Soundscape - Interactive Data Visualisations of Spotify's Top Songs

Abigail Espejo

Abstract— **In the era of music streaming platforms like Spotify, marketing tactics in the music industry have increasingly prioritised streaming activity to boost song popularity. Spotify stores extensive metadata about song characteristics, such as the measure of danceability. This particular measure evaluates a song's suitability for dancing by combining various musical elements like tempo and rhythm stability. Therefore, the availability of such complex and diverse data underscores the importance of effective data representation and analysis techniques. Soundscape, a React web application, aims to address the challenge of exploring available rich music ranking data by developing an interactive visualisation system using D3.js. By leveraging the song characteristics of the top 2,000 Spotify tracks spanning 1999 to 2019, Soundscape employs D3 and React JavaScript web-development tools to represent the data. The system's effectiveness and usability was evaluated through a heuristic evaluation. Interestingly, Soundscape was able to facilitate analytical reasoning through some of its interactive data visualisations.**

I. INTRODUCTION

SPOTIFY is today's most popular music streaming service with 220 million paying subscribers on the platform [1]. In 2015, for the first time in the United States (US), music streaming revenues exceeded those of digital downloads, becoming the most significant portion of the music industry's earnings [2], and currently account for over 67% of the total global recorded music revenue [3]. This phenomenon can be referred to as the 'streaming era', which highlights the shift in how people engage with listening to music, thus, how the music industry operates.

Ranking songs by popularity is a longstanding tradition in the music industry. Since 1958, Billboard magazine's "Hot 100" chart, one of the most prestigious music charts, has weekly ranked songs using radio airplay and sales data [4, 5]. At present, the Hot 100 also factors in a song's streaming activity in the US when determining its rank [5]. Spotify has its own incentives for artists to upload their songs on the platform. 'Playlisting' enables artists to submit their songs to Spotify's editorial teams for inclusion in the platform's curated playlists [6]. These playlists can either be followed by millions of users or be customised for specific listener profiles based on personalised data [6]. Through playlisting, the artist's goal is to be discovered, and ideally, see an increase in streaming activity for their songs.

It becomes evident that the streaming era allows for the availability of complex and rich music metadata, revolutionising music industry marketing tools that continue to influence how listeners choose what to listen to. Spotify's Web API allows developers to retrieve Spotify content metadata, obtain music recommendations, curate playlists, and manage playback from Spotify's streaming service [7]. In this report, the design, implementation, and evaluation of "Soundscape" is presented — a novel React web application for end-users who are interested in the music industry, focussing on mainstream artists and popular music chart performance. Illustrated in Fig. I, Soundscape takes a complex, high-dimensional tabular dataset (sourced from Kaggle [8]) that contains audio metrics of Spotify's top 2,000 songs from the years 1999 to 2019. Therefore, the aim of the project is to implement a web application consisting of user-friendly interactive data visualisations, powered by the D3 JavaScript library, to represent rich Spotify data in an easily comprehensible way. Consequently, Soundscape should promote an understanding and analysis of contemporary music trends for its end-users.

A. The Problem

The realm of data visualisation has gained prominence in response to the rapid growth of data [9, 10]. With the ease of data generation, it is imperative that appropriate systems are designed to accurately represent complex data. This project endeavours to demystify the rich Spotify dataset, spotlighting its artists, songs, genres, and intricate song characteristics. We achieve this through the development of interactive data visualisations integrated into a web application.

An effective, interactive data visualisation enables users to delve into the underlying phenomena within a dataset by engaging in analytical reasoning, which cannot be achieved through passively viewing a static display [10]. Interactive visualisation techniques can involve a user manipulating various data parameters whilst observing the visualisation, or panning around the viewpoint to reveal structure in the data [9]. This highlights the imperative of designing a system that requires a deep understanding of human perception and the role of user interactions in facilitating meaningful insights [9].

1) Non-functional System Requirements

a) **Usability**: The system should feature an intuitive and user-friendly interface, enabling users with no prior experience or knowledge to utilise it with ease.

b) **Web-based Application**: The system should be developed as a web-based application, accessible over the internet through standard web browsers.

This project was supervised by Stuart Marshall.

TABLE I
DATA STRUCTURE OF SOUNDSCAPE'S DATASET

Column name	Column description	Example value
artist	Track's artist.	Rihanna
song	Name of the track.	bad guy
duration_ms	Duration of the track in milliseconds.	211160
explicit	The lyrics or content of the track contain criteria which could be considered offensive or unsuitable for children.	true
year	Release year of the track.	2002
popularity	The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are.	24
danceability	Describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.	0.585
energy	A measure from 0.0 to 1.0 that represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.	0.842
key	The key the track is in. Integers map to pitches using standard Pitch Class notation. E.g. 0 = C, 1 = C♯/D♭, 2 = D, and so on. If no key was detected, the value is -1.	9
loudness	The overall loudness of a track in decibels (dB). Values typically range between -60 and 0 db.	-5.883
mode	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.	0
speechiness	Detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value.	0.0556
acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.	0.00242
instrumentalness	Predicts whether a track contains no vocals. The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content.	0.00686
liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.	0.0866
valence	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).	0.428
tempo	The overall estimated tempo of a track in beats per minute (BPM).	118.211
genre	Genre(s) of the track	pop, R&B

c) **Data fetching**: The system should incorporate a data storage solution that fetches data from a remote source, ensuring that users receive real-time updates and always access the most current information without manual intervention.

2) Functional System Requirements

a) **Temporal Trend Analysis**: Users should be able to analyse the progression of music trends over various periods, from short-term shifts to longer historical changes, spanning years or even decades.

b) **Genre-Specific Insights**: Users should be able to delve into specific music genres, discerning their relationships with song characteristics and understanding how they influence their popularity and evolution.

c) **Artist-Centric Exploration**: Users should be able to identify and track the careers of mainstream artists, noting their impact on the industry over different time frames and their longevity in popular charts.

d) **Cultural Depth Analysis**: While acknowledging the potential western bias due to Spotify's English-based nature, the system should facilitate exploration into the music of various cultures and underrepresented regions.

The final system's usability was evaluated and has several aspects that need improvement, detailed in Section V. While the system met most requirements, it fell short in fetching the latest song metadata from Spotify's Web API endpoint. Consequently, the 'Cultural Depth Analysis' requirement was also unfulfilled, as the system's data files lacked information on artists from underrepresented regions.

3) User Model

A user model was devised to holistically represent the attributes, behaviours, needs, and preferences of the system's target audience. Depicted in Fig. II, the user model personas played a pivotal role in guiding the design of the system's interactive visualisations and overall user interface (UI). In crafting these personas, various factors were considered, including potential use cases, daily tasks, current solutions, ongoing challenges, pertinent connections, and objectives [11]. Narratives, rather than itemised lists, proved to be more effective in portraying a persona, providing deeper insight into a user's mindset, needs, skills, and goals [11]. However, the user model is not without its shortcomings. Those outside of the human-computer interaction (HCI) community may view user modelling as an outdated technique, critiquing the framework's tendency for oversimplification. Moreover, the user model was not based on genuine individual data. Nonetheless, the user model was instrumental in comprehensively understanding how end-users would engage with the application's interface and visualisations, enabling them to extract valuable insights from the Spotify dataset.

B. Sustainability Goals

As the world grapples with complex challenges, from economic fluctuations to the urgent menace of climate change, the importance of sustainable development has been magnified. Specifically, "the post-pandemic recovery and the war in Ukraine resulted in a sharp rise in consumer prices in early 2022, particularly in food and energy [12]". Within this context, it is essential to evaluate how specific initiatives align with broader sustainability goals, such as the United Nations' 17 Sustainable Development Goals (SDGs) [13]. While Soundscape does not directly address these global objectives, Section VI, B offers suggestions on how the implemented system could potentially be steered towards the SDGs in the future by going beyond the system's local dataset.

Goal 12 of the SDGs underscores the importance of ensuring sustainable consumption and production patterns [13]. Assessing Soundscape in this context, it operates in a manner that does not strain resources. Its JavaScript software is designed for local execution on standard computer devices, compatible with commonly used browsers like Google Chrome. Moreover, it was not designed to support asynchronous usage by numerous users, thereby minimising potential consumption spikes. The data files integral to the system are minimalistic and less than 1MB in total with sizes of 254KB, 4KB, and 225KB respectively, further highlighting the system's lean resource footprint.

II. RELATED WORK

Soundscape is a project centred on designing, developing, and evaluating a web application that utilises Spotify music data to produce interactive visualisations. Its primary objective is to cater to users interested in uncovering trends in popular music. This section discusses related works to draw from their methodologies. Firstly, we examine the technical approaches of data visualisation implementation, then we aim to understand the intended user experience (UX) while engaging with such visualisations when integrated within a web application.

TABLE II
PERSONA 'JAKE' FOR SOUNDSCAPE USER MODEL

Background	Jake is a 23-year-old university student majoring in Music Business. He is tech-savvy, frequently listens to music, and is always curious about the ever-changing dynamics of the modern music industry.
Goal	Jake wants to understand the prevailing trends in the music industry and the performance of various genres to better inform his career choices post-graduation.
Scenario	While working on a project about the impact of audio metrics on popular song charting, Jake stumbles upon Soundscape. Through its user-friendly interface, he dives into the different trends and recognises the rise of specific genres over the past two decades. He also notes how certain artists have dominated the charts and is especially interested in the representation of underrepresented communities in the industry.

A. MusicVis

Designed by Guedes and Freitas, MusicVis is a web-based application consisting of a menu and a main data visualisation area [14]. MusicVis' system architecture and dataset is very similar to the nature of Soundscape. Firstly, it offers visualisations that include an interactive sunburst diagram, node-link tree, tree map, and bubble chart, which were developed using the D3 JavaScript library. These visualisations are shown in Fig. 3. Guedes and Freitas opted for such visualisations to deviate from traditional tabular formats. Secondly, MusicVis utilises music ranking datasets from both Billboard and Spotify. Additionally, users have the option to upload their personal Last.fm data.

Guedes and Freitas developed MusicVis with a user-centred design approach. The platform had the following goals: (i) allow users to compare and filter the performance and position of artists, tracks, and genres, and (ii) compare their personal Last.fm data with traditional music rankings. Focusing on the user's needs, preferences, and goals, Guedes and Freitas conducted a remote user survey prior to beginning development. By gathering data on users' musical preferences and listening habits, Guedes and Freitas were better informed in their design decisions, determining which visualisations, techniques, and datasets to incorporate into MusicVis.

Guedes and Freitas conducted a user evaluation study that aimed at assessing the usability and learnability of MusicVis. The study revealed that there were varying opinions regarding the advantages and disadvantages of each visualisation. The sunburst visualisation was the most favoured with 35.1% of users expressing their preference for it. The node-link tree visualisation followed closely with 31.9%. Only 16% of users liked the tree map visualisation.

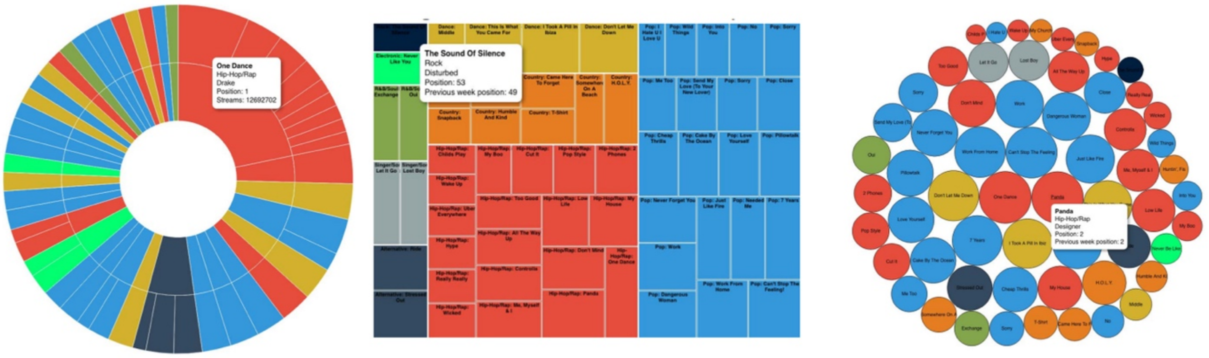


Fig. 3. Interactive data visualisations as part of the MusicVis web-application. From left to right: sunburst diagram, tree map, and bubble chart.

There are certain drawbacks in utilising Guedes and Freitas' user study for informing the design of the project's usability The questionnaire employed in the study lacked a specific focus on enhancing users' data comprehension by primarily relying on vague Likert scales with prompts such as "I found the chosen visualisations good" or "I liked the data search on the filtering tab." The evaluation of Soundscape, discussed in Section V, aims to address the limitations of MusicVis' user study by conducting a Heuristic evaluation.

B. Hyper Word Clouds

The aforementioned Last.fm online music platform allows users to track their listening habits. Users scrobble (track) the music they listen to then this information is transferred to Last.fm's database through a range of music streaming platforms such as Spotify, Apple Music, SoundCloud, or a plug-in installed into the user's music player [15]. Hyper Word Clouds is an interactive visualisation system that allows users to analyse the intricate relationships across multiple tables in Last.fm data [16], depicted in Fig. 4. Similar to Soundscape, the system was implemented as a web application, employing jQuery and D3.js for its visualisation. The main visualisation represents the Last.fm dataset as parallel word clouds, where each cloud corresponds to a specific data table and contains a column of words. Nguyen and Le's technique extend beyond set, connection, and quantitative relations by incorporating anchor paths. These paths serve as references for data loading, interaction selection, as well as filtering, allowing for the visual examination of ordered relations. Hyper Word Clouds allows users to explore the interconnectedness within the dataset that encompasses songs, album, artists, usernames, and other relevant information. Its interface includes the described visualisation as well as a menu bar for data loading and interaction handling.

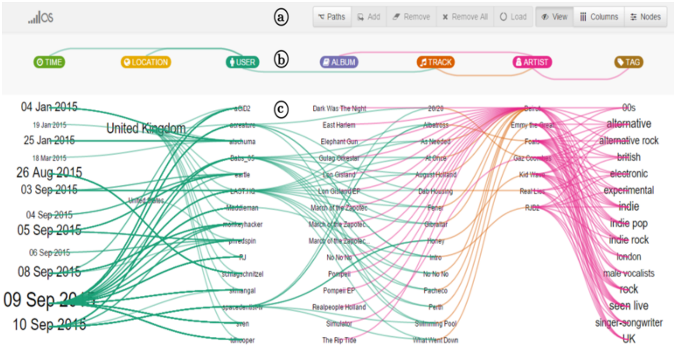


Fig. 4. The Hyper Word Clouds interactive data visualisation features equal-sized columns with words aligned horizontally across the screen. These words are colourised, resized, and displayed with a limited character count, and arranged according to a predefined criterion.

Given that the data primarily focuses on Last.fm's user activity, it cannot be directly leveraged for the project's specific focus on the complexities of music ranking data. Moreover, the absence of a user study presents a further challenge in assessing the system's effectiveness in facilitating interactive selection, filtering, highlighting, and comparison of Last.fm data and relationships. However, a key takeaway from this tool is the importance of employing existing interactive visualisation technologies. Even with novel techniques, such as parallel columns of word clouds with anchor-links for text-based data, the authors avoid unnecessarily "reinventing the wheel" by leveraging D3.

C. The Art of Storytelling with Data Visualisations

Upon reviewing the two prior works, MusicVis and Hyper Word Clouds, we assess the pros and cons of their technical strategies. Specifically, we focus on their technical approaches for developing a web application that interactively visualises data from music streaming services. Both works leveraged the D3 JavaScript library to develop custom interactive data visualisations tailored to the needs of specific end-users. Yet, these examples do not offer a clear solution for narrating the intrinsic patterns within their datasets. Soundscape, on the other hand, seeks to address this gap by presenting the dataset illustrated in Fig. I through a positive user experience.

Therefore, we are inspired by two existing, public interactive data visualisation websites: *Film Dialogue from 2,000 screenplays, Broken Down by Gender and Age* [17], and *The New Normal* [18]. While they do not deal with data related to music, they both boast a sophisticated, visually-appealing, user-friendly interface. Both websites utilise "infinite scrolling", a prevalent and user-friendly navigation method where users simply scroll continuously to access all the content on the webpage. This approach streamlines website navigation [19]. Additionally, the storytelling element of these sites is enhanced by well-curated headings and descriptive text, providing users with the context behind the data. For instance, the Film

TABLE V
OVERVIEW OF SHNEIDERMAN’S VISUAL-SEEKING MANTRA

Overview	Provide an overview of the entire dataset to understand its scope and structure.
Zoom	Enable users to zoom in and focus on specific items for closer examination.
Filter	Enable users to filter out or exclude uninteresting items to streamline the visualisation.
Details on Demand	Enable users to select an item or group to access detailed information when needed.
Relate	Enable users to view and explore relationships between different items in the dataset.
History	Support a history of user action, enabling features like undo, replay, and progressive refinement.
Extract	Provide the ability to extract sub-collections or specific query parameters for further analysis.

Dialogue website displays *The prevailing theme: white men dominate movie roles* as the first sentence on the UI. This assertive proclamation immediately captures the user’s interest, compelling them to delve deeper into the data and understand the reasoning behind the claim itself and the purpose of the website. This strategy, often termed ‘clickbait’ on the internet, is effective in grabbing the user’s attention [20].

Conversely, The New Normal is an interactive data visualisation website that explores the influence of the COVID-19 pandemic on Google search trends [18]. Shown in Fig. 6, the landing page of the website prominently displays an animated line chart visualisation, which tracks the frequency of searches for various items, including toilet paper, Tequila, socks, and bleach, from 2019 to 2022. Crucially, the x-axis highlights the point at which the World Health Organisation declared COVID-19 a global pandemic, offering context for the data.

D. *Visual-Seeking Mantra*

The “Visual-Seeking Mantra” proposed by Shneiderman in 1996 [21], is a frequently cited guideline for designing interactive visualisations, whereby researchers who develop novel interactive techniques employ the Mantra as a justification for their methodological approaches [22]. Fig. V provides an overview of the Mantra. Craft and Cairns argue that validation by further researching the usability of a system is critical for the development of visualisation methodology. The project addresses this in Section V by conducting a Heuristic evaluation of the system. Craft and Cairns also critique of using the Mantra’s guidelines by itself. They claim it is inadequate and a strong methodological approach should tackle a broad range of visualisation design issues [22]. Craft and Cairns propose a holistic design methodology advocated to address the identified issues; incorporate useful techniques from guidelines and patterns, possess measurable validity, be rooted in a user-centred development framework, offer a systematic procedure, and cater to both beginners and experts [22].

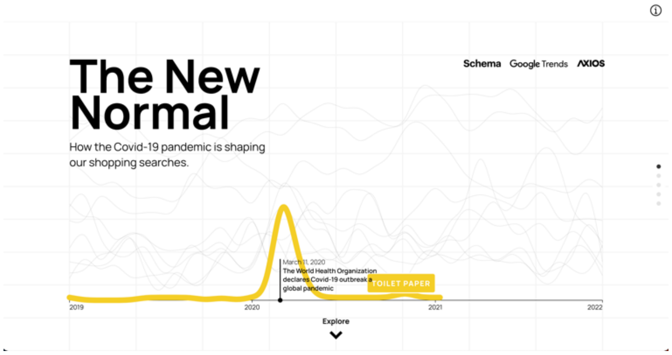


Fig. 6. The New Normal interactive data visualisation is a public website. On the site’s landing page, there is an animated multi-series line chart, depicting how frequent various items were searched for on Google in relation to when COVID-19 was announced as a global pandemic.

III. DESIGN

This section delves into the proposed design solution for the problem discussed in Section I, A.

A. *System Architecture*

The illustrated diagram in Fig. 7 displays the proposed system architecture of our web-based application, sectioning it into front-end and back-end components. This architectural design aligns with the web-based requirement specified in Section I, A, 1), b).

In the front-end, users are presented with four interactive data visualisations: a bar chart race, correlation heatmap, treemap, and a multi-series line chart.

On the back-end, song metadata is fetched from Spotify’s restful Web API endpoint for the data fetching requirement outlined in Section I, A, 1), c). Once retrieved, this unprocessed data is processed and manipulated using specific data algorithms. The prepared data is then utilised to render the data visualisations in the front-end. Furthermore, the diagram highlights that users can access the web application via their computing devices. This setup allows users to engage interactively with the visualisations, necessitating real-time updates in data logic and parameters based on their interactions.

B. *User Interface Theme*

To meet the usability requirements outlined in Section I, A, 2), the UI was designed to have a theme and aesthetics similar to conventional web pages. A wireframe of the main page of the system’s UI is illustrated in Fig. 8. Additionally, the layout is designed for straightforward navigation, utilising a single-page format where users can scroll without feeling disoriented. Consequently, the UI components are methodically arranged, echoing the principles from related works in Section II, C, which underscore the significance of narrating a data-driven story.

C. *Bar Chart Race*

Shown in Fig. 8(b), the initial interactive visualisation designed for the UI is a bar chart race, highlighting the top ten artists with the highest number of songs over the years. The bar

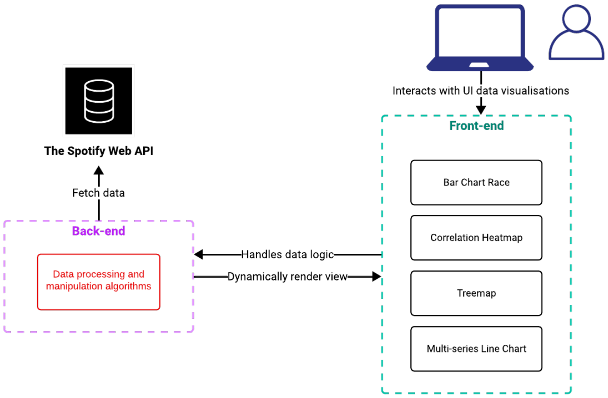


Fig. 7. The proposed design of Soundscape’s system architecture to address the system requirements outlined in Section I, A.

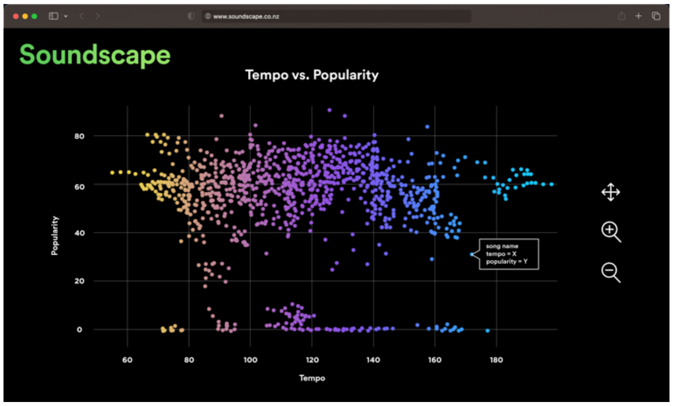


Fig. 8. A wireframe of an interactive scatterplot graph visualisation to analyse the relationships between two song characteristics found in the system’s dataset. Please note that the data is not accurate or truthful, given that the wireframe was designed in Photoshop.

chart race serves as an effective visualisation because it provides a comparative view of the artists’ changing successes over time, which is of keen interest to the end-users discussed in Section I, A, 3).

The project’s user experience was designed in a similar vein of the interactive data visualisation system, The New Normal, discussed in Section II, C. The system aims to include the bar chart race visualisation as the first component of the UI, priming the user for deeper exploration into the intricacies of the music data, such as the reasons certain artists dominate in Spotify hits.

The New Normal also commences with an animated line chart prominently displayed on its homepage and this strategic placement effectively captures the user’s attention, offering an immediate overview of the COVID-19 global pandemic’s impact on Google search trends. By doing so, it lays the foundation for users, clarifying the core intent of the visualisation web-application.

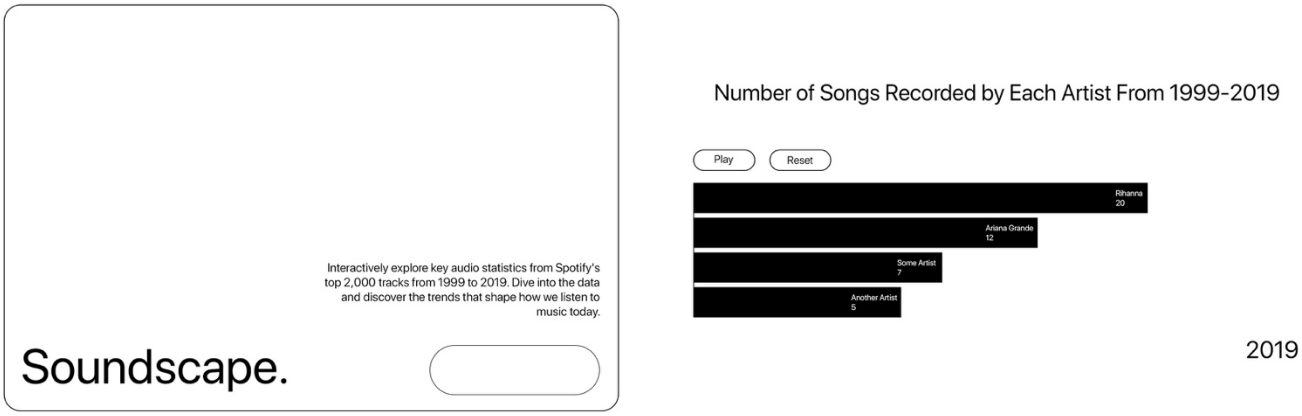
Originally, the bar chart race was designed to be animated (reminiscent of the introductory line chart visualisation in The New Normal) complete with play, pause, and reset functionalities to cycle through the animation from the start. However, by following Shneiderman’s Visual-Seeking Mantra, a strategic design decision led to adding an interactive slider below the bar chart. This slider, spanning from 1999 to 2019, empowers users with greater autonomy, granting them the ability to navigate through the years at their own pace. Such an approach not only fosters deeper user engagement but also allows a closer examination of change in data, that is, the artists who did and did not consistently secure top ten positions across two decades. The Mantra emphasises that users should have the capability to swiftly revert to prior stages of data exploration because comparing the current view with previous states often enhances data understanding [21]. Furthermore, an ideal interface offers a tool for state reversal, facilitating iterative data exploration refinement [21].

D. Multi-Series Line Chart

The interactive bar chart race emphasises shifts in data over time, highlighting dominant artists in the music industry. To complement this, a user-friendly visualisation was designed to let users delve into the relationships among Spotify’s popular songs and its metadata regarding audio characteristics, as depicted in Fig. I. This aligns with the functional requirement of allowing users to learn about trends about different genres and their relationship with audio metrics outlined in Section I, A, 2), b). Additionally, the Visual-Seeking Mantra states that facilitating the identification of relationships is crucial when comparing characteristics of distinct data objects within the display [21]. Initially, a scatterplot graph, shown in Fig. 8, was considered to represent the relationship between two selected audio characteristics. However, it was later decided that a multi-series line chart would be employed to maximise the amount of data relationships that could be viewed under one interactive data visualisation by including genres as well.

The interactive multi-series line chart aims to let users delve into the relationship between the average audio characteristics of different genres over the years. The wireframe for the multi-series line chart is shown in Fig. 9(d). The design of the chart would include lines that each represent a distinct genre, accompanied by an additional line portraying the comprehensive average for that particular year. Users are empowered with the option of a dynamic filter, allowing them to pick their desired audio characteristic from a dropdown menu. The design of this feature was driven by the Visual-Seeking Mantra, allowing users to quickly see how the changed variable affects the data representation [21].

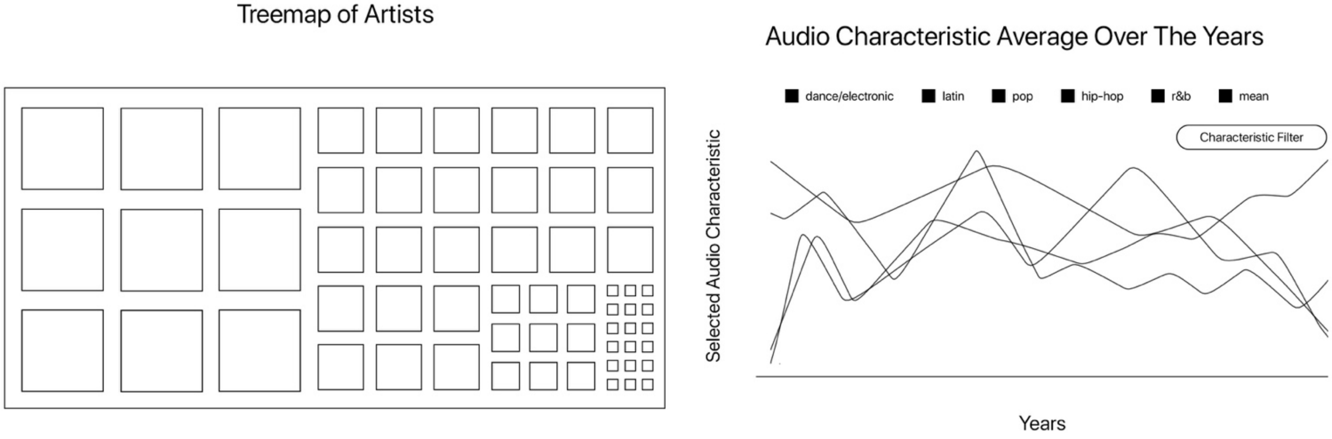
Moreover, the line chart was designed so that users have the choice to determine which genres are displayed on the line chart. This design decision was motivated by the Mantra’s principle of “details-on-demand” [21]. Delivering exhaustive data about every item displayed is not feasible [21]. Hence, the ability to select or deselect genres becomes a vital option, ensuring users receive pertinent information without requiring a change of view of the visualisation. The essence of details-on-demand is further realised through the design of mouse hover interactivity incorporated within the chart. Specifically, when a user hovers over a line using their mouse, it would perform two tasks: (i) the selected line on the chart is highlighted while dimming the others, and (ii) a tooltip is revealed, providing precise average values, genre information, and the year in question. This design decision ensures users are able to



Interactively explore key audio statistics from Spotify's top 2,000 tracks from 1999 to 2019. Dive into the data and discover the trends that shape how we listen to music today.

(a) The landing page.

(b) The bar chart race visualisation.



(c) The zoomable treemap.

(d) The multi-series line chart.

Fig. 9. The wireframes of Soundscape's interactive data visualisations that would appear on the main page of the UI. Users would scroll down the page to view each component in the following order: (a), (b), (c), (d). The correlation heatmap visualisation was omitted from the wireframe designs but would appear after the multi-series line chart on the UI.

seamlessly pinpoint specific data elements of interest, maintaining the chart's holistic view [21].

E. Correlation Heatmap

Fig. 8 presents a high-level wireframe of a scatterplot graph, originally designed to represent the relationships between two audio characteristics within the Spotify dataset. However, this design was replaced with a correlation heatmap too. Unlike the scatterplot, a correlation heatmap uses a colour-coded matrix to vividly depict how closely related multiple variables are, making it appropriate for displaying the assortment of audio characteristics found in Spotify's song metadata. The correlation heatmap is designed to enable users to delve deeper into the ties binding these metrics, irrespective of genre. Furthermore, the visualisation will be placed after the multi-series line chart in the system's UI, providing a seamless transition to delve deeper into the exploration of audio characteristics.

F. Zoomable Treemap

The final interactive data visualisation that was designed for the user interface is a zoomable treemap, which provides an overview of the dataset. The zoomable treemap is designed to display artists, their corresponding songs, genres, and overall popularity. The popularity is calculated by the total popularity score of an artist's songs. The visualisation would unfold in three distinct layers; (i) the initial view presents all artists alongside their popularity scores, (ii) a click on an artist's section expands to reveal their songs displayed as cells and (iii) finally, selecting a song cell provides a detailed view of its associated genres.

Shneiderman's Visual-Seeking Mantra states that an overview offers a broad context to grasp the dataset, presenting a holistic image of the data [21]. This comprehensive perspective reveals significant patterns, relationships, and features that might not be noticeable from a partial view [21]. By highlighting these elements initially, users can effectively filter out unnecessary information, making their tasks more efficient.

IV. IMPLEMENTATION

The previous section presented an overview of the system architecture, the interactive data visualisations within the user interface, and the system's intended user experience. This section will discuss the tools and methodology used to implement the proposed system design, a web-application named Soundscape, including how each visualisation was realised.

A. *Tools and Methodology*

1) *D3*

D3.js is a JavaScript library explicitly designed for dynamic and interactive data visualisations in web browsers that offers an extensive array of capabilities for manipulating and visualising data. Data can be seamlessly bind to visual elements, allowing developers to have precise control over all aspects of the visualisation process, such as data loading, manipulation, rendering, and interaction. This flexibility enables the creation of bespoke visualisations tailored to effectively communicate intricate data insights to users.

2) *React*

React is a JavaScript library for building the system's user interface. React provides reusable UI components for the front-end development of the web application. It follows a component-based architecture where the user interface is divided into modular, self-contained components that can be easily composed to create more complex UI structures. The system's UI consists of a homepage and a navigation bar.

3) *HTML/CSS*

HTML/CSS are fundamental technologies for building web-application's webpages. HTML structures the content and defines the layout, whereas CSS is responsible for styling and presentation. Together these tools are used to develop the user interface and visual aesthetics of the system.

4) *Node.js*

Node.js provides the runtime environment to execute JavaScript code outside of the browser. This is essential for various build and development tools that are written in JavaScript. Node.js comes with npm, which is a package manager used to install libraries and tools. React and many other associated libraries are available as npm packages.

Overall, D3 handles the data operations, while React efficiently manages the rendering and state management, resulting in a seamless integration of data-driven visualisations within React applications. The data processed by D3.js can be passed as props to React components, which then render the visual elements based on the provided data. React's virtual DOM efficiently updates and re-renders only the necessary parts of the visualisation when the data or state changes.

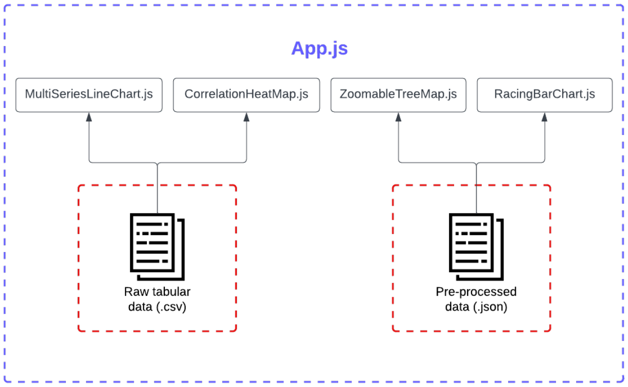


Fig. 10. The locally-run implemented web application system architecture of Soundscape. App.js is the main React component that is run, containing the visualisations. The data is locally stored in the application's files.

B. *Web Application and Data Files*

The React web-application was created using Create React App [23, 24]. Consequently, locally running the app on a web browser requires Node 14.0.0 on the local machine [24]. Creating a React application is simple and can be done by running the command `npx create-react-app my-app`.

The Spotify data was downloaded from Kaggle [8] as a tabular CSV file and stored locally within the web-application's project files. This data contained audio metrics of the top 2,000 Spotify tracks from 1999 to 2019. While some visualisations dynamically read and process the data using algorithmic processes, other visualisations read and process pre-processed JSON data files. These pre-processed JSON data files were generated using Python scripts that processed the raw CSV data file, which will be discussed in the implementation sections of the treemap and bar chart race visualisations. Illustrated in Fig. 11 is the structure of the main React project file components that was ultimately implemented as a proof-of-concept prototype.

C. *Bar Chart Race*

The bar chart race visualisation, was developed using an open-source, fully customisable React component [25], as depicted in Fig. 12. Notably, its associated Github repository had not seen updates in several years [25]. As a result, modifications were made to its codebase, aligning with a modern JSX syntax.

The bar chart race interactive visualisaiton is structured into three primary React components. BarChart acts as the parent component, facilitating data flow to its child component, RacingBarChart. Within BarChart, there are as many Bar components as specified in the maxItems property, each dictating the styling for the individual bars in the chart. Additionally, BarChart incorporates a slider, allowing users to traverse through different years in the specified timeframe. This slider was an addition, not present in the original open-source component, tailored specifically for this project as discussed in Section III, F. The BarChart component expects a data property; therefore, the RacingBarChart.js component passes a pre-

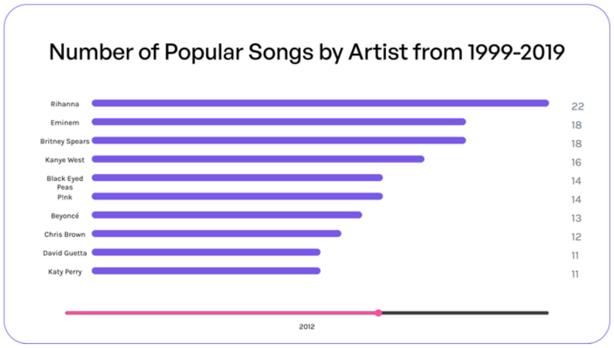


Fig. 11. Soundscape's bar chart race interactive visualisation.

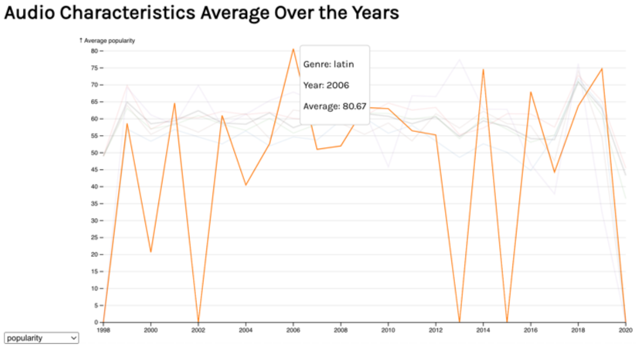


Fig. 12. Soundscape's multi-series line chart interactive visualisation. Its tooltip appears when a user hover's their cusor over a line on the chart.

processed JSON data file. This file is derived from a Python script which processes the original CSV data of the top Spotify hits dataset.

D. Multi-Series Line Chart

Shown in Fig 12, the multi-series line chart was implemented using the D3.js library to visualise the average of various audio characteristics over the years by genre.

First, the necessary dependencies are imported. This includes React hooks, the D3 library, and the CSV dataset. State variables, constants for dimensions, genres, and audio characteristics are then defined. The component relies heavily on React's useEffect hook for various lifecycle events and data manipulation operations.

Upon component mount, the first useEffect is triggered. It fetches the dataset using D3's csv method and sets the returned data to the data state. Subsequent to the data being fetched, another useEffect performs a data processing algorithm to calculate the averaged values of the selected characteristic for each year. For each year, it first determines the overall average for the characteristic and then computes averages for each genre. The min and max values for the selected characteristic are also dynamically computed, ensuring the y-axis scale is adapted to the range of the data.

Once the averaged data is set and the y-axis scale values are determined, another useEffect is invoked. This effect focuses on rendering the line chart using D3. First, any existing SVG within the chart's container is removed to ensure a fresh render. The SVG canvas is then appended to the container. Scales for the x and y axes are defined based on the data.

Two significant sections in this rendering process involve plotting lines: one for the overall average and others for individual genres. Both use D3's line generator. An animation is applied to each line to make it appear as if it's being drawn on the line chart for a pleasant user experience. Event listeners are attached to each line to enable interactions. On hovering over a line, other lines' opacity is reduced, and a tooltip displaying information about the hovered year is shown.

Finally, the x and y axes are appended to the SVG. For the y-axis, a conditional formatting is applied to the tick labels based on the selected audio characteristic.

The rendered component consists of a title, a dropdown to select an audio characteristic, and the line chart itself. The dropdown allows users to switch between various audio characteristics, triggering a re-render of the chart with updated data. The detailed approach in this component showcases a typical pattern in combining React with D3. React manages the state and lifecycle events, while D3 is leveraged for its powerful data-driven document manipulation capabilities, ensuring the visualisation is both dynamic and interactive.

E. Correlation Heatmap

The correlation heatmap visualisation is a React functional component that employs the D3.js library to create a correlation heatmap based on data from a CSV file. Upon importing the necessary dependencies from both React and D3.js, the dataset CSV file is imported. Inside the component, there are three main state variables: data which holds the CSV data, correlations that will keep the correlation values of the various fields in the dataset, and svgRef which is a reference to the SVG element where the heatmap will be rendered.

The first useEffect hook is responsible for data loading. It leverages D3's csv method to read and parse the dataset. Once the data is loaded, it's stored in the data state variable.

To compute the correlation between various fields, a defined list named interestedFields is utilised, containing field names from the dataset like "duration_ms", "year", and "popularity" among others shown in Fig I. A helper function, corr, calculates the correlation between two given fields.

Subsequently, another useEffect hook computes the correlations between pairs of fields. It uses D3's cross method to generate combinations of fields and then computes their correlation using the corr function. The results are then stored in the correlations state variable.

Rendering is done through another useEffect hook. This section is tasked with visualising the correlation data onto the heatmap. First, it clears any existing SVG contents to ensure a fresh render. It establishes xScale and yScale to map the data fields to screen coordinates, while a colorScale determines the colour of each cell in the heatmap based on its correlation value, using a sequential colour scale ranging from red to blue. The heatmap itself is formed by SVG rectangles (rect elements) that represent each correlation value, with colors indicating the strength and direction (positive or negative) of the correlation. Overlaying the heatmap, SVG text elements (text) display the correlation values, with text colour adjusted for readability based on the correlation value.

The component finally renders a header, "Correlation heatmap", followed by the SVG canvas, where the heatmap will be visualized. The canvas is sized at 1000 by 1000 pixels, ensuring sufficient space for detailed representation.

F. Zoomable Treemap

The ZoomableTreemap visualisation, detailed in the provided code, presents an advanced data representation technique enabled through a combination of React and D3. These libraries work cohesively, with React handling the component lifecycle and D3 providing the advanced data visualisation capabilities.

The component initialises with default width and height values, though these can be overridden by parent components. The data is sourced from a JSON file, "output_file.json", and is utilized within the component's effect, which is triggered post-render. Notably, the absence of data will halt the effect's execution, ensuring error-free operation.

An important feature of this visualisation is its custom tiling function. This function is responsible for adapting the default binary tiling of D3 to adjust to the appropriate aspect ratio when the treemap is zoomed in. This ensures that the zoomed-in view maintains a clear and proportional representation.

To ensure data is effectively visualised, the component employs a hierarchical D3 structure which then undergoes a treemapping transformation. Essential scaling operations are performed using D3's linear scaling, allowing the component to fit and adjust visualisations within the specified width and height parameters.

The visualisation's main display is an SVG element, which is dynamically adjusted based on the data. This SVG contains multiple groups of visual nodes. Each node can represent a parent or child entity and can be zoomed into or out of, providing a granular view of the data.

Additionally, for enhancing the user experience, the treemap supports zooming functionality. Two distinct functions, "zoomin" and "zoomout", manage these zoom operations, each ensuring a smooth transition with the aid of D3's built-in transitions. Not only does this zooming allow users to dive deeper into specific data segments, but it also provides an aesthetically pleasing and interactive experience.

Lastly, the use of universally unique identifiers (UUIDs) within the component ensures each leaf node maintains a distinct identity, which is crucial for accurate referencing and rendering, especially when operations like zooming are applied.

V. EVALUATION

The implementation section outlined how the proof-of-concept prototype, Soundscape, was developed. In this section, we discuss how the system's usability and effectiveness to facilitate insights about the data was assessed by conducting a heuristic evaluation.

Three participants, all students from Victoria University of Wellington (VUW), served as evaluators by identifying potential usability and experience issues in the system. These participants had previously taken either SWEN303, a 300-level User Experience Engineering course, or SWEN422, a 400-level Human-Computer Interaction course, equipping them with a strong understanding of UI practices and allowing them to assume 'expert' roles. Under VUW's Human Ethics Committee, this evaluation has been approved under the application ID 0000029386.

A. Methodology

A heuristic evaluation is an assessment to pinpoint design issues in a user interface [26]. Evaluators compare the UI design to certain guidelines (the heuristics) that enhance system usability [26]. The evaluators followed Jakob Nielsen's ten usability heuristic principles for UI design [27] (as depicted in Fig. XIII) and Shneiderman's Visual-Seeking Mantra guidelines for crafting interactive data visualisations (illustrated in Fig. V). This dual-guideline approach was adopted in response to Craft and Cairns' critique, which suggested that solely relying on the Mantra' guidelines could be ineffective for assessing the entirety of a system [22]. Furthermore, Nielsen Norman Group's suggest incorporating other domain-specific heuristics for niche UI areas or specific usability evaluations, such as the interactive visualisations [26]. During the system evaluation, evaluators adopted the personas developed for the user model and simultaneously approached the task as UI design experts, offering rigorous critiques.

B. Evaluator 1

Evaluator 1 found the "Get Started" button on the landing page, depicted in Fig. 14, to be particularly effective. This button served as a shortcut, making the users scroll down to the bar chart race, the initial visualisation. Evaluator 1 felt that the button acted as a streamlined pathway, quickly directing them to the subsequent primary UI element. This fulfilled the 7th usability guideline.

Upon interacting with the bar chart race visualisation, Evaluator 1 was impressed by the dynamic animations of bars shifting positions with changing years. This fluid motion enabled them to discern changes in the data, such as identifying artists who frequently ranked among the top ten, fulfilling the 'Filter' guideline of the Visual-Seeking Mantra. However, the uniform purple shade of all the bars was a point of contention for them. Evaluator 2 remarked "I have to look at the labels of the artists' names to know who's who." Notably, Evaluator 1's reliance on text labels for each bar representing an artist aligns with the 6th usability guideline, "Recognition Rather Than Recall," which ensures elements and content remain visible to minimise the user's memory load.

However, Evaluator 1 highlighted a potential UX misstep concerning the audio characteristics documentation. For instance, while exploring the multi-series line chart, Evaluator

TABLE XIII
OVERVIEW OF JAKOB'S TEN USABILITY HEURISTICS

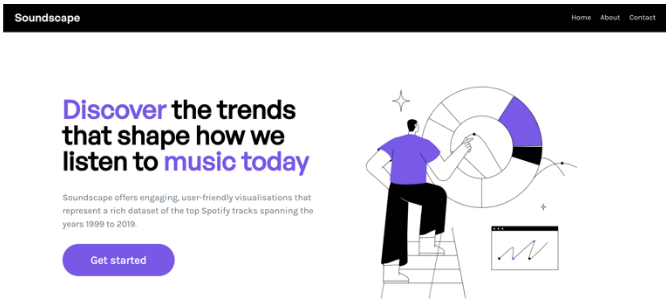
1. Visibility of system Status	Designs should keep users informed about what is going on, through appropriate, timely feedback.
2. Match between System and Real World	The design should speak the users' language. Use words, phrases, and concepts familiar to the user, rather than internal jargon.
3. User Control and Freedom	Users often perform actions by mistake. They need a clearly marked "emergency exit" to leave the unwanted action.
4. Consistency and Standards	Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.
5. Error Prevention	Good error messages are important, but the best designs carefully prevent problems from occurring in the first place.
6. Recognition Rather Than Recall	Minimise the user's memory load by making elements, actions, and options visible. Avoid making users remember information.
7. Flexibility and Efficiency of Use	Shortcuts — hidden from novice users — may speed up the interaction for the expert user.
8. Aesthetic and Minimalist Design	Interfaces should not contain information which is irrelevant. Every extra unit of information in an interface competes with the relevant units of information.
9. Recognise, Diagnose and Recover from Errors	Error messages should be expressed in plain language (no error codes), precisely indicate the problem, and constructively suggest a solution.
10. Help and Documentation	It's best if the design doesn't need any additional explanation. However, it may be necessary to provide documentation to help users complete their tasks.

1 struggled to recall the specifics of each audio characteristic. The brief descriptions provided by the chart's tooltip on mouse hover proved insufficient for a comprehensive understanding. This severely broke the 10th usability guideline of "Help and Documentation," given that the placement of the documentation appearing after the bar chart race meant that users lacked immediate context when they most needed it.

Regarding the correlation heatmap and treemap, Evaluator 1 expressed feeling somewhat adrift due to the absence of clear labels or guiding instructions. This feedback was anticipated, given that the complete functionalities for these visualisations hadn't been integrated yet.

C. *Evaluator 2*

Evaluator 2 found the UI theme of Soundscape effective, noting that it centred on crucial elements in its design and content. Yet, they believed that the visualisations lacked a unified theme. To address this, they suggested adding brief explanations to clarify the present view of each visualisation. For instance, upon engaging with the multi-series line chart and adjusting the song characteristic to 'popularity' using the drop-down filter, they remarked, "It would be helpful if there was some context to explain why the rock genre reached its peak popularity in 2013." Evaluator 2's preference appears to be influenced by adopting the personas from the user model—end-users who are inherently intrigued by music industry trends. While Evaluator 2 didn't specifically identify this as a UI usability issue, the 10th usability guideline, "Help and Documentation," suggests that users might need documentation to assist them in their tasks. In this context, the task that Evaluator 2 wanted to complete pertains to facilitating the analysis of phenomena within the dataset.



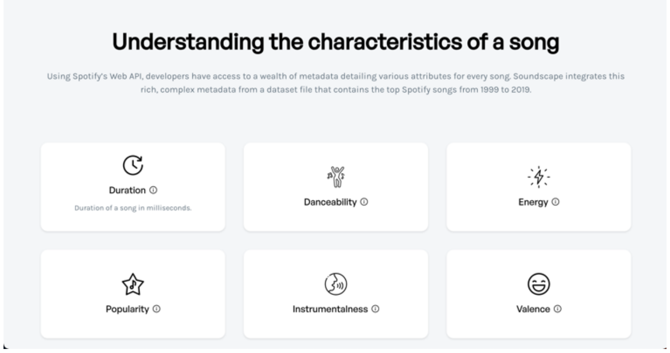
Discover the trends that shape how we listen to music today

Soundscape offers engaging, user-friendly visualisations that represent a rich dataset of the top Spotify tracks spanning the years 1999 to 2019.

Fig. 14. The landing page of Soundscape's user interface. Clicking on the "Get Started" button scrolls the user down the page to the bar chart race visualisation shown in Fig. 12.

For the UI design detailing song characteristics shown in Fig. 15, Evaluator 2 thought the design was striking. However, they encountered challenges accessing descriptions for each characteristic. They observed, "The tiles for each characteristic seemed clickable because of the changing cursor but nothing happened when I clicked [it]." They advised making the entire tile responsive to clicks, rather than just the icon adjacent to the characteristic's title. Moreover, they said that there was not enough information to comprehensively understand what the characteristics meant.

Similar to Evaluator 1, Evaluator 2 was confused by the correlation heatmap visualisation, describing it as a "grid filled with colourful numbers lacking context." This reaction was expected since the heatmap's functionality was not yet fully realised. The treemap, on the other hand, also puzzled Evaluator 2, but they were appreciative of its zooming animations, noting its effectiveness in visualising data structures. This adheres to the Mantra's guidelines of "Zoom" and "Overview." Additionally, they expressed surprise at the prominence of Selena Gomez's cell within the treemap. Delving further by clicking on her cell, they recalled, "Oh, I remember the song 'Come & Get It.' That song was my favourite when I was younger." However, once inside the detailed view of Selena Gomez's song within the treemap, Evaluator 2 found no clear way to navigate back to the overarching artist view and felt the UI didn't make the return path evident. This severely breaks the 1st usability guideline of "Visibility of System Status."



Understanding the characteristics of a song

Using Spotify's Web API, developers have access to a wealth of metadata detailing various attributes for every song. Soundscape integrates this rich, complex metadata from a dataset file that contains the top Spotify songs from 1999 to 2019.

Duration of a song in milliseconds.

Fig. 15. Soundscape's documentation section of the song characteristics within the system's dataset. A user can click the (i) next to a label to reveal or hide the characteristic's description.

D. *Evaluator 3*

Evaluator 3 was impressed with the main heading of Soundscape's landing page, commenting on its clarity and the evident objective of the website. This achieves the 8th guideline "Aesthetic and Minimalistic Design", which aims to only contain relevant information in the UI.

While they opted to scroll down to the bar chart race visualisation rather than clicking the "Get Started" button, they found its design to be both "simple and effective." As they interacted with the visualisation, Evaluator 3 made a keen observation, noting that it was "interesting to see the Black Eyed Peas' decline after 2012." This is significant as the Evaluator was able to engage in analytical reasoning interacting with the bar chart race visualisation.

Evaluator 3 felt that the visualisations appeared somewhat disjointed, each seeming to have its "own objective." To address this, they suggested replacing the infinite-scrolling feature with a main hub and buttons to access the different visualisations for better navigation. When the idea of retaining infinite-scrolling but incorporating captions and contextual text was proposed, Evaluator 3 believed this could indeed enhance the overall cohesion. Given that Soundscape aimed to tell a story using the Spotify data, the implementation of infinite-scrolling was favoured in the design phase.

Evaluator 3 found the multi-series line chart sophisticated in its features. However, they felt that relying solely on tooltips made it challenging to decipher the meaning of each line on the chart. This breaks the 6th usability guideline of "Recognition Rather Than Recall." As a fix, they proposed integrating a colour-coded key to denote what each line represents, along with options to toggle their visibility. Although this design feature had been discussed in Section III, G, it was not actualised during the implementation phase. Nevertheless, Evaluator 3 was particularly fond of the animated drawing of lines onto the chart, feeling it enriched the user experience when adjusting data parameters.

E. *Summary of Evaluation Findings*

In this section, we highlight the common findings identified by the three evaluators assessing the usability of the system and its interactive data visualisations.

Upon entering the Soundscape's landing page, Evaluator 1 and 3 had varied interactions with the "Get Started" button displayed in Fig. 14. While Evaluator 1 clicked the button, directing them to the bar chart race visualisation, Evaluator 2 chose to scroll down to manually locate what they though was the subsequent UI component. This difference suggests that the "Get Started" button offers expert users a clear navigational shortcut according to the 7th usability guideline of "Flexibility and Efficiency of Use." Further user tests could explore how those unfamiliar with the system would approach this navigation feature.

Both Evaluator 1 and 3 expressed a preference for the bar chart race visualisation. Evaluator 1 was particularly impressed with the animations of the bars when adjusting the year using the slider. Evaluator 3, on the other hand, was able to draw insights from the data, noting trends related to the Black Eyed Peas over time. Evaluator 3 also found the treemap's zoom animations helpful in understanding the data structure, and Evaluator 2 recognised music trends related to Selena Gomez using the Zoomable Treemap. This suggests that animations highlighting changes or structures in data enhance users' comprehension.

Evaluators 1 and 2 were confused after encountering the correlation heatmap within the system. Given that the heatmap did not turn out as initially planned in the project, this feedback was expected. From Evaluator 2's feedback, it's evident that the heatmap's colour scheme and values need clearer explanations.

Both Evaluator 2 and 3 highlighted issues concerning the lack of context for some visualisations. Evaluator 2 sought more detailed explanations for the multi-series line chart, while Evaluator 3 believed that individual visualisations in Soundscape should each convey a distinct point rather than trying to craft a singular narrative from the data.

VI. LIMITATIONS AND FUTURE WORK

Soundscape has certain limitations, which we will address in this section. We evaluate Soundscape in the context of being a proposed solution to the problem statement presented in Section I, A. Further, based on the conducted heuristic evaluation involving three evaluators—a number aligned with the Nielsen Norman Group's recommendations—we have pinpointed areas where Soundscape's usability could be enhanced.

A. *Improving the System's Usability Through Context*

The findings of the heuristic evaluation conducted on Soundscape were anticipated, given that the design of the multi-series line chart, zoomable treemap, and correlation heatmap were not fully realised. Further, there was a lack of context provided for the evaluators to grasp understand the overarching "plot" of the dataset. This was a design choice aligned with the related works under Section II, C. Offering a clear, deeper narrative for the dataset would entail adding supplemental details behind the interactive data visualisations. This point was highlighted during the evaluation, with Evaluator 2 recommending clarity on the rationale behind the music data's presentation in the multi-series line chart. To address this, the visualisations could be improved by adding relevant, bite-sized, user-friendly "fun facts" that are displayed next to the visualisations. For example, Evaluator 2 questioned the reasoning behind *why* a certain genre on the multi-series line chart was popular during a given year. This design improvement aligns with the system's user model depicted in Section I, A, 3, appealing to end-users with a keen interest in dissecting music trends.

B. *Social Awareness for Underrepresented Communities*

In future work, Soundscape aims to elevate social awareness of underrepresented communities within the music industry by integrating data directly from Spotify's streaming platform. The "Cultural Depth Analysis" system requirement underscores the need to spotlight lesser-known artists from underrepresented, non-English speaking communities, such as Māori or Pasifika artists. Given the predominantly western and English-based focus of Soundscape's current dataset, this requirement remains unmet. Nonetheless, by adopting the solution proposed in Section III, A, Soundscape has the potential to address this gap. This approach entails leveraging Spotify's Web API for data acquisition, rather than relying on static local CSV or JSON files. By doing so, Soundscape can offer users real-time interactions with up-to-date data. For instance, the multi-series line chart could be enhanced with filters, enabling users to explore artists from diverse regions beyond just the top-charting songs.

VII. Conclusion

In this report, we introduced Soundscape, a proof-of-concept React web application prototype designed to simplify the comprehension of Spotify's vast song metadata through interactive data visualisations, developed using D3. Targeted at individuals keen on exploring the evolving trends of mainstream artists and songs over time, Soundscape represents a crucial stride toward data-driven music exploration.

Our envisaged system architecture for Soundscape, detailed in Section III, A, intended to fetch real-time music data via Spotify's Web API. However, the current prototype operates locally, utilising integrated local data files—some of which were refined using Python scripts. D3's capabilities in Soundscape shine through its multi-series line chart, which adeptly processes and displays raw CSV data obtained from Kaggle. While certain visualisations harness pre-processed JSON files, the bar chart race visualisation leverages an open-source React component, bypassing D3's complexities. D3's intricate nature, rooted in its close alignment with foundational web technologies, offers significant customisation but demands a deeper understanding and often, more extensive coding.

Looking ahead, it is imperative to evolve Soundscape to directly interface with Spotify's Web API, ensuring users can continuously engage with fresh data. Such enhancements will not only extend the data's temporal scope beyond 1999-2019 but also illuminate lesser-known artists. Feedback from our heuristic evaluation underscored Soundscape's usability and highlighted its straightforward UI. However, the evaluation's scope had constraints, with evaluators already proficient in UI navigation. Thus, future endeavours should involve user studies, incorporating feedback from participants unfamiliar with advanced web application interfaces, ensuring a more comprehensive and user-centric evolution of Soundscape.

Acknowledgment

I extend my heartfelt gratitude to my supervisor, Stuart. Your belief in me, consistent guidance, support, and motivation have been invaluable. I genuinely feel fortunate to have had you as my guiding light throughout this journey.

My deepest appreciation goes out to Sahil, Michelle, and my family for their unwavering support. And a special shoutout to my friends who are also navigating the world of engineering with me; having you all by my side made it possible to see this project to its conclusion.

References

[1] Spotify, "Spotify Reports Second Quarter 2023 Earnings," Jul. 2023, https://newsroom.spotify.com/2023-07-25/spotify-reports-second-quarter-2023-earnings/, (Accessed on 1 July 2023).

[2] J. P. Friedlander, "News and notes on 2015 RIAA shipment and revenue statistics," *Washington, DC: Recording Industry Association of America*, 2016, https://www.riaa.com/wp-content/uploads/2016/03/RIAA-2015-Year-End-shipments-memo.pdf, (Accessed 10 Oct. 2023).

[3] M. C. Götting, "Music streaming revenue worldwide from 2005 to 2022," May 2022, https://www.statista.com/statistics/587216/music-streaming-revenue/, (Accessed 1 July 2023).

[4] Billboard, "The Hot 100 Chart," *Billboard*, 2023, https://www.billboard.com/charts/hot-100/, (Accessed 1 July 2023).

[5] M. Picard, "Visualized: Can we Quantify the Most Popular Music?," *Displayr*, Sep. 2018, https://www.displayr.com/most-popular-music/, (Accessed 29 June 2023).

[6] Spotify for Artists, "Playlisting – Spotify for Artists," https://artists.spotify.com/en/playlisting, (Accessed 9 October 2023).

[7] Spotify for Developers, "Web API | Spotify for Developers," https://developer.spotify.com/documentation/web-api, (Accessed 17 May 2023).

[8] M. Koverha, "Top Hits Spotify from 2000-2019," https://www.kaggle.com/datasets/paradisejoy/top-hits-spotify-from-20002019, (Accessed April 2023).

[9] K. E. Barkwell *et al.*, "Big Data Visualisation and Visual Analytics for Music Data Mining," *2018 22nd International Conference Information Visualisation (IV)*, Jul. 2018, doi: https://doi.org/10.1109/iv.2018.00048.

[10] R. S. Kalawsky, "Gaining Greater Insight through Interactive Visualization: A Human Factors Perspective," pp. 119–154, Jan. 2009, doi: https://doi.org/10.1007/978-1-84800-269-2_6.

[11] P. T. A. Junior and L. V. L. Filgueiras, "User modeling with personas," *Proceedings of the 2005 Latin American conference on Human-computer interaction - CLIHC '05*, 2005, doi: https://doi.org/10.1145/1111360.1111388.

[12] World Bank, "Europe and Central Asia Economic Update, Fall 2022: Social Protection For Recovery," *World Bank Publications*, Oct. 2022, doi: https://doi.org/10.1596/978-1-4648-1928-5

[13] United Nations, "The 17 sustainable development goals," *United Nations*, 2015, https://sdgs.un.org/goals, (Accessed 15 June 2023).

[14] L. Guedes, C. M. D. S. Freitas, "Exploring music rankings with interactive visualization," *Proceedings of the Symposium on Applied Computing,* Apr. 2017, doi: https://doi.org/10.1145/3019612.3019690.

[15] "Track My Music," *Last.fm.*, https://www.last.fm/about/trackmymusic, (Accessed May 2023).
[16] D. H. Nguyen and D. D. Le, "Hyper Word Clouds," Jan. 2016, doi: https://doi.org/10.1145/2857546.2857613.
[17] H. Anderson and M. Daniels, "The Largest Analysis of Film Dialogue by Gender, Ever," *The Pudding*, Apr. 2016. https://pudding.cool/2017/03/film-dialogue/, (Accessed June 2023).
[18] Schema Design, Google Trends, Axios. "The New Normal," *The New Normal*, https://thenewnormal.is/, (Accessed June 2023).
[19] T. Neusesser, "Infinite Scrolling: When to Use It, When to Avoid It," *Nielsen Normal Group*, Sep. 2022, https://www.nngroup.com /articles/infinite-scrolling-tips/, (Accessed 14 October 2023).
[20] J. A. Lischka and M. Garz, "Clickbait news and algorithmic curation: A game theory framework of the relation between journalism, users, and platforms," *New Media & Society*, vol. 25, no. 8, pp. 2073–2094, Jul. 2021, doi: https://doi.org/10.1177/14614448211027174.
[21] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," *Proceedings 1996 IEEE Symposium on Visual Languages*, 2020, doi: https://doi.org/10.1109/vl.1996.545307.
[22] B. Craft and P. Cairns, "Beyond Guidelines: What Can We Learn from the Visual Information Seeking Mantra?," *Ninth International Conference on Information Visualisation (IV'05)*, doi: https://doi.org/10.1109/iv.2005.28.
[23] "Create React App · Set up a modern web app by running one command.," Create-react-app.dev, 2019. https://create-react-app.dev/, (Accessed 11 October 2023).
[24] Facebook, "facebook/create-react-app," *GitHub*, Dec. 09, 2019, https://github.com/facebook/create-react-app, (Ac-cessed 11 October 2023).
[25] B. Chao, "chart-race-react," *GitHub*, Oct. 11, 2023, https://github.com/bchao1/chart-race-react, (Accessed October 2023).
[26] J. Nielsen, "Heuristic Evaluation: How-To: Article by Jakob Nielsen," *Nielsen Norman Group*, Nov. 01, 1994, https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/, (Accessed October 2023).
[27] J. Nielsen, "10 Heuristics for User Interface Design," *Nielsen Norman Group*, Nov. 15, 2020, https://www.nngroup.com/articles/ten-usability-heuristics/, (Accessed October 2023).