**Pymiles**

Contents

[Introduction 2](#_Toc111914801)

[Roadmap of the report 2](#_Toc111914802)

[Background 3](#_Toc111914803)

[Project Requirements 3](#_Toc111914804)

[Non-technical requirements. 3](#_Toc111914805)

[Non-technical resources 4](#_Toc111914806)

[Technical requirements 4](#_Toc111914807)

[Design and architecture 5](#_Toc111914808)

[IMPLEMENTATION AND EXECUTION 6](#_Toc111914809)

[Development approach and team member roles 6](#_Toc111914810)

[Tools and libraries – packages – see above 7](#_Toc111914811)

[Implementation process (achievements, challenges, decision to change something) 7](#_Toc111914812)

# Introduction

The Pymiles project is a project to capture personal mileage expenses using Google Maps history, select the required items and then send to accounting department or accountant. The idea is to reduce work for people who already lead busy lives and perform the calculation automatically.

Objectives/Aims

The object of project Pymiles was to automate UK expense claims.

For employees, company directors and self-employed people,

recording business mileage on a daily basis is an unwanted chore, which is difficult to fit in to a busy daily schedule.

If it is missed out, then several weeks later, it may be necessary to resort to painstakingly tracking back through Google maps data by searching one’s phone and then entering mileage information into an excel spreadsheet.

The object of project Pymiles was to automate this process by displaying the previous month’s data, allowing selection and then automatically email and calculate the expense claim.

Alongside this, further aims were to develop some project planning skills, implement some object-oriented programming, display the use of skills learned in the course, and to push oneself to learn new skills and find out what else is possible using python.

# Roadmap of the report

A picture containing background pattern

Description automatically generated

Timeline

Description automatically generated

# Background

In the UK the government tax body, known as HMRC, allows mileage costs to be tax deductible.

For an employee of a company, this means that the company will refund mileage expenses when the employee uses their own car on company business

The employee submits a report with mileage details, often in a spreadsheet to the payroll or accounts payable department.

Company directors or self-employed people must keep a record in the same way as employees, although they would reimburse themselves from their self-employed earnings or company account, thus the money would be tax deductible. For these

types of people, the expenses records might be sent to an accountant rather than accounts payable.

Currently in the UK, the first 10,000 miles are reimbursed at 45p (i.e. 0.45 of a British pound) per mile up to 10,000 miles and at 25p per mile for mileage that is over 10,000 miles

# Project Requirements

Scope of the project

To produce a beta version which can run over a downloaded monthly Google maps data file on a personal or company Windows or Mac OS operating system

Out of scope

Phone or tablet operating system.

The user will be responsible for downloading the own Google maps history file and renaming it to “Google\_history\_download\_monthly.json”

# Non-technical requirements.

1. Software should extract data from the above json file selecting “IN\_PASSENGER VEHICLE” data to exclude non car data

Sort and extract the relevant data from the json input, as Google data records any travel at all including, walking, cycling, bus etc. then for the purpose of this project select only the travel in a passenger vehicle must be extracted.

1. Data to be displayed in a GUI tree structure displaying from and to locations, the distance in metres, and time and date
2. User able to de-select non work-related records by selecting a line and clicking delete button
3. Use able to click Submit, this will initiate background processes to create a
4. File named new\_monthly\_claim.csv
5. Submit will then load the database file “Mileage” from “new\_monthly\_claim.csv”In addition to this produce an option to email the accounts department or account and attach the file created in the form of a .csv file.
6. A second GUI will allow the calculation process to be submitted and this will update the Personclaims database file in the following manner:

“claimed\_this\_month” to overwrite “claimed last month

“claimed\_this\_month to be updated with calculated claim amountTotal\_claimed to be accumulated ny adding on claimed this monthTotal miles to be accumulated using this month’s milage

8.Statistics button can be pressed to display a chart generated from the database shoeing amount claimed by each person for this month

Future requirements

User log in and registrationMasking of home data for security – setting of home postcode to GDPRValidation of home email for submit button. The email is currently mocked in the codeImprovement of matplotlib functionProper secure processing of the email function or removal of email function

# Non-technical resources

In an ideal world this project would have required a Product Owner, Scrum master a developer and some testing resources.

# Technical requirements

Python 3.9 or greater

Mysql 8.0

Both a Windows and a MAC were used in production of this project, although this added time as the GUI produced on Windows did not function on the MAC hence different versions were produced

Pycharm was used as IDE

Python packages required are:

Csv,Json,deepcopy

Os,tkinter,on MAC OS

tkMacosx – not required on windows

yagmail,mysql.connector,matplotlib

# Design and architecture

Architecture is demonstrated below

Diagram

Description automatically generated

Program flowDiagram

Description automatically generated

# IMPLEMENTATION AND EXECUTION

# Development approach and team member roles

Agile sprints were used, and 4 weekly prints were planned using the free online tool Click Up as follows:

Graphical user interface, text, application, email

Description automatically generated Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application

Description automatically generated

PLEASE NOTE

I used 4 sprints but had an issue with Git for the fourth, so unfortunately

It does not appear as a 4th sprint in GitHub

Team member roles

As I worked alone I did not have team member roles but if I had then I would have liked to work alongside a second software engineer for the coding and had a scrum master to organise the team and create the agile sprint documentation.

A person to oversee testing and objectively review and test my code would also have improved the project, and a product owner would have been most useful for the next phase of development

It was an achievement to produce this myself but I think that having help with decision making and when to STOP pursuing an avenue would have been the most useful asset to this project, as I tried many different thing and a second person to help see a more global view of the project would have been more productive.

Development was iterative but particularly testing as every time a change was made or a function run then testing was carried out. I could not think of a way to employ unittest with the GUI approach and therefore decided to use an End to End test instead. This is documented in Github

# Tools and libraries – packages – see above

Github MANY Python websites,In particular stack overflow and Codemy

# Implementation process (achievements, challenges, decision to change something)

Security and accuracy generally could be improved upon

Testing could be improved upon

The biggest challenge I had was that I began work on a Windows machine and later continued on a Mac and found to my horror that a few functions did not work on a Mac and had to refactor them The decision to create a GUI was based on it being sensitive data relating to personal tax information and learning the security side of web development would take too long and be outside of the scope of this project.

The thought that learning front end development for a project to be created by one person would be too much given the time allowed. That turned out to be incorrect as in fact, learning and implementing a GUI took a long time and was very challenging particularly as it had not been covered at all on the course.

However, once the GUI side had been achieved then it was great fun to choose the Code First Girls colours for use in the GUI.

Another challenge was trying to implement OOP in the project, which is something that I would improve on for future releases. I did try to design my Buttons as children of an abstract class but unfortunately this did not work out with a GUI although it could have worked with back-end functions much more easily

**TESTING AND EVALUATION**

Time constraints meant that I was unable to implement unit tests, I also couldn’t think how it would be possible while using a GUI

Instead I opted for a End to End testing approach

System limitations

I would have liked to design a login screen.

The email was not secure and not the most correct way to do an email

**CONCLUSION**

Many improvements could be made to this small project but I do feel that the functionality and database fit together reasonably well and the appearance is reasonable. It also fulfils a useful function.

Python coding skills were improved along with design skills and an over grasp of project management was attained.