**A Level computer Science**

Component 3

£xpense Tracker

(Expense Tracker)

Perhaps a Logo / Picture

By: Cameron McQuade

For: A. Issa

**Stoke on Trent Sixth Form College**

Table of Contents

[Chapter One: Analysis of the problem 3](https://stokesfc.sharepoint.com/sites/Computing/Shared%20Documents/Computing%20Project/Programming%20project%20Templates/NEW%20Analysis%20Section.docx#_Toc136004608)

[1.1 Introduction 4](#_Toc136004609)

[1.2 Problem Identification 5](#_Toc136004610)

[1.3 Possible Computational Methods 6](#_Toc136004611)

[1.4 Stakeholders analysis 7](#_Toc136004612)

[1.4.1 who are Stakeholders ? 7](#_Toc136004613)

[1.4.2 How they make use of the solution? 7](#_Toc136004614)

[1.4.3 Why the solution is appropriate to them? 7](#_Toc136004615)

[1.4.3 Stakeholders’ involvement (interview and conclusion) 7](#_Toc136004616)

[1.5 Research of solutions for similar problems 8](#_Toc136004617)

[1.6 Hardware and software requirements 9](#_Toc136004618)

[1.7 The requirements of the solution 10](#_Toc136004619)

[1.8 Features of the solution 11](#_Toc136004620)

[1.9 Success Criteria 12](#_Toc136004621)

[1.10 Limitations of the solution 13](#_Toc136004622)

# Chapter One: Analysis of the problem

## Introduction

After hearing concerns from my parents about tracking their business’s expenses, I knew I had to create a perfect solution. They informed me how often expense software crashes, or inputting lots of data can be tedious, or an error can occur – which sometimes removes all their tracking progress. My solution tries to be a lightweight, user-friendly app with an interactive GUI, reducing this problem into a simple task. Furthermore, the information displayed on a typical expense tracker can be limited, or hard to understand. These problems often mean that business owners try other methods, such as storing all expenses in one spreadsheet, which can be hard to navigate or the file is so large, modifications cannot be made without crashes or severe lagging is experienced. My app will include a method to add, remove, edit or download expenses as well as graphs and tables which will allow the user to see yearly or all expenses. There will also be a predictor that uses machine learning to predict how the expenses and expense amount will change in the future. There will be two types of users: default users and admins. Admins will have access to more features, such as a simple yet effective account system which allows admin users to add, remove, edit or view all users. Admins will also be able to use money managing/ budgeting methods which will include a loan calculator, a budgeting calculator, a currency converter and an income graph. The ability to manipulate the database like adding test data or wiping all data is also a feature specific to admin users. My solution will be best suited for my stakeholders, allowing me to create these great features with lots of customizability. This will create the perfect experience for the user.

## 1.2 Problem Identification

Nowadays, software has become overly bloated and lots of memory is used for even the simplest task. This becomes apparent in apps such as trackers. Many bugs and severe lag cause users to opt for better solutions, such as inefficient drag and drop programs or huge spreadsheets – which usually take too much time or too much space in the memory. This quickly becomes unorganised, while the user loses motivation; problems like this can cause the whole business to become disorientated and unregulated. Furthermore, generic applications do not contain specialised features – making the process feel boring.

A tailored piece of software will allow my shareholders to easily track, edit and view their expenses much simpler meaning budgets or targets can be seen and met far easily – compared to overly complicated software, which often reduces performance and improvement is not seen. In my software, lots of data will be displayed graphically or in tables – which allows for better visualisation of where they are economically overall or within the year, month, or day.

My main aim is to create a useful desktop application, where the main features consist of adding, removing, editing and downloading expenses, view expense graphs and tables, use a future expense prediction method, adding, removing, editing and viewing accounts for other employees to use, calculators which help the user with budgeting compared to monthly expenses and calculate how much is needed to paid with a loan calculator, a currency converter, income graph, database testing methods, and a system for users to retrieve passwords if they are forgotten. There will be other features too: changing the style/ theme of the application, adding an FAQ page to help users who are unsure or stuck, adding admin accounts who have permissions to add or delete accounts, being able view expenses in a table format, and being able to change the graph type in the expense graph screen. These features will create a unique platform for users to experience a well-rounded, and easy to use expense tracker.

Another feature to be added is a target budget setter, where an admin-level user can enter an amount of money which the company will try to stay under or go over – depending on the settings chosen. My application will automatically calculate, using data extrapolation, if the business is on track to reach this target, or if they are not. This will hopefully see an assistance in money-spending visualisation, allowing the user to feel like they are progressing properly.

Problems and the lack of features in modern software creates businesses to become unaware of their spending habits or if they are over-budget. This becomes a further problem if they require a certain amount of money at the end of each year or quarter. Removing these issues and creating a positive environment will be key in my software – as I want the application to be trouble-free and a smooth experience.

## Possible Computational Methods

My application will use a variety of computational methods due to the type of data, and how it is stored and organised. A method my app could use is decomposition:

Decomposition involves breaking down a complex problem or system into smaller parts that are more manageable and easier to understand. I can use decomposition to divide up my project into easy, manageable pieces – which is more efficient to create. Furthermore, decomposition will be used inside the code when expenses are shown on a graph; data is searched and retrieved, it is placed into the lists which will be used for the different axis, then the data is displayed visually. This process of modular programming (breaking down the data into separate components) will be more efficient and is perfect use of a computational method.

Another method my app could use is abstraction:

Abstraction is the process of removing unnecessary details, and only focusing on the important, essential details. This will help to declutter the code; maintaining efficiency, ease of use, and reducing pointless features which only result in confusion on the user’s end. This ‘breaking down’ of the code allows my application to be accustomed to an object-oriented approach, further showing how it is great for a computational method solution.

Furthermore, a method that will be used is iteration:

Iteration means repeating a block of code for several times or until a condition is met. This is in different scenarios in my app. An example is adding multiple expenses in one go. The code will iterate repeatedly, adding these expenses into the database. Another example is when the expenses are being viewed graphically, as the graph type may change or the time frame (yearly, monthly) may change, so the graph must be changed. This task will take longer if each piece of data is fetched one by one. It is much more efficient to use iteration to take lots of data, which will be displayed graphically much faster.

A final method to be used is efficient storage of data:

Storing lots of data and expenses in one text file or one spreadsheet swiftly becomes unorganised and inefficient to search through. It is much more effective to store this information in a database. Multiple tables can be using in a relational database for a fast, useful way to get the data back. In addition, data can be organised much quicker, while also reducing data redundancy or data repetition. This increase in performance and structure allows business to quickly see expenses. Therefore, this is another computational method which will severely benefit the app.

## 1.4 Stakeholders analysis

### 1.4.1 who are Stakeholders ?

My stakeholders will be my parents. They operate a conveyancing company and have lots of clients who heavily rely on the business to help them with taking care of the legal aspects of transferring property ownership. This application can help them store the money spend, used for quotes or accessing deeds to properties. Furthermore, this program could be used to store the money coming into the business, helping them keep track of customers.

My father, director of the firm, will be a great stakeholder for this software, since he can try it out and give his feedback back to me easily. He will test all the features and view if it is of benefit to the business. If it is of use, it may be viable to make the application more suitable to a wider audience, allowing more companies to easily track their expenses and budget goals. There are plenty of computers in the office, so the application can be installed on all the machines – so it can be accessed by anyone at any time within the company.

My mother, practice founder and director, will also be a stakeholder. She said she will take great interest in looking at the statistics visually since it will be of great help. In addition, the fee earners within the company will be able to better track the financial setting of the business. Being able to efficiently track all money in and out of the business could skyrocket their performance, and being able to retrieve and view all the data will save lots of time which would have been spent looking hopelessly through a spreadsheet.

### 1.4.2 How they make use of the solution?

Each person in the business will be made a unique account, in which they can add and remove expenses at their own will. This application will make this experience much faster whilst also being very efficient at viewing this inputted data. Whilst remaining light, this program will offer a variety of key features, which each employee can freely use, and make good use of.

### 1.4.3 Why the solution is appropriate to them?

As explained by my father, multiple solutions have been tried and tested previously. From spreadsheets to drag and drop databases, nothing has worked. Unorganised data and huge file sizes cause many errors, leading to mistakes which the business must deal with. ‘A neat, unambiguous program is what the business needs’ in his words sparked a flare in my mind; I want to put a stop to these problems. Every computer will be effortlessly able to run and operate my solution, allowing for lots of time to be saved which can be used helping clients.

### 1.4.3 Stakeholders’ involvement (interview and conclusion)

I had some questions written down to ask my parents, to help me better understand the problem and allow me to create the best program for them:

Q1. How do you currently track your expenses?

Q2. What are the pros and cons of this software?

Q3. Why do you need to track your expenses?

Q4. How do you want to track your expenses?

Q5. What key features should a great expense tracker include?

Q6. What is the most important feature you would want and why?

Q7. How many people currently track the businesses expenses currently?

I believe these questions are vital in helping me better understand what my stakeholders want from this project, and equally importantly how it will benefit them. Here is a summary of the interview of my stakeholders:

Q1: We’ve use multiple of Microsoft products, from excel to access, which each have their benefits, but nothing quite clicked. There were issues with both apps, and we eventually gave up as too much time was being wasted.

Q2: Microsoft excel is easy to start but hard to master, and problems quickly started when the file grew. It took ages to open, and when it did open it kept freezing or just crashed. Access was similar but took some time to first set up. Also, there was some issues where the data was being changed but not saved and the queries were sometimes returning data which we didn’t want.

Q3: We think it is beneficial to track the purchases we make and the money we receive from customers. It can help grow the business and it shouldn’t be a hard task to store.

Q4: We require a simpler application which can store lots of data without any problems. We want the app to be usable by everyone, since this will help us the most. The best apps we’ve seen include graphs and spreadsheets but always come at a price or a subscription. We don’t want this as if it does not suit us, it feels like a waste of money.

Q5: Key features should be graphs to visualise the purchases, a table of the expenses, a way to add and delete accounts, and a simple way to add and remove the expenses.

Q6: We think the most important feature will be the table of expenses, as it can be seen by all the employees and easy to edit. This will hopefully save time and money in the long run.

Q7: Currently, only my father and the fee earner in the business track the expenses. My parents hope this can change, so the job can be done by anyone. This will be much more efficient as people can be easily trained to use to application and track lots of purchases within minutes.

## 1.5 Research of solutions for similar problems

There is an abundance of expense trackers on the market, available for use but lots have a pay wall to access the best features. Here is a photo of a free expense tracker from Microsoft store called MoneyPoint: A screenshot of a computer

Description automatically generated

Lots of key features are shown here, which I will implement in my own software. However, I feel the main GUI may be too cluttered, or too complex for a new user to deal with. Lots of these features can be added in the ‘settings’ tab. MoneyPoint uses a variety of methods to store data, such as using excel to store purchase in a spreadsheet and makes use of databases to store data. The app has a feature to import ‘test data’, which will help me to improve my own app. Here is a photo of the test data in the ‘Account’ tab:

A screenshot of a computer

Description automatically generated

I showed my stakeholders this app, and they indicated that they liked it since there were a lot of features, but they said that they may not use it as it could take too much time to learn and set up. One way this is seen, is in the ‘Help’ tab, as it only offers short sentences of information to the user: A screenshot of a computer

Description automatically generated

My stakeholders also wanted an option to graphically see the expenses, which isn’t apparent in this application. However, an application which does make use of graphs is ‘Spending Tracker’:

A screenshot of a computer

Description automatically generated

My stakeholders agreed that the graphing system on this software was good, yet there were still some problems. The software had an ‘out of date’ feel to it, while also performing slowly, and having features such as ‘recurring expenses’ locked behind a paywall:

A screenshot of a computer

Description automatically generated

This is of no use to my stakeholders. Furthermore, there was no option to automatically add test data to the tracker, and the help button in settings did not work. There is also no option to change the colour of the UI. These issues swiftly made this program a no-go for my stakeholders. They told me that these apps had some key features, yet most lacked some details or specific elements, which make a great expense tracker.

## 1.6 Hardware and software requirements

Hardware:

* The user will need a computer with the ability to use python. My solution will be lightweight, so no high-end parts will be required for the computer. Furthermore, the graphs displayed will be easily shown using integrated graphics. RAM is not also a problem as the program will not be completing many calculations.
* The user will need some basic input devices, such as a mouse and a keyboard – so data can be inputted into the program. The user will also need a monitor to view the application.

Software

* An operating system which can run python 3.12 applications. Most operating systems can handle this such as: Windows 7 or later, Linux, and MacOS.
* Python 3.12 with some modules installed: Tkinter, MySQL, Matplotlib, Pandas, etc.

## 1.7 The requirements of the solution

There are many requirements to the solution. For example, the user interface:

The user interface is a key aspect to any program, so it is vital that it is simple to understand whilst keeping all the features. There will be clear and labelled buttons, and boxes to help guide the user to the best experience. Being customisable is also a key component to my program, not only so the style can suit the user – but my application needs to be accessible by people who may be colourblind, or struggle with some neurological disabilities such as dyslexia.

There will also be many features added to the solution, so the user can have a stress-free experience whilst also being efficient.

## 1.8 Features of the solution

My application will be packed with features, all to benefit the user. Some of them include:

- Adding and removing accounts so many employees can input expenses at the same time.

- Editing account information, such as passwords if they are forgotten or changing an account type from default to admin.

- Adding expenses, which is the whole premise of the application. Making this as efficient and effective as possible will be my primary concern.

- Removing or editing expenses if the data entered is wrong.

- Viewing the expenses in a graphical way, while also being able to change the graph type and time frame of the graph. This is a key aspect as it can allow the user to easily view their progress.

- Viewing the expenses in a table/ spreadsheet, as some users are more accustomed to using applications like excel. This accessibility will allow any user to easily view all expenses.

- Adding a target budget, which automatically works out when debts are paid off, or when the budget is reached. This is another key feature to my application since it will save time working this out manually.

- Input checkers, to ensure the data enters matches the correct format: this will keep the code more efficient since it is better to check locally, rather than returning an error from the database.

- A server database using MySQL. Having an online database allows multiple employees to work together in the tracker since if the database is local, every time an update is made, the database will have to be sent to everyone. Furthermore, it is efficient to use the correct type of database. I will be using a relational database as there will be multiple tables. This will aim to remove data repetition and redundancy.

## 1.9 Success Criteria

|  |  |  |  |
| --- | --- | --- | --- |
| Features | What makes it successful? | Proof of its success | How/ why is it successful? |
| User interface | A good user interface is easy to navigate whilst containing all features. It should be customisable and simple to use. | Screenshots will be provided, showing the user interface working as expected whilst having aa clear, concise design. | One of the most key aspects to any application is the user interface. It is the visuals of the program, and any user wants to have the best experience with it. |
| Account Login | A simple login screen boosts programs authenticity, which will be provided in a clear frame. Help buttons also guide the user to ensure it is an easy process to log in. | Screenshots of the program will show the input system, and the help system, showing its usefulness. | Another great aspect to a program. There will be a clear input system, guiding the user to enter their username and password. Buttons will also be provided if the user has forgotten their password, allowing for a better experience with the program. |
| Account creation | A clear design for the account creation, whilst having a secure system to store the accounts. Options to remove or edit accounts will also be available. | Screenshots of the account creation screen and information stored in the database will show how the application deals with information | The account will be stored in a database. The passwords will be hashed using a custom module, ensuring security within the program. Encrypting the passwords is key in an online application as businesses can lose lots of data if account information is compromised. |
| Entry boxes and buttons | Clear and designated widgets to enter information or proceed to another area of the program allows the user to learn this application faster | Screenshots of the GUI showing a clear layout with many buttons, and entry boxes in designated areas will portray how the program is made to keep simplicity. | Having a cluttered GUI full of buttons looks unprofessional. Widgets are the main way of navigating through the program, so staying unambiguous is a must. |
| Adding, removing an editing expense | Users should be able to efficiently enter lots of expenses, whilst also having a simple way to edit or remove them if mistakes are made. | Screenshots will show the effective way the user can input lots of data, while maintaining minimalism. | The main goal is keeping this task the easiest it can be, as it is the main task the user will be doing in the application. |
| Visually showing the expenses | Users can view the expenses either via graphs or in a table format. This will suit any user, who can change the type of graph or time frame of expenses. | Screenshots showing expenses in different graphical forms and in a table / spreadsheet. | This is another primary feature, as it allows users to keep the best track of their money. It needs to be clear and straightforward to navigate or change. |
| Settings | Users can access the settings at any point within the application, and it should be easy to change settings. There will be a variety of options, such as changing the theme of the UI, or changing the default graph type or time frame. | Screenshots showing the settings being easily accessible and containing all the settings for the user to have a customisable application. | The settings allow the user to make the application feel like their own. They need to be easily navigable and offer a variety of options. |
| Database | The database needs to be online, and accessible by the application so data can be efficiently retrieved for the user. | Screenshots showing the database, and how data is stored within it. | An online relation database will be essential for this application, so multiple people can access it at one time. It needs to be able to store account information, the account type, and expenses. |

## 1.10 Limitations of the solution

There may be some limitations of my application. For example, the online database may be hard to set up, and will require a device to constantly keep it running. Furthermore, it will require an internet connection to connect and make changes to. Most devices are always connected to the internet, yet in some areas this is not always an option.

Another limitation is the python modules. To keep my application lightweight but keep a professional feel, I have opted to use Tkinter. However, this isn’t as hugely customisable as some larger modules like PyQt or other versions of Tkinter, such as CustomTkinter. These modules allow for much greater looking applications at the cost of performance, where a better-quality device is required to run it. In addition, if the application was built as a website application using a module like flask, the customisation would be much larger for the user. however, Tkinter is more flexible and easier to implement into a program like mine.