version

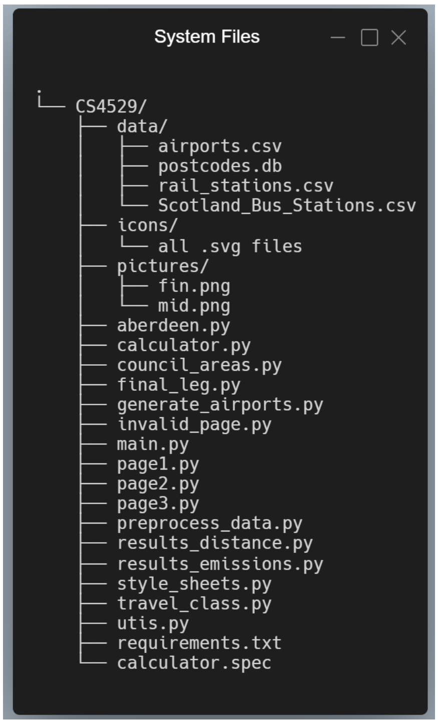

Figure B.1: System Files in Code Version of StudentGreenTravel

B.2.3 Space and Memory Requirements

- Required Space: 1GB
- Required Memory (RAM): 700MB

B.3 Source Code Files

File	Functionality
aberdeen.py	Contains methods that calculate distances from a home postcode to the university, and divide students according to travel assumptions for transport methods.
calculator.py	Brings together all the features and initiates the program's operation through a class called Calculator.
council_areas.py	Deals with getting the admin district for a list of postcodes and then grouping them by the admin district. Also finds the percentage of postcodes in each admin district with respect to all postcodes for the country.
final_leg.py	Contains the functions to calculate the travel distance for the final leg of the journey of each mode of transport for the students from Scotland and the rest of the UK.
generate_airports.py	Used to generate the "airports.csv" file which contains only the public UK airports and their longitude and latitude.
main.py	Combines the functionality of the other Back End files and produces the final results.
preprocess_data.py	Contains the functions used to read and prepare the postcodes of students for calculations.
style_sheets.py	Contains the style sheets for the application and its widgets.
test_code.py	Tests the main functions of the project with the unittest framework.
travel_class.py	Holds the Travel class methods used to perform initial and middle leg travel distances calculations.
utils.py	Accommodates functions which are used in multiple other files.
page1.py	Holds the MainPage class which contains the initial menu page components.
page2.py	Holds the Page2 class which contains the middle leg page components.
page3.py	Holds the Page3 class which contains the final leg page components.
results_distance.py	The first final page which contains the webview for the plotly heatmaps and the other widgets for the results.
results_emissions.py	The second final page which contains the webview for the plotly heatmaps and the other widgets for the results.
invalid_page.py	Holds the invalid data page class along with the methods for identifying invalid data points.

Table B.1: File Functionalities

B.4 Future Improvements

There are areas where the application can be further enhanced and expanded:

- The addition of visualizations of the routes chosen by students for their journeys. This could be implemented as a generalization for students from a certain country, or individual visualizations for each student. However, this may require additional data collection on student travel patterns.
- The software could be made cross-OS compatible. Currently, it does not function on Mac or Linux and the application could benefit from having dedicated versions for these operating systems. This could involve developing dedicated versions for these operating systems using new packaging methods, potentially opening doors for commercial use.
- Despite the tool providing detailed results on both country and council level, it might be a good addition to produce even more results. For instance, displaying council data on flights or offering even more graphical visualizations. This may require creating a new result page designs and developing new Back End methods.
- The software may benefit from larger data sets on transportation hubs and methods to improve the accuracy of distance calculations. For instance, there is a lack of data on Scottish bus stations, which could be addressed by utilizing an API.
- To prepare the product for a broader use beyond the University of Aberdeen, methods for calculating international travel distance could be implemented. However, this may require a major re-design and the utilization of an API for data retrieval.
- To achieve the last two suggestions for future work, the Google Maps API could be implemented to provide the needed data and distance measures. However, this requires additional funding as the API involves a fee for usage.

B.5 Bug Reports

There are three known bugs to the system:

1. Occasionally, during the initial data processing after clicking the "Prepare Data" button on the initial menu, the system may be labeled as "Not Responding." Despite this, the system continues to function properly, and upon completion, it displays a confirmation message.
2. Similarly, when calculating distances and emissions after clicking the "Calculate" button on the final leg assumptions page, the system may also be marked as "Not Responding." However, this issue does not impact any of its core functionalities.
3. Due to time and resource limitations, StudentGreenTravel is currently not signed by a certificate authority (CA). Consequently, Windows Defender may identify it as "potentially unwanted software" because the publisher is "Unknown". This is a security measure and doesn't indicate any malicious code within StudentGreenTravel. You can safely use the application.

Appendix C

Supplementary Materials

C.1 Wireframes

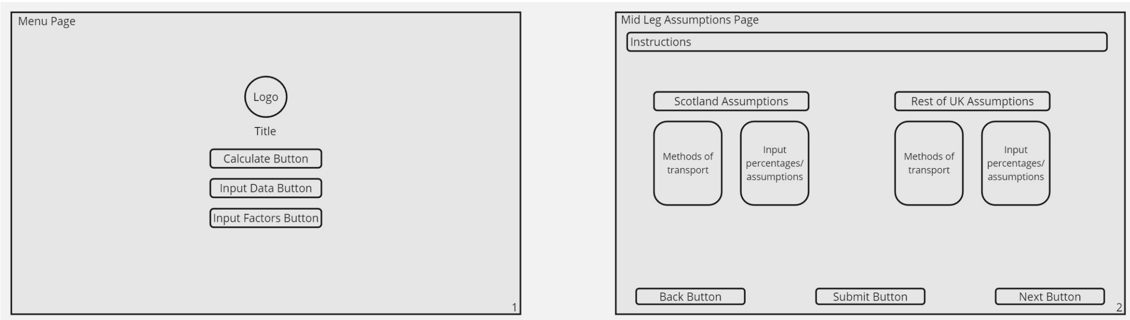

Figure C.1: Wireframes of Menu and Middle Leg page

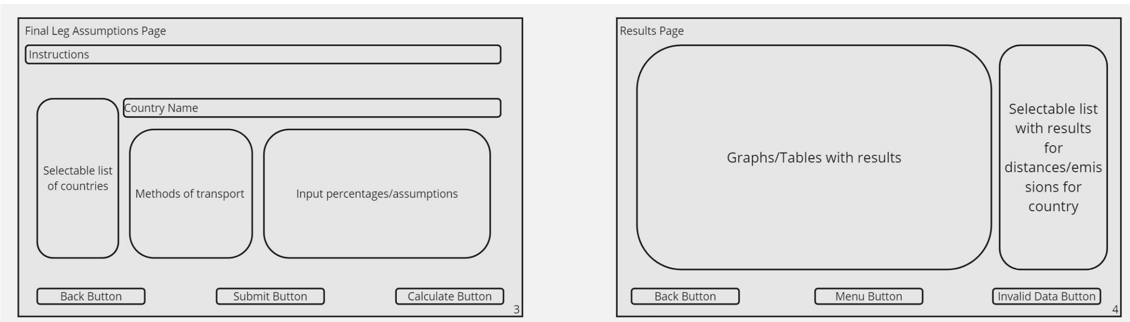

Figure C.2: Wireframes of Final Leg and Results page

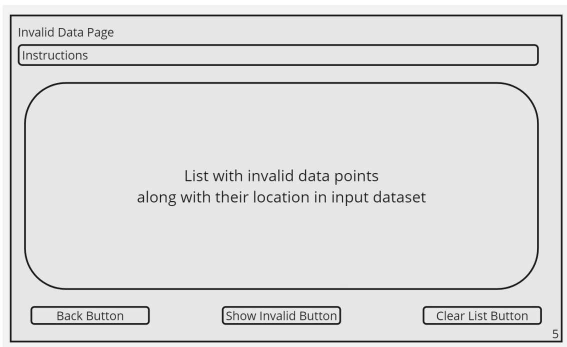

Figure C.3: Wireframe of Invalid Data page

C.2 Evaluation Tables and Results

C.2.1 SUS Scores

SUS Scores															
ID	Consent	Gender	TC	SUS1-A	SUS2-A	SUS3-A	SUS4-A	SUS5-A	SUS6-A	SUS7-A	SUS8-A	SUS9-A	SUS10-A	TOT-A	SUS-A
1	Consent	Female	High	0	0	0	3	1	2	0	0	0	1	7	17,5
2	Consent	Male	High	1	4	4	4	4	4	3	4	4	4	36	90
3	Consent	Female	High	3	2	3	3	4	4	4	2	4	3	32	80
4	Consent	Female	High	4	4	4	4	4	4	4	3	4	4	39	97,5
5	Consent	Female	Low	4	4	4	4	4	4	4	4	3	4	39	97,5
6	Consent	Female	High	1	1	0	3	1	1	1	0	1	3	12	30
7	Consent	Male	Medium	0	0	0	1	1	1	2	2	1	2	10	25
8	Consent	Female	Medium	1	3	2	3	2	2	1	2	1	2	19	47,5
9	Consent	Male	High	2	1	1	3	2	2	1	2	1	1	16	40
10	Consent	Male	Low	0	0	0	1	1	2	1	2	1	1	9	22,5

Figure C.4: SUS-A Scores

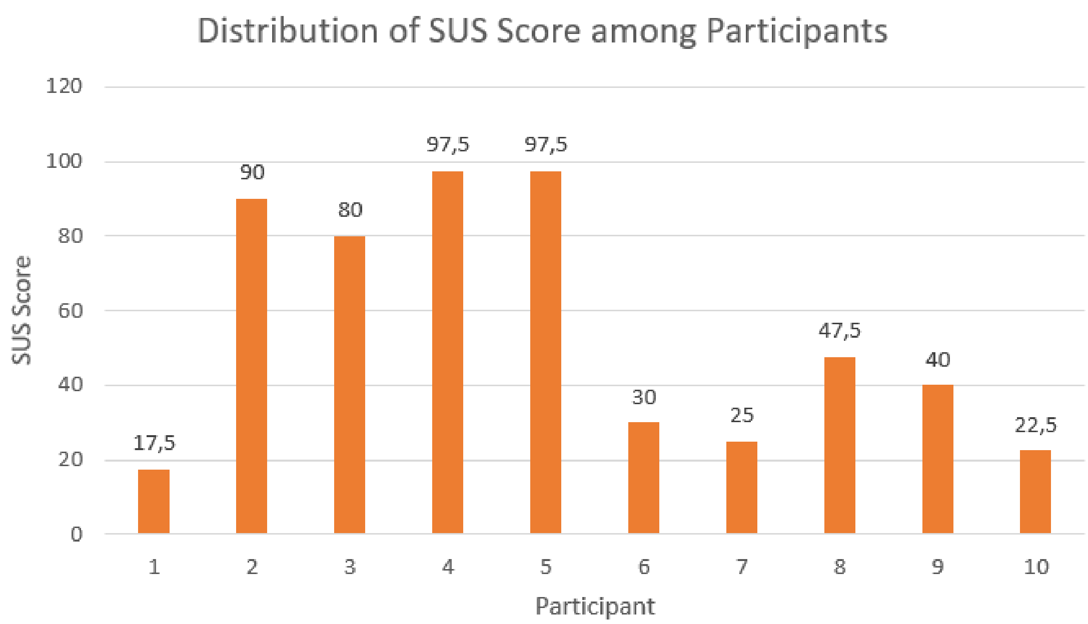

Figure C.5: SUS-A Scores Distribution

SUS Scores															
ID	Consent	Gender	TC	SUS1-B	SUS2-B	SUS3-B	SUS4-B	SUS5-B	SUS6-B	SUS7-B	SUS8-B	SUS9-B	SUS10-B	TOT-B	SUS-B
1	Consent	Female	High	3	4	4	4	4	4	4	4	4	4	39	97,5
2	Consent	Male	High	1	4	3	4	4	4	3	4	4	4	35	87,5
3	Consent	Female	High	4	4	4	3	4	4	4	4	4	4	39	97,5
4	Consent	Female	High	4	4	4	3	4	0	4	3	4	4	34	85
5	Consent	Female	High	4	4	4	4	4	4	4	4	4	4	40	100
6	Consent	Female	Low	3	4	3	3	4	4	3	4	3	3	34	85
7	Consent	Male	Medium	4	4	4	3	4	4	3	2	4	4	36	90
8	Consent	Female	Medium	4	4	4	4	4	4	4	4	4	3	39	97,5
9	Consent	Male	High	3	3	4	4	3	4	4	4	4	4	37	92,5
10	Consent	Male	Low	2	3	3	4	3	2	3	3	3	3	29	72,5

Figure C.6: SUS-B Scores

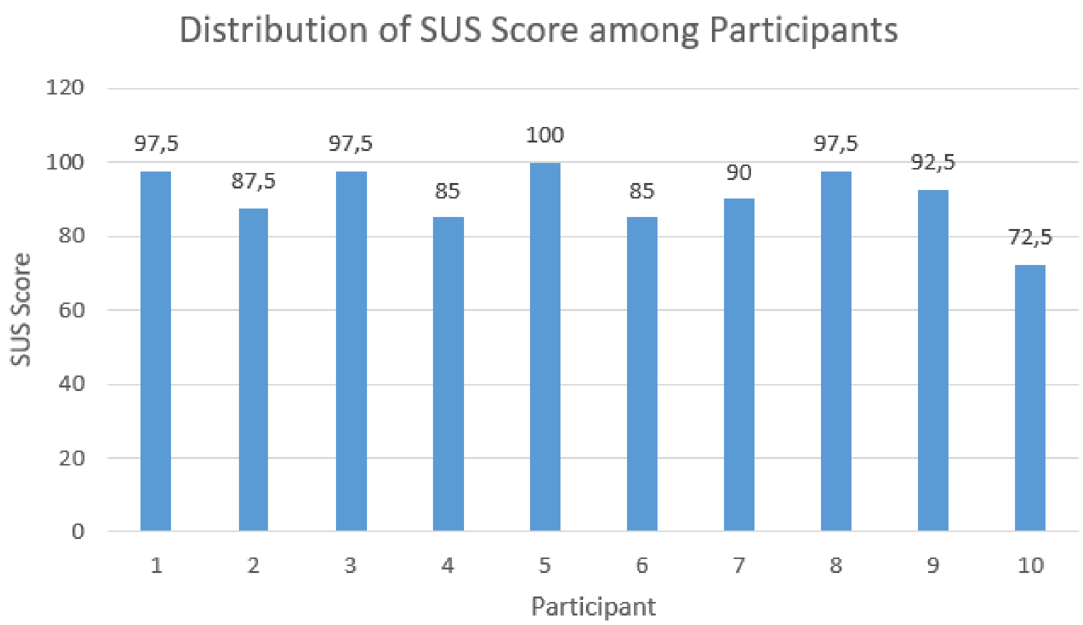

Figure C.7: SUS-B Scores Distribution

C.2.2 AQ Scores

Additional Questions (AQs)									
Participant	Consent	Gender	Technical Competence	AQ1-A	AQ2-A	AQ3-A	AQ4-A	AQ5-A	AQ6-A
1	Consent	Female	High	1	4	1	2	1	2
2	Consent	Male	High	2	3	2	4	1	3
3	Consent	Female	High	4	4	2	4	1	3
4	Consent	Female	High	5	5	4	5	1	5
5	Consent	Female	Low	5	5	5	4	3	5
6	Consent	Female	High	1	3	2	2	2	2
7	Consent	Male	Medium	4	4	1	3	1	1
8	Consent	Female	Medium	4	3	1	3	1	3
9	Consent	Male	High	4	4	1	3	1	3
10	Consent	Male	Low	4	4	1	2	1	1
				Averages for AQs					
				3,4	3,9	2	3,2	1,3	2,8

Figure C.8: AQ-A Scores

Additional Questions (AQs)										
Participant	Consent	Gender	Technical Competence	AQ1-B	AQ2-B	AQ3-B	AQ4-B	AQ5-B	AQ6-B	Final Question
1	Consent	Female	High	5	4	5	5	5	5	StudentGreenTravel
2	Consent	Male	High	4	4	3	5	4	5	StudentGreenTravel
3	Consent	Female	High	5	4	5	5	5	5	StudentGreenTravel
4	Consent	Female	High	4	4	4	5	5	5	StudentGreenTravel
5	Consent	Female	High	5	5	5	5	5	5	StudentGreenTravel
6	Consent	Female	Low	5	5	5	5	4	5	StudentGreenTravel
7	Consent	Male	Medium	5	5	5	5	5	5	StudentGreenTravel
8	Consent	Female	Medium	5	5	5	5	5	5	StudentGreenTravel
9	Consent	Male	High	5	5	5	5	5	5	StudentGreenTravel
10	Consent	Male	Low	4	4	4	4	4	5	StudentGreenTravel
				Average for AQs						
				4,7	4,5	4,6	4,9	4,7	5	

Figure C.9: AQ-B Scores

C.2.3 T-tests

SUS T-test					
SUS-A	**SUS-B**		**t-Test: Two-Sample Assuming Equal Variances**		
17,5	97,5				
90	87,5			*SUS-A*	*SUS-B*
80	97,5		Mean	90,5	54,75
97,5	85		Variance	70,55556	1081,181
97,5	100		Observations	10	10
30	85		Pooled Variance	575,8681	
25	90		Hypothesized Mean Difference	0	
47,5	97,5		df	18	
40	92,5		t Stat	3,331191	
22,5	72,5		P(T<=t) one-tail	0,001858	
			t Critical one-tail	1,734064	
			P(T<=t) two-tail	0,003716	
			t Critical two-tail	2,100922	

Figure C.10: SUS T-test

AQ T-test					
AQA	**AQB**		**t-Test: Two-Sample Assuming Equal Variances**		
3,4	4,7				
3,9	4,5			*AQ-A*	*AQ-B*
2	4,6		Mean	2,766667	4,733333
3,2	4,9		Variance	0,922667	0,034667
1,3	4,7		Observations	6	6
2,8	5		Pooled Variance	0,478667	
			Hypothesized Mean Difference	0	
			df	10	
			t Stat	-4,92351	
			P(T<=t) one-tail	0,000301	
			t Critical one-tail	1,812461	
			P(T<=t) two-tail	0,000602	
			t Critical two-tail	2,228139	

Figure C.11: AQ T-test

C.2.4 ANOVA Tests

Anova: Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
Technical Competence	10	24	2,4	0,711111
SUS-A	10	5,475	0,5475	0,108118

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	17,15878	1	17,15878125	41,89006	4,35E-06	4,413873
Within Groups	7,373063	18	0,409614583			
Total	24,53184	19				

Figure C.12: ANOVA test between SUS-A scores and Participants' Technical Competence

Anova: Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
Technical Competence	10	24	2,4	0,711111
SUS-B	10	9,05	0,905	0,007056

ANOVA

Source of Variation	SS	df	MS	F	P-value	F crit
Between Groups	11,175125	1	11,17513	31,12126	2,7E-05	4,413873
Within Groups	6,4635	18	0,359083			
Total	17,638625	19				

Figure C.13: ANOVA test SUS-B scores and Participants' Technical Competence

Bibliography

Abras, C., Maloney-Krichmar, D., Preece, J., et al. (2004). User-centered design. *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications*, 37(4):445–456.

Andrews, J. and Leach, A. (2024). Home | SIMAP — unhsimap.org. `https://unhsimap.org/home`. [Accessed 26-02-2024].

Arioli, M. S., Márcio de Almeida, D., Amaral, F. G., and Cybis, H. B. B. (2020). The evolution of city-scale ghg emissions inventory methods: A systematic review. *Environmental Impact Assessment Review*, 80:106316.

Ascui, F. and Lovell, H. (2012). Carbon accounting and the construction of competence. *Journal of Cleaner Production*, 36:48–59.

Bailey, G. and LaPoint, T. (2016). Comparing greenhouse gas emissions across texas universities. *Sustainability*, 8(1):80.

Bangor, A., Kortum, P. T., and Miller, J. T. (2008). An empirical evaluation of the system usability scale. *Intl. Journal of Human–Computer Interaction*, 24(6):574–594.

Birnik, A. (2013). An evidence-based assessment of online carbon calculators. *International Journal of Greenhouse Gas Control*, 17:280–293.

Boucher, C. and Altamimi, Z. (2001). Itrs, pz-90 and wgs 84: current realizations and the related transformation parameters. *Journal of Geodesy*, 75:613–619.

Brand, C. and Preston, J. M. (2010). '60-20 emission'—the unequal distribution of greenhouse gas emissions from personal, non-business travel in the uk. *Transport Policy*, 17(1):9–19.

Chen, J., Chen, C., Xing, Z., Xia, X., Zhu, L., Grundy, J., and Wang, J. (2020). Wireframe-based ui design search through image autoencoder. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 29(3):1–31.

Choudaha, R. and Chang, L. (2012). Trends in international student mobility. *World Education News & Reviews*, 25(2).

Commission, E. et al. (2019). The european green deal.

Cortese, A. D. (2003). The critical role of higher education in creating a sustainable future. *Planning for higher education*, 31(3):15–22.

Dautremont-Smith, J., Cortese, A., Dyer, G., and Walton, J. (2009). Implementation guide: Information and resources for participating institutions.

Davies, J. C. and Dunk, R. M. (2015). Flying along the supply chain: accounting for emissions from student air travel in the higher education sector. *Carbon Management*, 6(5-6):233–246.

Dias, A. C. and Arroja, L. (2012). Comparison of methodologies for estimating the carbon footprint–case study of office paper. *Journal of Cleaner Production*, 24:30–35.

Durojaye, O., Laseinde, T., and Oluwafemi, I. (2020). A descriptive review of carbon footprint. In *Human Systems Engineering and Design II: Proceedings of the 2nd International Conference on Human Systems Engineering and Design (IHSED2019): Future Trends and Applications, September 16-18, 2019, Universität der Bundeswehr München, Munich, Germany*, pages 960–968. Springer.

Earp, J. B. and Payton, F. C. (2001). Data protection in the university setting: employee perceptions of student privacy. In *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, pages 6–pp. IEEE.

Eggleston, H S, B. L. M. K. N. T. and Tanabe, K. (2006). 2006 ipcc guidelines for national greenhouse gas inventories.

for Energy Security, D. and Zero, N. (2020). Carbon calculator — gov.uk. `https://www.gov.uk/guidance/carbon-calculator`. [Accessed 26-02-2024].

Franchetti, M. J. and Apul, D. (2012). *Carbon footprint analysis: concepts, methods, implementation, and case studies*. CRC press.

Gaffron, P. and Niemeier, D. (2015). School locations and traffic emissions—environmental (in) justice findings using a new screening method. *International journal of environmental research and public health*, 12(2):2009–2025.

Gómez, N., Cadarso, M.-Á., and Monsalve, F. (2016). Carbon footprint of a university in a multiregional model: the case of the university of castilla-la mancha. *Journal of Cleaner Production*, 138:119–130.

Güereca, L. P., Torres, N., and Noyola, A. (2013). Carbon footprint as a basis for a cleaner research institute in mexico. *Journal of Cleaner Production*, 47:396–403. Cleaner Production: initiatives and challenges for a sustainable world.

Hanemann, U. (2015). *Transforming Our World: Literacy for Sustainable Development.* ERIC.

Helmers, E., Chang, C. C., and Dauwels, J. (2021). Carbon footprinting of universities worldwide: Part i—objective comparison by standardized metrics. *Environmental Sciences Europe*, 33:1–25.

HESA, H. E. S. A. (2014). Estates Management | HESA — hesa.ac.uk. `https://www.hesa.ac.uk/data-and-analysis/estates`. [Accessed 12-02-2024].

IEA (2020). *Energy Technology Perspectives 2020*.

ISO (2018). Iso 14064-1:2018. `https://www.iso.org/obp/ui/#iso:std:iso:14064:-1:ed-2:v1:en`. [Accessed 25-02-2024].

Jonsson, E. (2023). Research and development of a methodology to calculate student travel to study emissions.

Karney, C. F. (2013). Algorithms for geodesics. *Journal of Geodesy*, 87:43–55.

Kim, S. and Zhang, C. (2022). Factors influencing korean students' choice of study abroad destination short-term and long-term by destination country. *Asia Pacific Education Review*, 23(1):197–208. Publisher Copyright: © 2021, Education Research Institute, Seoul National University, Seoul, Korea.

Klein-Banai, C. and Theis, T. L. (2013). Quantitative analysis of factors affecting greenhouse gas emissions at institutions of higher education. *Journal of Cleaner Production*, 48:29–38.

Koester, R. J., Eflin, J., and Vann, J. (2006). Greening of the campus: a whole-systems approach.

Journal of cleaner production, 14(9-11):769–779.

Kusano, K., Nakatani, M., and Ohno, T. (2013). Scenario-based interactive ui design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 391–394.

Letete, T., Mungwe, N. W., Guma, M., and Marquard, A. (2011). Carbon footprint of the university of cape town. *Journal of Energy in Southern Africa*, 22(2):2–12.

Matthews, H. S., Hendrickson, C. T., and Weber, C. L. (2008). The importance of carbon footprint estimation boundaries.

Nature, S. (2020). Carbon Management & Greenhouse Gas Mitigation - Second Nature — secondnature.org. `https://secondnature.org/signatory-handbook/carbon-management-greenhouse-gas-mitigation/`. [Accessed 11-02-2024].

Nerur, S. and Balijepally, V. (2007). Theoretical reflections on agile development methodologies. *Communications of the ACM*, 50(3):79–83.

Ozawa-Meida, L., Brockway, P., Letten, K., Davies, J., and Fleming, P. (2013). Measuring carbon performance in a uk university through a consumption-based carbon footprint: De montfort university case study. *Journal of Cleaner Production*, 56:185–198.

Quin, F., Weyns, D., Galster, M., and Silva, C. C. (2024). A/b testing: a systematic literature review. *Journal of Systems and Software*, page 112011.

Ranganathan, J. and Corbier, L. (2011). Greenhouse gas protocol. *Sector Toolsets for Iron and Steel-Guidance Document*.

Ritchie, H. (2020). Cars, planes, trains: where do co2 emissions from transport come from? *Our World in Data*. https://ourworldindata.org/co2-emissions-from-transport.

Robinson, O. J., Tewkesbury, A., Kemp, S., and Williams, I. D. (2018). Towards a universal carbon footprint standard: A case study of carbon management at universities. *Journal of Cleaner Production*, 172:4435–4455.

Rosson and Carroll (2000). Scenario-based design of human-computer interactions. *Carroll [2000], Massachusetts Institute of Technology*.

Roy, A., Newman, A., Ellenberger, T., and Pyman, A. (2019). Outcomes of international student mobility programs: A systematic review and agenda for future research. *Studies in Higher Education*, 44(9):1630–1644.

Saabith, A., Fareez, M., and Vinothraj, T. (2019). Python current trend applications-an overview. *International Journal of Advance Engineering and Research Development*, 6(10).

Schaltegger, S. and Csutora, M. (2012). Carbon accounting for sustainability and management. status quo and challenges. *Journal of Cleaner Production*, 36:1–16.

Sekoai, P. T. and Yoro, K. O. (2016). Biofuel development initiatives in sub-saharan africa: Opportunities and challenges. *Climate*, 4(2).

Serafini, P. G., de Moura, J. M., de Almeida, M. R., and de Rezende, J. F. D. (2022). Sustainable development goals in higher education institutions: a systematic literature review. *Journal of Cleaner Production*, page 133473.

Shields, R. (2019). The sustainability of international higher education: Student mobility and global climate change. *Journal of Cleaner Production*, 217:594–602.

Simpson, W. (2009). Cool campus! a how-to guide for college and university climate action planning. *Association for the Advancement of Sustainability in Higher Education*.

Specification, P. A. et al. (2008). Specification for the assessment of the life cycle greenhouse gas emissions of goods and services. *Bsi Br. Stand. Isbn*, 978:580.

Thiruvenkatachari, R., Su, S., An, H., and Yu, X. X. (2009). Post combustion co2 capture by carbon fibre monolithic adsorbents. *Progress in Energy and Combustion Science*, 35(5):438–455.

Townsend, J. and Barrett, J. (2015). Exploring the applications of carbon footprinting towards sustainability at a uk university: reporting and decision making. *Journal of Cleaner Production*, 107:164–176.

UNH Sustainability Institute (2018). Simap user's guide. https://unhsimap.org/sites/default/files/user-uploads/SIMAP%20User%20Guide_DRAFT6.2_2.21.2018.pdf. [Accessed 27-02-2024].

Valls-Val, K. and Bovea, M. D. (2021). Carbon footprint in higher education institutions: a literature review and prospects for future research. *Clean Technologies and Environmental Policy*, 23(9):2523–2542.

Valls-Val, K. and Bovea, M. D. (2022). Carbon footprint assessment tool for universities: Co2unv. *Sustainable Production and Consumption*, 29:791–804.

Vásquez, L., Iriarte, A., Almeida, M., and Villalobos, P. (2015). Evaluation of greenhouse gas emissions and proposals for their reduction at a university campus in chile. *Journal of Cleaner Production*, 108:924–930.

Walker, P. (2014). International student policies in uk higher education from colonialism to the coalition: Developments and consequences. *Journal of Studies in International Education*, 18(4):325–344.

Whitmarsh, L., Seyfang, G., and O'Neill, S. (2011). Public engagement with carbon and climate change: To what extent is the public 'carbon capable'? *Global environmental change*, 21(1):56–65.

Williams, I., Kemp, S., Coello, J., Turner, D. A., and Wright, L. A. (2012). A beginner's guide to carbon footprinting. *Carbon Management*, 3(1):55–67.

World Resource Institute (2016). Data Explorer | Climate Watch — climatewatchdata.org. https://www.climatewatchdata.org/data-explorer/historical-emissions?historical-emissions-data-sources=cait&historical-emissions-gases=co2&historical-emissions-regions=All%20Selected&historical-emissions-sectors=total-including-lucf%2Ctransportation&page=1&sort_col=country&sort_dir=ASC. [Accessed 29-02-2024].

WRI and WBCSD (2013). Technical guidance for calculating scope 3 emissions. *Supplement to the Corporate Value Chain (Scope 3), Accounting & Reporting Standard, in partnership with the Carbon Trust*.