EtnoFletno: Slovenian folk songs on the web and mobile devices

Sandi Gec, Tadej Mittoni, Ciril Bohak, Matija Marolt

University of Ljubljana, Faculty of Computer and Information Science
E-mail: {sandi.gec, matija.marolt}@fri.uni-lj.si

Abstract

In our paper we present a mobile and web based platform for presentation of Slovenian folk songs. Our focus was to develop a fast, responsive and functional application available on all major mobile platforms, as well as through web browsers. The application is build on a three-tier architecture and it is developed in scalable way. It offers standard browsing and search features over a collection of Slovenian folk songs. In addition, several more advanced search options are implemented, most notably efficient melody-based search and a query by humming algorithm.

1 Introduction

Folk songs are an inherent part of a nation's history and identity, however in modern times, they are becoming a lost and forgotten item for the younger generations, overexposed to popular trends. The aim of the EtnoFletno project, presented in this paper is to awaken the connection and interest of individuals for folk songs and music. It aims to become a resource reflecting the cultural and historical identity of the nation, old customs and traditions and to revive the past ways of folk singing and playing. As nowadays the majority of people, irrespective of their age, own a mobile phone and/or a computer we aim to achieve these goals by developing an application available on the web as well as the three major mobile platforms: Android, iOS and Windows Mobile. We are developing a user-friendly application that presents part of the cultural legacy gathered by the Institute of Ethnomusicology, the oldest institute of the Scientific Research Centre of the Slovene Academy of Sciences and Arts.

Music is ubiquitous in today's culture, and in the last years, its distribution in the form of physical media is being superseded by Internet streaming. For classical and popular music, several applications are offering millions of music items, amongst the more popular are iTunes [6], Spotify [5], Deezer [7] and SoundCloud [3]. However, for folk music, the situation is not so rosy. There are several web sites presenting folk music, such as Ethnomuse [14], Hymnary [9], Folktunfinder [10] and Themefinder [11], however they are only available on the web and their usability is poor when compared to commercial popular music sites and they are focused more on the needs of ethnomusicologists and folklorists.

Our main goal it thus to develop a user-friendly mobile and web application. It should present folk songs in a modern way, incorporating different search engines to improve user experience, encourage singing and in addition present historical and geographical information interactively. It is intended to be used by a wide array of users and should also be suitable for use in an educational context as an interactive learning tool.

The main features of our solution are similar to the offerings current commercial music sites. We used the latest technologies for its development, such as the indexed database Elasticsearch to obtain high search performance, and multi level architecture for better security and scalability. In addition, alternative search algorithms are provided. Melody-based search, also implemented in other folk song applications, such as the Ethnomuse, Hymnary, Folktunfinder and Themefinder, is based on the algorithm presented in [12]. The algorithm uses efficient hashing of symbolic n-grams to enable fast approximate queries based on melody, as well as rhythm. We have also implemented a geolocation based search, which uses the user's current location as a query and map-based visualization of results, such as in popular sites as the TripAdvisor [13]. Finally, we also offer Query-by-Humming searching, where we used the Probabilistic YIN algorithm [8] to extract pitches from an audio signal and a flexible approximate search for finding the best matching melodies [22].

2 System overview

Our solution is based on a three-tier architecture: (i) presentation tier, (ii) logic tier and (iii) data tier. It is available on the three major mobile platforms (Android, iOS and Windows Phone), as well as the web, thus covering a mobile market share up to 99% [2]. An overview of the entire solution is shown in Figure 1.

To develop the presentation tier, we use PhoneGap, which is a cross-platform development environment from Adobe that uses the same source code to generate hybrid mobile applications for multiple platforms [15]. It is based on the open source Apache Cordova framework [16]. In principle this framework is used for building native mobile applications using noted technologies like HTML5, CSS and JavaScript. Those technologies are also the basis of web development, so it is possible to al-

most simultaneously develop for mobile and web reusing most of the source code. In our case, our source code is up to 95% same for all platforms. Most differences lie in the processing of events (mouse and touch), libraries and graphical user interface (GUI) specifics.

The logic tier is build around two different servers, that we developed for different purposes. A NodeJs Express server is used to manage Asynchronous JavaScript and XML (AJAX) requests from the client [17, 18]. A Hypertext Transfer Protocol (HTTP) server written in C# programic language also manages AJAX requests but its main purpose is that is runs that the querying algorithms (melody search and Query by Humming). It also performs user management and access to WebAAI [20] that allows users to login using accounts from federal organisations.

The central point of the third tier is the high performance Elasticsearch server database [19]. It is a full-text search engine, which we use to index the textual metadata in our database, enabling fast and simple queries. The database structure originates from the Ethnomuse project, but it is adapted for the project's specifics.

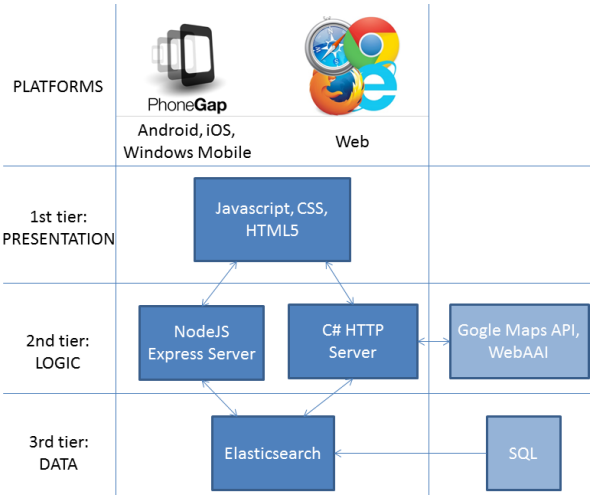

Figure 1: Overview of the solution architecture. The first tier is implemented using Javascript, CSS, HTML5 and other libraries like Bootstrap, Bootbox, internationalization (i18next), Waypoints and others. The second tier consist of two servers (Javascript and C# based) and web services calls (Google Maps API and WebAAI). In the third tier we use the Elasticsearch database linked to the database from the Ethnomuse project.

3 Functionalities

Functionalities of the application enable exploration and search over a collection of folk songs in the Ethnomuse database, presentation of song metadata and scores, and playing of folk song recordings. Search can be text-based, melody-based, location-based or singing-based, more details are provided in the next section. Songs can be shared through social and other communication networks, they can be added to personal favorites and grouped into public playlists. Web users can also add their own materials and comments to the database. The creation of playlists and addition of new material enables uses of the application for educational and research purposes. The mobile user interface is shown in Figure 2.

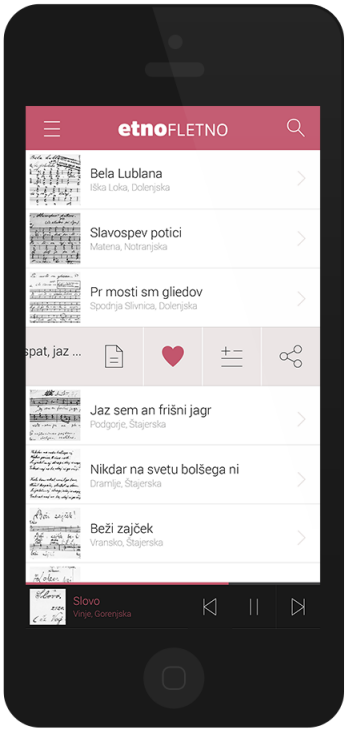

Figure 2: Mobile user interface showing a list of songs with actions for one: detailed info, add to favourites, add to playlist and share song.

4 Searching

Since the nature of our data is multi-modal, we also wanted to allow for different search modalities. These modalities improve user experience and increase the possibility of finding relevant songs. Users can search songs without knowing their title, based on other textual attributes, geolocation, or melody. Visual interfaces of the main search interfaces are shown in Figure 3.

4.1 Text search

We use the ElasticSearch engine to index all textual attributes of songs (e.g. title, year, region, performer names, description) and thus enable efficient querying by any attribute. Queries take advantages of Elasticsearch's full-text indexing engine, retrieving relevant documents with $O(n \cdot logn)$ time complexity. So even searching through a large amount of data keeps responsiveness high.

4.2 Melody search

Melody search enables melody-based querying. The user can specify a given melody by using a virtual keyboard or entering a textual note query, and the querying engine efficiently searches a database of annotated melody with an approximate search algorithm described in [12]

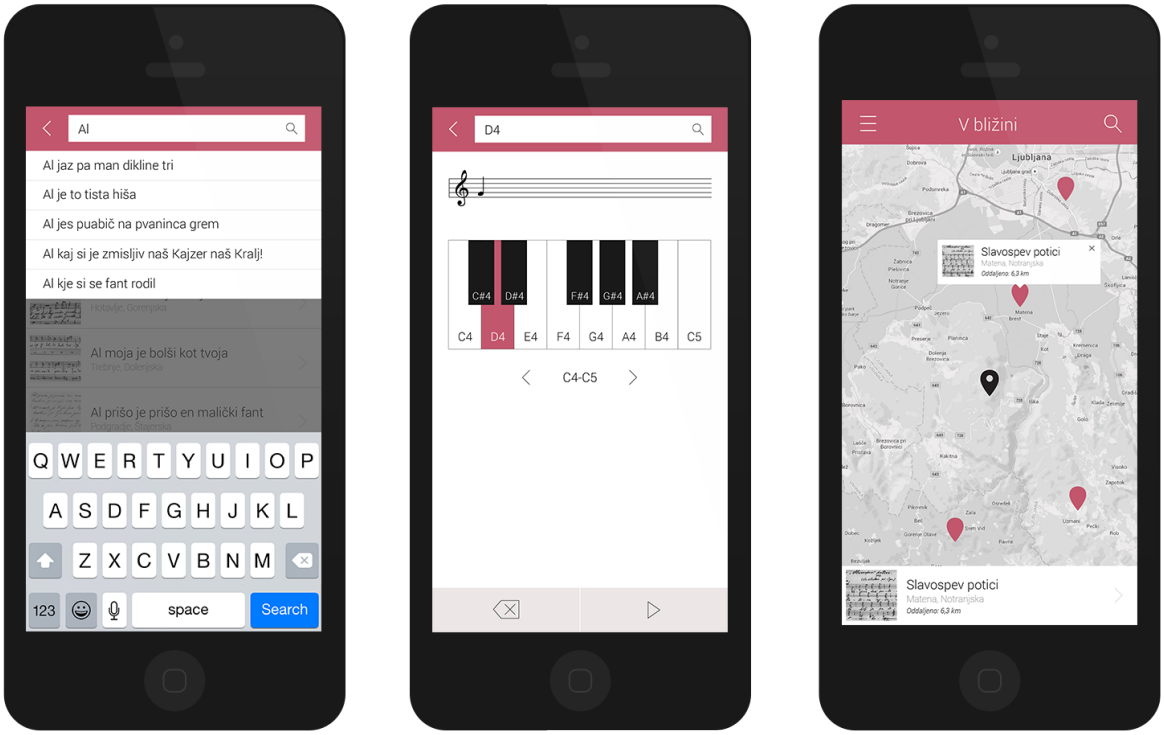

Figure 3: Visual interface examples of three search engines: text search (left), melody search (middle) and geolocation search (right).

and implemented by [1]. The algorithm is suitable for monophonic melodies or for those with separate voices and enables melody-only, as well as melody and rhythm based exact or approximate and optionally transposition-invariant searches. It is based on a melodic signature index that is built from algebraic signatures of single melodic n-grams. This signatures fuse values of distances between tonal heights on each n-gram into a single value. Similarity search is performed by comparing melodic n-grams. The higher the difference, the less similar melodies are.

4.3 Geolocation-based search

Search based on geolocation enables for searching of songs based on user's current location. User's current location is obtained from their current GPS location or IP based network location. It is also possible to set the search location manually. Search is performed by calculating the distance of the query point to locations of songs in the database, which represent the places songs were written down by ethnomusicologists. This type of search is made responsive by the use of Elasticsearch and asynchronous catching of data (calculated distances). Users can pick the maximum radius of search results between 0 and 100 kilometers. Results are shown in two ways: as a list of songs sorted by their distance to the query point, and on a map. For the latter the Google Maps API is used, as it is familiar to users, offers detailed maps and allows for interactivity with resulting markers.

4.4 Query By Humming

Query by Humming enables users to sing or hum a melody into their phones and matches the recorded sound with melodies in the database. Our solution is composed of two distinct algorithms. In the first stage, the sung melody is transformed into a sequence of notes by the probabilistic Yin algorithm [8]. The algorithm is an adaptation of the well-known Yin algorithm for fundamental frequency estimation, which is based on auto-correlation. Instead of estimating a single fundamental frequency per frame, it keeps several alternatives, and then uses a hidden Markov model to smooth the resulting pitch track. The reasoning behind using a HMM is its ability to favour smooth pitch tracks over discontinuous ones as well as favouring few changes between unvoiced and voiced states. The final step of the algorithm is to extract note features, which yield a segmentation of the continuous pitch track into a sequence of notes with estimated frequency, start time and duration.

To match the queried sequence of notes to songs in the database, a subsequence melody matching framework is used that allows for gaps in both the query and target melodies, with variable matching tolerance levels efficiently tuned for each query and target sequence, and allows for constrains on the maximum match length [22]. The framework is based on a space and time efficient dynamic programming method that given a short melodic sequence and a large database, efficiently identifies the subsequence of the database that best matches the query

and bounds the number of consecutive gaps in both sequences.

5 Conclusion

In this paper we presented our work within the project EtnoFletno where the aim is to bring the almost forgotten folk culture to the public by means of a modern mobile and web application. We described the system architecture, application functionalities and the various search mechanisms implemented.

In our further work, we plan to make an evaluation of the application with target users, which consist of folklore groups, educational institutions, researchers and broader public and based on the evaluation improve the overall user experience. After launching the EtnoFletno platform, we will make extensive efforts to spread the word over different channels and in cooperation with the Institute of Ethnomusicology, hoping to reach a wide audience interested in this part of our culture.

6 Acknowledgements

The work presented was made possible by the project EtnoFletno: Slovenian folk song and music on mobile devices, a Structural Funds Project founded by Ministry of Education, Science and Sport, Republic of Slovenia.

References

[1] E. Ivnik, "Indeksiranje in iskanje po simboličnih glasbenih zbirkah", Faculty of computer and information science, Ljubljana, Slovenia. 2014.

[2] Smartphone OS Market Share. (2015). Avaiable: http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[3] SoundCloud. (2015). Avaiable: http://www.soundcloud.com

[4] Shazam. (2015). Avaiable: http://www.shazam.com

[5] Spotify. (2015). Avaiable: http://www.spotify.com

[6] iTunes. (2015). Avaiable: http://www.itunes.com

[7] Deezer Music. (2015). Avaiable: www.deezer.com

[8] M. Mauch and S. Dixon, "pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions", in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014), 2014.

[9] Hymnary. (2015). Avaiable: http://www.hymnary.org/

[10] FolkTuneFinder. (2015). Avaiable: http://www.folktunefinder.com/

[11] Themefinder. (2015). Avaiable: http://www.themefinder.org/

[12] Camelia Constantin, Cédric du Mouza, Zoé Faget, Philippe Rigaux, "The melodic signature index for fast content-based retrieval of symbolic scores", The 12th International Society for Music Information Retrieval Conference, Miami, USA, October 2011, pp. 363-368.

[13] TripAdvisor LLC. (2015). TripAdvisor. Available: http://www.tripadvisor.com/

[14] M. Marolt et al., "Ethnomuse: Archiving folk music and dance culture," in Proc. Eurocon 2009, St. Petersburg, Russia, 2009.

[15] PhoneGap. (2015) Available: http://phonegap.com/

[16] Apache Cordova. (2015) Available: https://cordova.apache.org/

[17] NodeJS. (2015) Available: https://nodejs.org/

[18] ExpressJS. (2015) Available: http://expressjs.com/

[19] Elasticsearch. (2015) Available: https://www.elastic.co/products/elasticsearch

[20] ArnesAAI. (2015) Available: https://aai.arnes.si/

[21] A. Anžič, "Spletna aplikacija za pregled zbirke slovenske ljudske glasbe", Faculty of computer and information science, Ljubljana, Slovenia. 2014.

[22] A. Kotsifakos, P. Papapetrou, J. Hollmén, D. Gunopulos, V. Athitsos and G. Kollios, "Hum-a-song: A Subsequence Matching with Gaps-Range-Tolerances Query-By-Humming System", Proceedings of the VLDB, Volume 5 Issue 12, August 2012.