CONTEXTPROJECT

Emergent Architecture Design, TI2806

Authors:
S.A. BOODT, sboodt, 4322258
T. HEINSOHN HUALA, theinsohnhuala, 4326318
A. HOVANESYAN, ahovaneyan, 4322711
D. SCHIPPER, dschipper, 4155270
J.R. WEIJGERTSE, jweijgertse, 4247124

Thursday 7th May, 2015

Contents

1

Introduction

The report consists of the following components. Firstly we will discuss the design goals of our project, next in chapter 2, the software architecture will be described from different views. Finally a glossary is defined in chapter 3.

1.1 Design goals

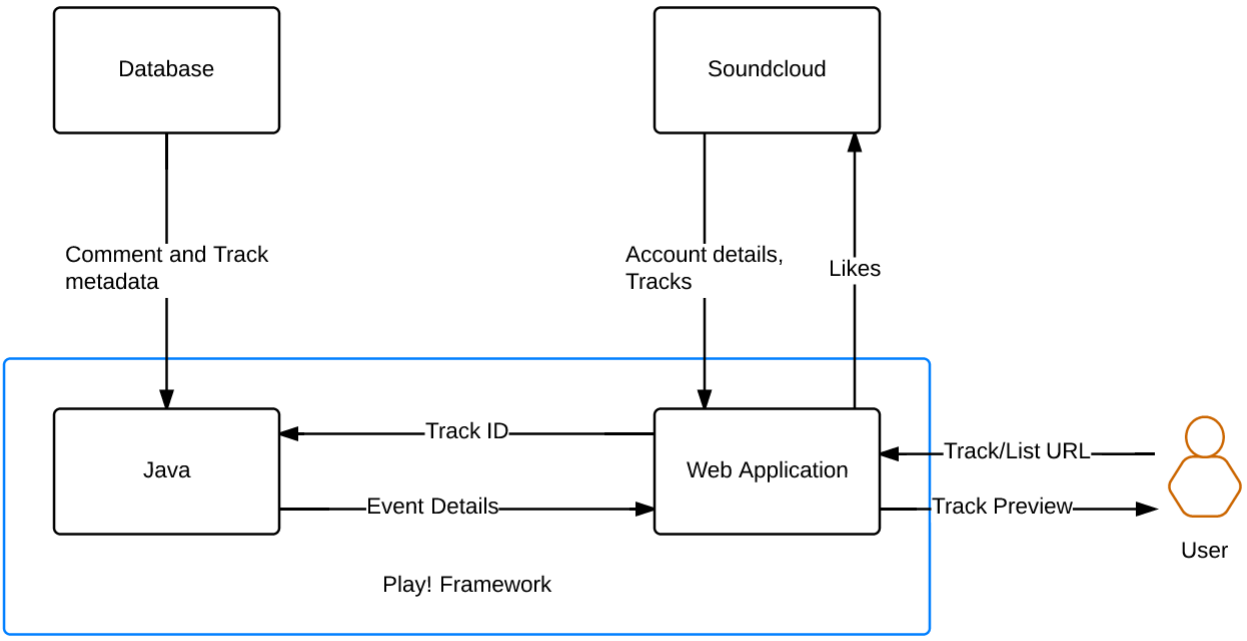
Besides the goals decsribed in the Product Vision document, we also have goals concerning the architecture of the product. Firstly we want to produce a working version of our product every week with code quality up to the normal standards. The client can in this way view the progress of our product every week and critize features if it is not to his likings. The code quality will be discussed in chapter 2.5
The product consist of two parts, the webpage where an user can play a snippet and the underlying component which decides what the snippet should be. How these components interact with eachother is described in chapter 2.4.

- Webplayer
  The webplayer will be the interface of our product.
- Back end system This subsystem will receive a song and return a snippet.

2

Software architecture views

2.1 Subsystem decomposition



- Web application: The web application holds the soundcloud's widget and buttons to control it. Through the interface the user can pick a song and play a preview/snippet. If the user is logged in he can then add the song to his collections if he likes the song. Being logged in gives the user the opportunity to see his collection.

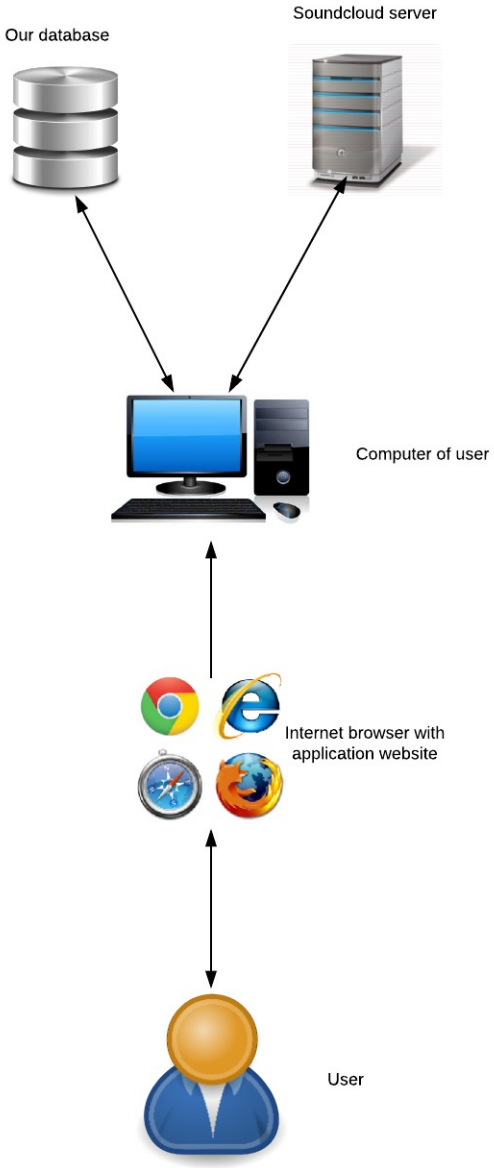
- Java: The java part of the program processes the soundcloud's track and comment meta-data to calculate the song's preview start-time with a given window. To be able to do this it receives a track id from the user interface which is in this case the web application. After processing the data a responsible java object outputs a start-time and a window for the given track.

- Database: The database contains all the meta-data of the tracks including the comment meta-data. When the application receives a track ID it searches the database for the corresponding comments and other details.

- Play! Framework: We use Play! to be able to run the java and web applications alongside each other. It's biggest responsibility is to make sure the data, such as track information and event details(start and end times), is exchanged.

2.2 Hardware/software mapping

The hardware/software mapping contains these parts: At first we have the user, the user will be able to use the application through a internet browser. Then the internet browser makes use of a connection to the soundcloud servers for the accounts and the internet browser makes use of our own database with some metadata on it.



2.3 Persistent data management

2.4 Concurrency

3

Glossary

word description