**Timeline Gantt Chart**

A screenshot of a project

Description automatically generated

*(Made using* [*https://www.onlinegantt.com/#/gantt*](https://www.onlinegantt.com/#/gantt)*)*

Steps:

1. Set up GitHub
2. Make development Schedule
3. Write problem Outline and Description
4. Outline Requirements for the Project
5. Discuss Ethical, Legal, and Security Implications
6. Discuss factors on qualities of data
7. Create ERD
8. Create Relational notation
9. Create Data Dictionary
10. Create Database
11. Insert Data into Database
12. Create Frontend
13. Create Sample SQL Queries
14. Product Evaluation
15. Retrospect on development process

**Problem Outline**

A family member is starting a new online business and needs a software solution to manage customers and record sales information for an online music streaming service. They require a backend for storing music data, and a basic frontend for user and administrator use. They are calling their new business: “Sound Shift”

This includes, creating a database in SQLite and filing it up with some sample data, alongside creating a python frontend that can interact with and query this database. The backend must track customer data, sales information, and follow legal requirements such as the adequately safeguarding customer data.

**Problem Description**

I have been tasked with developing a front and backend for an online music streaming service called “Sound Shift” by a family member. The backend database built in SQLite of this software, must be able to manage customer data, record sales information, and store music track and album data, along with keeping user data confidential. The frontend must be able to make SQL queries to the database and display the outputs of said queries to the user. Ethical, legal, and security issues must be considered when creating this software solution, to take protect customer data.

**Problem Scenario**

A family member has asked me to assist them in the creation of their business by developing a software solution in the form of a frontend and backend for their online music streaming service. Customers can subscribe to the service, and through their account, access all the music stored within the database through an intuitive frontend, just by logging in, and searching up the song title, or genre. They want the backend to be able to securely store customer data and accounts for the customers, for ease of use. They want to be able to access customer data, and record subscription figures with ease.

**Requirements for the Solution**

**Frontend –**

**Functional:**

* Users need to be able to sign in to their account on the software
* The user must be able to change and/or delete their customer data
* Customers can browse songs based on certain input criteria
* Results of any queries made to the database must be displayed to the user

**Non-Functional:**

* An intuitive interface for users to use
* SQL code should never be shown to the user
* The user should never see a non-custom error message
* The program should never crash on the user
* Fast response time on the frontend

**Backend –**

**Functional:**

* This account data must be stored, and accessible to the user
* All of the songs must be stored in the database, along with their respective attributes
* The database must be in third normal form to reduce data anomalies
* The frontend python code must be able to query the backend SQLite database
* Must store all subscription data, on customers
* Data in the database must be validated using appropriate constraints

**Non-Functional:**

* All inputs should be validated, so there are no SQL injections
* User data in the database is updated every once in a while, by asking them if it is up to date.
* User data in the database is encrypted
* User passwords are stored as hashes
* The code is clean, and easy to read

**Legal issues**

If Sound Shift makes an annual turnover of $3 million of more, or if they decide to opt in, they are covered by the Australian Privacy Principal Act of 1988. This subjects them to the 13 Australian Privacy Principles which govern, how the user’s private data can be used, obtained, and stored.

Privacy principle 1 dictates that businesses must be open and transparent with how their user’s data is being used. This is normally done through having a terms and conditions, document that users agree to, when they sign up to the site. This document details what data will be stored, how they will collect this data, the reasons they must have this data, how this data will be used, how the user can access and change this data, how to lodge a complaint about this the above, and if the data will be disclosed overseas. So, Sound Shift will need to have a publicly available privacy policy, that users agree to when they sign up.

Privacy principle 2 states that businesses must give users the option of anonymity. This can be done, by instead of asking for a first and last name for the database, asking for a required username to put their account under, with optional first and last name fields.

Privacy principle 3 dictates that businesses must only collect personal information that is directly related to its use, or that assists the business’s functions, and that it can only be collected with the consent of the owner of the personal information. This can be done, by not asking for any unneeded personal information e.g. while Sound Shift can ask for credit card information to pay for the subscription, we can’t ask for a driver’s licence number, as it is not related to streaming music. The part about how only the owner of the personal data can disclose it, can be solved by only asking the user for the data, and having them tick a check box saying “This is my own personal data, and not anyone else’s”

Privacy principle 4 states that businesses must destroy or deidentify any unsolicited data. This means that is Sound Shift is given someone’s driver’s license number, they should delete said information, as it is not relevant to the business’s activities.

Privacy principle 5 dictates that businesses must notify any users they collect the personal data of, how to contact the business, the circumstances of the collection, the purposes that data was stored, who the business usually discloses the information to. A link to the business’s privacy policy, and if the information is likely to be sent overseas. These requirements can be settled within the privacy policy terms and conditions that the user must agree to while signing up to Sound Shift.

Privacy principle 6 states that businesses can only use or disclose the collected personal information for the purpose it was collected, and for any use the user consented to, or would reasonably expect. Sound Shift can do this by only using the data in the ways described within the privacy policy, that users agreed to.

Privacy principle 7 dictates that businesses may only continually advertise to users or add them to an email list if the user consents to it. Sound Shift can follow this by having a second checkmark that the user needs to opt-out of the email list for marketing.

Privacy principle 8 states that businesses sending personal information overseas must ensure that the recipient will not break any of the privacy principles. Sound Shift will comply with this due to being a small national business, and not sending any information overseas.

Privacy principle 9 dictates when businesses can use government identifiers. Sound Shift is not regulated by this, as all government identifiers are unsolicited information, and so will be deleted anyway.

Privacy principle 10 states that businesses must take reasonable steps to ensure that any personal information they collect is accurate, up to date, and complete. Sound Shift can ensure this, by saying on sign in that “By signing up you agree that to the best of your knowledge, all of this information is accurate, up to date, and complete”. It can then email users once a year, asking them if their data is still up to date.

Privacy principle 11 dictates that businesses must protect the personal information it collects. How Sound Shift protects this data, can be viewed under the “*Security Issues*” heading below.

Privacy principle 12 states that businesses must give the user access to any information covered by privacy principle 11, whenever they want. Sound Shift can follow this, by having it be accessible of the users ‘account info’ page.

Privacy principle 13 states that the user must be able to amend and change the information provided under privacy principle 12 whenever they want. Sound Shift can do this by allowing the user to change the personal information displayed to them on their ‘account info’ page.

The 2017 privacy amendment ensures that if businesses experience a data breach, they must notify the afflicted users the details of the breach, what data was breached, and recommendations on how to respond to the breach. Sound Shift can follow this, by having a response plan in place, to handle data breaches, and the notification of users.

The copywrite of each of the songs must be ensured by identifying that only the owners of a song may upload it and allowing users to file complaints against each song claiming copy write infringement. These claims must then be investigated by employees to check if copyright laws are being followed. If not, then it will be taken down.

**Security Issues**

As privacy principle 11 states, any personal information of the users must be adequately protected and secure. This will be done through numerous security measures enacted upon the database. To ensure the protection of user’s data a few steps will need to be taken:

1. The database will need to be encrypted server side using a symmetric encryption algorithm such as AES (Advanced Encryption Standard), so that even if the database is leaked, it is still encrypted and so the user’s data is not public.
2. Access Control Lists (ACLs) will need to be implemented so that only the admins who need access to user’s data can access it, and other staff, must file a request to view/edit it.
3. The passwords of users will need to be hashed server side, and then stored as a hash, the hashing algorithm will be SHAR256. This will ensure that yet again, if the password field is somehow leaked, only the password hash is discovered, not the actual password.
4. All client inputs will need to be sanitised and validated before being enacted upon, to ensure that any client-side manipulation of the website cannot allow users to interact with the server in unintended ways.
5. Communication between the user and server, should be done through HTTPS, so that it is TLS encrypted. Sound Shift can achieve this, by obtaining an SSL certificate, which allows them to host on ‘https’ instead of just ‘http’. All attempts to access the site through ‘http’, should be redirected to the ‘https’ version of the site.
6. An Intrusion prevention system (IPS) should be set up for the server, to stop any attempted DDoS attacks.
7. A firewall should be set up that does packet inspection to check for any suspicious data, and that blocks unnecessary ports into the server e.g. port 20 which handles FTP.
8. Physically, the building housing the servers, should have cameras, locks on all the doors, and the servers should be behind a locked faraday cage that is locked with a combination lock, a lock that needs a physical key, and an RFID card. The server which stores the security camera footage, should also be within this faraday cage.
9. A digital security should be hired, that are on call in case of any data breaches, malware, or hacks to the system. This team should be able to respond to any digital security issue.
10. Once every two years, a group of penetration testers should be hired to test the network security, without the knowledge of the security team, to simulate a real attack, and test the countermeasures/defence plans in place.

**Ethical Issues**

These are not hard rules that Sound Shift needs to follow, but guidelines that the business should follow in good practise and to be morally sound.

Sound Shift should respect artists rights, and if the artists, producers, lyricists, or composers of songs are protesting, and on strike, then the songs that are attributed to them, will not be able to be streamed through the service, to support them in solidarity. And to respect their right to control just how their work is distributed and monetised.

When a song is uploaded onto the site from an artist, the producer, composer, lyricist, and any other people that worked on the song should be attributed, and their names, along with titles, accessible in the song’s description.

When users sign up, there should be a tick box asking the user if they want to donate another dollar each month, to go to charities that help protect artists against discrimination or being taken advantage of.

**Factors Impacting the Quality of Database Data**

There are many factors affecting data quality, these factors are: currency; authenticity; relevance; accuracy.

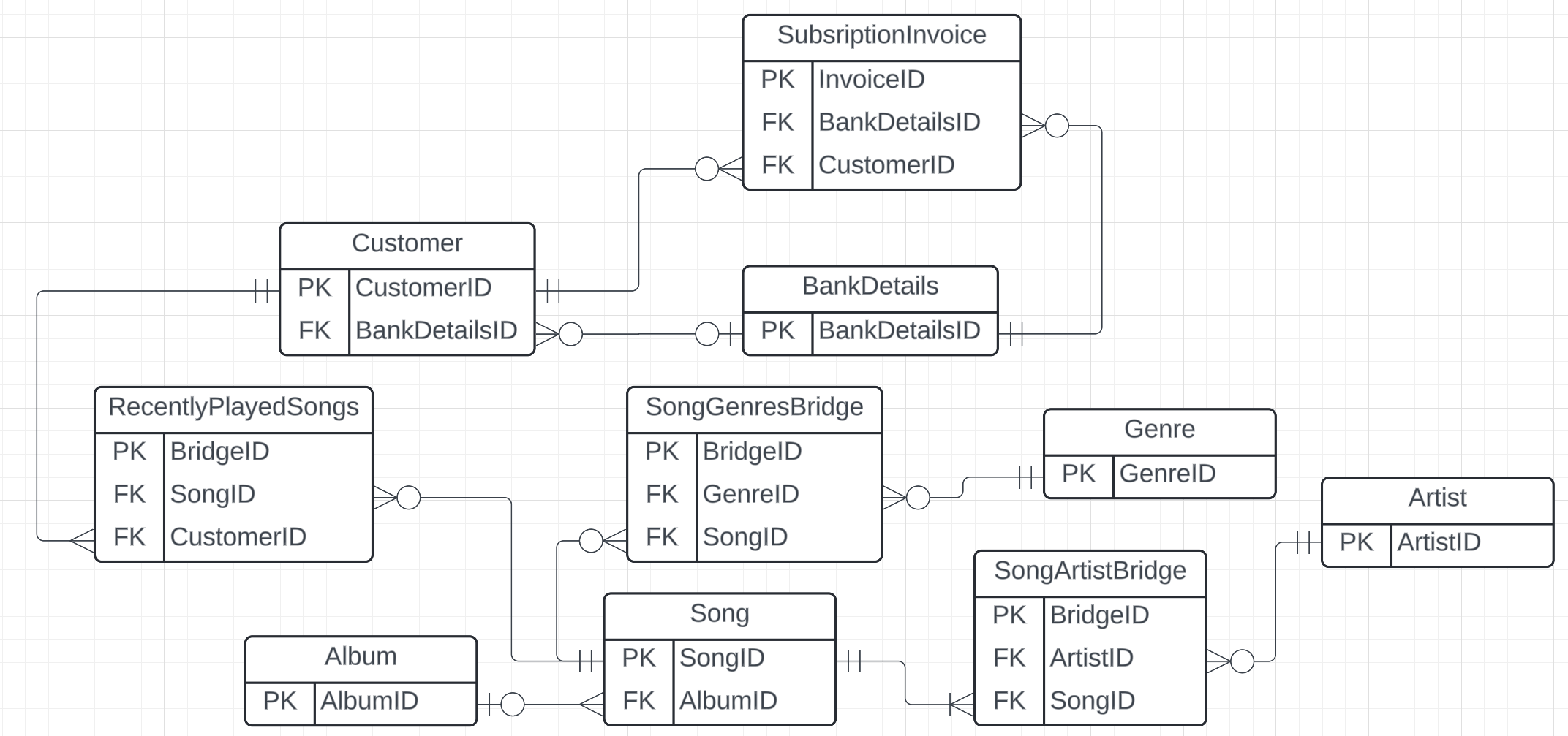
Currency refers to how current, or up to date the data is. The more recent the data, the better, as recent data is more relevant to any current problem then out of date data. This ensures that decisions are made based off of current data instead of outdated data. Sound Shift can check on this, by storing in records, when the last time it was updated was, as while songs are very likely to not be updated, it is better to make sure that users know what data the data was current at. To ensure currency can take place, any data on songs may be subjected to editing by the publisher at any time, in case e.g. the artist changes their name.

Authenticity refers to if the inputted data was from a reliable source, and so was genuine in the first place when input. This helps prevent the spread of misinformation. There isn’t much of a way to ensure that publishers input correct data about their songs or that the data is from a reliable source, other than having complaints from customers that the information is wrong or tracking down each producer and investigating the song. So Sound Shift will merely have a checkbox stating “This data is accurate and up to date” whenever publishers submit anything.

Relevance refers to how important the data is to the use cases of said data. This helps make sure that useless data does not clutter the system. To ensure this, Sound Shift will make sure to clearly ask very structured questions for any inputted data.

Accuracy refers to how close the data is to the real data. The only way to ensure this is to double check any data with relevant and reliable sources, but that would be extremely tedious. So Sound Shift will just rely on the publishers to input accurate about their songs/albums data, as falls under the authenticity safeguards.

**ERD**

****

*(Created using* [*https://lucid.app*](https://lucid.app)*)*

**Relational Notation**

Customer(CustomerID, Username, BankDetailsID, Email, Password)

BankDetails(BankDetailsID, CardNumber, CardHolderName, ExpirationDate, CVV)

SubscriptionInvoice(InvoiceID, SaleDate, BankDetailsID, CustomerID, AmountCharged, SubscriptionLengthBought)

RecentlyPlayedSongs(BridgeID, SongID, CustomerID, DateListenedTo)

Genre(GenreID, Name)

Album(AlbumID, Name, ReleaseDate)

Artist(ArtistID, StageName, FirstName, LastName)

Song(SongID, Name, Length, AlbumID, ReleaseDate)

SongArtistBridge(BridgeID, SongID, ArtistID)

SongGenreBridge(BridgeID, SongID, GenreID)

**Data Dictionary**

**Customer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size/Format Default | Constraints | Description |
| CustomerID | Integer |  | Unique, not null | The customer’s primary key identifier |
| Username | Text | 20 | Not null | How the customer will be referenced/called in any records or contact with them |
| Email | Text | 50 | Not null, requires an ‘@’ | The email that a customer can be contacted through |
| Password | Text | 32 | Not null | The hashed version of the password users can use to sign in to their account |
| BankDetailsID | Integer |  |  | A foreign key linking the customer to their banking account that they pay with |

**BankDetails**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size/Format Default | Constraints | Description |
| BankDetailsID | Integer |  | Unique, Not null | The Bank Details’ primary key identifier |
| CardNumber | Integer | 19 | Not null | The payment card number of the payment card the customer uses to pay with |
| CardHolderName | Text | 50 | Not null | The name on the payment card, of the owner |
| ExpirationDate | Text | 5 | Not null, contains a ‘/’ | The expiration date of the payment card |
| CVV | Integer | 4 | Not null, length of 3/4 | The three/four number physical code on a payment card |

**SubscriptionInvoice**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size/Format Default | Constraints | Description |
| InvoiceID | Integer |  | Not null, Unique | The Invoice’s primary key identifier |
| SaleDate | Date | 10 | Not null | The date on which the sale took place |
| BankDetailsID | Integer |  | Not null | The Foreign key to the bank details that were used to buy the subscription |
| CustomerID | Integer |  | Not null | The Foreign key to the customer that bought the subscription |
| AmountCharged | Float |  | Two decimal places, Not null, greater than ‘0’ | The amount charged to the payment card for the subscription |
| SubscriptionLengthBought | Integer |  | Not null | The amount of months, that the subscription has been bought for within the transaction |

**RecentlyPlayedSongs**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size/Format Default | Constraints | Description |
| BridgeID | Integer |  | Not null, Unique | The primary key identifier of the Recently played songs entity |
| SongID | Integer |  | Not null | The Foreign key that connects which song was listened to |
| CustomerID | Integer |  | Not null | The Foreign key that connects to which customer listened to the song |
| DateListenedTo | Date | 10 | Not null | The date on which the song was listened to, and subsequently inputted into the ‘recently played songs’ entity |

**Genre**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size/Format Default | Constraints | Description |
| GenreID | Integer |  | Not null, Unique | The primary key identifier of the genre |
| Name | Text | 50 | Not null | The name of the genre |

**Album**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size/Format Default | Constraints | Description |
| AlbumID | Integer |  | Not null, Unique | The primary key identifier of the Album |
| Name | Text | 50 | Not null | The name of the Album |
| ReleaseDate | Date | 10 |  | The date in which the song was released |

**Artist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size/Format Default | Constraints | Description |
| ArtistID | Integer |  | Not null, Unique | The primary key identifier of the Artist |
| StageName | Text | 50 | Not null | The known stage name of the artist |
| FirstName | Text | 50 |  | The artists actual first name |
| LastName | Text | 50 |  | The artists actual last name |

**Song**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size/Format Default | Constraints | Description |
| SongID | Integer |  | Not null, Unique | The primary key identifier of the song |
| Name | Text | 50 | Not null | The name of the Song item |
| Length | Integer | 4 |  | The length of the song in seconds |
| AlbumID | Integer |  | Not null | The Foreign key identifier of the album the song is part of |
| ReleaseDate | Date | 10 |  | The date in which the song was released |

**SongArtistBridge**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size/Format Default | Constraints | Description |
| BridgeID | Integer |  | Not null, Unique | The primary key identifier of the bridge |
| SongID | Integer |  | Not null | The Foreign key identifier of the song |
| AlbumID | Integer |  | Not null | The Foreign key identifier of the album |

**SongGenreBridge**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size/Format Default | Constraints | Description |
| BridgeID | Integer |  | Not null, Unique | The primary key identifier of the bridge |
| SongID | Integer |  | Not null | The Foreign key identifier of the song |
| GenreID | Integer |  | Not null | The Foreign key identifier of the genre |

**Entity Overview**

|  |  |
| --- | --- |
| **Entity** | **Description** |
| Customer | Stores data on customers, such as their password, which set of bank details they are connected to, their username, and their email to be able to contact them |
| Bank Details | Stores all the necessary data in order to charge a customer’s credit/debit card for any purchases they make |
| Subscription Invoice | Stores the invoices in each record, of each purchase of the subscription made. Customers must purchase a subscription either monthly, or every six months, in order to continue streaming songs. This entity stores all the necessary sales information of that purchase |
| Recently Played Songs | This entity tracks which songs a customer has recently listened to, this therefore needs to contain, which customer listened to it, which song they listened to, and when it was listened to |
| Genre | Stores all the genres that songs can be e.g. ‘Rock’, ‘Jazz’, ‘Hip-Hop’, ‘Pop’ |
| Album | Stores all the albums that the streaming service provides, albums are defined for this purpose, as a collection of songs |
| Artist | Stores all artists that create songs on the platform, for this, their ‘stage name’ is needed, for them to be publicly credited for their songs |
| Song | This stores all the songs on the platform, along with their length, and which album they were released as a part of |
| Song Artist Bridge | This bridging table resolves the many to many relation of ‘Song’s and ‘Artist’s, as naturally an artist can release many songs, and as artists collaborate, a ‘Song’ can have multiple artists that contributed to it. So, this bridging table resolves this many to many relation |
| Song Genre Bridge | This is a bridging table between the entities ‘Song’ and ‘Genre’, this resolves the many to many relation of the two. As multiple songs share the same genre. And a Song can have multiple genres such as ‘Pop rock’ |

**SQL Quires in English**

* Insert a new record into the customer table
* Get all the songs that are of the “rock” genre
* Get all of a customer’s recently played songs
* Get all the songs in an album
* Use the date in the invoice and the length of subscription bought to figure out when a customer’s subscription ends
* Get all of an artist’s songs
* Add an artist
* Add a song
* Link the previously aforementioned artist and song
* Change a customer’s password

**Appendix**

**References**

*Best ways to ask for credit card information in online forms*. (2024, August 9). Zuko Form Optimization and Tracking Analytics Tools. <https://www.zuko.io/blog/asking-for-credit-card-information-in-online-forms>

ChatGPT. (2024, August 11). Name for the business. OpenAI. Prompt: “I am making a music streaming service that users subscribe to and can listen to any music available on the platform, just like Spotify. Come up with 50 possible names for the business.”

*Free online Gantt chart software*. (n.d.). Free Online Gantt Chart Software. <https://www.onlinegantt.com/#/gantt>

Sullivan, B. (2024, June 17). *ethics and law, ethical hacking hacking*. https://cewaedu.sharepoint.com/sites/Section\_8615-ATCSC\_1-2024/\_layouts/15/Doc.aspx?sourcedoc={7efdbbfd-a216-4eae-a831-42c726fac0cf}&action=view&wd=target%28\_Content%20Library%2FT2W10.one%7Cb3e9cecc-8cf7-449f-898f- 28724e241c64%2FEthics%20and%20Law%2C%20Ethical%20Hacking%20Hacking%7Cecc5966d-16d6-4642-bc7d-8a09dca2feb9%2F%29. [https://doi.org/2024/\_layouts/15/Doc.aspx?sourcedoc={7efdbbfd-a216-4eae-a831-42c726fac0cf}&action=view&wd=target%28\_Content%20Library%2FT2W10.one%7Cb3e9cecc-8cf7-449f-898f-28724e241c64%2FEthics%20and%20Law%2C%20Ethical%20Hacking%20Hacking%7Cecc5966d-16d6-4642-bc7d-8a09dca2feb9%2F%29](https://doi.org/2024/_layouts/15/Doc.aspx?sourcedoc=%7b7efdbbfd-a216-4eae-a831-42c726fac0cf%7d&action=view&wd=target%28_Content%20Library%2FT2W10.one%7Cb3e9cecc-8cf7-449f-898f-28724e241c64%2FEthics%20and%20Law%2C%20Ethical%20Hacking%20Hacking%7Cecc5966d-16d6-4642-bc7d-8a09dca2feb9%2F%29)

Sullivan, B. (2024, August 1). *factors affecting quality of data*. Computer Science - ATAR Y12 - ATCSC 1 - 2024 - 8615 Notebook. [https://cewaedu.sharepoint.com/sites/Section\_8615-ATCSC\_1-2024/\_layouts/15/Doc.aspx?sourcedoc={7efdbbfd-a216-4eae-a831-42c726fac0cf}&action=view&wd=target%28Joshua%20Finlayson%20%28Prendiville%20Catholic%20College%20-%20Ocean%20Reef%5C%29%2FT3W3.one%7C75e116d1-2e05-4f2d-a6c8-b7a9e66fcac8%2FFactors%20Affecting%20Quality%20of%20Data%7Cbb1a6e51-781c-437a-a044-6bef1beb6ca7%2F%29&wdorigin=NavigationUrl](https://cewaedu.sharepoint.com/sites/Section_8615-ATCSC_1-2024/_layouts/15/Doc.aspx?sourcedoc=%7b7efdbbfd-a216-4eae-a831-42c726fac0cf%7d&action=view&wd=target%28Joshua%20Finlayson%20%28Prendiville%20Catholic%20College%20-%20Ocean%20Reef%5C%29%2FT3W3.one%7C75e116d1-2e05-4f2d-a6c8-b7a9e66fcac8%2FFactors%20Affecting%20Quality%20of%20Data%7Cbb1a6e51-781c-437a-a044-6bef1beb6ca7%2F%29&wdorigin=NavigationUrl)

Office of the Australian Information Commissioner. (2023, October 9). *Australian privacy principles quick reference*. OAIC. <https://www.oaic.gov.au/privacy/australian-privacy-principles/australian-privacy-principles-quick-reference>

Office of the Australian Information Commissioner. (2023, March 23). *Small business*. OAIC. <https://www.oaic.gov.au/privacy/privacy-guidance-for-organisations-and-government-agencies/organisations/small-business>