**Tanzil Chowdhury**  
**Computer Science NEA**

**Analysis**

**Problem Definition and Stakeholders**

I will be making a music player app for my main client, Bob a Year 13 student who spends a lot of time studying for his A Level exams and uses music to help him focus whilst studying. My stakeholders will be other Year 13 students who also listen to music.

Many students struggle to focus on their work due to a variety of reasons such as distractions from other sources or tinkering with their environment to get it just right. With more work than ever before students are expected to do, it can demand even more time from them. This can make it more difficult to even start the work as it just makes it seem worse and more tiring. Whereas, with music it makes it much easier to get started on the work and carry on doing it for a much longer time period as you get less fatigued from the work and it seems less daunting to start when you can listen to your favourite songs whilst doing so.

I intend on creating a Python desktop application which lets users listen to and create playlists, upload their own songs, track frequently played songs, search for songs by genre and artist, shuffle playlists as well as download songs directly within to app to the user’s device. It will also have a GUI which aims to minimise the number of interactions needed with the app one they are setup as the main goal is to get people working rather than maintain their attention.

**Use of computational methods**

A computer is needed for this problem as physical solutions often require a lot of interruption to change out songs and setup as well as often being very clunky and lacking portability. It is therefore much more efficient for students to be able to open an application on their computer and click one button to start playing their playlist than use any external machine.

**Thinking abstractly**

This program will include abstraction to hide away detail the user does not need to directly see when interacting with the menu such as file formats or file metadata. Words can also be abstracted to single character symbols which convey the same meaning to make it easier to quickly identify what everything is. For example, the button for pausing a song can be abstracted from ‘Pause’ to ‘⏸’ which is much quicker to understand what the button does at a glance than reading the text. I will also simplify the GUI and have less buttons to make it less distracting, so it does not take away focus from the user’s task.

**Thinking ahead**

When designing this program, I need to look ahead and consider what features my client would want in the program. To do this I will interview my client to better understand what he wants and needs this program to be. Furthermore, to better understand the stakeholders desires I will also survey them to better understand what sorts of things they are looking for in this program. Before starting the programming of the application I will plan out how the systems should work and interact with each other through diagrams and pseudocode which will help me understand the expected behaviours when programming reducing logical errors and unconsidered interactions between systems.

**Thinking procedurally with decomposition**

I can break up my program which has many different features into smaller sections to make development of the program easier as creating smaller more general solutions allows for more reusability saving on development time. It also makes debugging a problem easier as if they are broken down into smaller chunks the program can be followed easier to reach the source. The program can be broken down into 3 main sections the music player, file management and downloading new songs. Each of these three main sections can then be broken down into smaller subsections.

**Thinking Concurrently**

The program will have to be able to run many processes concurrently and efficiently in the background limiting usage of computer resources whilst also ensuring the program runs smoothly. For example, the program should be able to play music in the background whilst also downloading a playlist and listening for global hotkeys to pause or rewind the music.

**Thinking Logically**

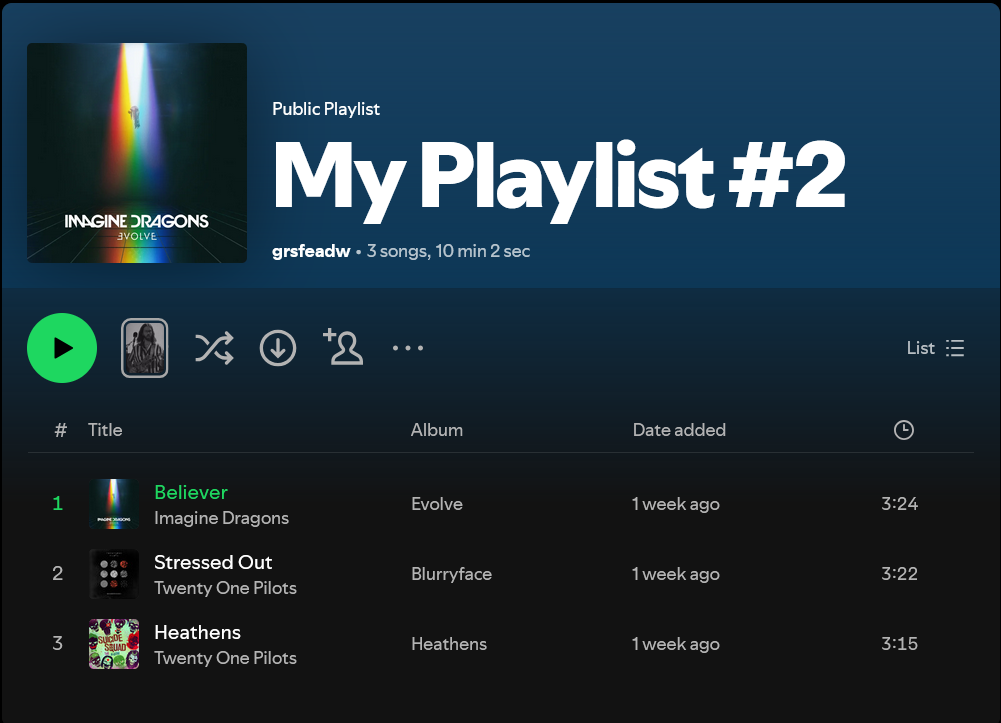
The application should be designed in a way where it is easy and intuitive to understand where you need to go to find an option, feature or setting minimising the amount of time spent fiddling with the application. To do this I should have a clear idea as to what my client and stakeholders would want in the program. This requires preplanning the features I would need in the application and what they would require. For the GUI I would use tkinter with ttkthemes as it allows me to make efficient native GUI’s which are also modern looking. For the database I will use SQLite as my database will be fairly small, and the database will be held locally on the machine.

**Research**

For my research I have decided to look at music player apps and websites and see what features I enjoy about each of them and wish to implement in my own application and which features I find hinders the user experience and should avoid.

**App 1: Spotify** [<https://open.spotify.com/>]

Spotify is a music streaming platform which lets users listen to songs from anywhere where they have an internet connection. They stream songs so are mostly dependant on the user having an internet connection so they can connect to their closest Spotify servers where they are then served the song in small 15 second chunks.



All Songs listed with artists & thumbnail

Song Length

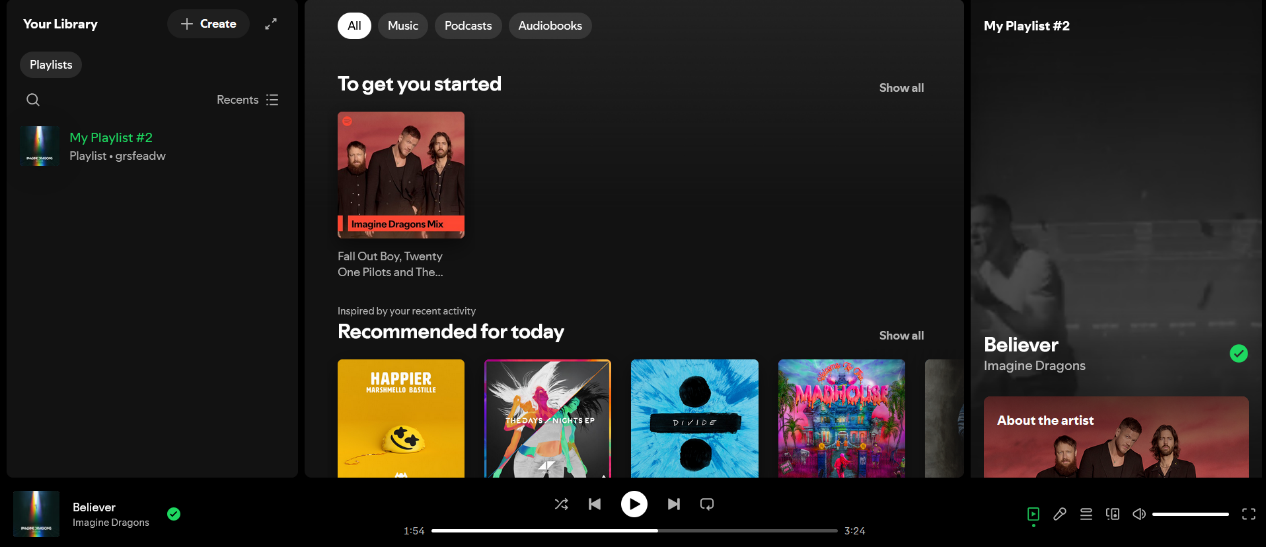
Name of album Song originated from

Song Count and length of playlist

Playlist Thumbnail

Playlist Naming

Shuffle to randomize playlist order and download songs locally



All play options/info shown here in one compact menu.

Volume Controls, queue and lyrics

Current Song playing and artist.

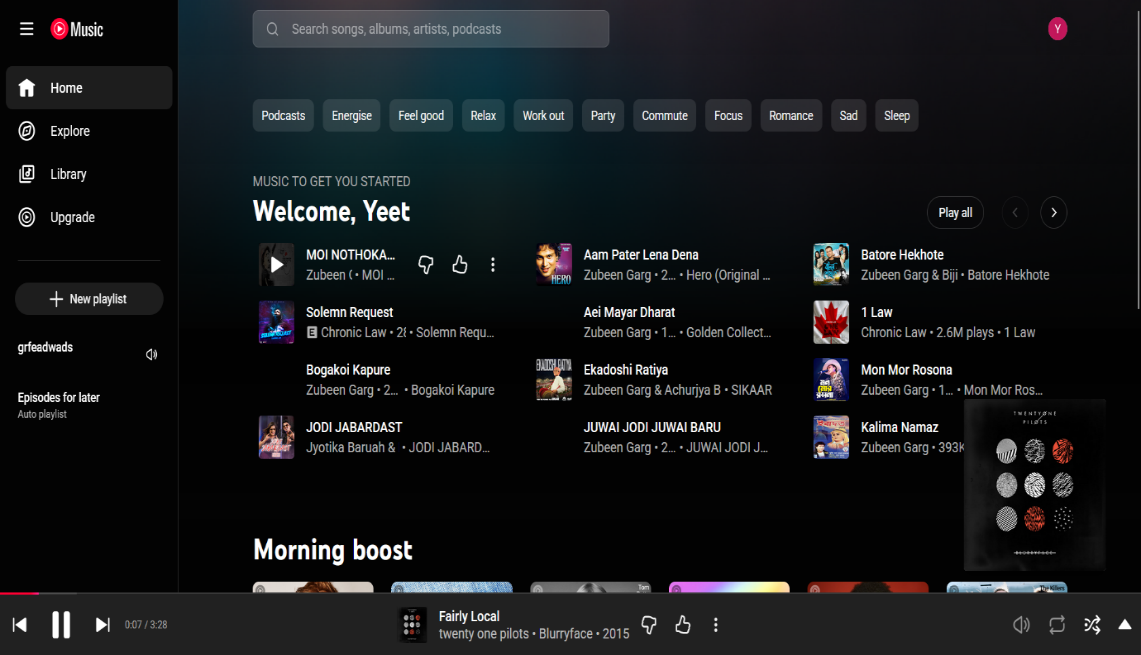
Recommendations for possible songs / playlists tailored to the user

All playlists for shown here for easy access

|  |  |
| --- | --- |
| Pros (+) | Cons (-) |
| Symbols which abstract away unnecessary words not necessary to understand a button’s functionality. For those unfamiliar with the symbols, if the user hovers over the symbol, a box explains what the button is in 1 or 2 words. Makes it easier to find what the user is looking for, minimising interaction with the app. | Some features are unintuitive not being tied to a GUI button and instead being tied to unexplained keyboard/mouse shortcuts. For example, to select multiple songs as once the user must click on one song than shift click on another to select all the songs between them (inclusive). This slows down adding songs to a playlist. |
| Black/Green Colour scheme is undistracting and easy to look at not overwhelming the user letting the user do what they need to do on the app quickly before going back to their work. | After every few songs, Spotify will play an advert sometimes not skippable. This becomes very distracting for the user as it takes attention away from their work and cause them to lose focus. |
| Ability to download songs locally for the user to listen to when offline as well as being able to listen to locally stored audio files. As users may not always have an internet connection when working. For example: Bad Wi-Fi signal at school. | Free users can only skip up to 6 songs per hour which is very limiting if a user has a song they don’t want to listen. Furthermore, they block the ability to directly select songs on a playlist for free users causing a more distracting and worse user experience. |
| Playback controls are all together in 1 place allowing to user to quickly adjust playback setting quickly. Useful for quickly skipping a song or pausing a song minimising disruption to focus. | The Spotify smart shuffle is forced upon free users so they can not select the order their songs are played in. Similarly at the end of a playlist instead of looping Spotify starts playing recommended songs it think you will like which may not be correct and end up distracting the user |
| A website, mobile app, and native desktop app so users have multiple ways to use the app. If they do not have their laptop or computer on them they can use their smartphone and still use the app. | Songs downloaded through Spotify require a monthly subscription to use and cannot be accessed to a regular user outside of the app as they are hidden in an encrypted folder. Furthermore the user must go online at least once per month to retain access to them. |

**App 2: YouTube Music [**[**https://music.youtube.com/**](https://music.youtube.com/)**]**

YouTube Music is a specialised app which takes from YouTube’s extensive music/audio catalogue and adds extra features to make it specialised for playing audio. Similar to Spotify the songs are streamed to the user over the internet where the song is sent in small increments requiring a user to have an internet connection.



Categories based on genre of song.

Main playback options close together

Name of Current playing song

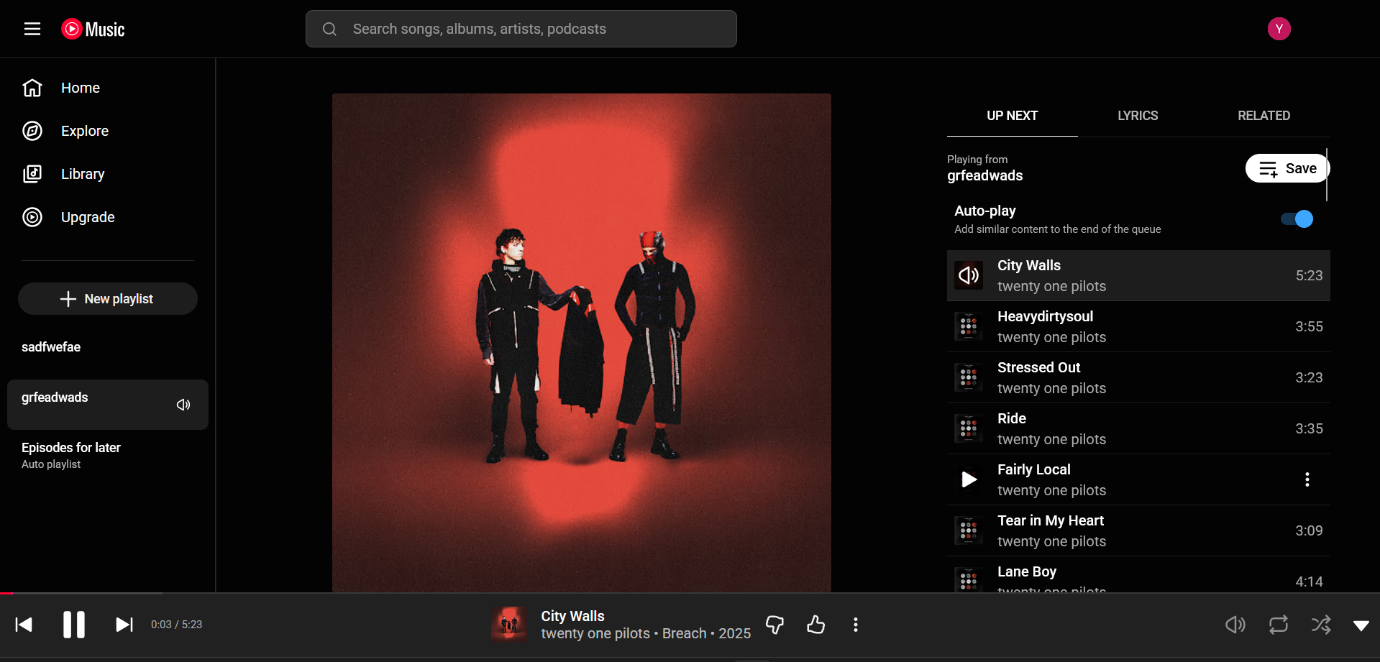
Volume, loop, shuffle, pull up menu buttons

Album Cover/video playback for the song

Recommends based off previously listened to songs

Playlists all listed here. Custom names.

Search bar to find specific songs



Song duration and progress bar

Song thumbnail displayed

All songs listed here for easy jumping between songs.

|  |  |
| --- | --- |
| Pros (+) | Cons (-) |
| Easy to look at GUI with Red and Black Colour Scheme which makes it less distracting for users. Also easy to use not requiring any technical knowledge to utilise it’s features. This means users can spend less time fiddling with it | No mass select for songs to add to playlist which wastes lots of time when compiling them as users have to individually select every song they want to add which for a large playlist ends up being quite monotonous and time consuming decreasing the amount of time the user spends actually doing their task. |
| Large catalogue of songs as it pulls from regular YouTube’s large catalogue of videos and being the largest video hosting service most if not all songs are available. | Advertisements play after/before some songs which distract the user from their task in order to skip the advertisement causing them to lose focus on their work and perhaps get distracted. |
| Ability to listen to audio files found locally on the device. Furthermore, users can download songs on the app which can then be listened to offline. | No background play without a subscription fee on mobile which massively hinders a user’s ability to be able to do work and listen to music simultaneously |

**App 3: Music Player – DD Music**

A screenshot of a phone

AI-generated content may be incorrect.

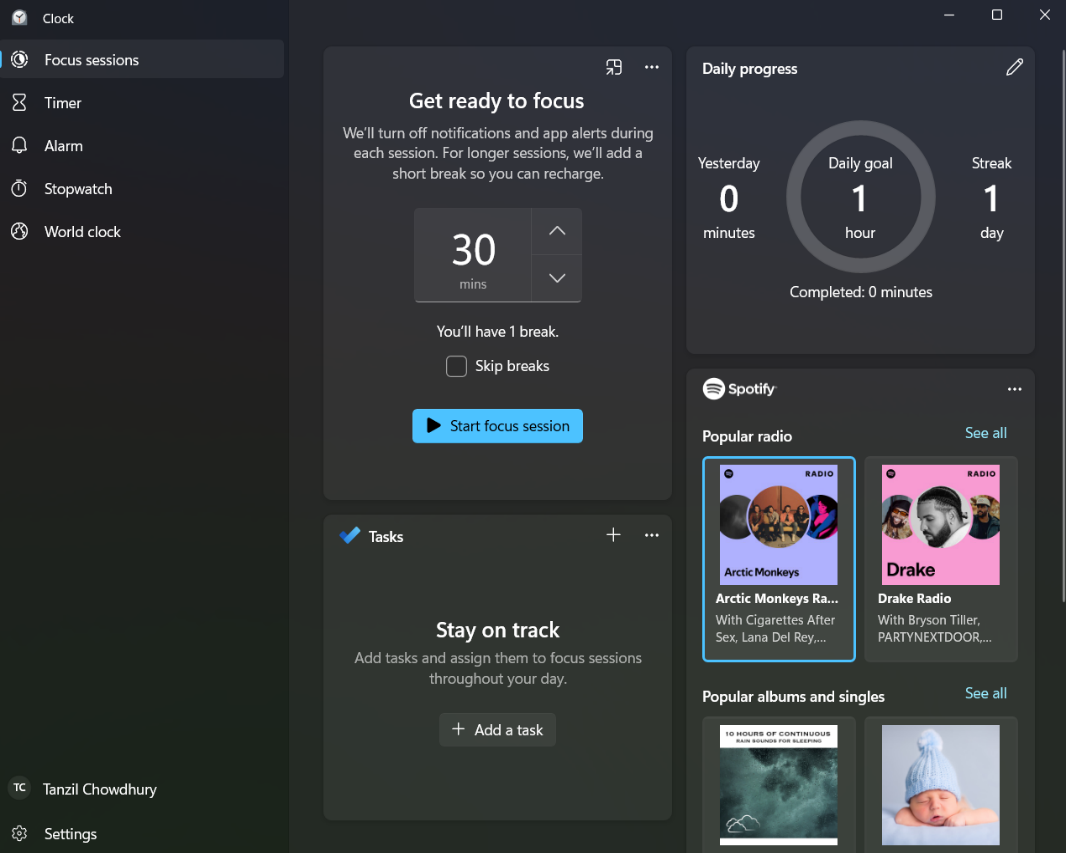
Quick access to pausing/playing song and skipping song

Button to find songs stored locally on the device

Easily able to scan for songs

|  |  |
| --- | --- |
| Pros (+) | Cons (-) |
| No ads during or in between songs so users are not distracted and don’t have to open the app whilst working to skip any advertisements to be able to carry on listening to their songs. | Banner ads are placed to make you accidentally click them when interacting with the app. Furthermore, when opening the app the user has advertisements blocking ability to interact with the app causing frustration and a loss in focus in the work. |
| Allows users to listen to their own locally downloaded songs and supports a variety of audio file formats such as MP3, WAV, FLAC, AAC, 3GP, OGC, | The advertisements can also often end up being inappropriate and unsafe due to a lack of filtering because it uses Google’s Advertising API. |
| Clean and easy to look at GUI which minimises use of eye catching colours to not take too much attention ensuring that the user stays focused on their task. | The app sometimes experiences quite slow load times (sometimes up to a few minutes) when initially booting it which can disrupt the flow of work making it harder for users to get started working. |

**App 4: Clock for Windows**



To do list to keep track of tasks

Way to track daily progress on studying / work

Pomodoro timer so users can have timed study sessions using the pomodoro technique

|  |
| --- |
| Pros (+) |
| Pop out window for pomodoro timer so users can quickly see how much time they have left before the end of the portion of the session. |
| To do list which keeps track of tasks a user puts in. Makes it easier for users to plan what they want to do next during their study sessions. |
| Study time tracker to keep track of how much a user is studying each day and users can set goals which helps motivate them to work a certain amount each day. Also has streaks so users are encouraged to keep studying each day. |

**Client interview**

I also conducted an interview with my client, Bob, about what features he was looking for in the app and what sort of look and feel he wanted for the app.

Q1: How often do you listen to music whilst studying?

A: “I listen to music whilst studying most of the time. Quite often some of the work I have to do is quite long or mundane and doesn’t require a lot of focus. For example: I get set 8 pieces of homework each week which often take over 40 minutes per piece. This alone is just over 5 hours which is often fairly boring. Being able to listen to music makes it much easier to keep focused once I start. I also find music quite helpful for when I am doing practice question which often are on one topic so can end up being fairly repetitive.”

Q2: What apps do you use to listen to music and what do you think of it?

A: “I generally use Spotify and YouTube. I particularly like Spotify because it has an intuitive GUI for music they stream. They just sort of work and have what I need when I need it and do most of the things I need then to do. Another thing, I enjoy about both is the very large catalogue of songs available to the user so most if not all the music I need is available when I need it.”

Q3: What problems do you have with the apps you use?

A: “There are many paywalled features on Spotify that used to be free and feel like arbitrary blocks on a user’s ability to use the app which feel very annoying when I am inconvenienced by them.

One such example is the limited amount of skips you get which can occasionally distract me when I try to skip a song and it fails. The support for offline playback or listening to your own audio files is horrible on Spotify as I am forced to use their shuffle and can’t choose the order, I listen to songs in.

The advertisements are also often quite frequent causing me to lose focus on my task in order to be able to skip them. Especially on YouTube where every other song has an ad play either before or after it. ”

Q4: What other specific features would you like for the app?

A: “Most music streaming services do offer a way to download songs to listen to offline but the ones that do often do not actually allow me to access the files, only being able to listen to the song on the app. I would like to be able to access the songs I am downloading easily outside of the app.

I should also be able to listen to my audio files in the app easily. They should just quickly all appear with one or 2 easy to find button presses or allow me to drag and drop in the songs I want to listen to into the app.

I would also find it convenient to have a way to import my Spotify playlists to the app as it would make the transition easier as my playlists are quite large.”

Q5: What sort of design do you want for the app?

A: “The app should have a default black and purple colour scheme because the colours are easy on the eyes which makes using the app easier, cosier, and less bright which is great for late night studying. However, not all users may enjoy that specific colour scheme so I think that it should have some customisability such as a light/dark mode toggle for the primary colour (default being black) and a completely custom secondary colour (default being purple).”

Q6: What study features do you want for the app?

A: “One app which I have found particularly useful whilst studying is the timer app built into Windows 11. It has a pomodoro timer which is really nice for helping me keep focused whilst revising as it reminds me to take breaks so revising doesn’t become a chore and I am not sat in my chair for too long. It also has a to do list which is a nice way to keep track of the homework and I am able to check off tasks which I have finished as I go along.”

**Client Interview Analysis**

|  |  |
| --- | --- |
| Question | Analysis |
| 1 | My client listens to music often and for long periods of time when they do.  Therefore, the app must be able to load quickly to minimise the time spent waiting on the app and fiddling around with it. |
| 2 & 3 | My client for the most part liked the apps he was currently using and believed they had the features he needed them to have. However, the limitations of the app were causing him to lose focus.  Therefore, the main core of the app should be fairly similar to those ones he currently uses but remove those limitations where possible. |
| 4 | My client wants to be able to access his downloads easily and be able to bring in his own audio files into the app easily. He also wants an easy way to transition from his current platform of choice (Spotify) to this app.  Therefore, the app should let the user choose where songs are stored when downloaded in a common audio format. The app should also automatically try to find songs when the user chooses so. The app should also have a way for the user to log into their Spotify account then the app should be able to find their playlists and start downloading them. |
| 5 | My client prefers a darker colour scheme for it being easy on the eyes. However, he also recognises that people have different colour preferences and prefer different colour schemes.  Therefore, the app should have a light/dark mode toggle so an individual user can decide which they prefer at certain times. The app should also have a customisable secondary colour so users can have some freedom to choose what they prefer best. |
| 6 | My Client wants to be able to track his study time, have a pomodoro timer and a basic to do list.  The app should be able to hold the time studied per day/week and hold goal times for tracking study time. The to do list should be able to hold tasks which ned to be done and keep track of the state of the tasks (completed or not). The pomodoro timer should be able to keep track of how far the user is through their study session. |

**Stakeholder survey**

I then conducted a survey sent out to all stakeholders to gain better insight on what sort of features and design should be included as well as opinions of certain features not found in other apps.

Questions:

Computer Spec Questions: (used to understand what the average user would be using)

1. What Operating System do you use on your Computer/Laptop?
2. Windows
3. Mac
4. Linux

2. How strong is your computer?

Slider scale from 1 to 10.

1 was described as functioning with lots of lag in basic browsing.

10 was described as being able to switch between many browser tabs and programs all in active use.

3. Do you listen to music?

a. Yes

b. No

4. If yes do you listen to it whilst studying?

a. Yes

b. No

c. Sometimes

6. Which features would you say are important for the app? (Multiple choices allowed)

Toggleable Light/Dark mode

Ability to download songs from within the app

Support for Large Amount of File Formats

Customisable hotkeys to access playback controls

Large Catalogue of Songs

Infinitely Looping a Song

Pre-Installed Relaxing Songs

Streaming

7. What other features not listed do you want for the app?

(Self-response question)

**Stakeholder Survey: Results**

A blue circle with a red triangle

AI-generated content may be incorrect.

A majority of the users use Windows so I will prioritise development for Windows. I will also try to ensure that the program functions effectively on windows. However, the windows user experience will be prioritised.

A graph with blue lines

AI-generated content may be incorrect.

A majority of users seem to have computers strong enough where performance will not seem to be an issue. However, to account for the users who may be running weaker computers I will try to minimise background processes whilst the program is not directly in focus so any performance drops are not hindering the user studying.

A blue circle with a red triangle

AI-generated content may be incorrect.

A very large majority of the stakeholders do listen to music suggesting that this could be a potential avenue of interest for them of listening to music whist studying to help them focus.

A pie chart with a few different colored circles with Crust in the background

AI-generated content may be incorrect.

A majority of users do listen to music whilst studying. However a sizable portion only do so some of the time. When enquired further as to why it was often task dependant as to whether music would be used. Some tasks such as going over content users found music a hinderance. Whilst they would instead use it when doing things like practice questions.



Streaming was not a popular option amongst users and is quite difficult and expensive to implement so will not be included in the application. Similarly, there was also little interest in support for a large amount of file types and pre-installed songs. A few pre-installed songs are much easier to implement so may be included for the people who want them.

A lot of users were interested in having a large catalogue of songs and being able to download songs they like for offline listen so I intend on implementing. Many also expressed interest in features such as infinite looping of a song and a light/dark mode toggle. Some people also expressed interest in the hotkey idea of being able to access the playback settings whilst not tabbed into the program.

**What other features not listed earlier do you want for the app?**

This question was open response to see if users expressed interest in any other ideas not asked earlier. Many users expressed interest in a darker colour scheme and some amount of customisability in the colours. Quite a few users also expressed interest in adding a Pomodoro timer and a to-do list features which would be fairly simple to implement.

**Features of Proposed Solutions**

|  |  |  |
| --- | --- | --- |
| Feature | Justifications | Evidence |
| Dark Colour Scheme | To make the app easier to look at as lighter colours were found to be more distracting and annoying. | App 1,2,3,4 Analysis  Client interview Q5  Survey Q6/7 |
| Customisable Colour Scheme | Many people prefer different colour schemes for many semantic reasons. Customisability is better for the user. | Client Interview Q5  Survey Q6/7 |
| Light/Dark Mode | Some users may prefer to have a light/white primary colour so it should be available to those who want it. | Client Interview Q5  Survey Q6/7 |
| Easy to use GUI | Minimises the amount of technical knowledge needed to use the app. | Client Interview Q5 |
| Large Catalogue of Songs | Users will have a large variety in music tastes so they should be able to access any song they may want to listen it | App 1,2 Analysis  Survey Q6  Client Interview Q2 |
| Progress Bar | Let’s the user easily see how far into the song they are. | App 1, 2, 3 Analysis |
| Skip forward/back | Lets the user easily jump forward in a song to skip a part or jump ahead. | App 1, 2, 3 Analysis |
| Loop Song/Playlist once/infinitely | User may want to listen to a song / playlist many times without having to go back to the app and start the song / playlist again. | App 1, 2, 3 Analysis  Client Interview Q3 |
| Shuffle | Users may not want to listen to the music in the exact same order every time so it should be randomised so the playlist doesn’t feel repetitive. | App 1, 2, 3 Analysis  Client Interview Q3 |
| Song skip/go back | Users may not want to listen to a song or may wan to go back to a song so should be able to quickly jump to the previous/next song | App 1, 2, 3 Analysis  Client Interview Q3 |
| Playback queue | Users should be able to see the next / previous songs played and be able to quickly jump head to that song. | App 1, 2, 3 Analysis |
| Global Hotkeys for accessing playback controls | This allows the user to quickly access the controls for the music without having to take focus away from their task to open the app and do it. | Survey Q6 |
| Custom Playlists | Lets users sort music in ways they want for specific scenarios such as a study playlist and a relaxing playlist. | App 1, 2, 3 Analysis |
| Mass Add to Playlist | Makes creating playlists especially for people with larger song libraries much faster and less tedious. | 3 Analysis |
| See Playlist length | Idk man it’s cool and fairly easy | App 1 Analysis |
| Custom Playlist Names | Lets users quickly identify what each playlist is without having to look though each individual one. | App 1, 2, 3 Analysis |
| Playlist / Song Thumbnails | Idk man | App 1, 2, 3 Analysis |
| Able to Download Songs | Lets users listen to songs without an internet connection. | App 1, 2 Analysis  Survey Q6 |
| Access to Downloaded Song files | Lets users easily modify and distribute the files between their devices. | Client Interview Q4 |
| Importing playlists from Spotify | Lets users easily set the app up making transitioning to the app much faster and efficient. | App 1, 2, 3 Analysis  Client Interview Q4 |

**Hardware/Software Requirements**

User and Developer Hardware Requirements:

CPU: 1.8+GHz (64 bit processor)

RAM: 8GB+

GPU: Integrated Graphics

Developer Software Requirements:

OS: Windows 11/10 is required for a smooth experience

Python 3.14 or later – I am using Python because it is what I am most familiar with reducing development time and it has large amount of library support assisting me in the development of the program.

IDE – PyCharm Community Edition by JetBrains.

Libraries:

SpotifyDL/ytdlp (pick one later) – This library will handle the downloading of the songs which will allow the app to theoretically have a very large catalogue of accessible songs.

SQLite – This will handle the storage of information about song data (such as frequently accessed metadata and file path locations) and study data (how long user has studied for, study goals, to-do list)

CustomTkinter – This will handle the GUI elements of the program to make it easy to use for the user

Fill rest of libraries in later as idk rn

User Software Requirements:

The user will require Windows 11/10 and Python 3.14 or later as well as all the Libraries listed in the developer section. If possible the program will be bundled with an installer so this can be done automatically for the user.

**Limitations of Proposed Solution**

* The app with only be able to run locally on a user’s PC/Laptop. So the user will have to download the application to be able to use it. This also means that the user cannot use this app on a mobile phone, tablet or anything of the sort. This is because the app primarily works off of playing locally downloaded music which is more difficult to do with a website. Development for mobile phones and tablets would be too difficult due to a lack of experience in developing for them.
* All songs must be downloaded to the users device to listen to. As this app will not stream music to the user, it must be downloaded into storage before it can be listened to. This is because implementing streaming would be both quite time consuming and quite expensive as servers would be needed if the userbase grows too much.
* This program will only be guaranteed to run smoothly on Windows 10/11 devices as that is the only operating system I have available to me to be able to test the program on so I cannot test on other Operating Systems such as MacOS or Linux.
* Some features such as searching a song to download may be limited due to Spotify/YouTube API limitations for free access (Youtube has a limit of 10k per day and Spotify is a bit confusing) as I am not able to pay for extra API calls. I would have to limit the number of results which appear which could make it more difficult to find more smaller songs which do not appear as highly in recommendation algorithms.

**Success Criteria**

Doing afterwards as idk rn