

HelaBeat: An Extensible Audio Streaming Mobile Application.

Dushani Perera¹, Maneesha Rajaratne¹, Shiromi Arunathilake¹
¹University of Colombo School of Computing, Colombo, SRI LANKA
¹{2015cs102.stu, 2015cs107.stu, sda}@ucsc.cmb.ac.lk

Abstract. With the increasing demand for applications supporting mobility, well-structured and competent mobile applications are a growing need. The music industry is one of the prominent sectors which is expanding its services to mobile platforms. This paper presents a novel design of a Mobile Music Streaming Application which provides music streaming services to users efficiently and effectively.

Keywords: Human-Computer Interaction, Music Recommender System, User-centered Design, Usability and User Experience, User Customization, Audio Streaming, Machine Learning

1 Introduction

Music streaming has offered the music industry and its artists a method for sharing and circulating music to users all over the world [1]. An efficient streaming application does not solely depend on the different features and components it has, as the efficiency and effectiveness of these components also play a major role. This paper presents a detailed description of the music streaming application named ‘HelaBeat’ which is the outcome of this research project. This application is currently available on the Google Play Store. This application focuses on providing efficient search suggestions, suitable recommendations and quick streaming. In order to achieve these features, research on each of the areas was carried out. Methods and techniques identified from this research were analyzed to choose the best way of implementing the new application. With this development based on research and a comprehensive literature review, ‘HelaBeat’ Android application is able to serve the community with convenient access to music and the artists with a platform to reach the public efficiently.

2 Overview of The Mobile Application

The ‘HelaBeat’ Android application has four main components: A Music Streaming Component, An Autosuggest and Auto-completion Search Suggestions Component, User Preference-based Recommendations, and User Analytics. The ‘Music Streaming Component’ develops a platform to play music over the internet for a compelling and stable experience. This component handles streaming of music and at the same time, it blocks unauthorized interceptions to the music stream. It keeps buffered data to provide a better experience to the user. Keeping cached data is useful in this component since the need for searching for content again is then eliminated. The ‘Autosuggest & Autocomplete suggestions Component’ focus on developing functionality to provide

efficient as-you-type suggestions to speed up human-computer interactions. Users will get suggestions for their search query in the areas that are preferred by them. The main advantage of this component is that the suggestions users get are tailored for them. Hence users would not consider moving to another application. The 'Recommendations Component' provides personalized recommendations to users. This is the second major research component of the project. This focuses on providing relevant recommendations to users. It involves research to identify and understand users' preferences. These preferences lead to the generation of results which keep the user intrigued and satisfied. Hybrid method was selected as the most suitable method. The 'User Analytics Component' analyses song streaming and user preferences. This generates analytics of the songs in the system. It also looks into providing a better view of what type of work the users expect to see from the artist.

3 Case Study on Similar Systems

3.1 Spotify

Spotify was first launched in October 2008 in Europe. Music can be browsed utilizing a search device by track name, artist, album etc. Users don't yet have the feature to explore and suggest users towards new music. Users may make a collection of most loved tracks committed to a specific topic, time, type, artist etc. Spotify gives search recommendations to users through auto-suggest and auto-complete alternatives [6].

3.2 Apple Music

Apple Music has become one of the biggest competitors to Spotify over the last year. Apple Music connects users with the artists they love [7]. Apple Music also automatically compares every track in a user's music collection to Apple's music library in order to understand what the user wants to hear. Apple Music also provides autosuggestions.

3.3 Pandora

Pandora is a USA based audio content streaming provider. Each user receives a personalized stream of songs. This is a complex labelling process with precisely defined methodology.

4 Chosen Development Methodology

The Software Development Life Cycle used for the project is the Feature Driven Development (FDD) approach. The 'HelaBeat' app includes several novel research components as well as a large number of development features. Thus, continuous development, integration and testing should be done. This approach was chosen over other methods as FDD mixes best practices that are all driven by importance. It is an iterative and incremental agile methodology with the objective of delivering software

quickly and efficiently. This project includes multiple features upon which the project would be built incrementally until its completion.

5 Components of The System

5.1 Music Streaming Component

This component includes both streaming and streaming analytics functionality. The main focus of this component is preserving the quality of the audio file as practically as possible within a data stream arriving over a network connection. Android APK has its own Media Player API which can be used to control the playback of audio files and streams. This is used to create a music player alongside the streaming application where the user can get a good streaming experience. The streaming behavior (Fig 2) is developed according to the best practices mentioned in the Android developer guide for developing a media application. Given the options of selecting between using Android inbuilt media player class or the third-party library Exoplayer [10][11], Exoplayer was selected to build the customized streaming component on top of the Android APK. The request for streaming a song will go through the ‘Endpoint 1’ shown in Fig 1. Thereafter, the streaming service in the backend opens a web socket to stream the particular song from Amazon Simple Storage Service (S3). Android media service will then respond to the data stream and will start playing the song.

5.2 Auto Complete Suggestion

Autocomplete suggestions will be provided in the system by combining users’ search preferences and search history [17]. Existing approaches to query auto-completion can be classified into two broad categories: heuristic models and learning-based models [18][19]. Heuristic approaches seek to compute a score directly by considering different sources for each possible query completion that indicates how likely it is that this query

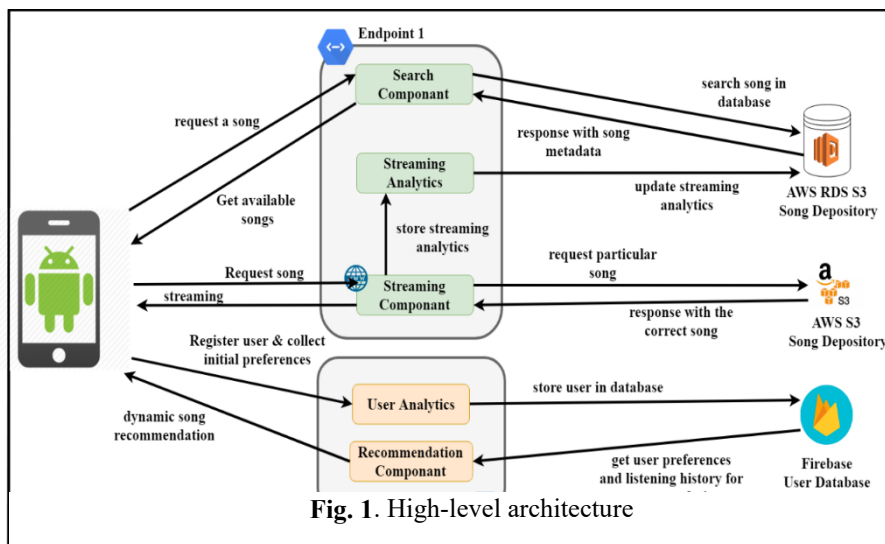


Fig. 1. High-level architecture

would be issued [20][21]. In contrast, learning-based approaches, are based on learning algorithms and aims to extract dozens of reasonable features to capture the characteristics of each query completion. This learns the behavior of each user and query. These two approaches can be split into two groups: time-sensitive and user-based [23]. Time may affect a query’s popularity, hence time-related aspects have been studied extensively for auto-completion. The user-centered approach is mainly focused on a user’s previous search history and a user’s profile information, such as age and gender, respectively [22].

5.3 Recommendation Engine

The system is developed in two versions: beta and alpha. The app is released to the Google Play Store and users’ listening behavior is analyzed and used to create and improve the recommendation system. Similar to the first iteration where user profiles were built using age, profession, and preferred music type, additional preference data will be collected from a sample community of users and the initial recommendation model is modified according to that data. When developing the recommendation engine, three main components are addressed: Dynamic playlist generation, Next track recommendation and Automatic playlist continuation.

Dynamic Playlist Generation. Users typically prefer certain categories of media items at different times & occasions. Recommending playlists for users is an efficient method to deal with efficiency [2][3][5]. By looking at the results of approaches that different researchers have taken and also by considering the limitations of the studied approaches [15][16], Neighborhood based methods and Matrix factorization techniques with content-based approach (metadata) were selected to recommend songs and to generate playlists for users. The way of development is using KNN (K-Nearest-Neighbors) approaches in machine learning with similarities between playlists most likely to provide sufficient accuracy in the recommendation [12][13].

Next Track Recommendation. Next-track music recommendation is a specific form of music recommendation that relies mainly on the user’s recently played tracks to create a list of tracks to be played next [12][9]. Session-based approaches are a good way to predict the user’s immediate next actions [4]. Computationally and conceptually simple methods often lead to predictions [8] that are similar and accurate better than

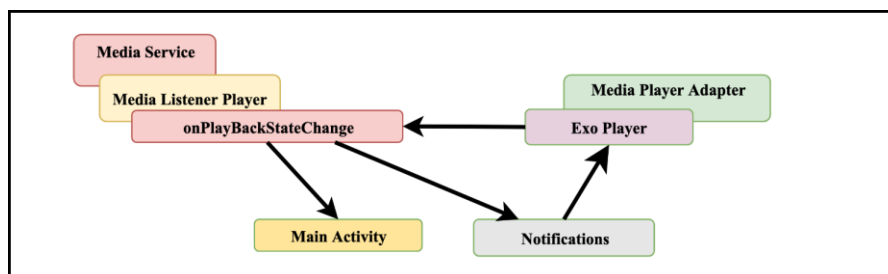


Fig. 2. Overview of the Android Streaming Component

deep learning-based models. Session-based KNN (K-Nearest-Neighbors) is selected to build the Next-track recommendation components.

Automatic Playlist Continuation. Automatic Playlist Continuation is a more challenging and the newest research area in the Music streaming industry. Automatic playlist continuation shares characteristics of session-based recommendation scenarios.

5.4 User Analytics

User analytics is the most important metric of the system. The analysis is done based on user profiles. Some personal data (name, age, gender, profession) and general user preferences about artists and music styles are collected at the signup for this purpose. Analytics will be used in future for song predictions (in search auto-suggestion) and dynamic playlist creations (recommendation). There after the users listening history is collected for further analysis. All user-related data (user profile data) is stored in a Firebase document database. This gives fast access and querying of user data. This data will provide a platform for artists to understand how their music is consumed and help in future productions.

6 FUTURE WORK

The app 'HelaBeat' Android application is currently available on Google Play Store. This application has the ability to expand into new horizons. One such form of novelty and capability which may be integrated to the application is the fingerprinting and identification of music items. This can provide the ability to generate recommendations based on music characteristics such as melody, harmony, rhythm, timbre and form.

7 CONCLUSION

The 'HelaBeat' Music Streaming application utilizes a novel approach in providing user preference-based recommendations and search suggestions. Users' data are continuously collected from explicit and implicit feedback methodologies [14]. This data is used to gain knowledge of habits, emotional state, general goals, age, profession of the users which leads to providing context-aware recommendations and suggestions to searches. Autosuggestions and recommendations are provided based on these data which makes them tailored to each user. 'HelaBeat' uses a hybrid approach of these methods which makes efficient suggestions and effective recommendations and allows artists to improve their future productions using user analytics.

8 References

1. A. Coffey, "The impact that music streaming services such as Spotify, Tidal and Apple Music have had on consumers, artists and the music industry itself .," 2016.
2. Nielsen Music, U.S. MUSIC 360: 2018 REPORT HIGHLIGHTS, 2018.
3. Bang, Sung-Woo & Jung, Hye-Wuk & Kim, Jaekwang & Lee, Jee-Hyong, "An Auto Playlist Generation System with One Seed Song", International Journal of Fuzzy Logic and Intelligent Systems, 2010.

4. Quadrana, Massimo & Cremonesi, Paolo & Jannach, Dietmar, Sequence-Aware Recommender Systems. *ACM Computing Surveys*. 51. 10.1145/3190616, 2018.
5. Bonnin, Geoffray & Jannach, Dietmar, "A comparison of playlist generation strategies for music recommendation and a new baseline scheme", 16-23, 2018.
6. K. Swanson and K. Swanson, "SPEA UNDERGRADUATE HONORS THESIS A Case Study on Spotify Exploring Perceptions", pp. 1–38, 2013.
7. A. Coffey, "The impact that music streaming services such as Spotify , Tidal and Apple Music have had on consumers, artists and the music industry itself", 2016.
8. Ludewig, Malte & Jannach, Dietmar, "Evaluation of Session-based Recommendation Algorithms. User Modeling and User-Adapted Interaction", 10.1007/s11257-018-9209-6, 2018.
9. Ludewig, Malte & Kamehkhosh, Iman & Landia, Nick & Jannach, Dietmar, "Effective Nearest-Neighbor Music Recommendations", 1-6. 10.1145/3267471.3267474, 2018.
10. Media app architecture overview: Android Developers. Retrieved from <https://developer.android.com/guide/topics/media-apps/media-apps-overview>
11. ExoPlayer: Google Developers. ExoPlayer. Retrieved from <https://exoplayer.dev/>
12. Iman Kamehkhosh, "Advances in Next-Track Music Recommendation", 2017.
13. Su, Xiaoyuan & M. Khoshgoftaar, Taghi, "A Survey of Collaborative Filtering Techniques. *Adv. Artificial Intelligence*", 2009.
14. J. O'Bryant, "A survey of music recommendation and possible improvements," no. April, 2017
15. Hariri, Negar & Mobasher, Bamshad & Burke, Robin, "Context-Aware Music Recommendation Based on Latent Topic Sequential Patterns", 10.1145/2365952.2365979.2014
16. Celma, Òscar, "Music Recommendation" 10.1007/978-3-642-13287-2_3.2010.
17. Milad Shokouhi and Kira Radinsky, "Time-sensitive query auto-completion" In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '12*, pages 601–610, New York, NY, USA, 2012. ACM.
18. Fei Cai, Shangsong Liang, and Maarten de Rijke, "Time-sensitive personalized query auto-completion." In *Proceedings of the 23rd ACM Conference on Information and Knowledge Management, CIKM '14*, pages 1599–1608, New York, NY, USA, 2014b. ACM.
19. Stewart Whiting and Joemon M. Jose, "Recent and robust query auto-completion", In *Proceedings of the 23rd International World Wide Web Conference, WWW '14*, pages 971–982, New York, NY, USA, 2014. ACM.
20. Steffen Bickel, Peter Haider, and Tobias Scheffer." Learning to complete sentences", In *Proceedings of the 16th European Conference on Machine Learning, ECML '05*, pages 497–504, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
21. Katja Hofmann, Bhaskar Mitra, Filip Radlinski, and Milad Shokouhi. "An eyetracking study of user interactions with query auto completion", In *Proceedings of the 23rd ACM Conference on Information and Knowledge Management, CIKM '14*, pages 549–558, New York, NY, USA, 2014. ACM.
22. Liangda Li, Hongbo Deng, Anlei Dong, Yi Chang, Hongyuan Zha, and Ricardo Baeza-Yates, "Analyzing user's sequential behavior in query auto-completion via Markov processes", In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15*, pages 123–132, New York, NY, USA, 2015. ACM.
23. Bhaskar Mitra, "Exploring session context using distributed representations of queries and reformulations". In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '15*, pages 3–12, New York, NY, USA, 2015.