Project report
Web Engineering 2019
Group 14

E. Waterink (s3417611), T. de Vries (s2367610), I. Oralin (s4171446)

October 2019
v1.0

Introduction

This document contains information regarding the architectural choices of the project.

Note that currently, this document is a work-in-progress as the technology we use evolves to fit the requirements of the project.

Technology Stack

It was decided to use programming language Python as two members of the team have experience with it and the third member can fast and easily learn it due to its simplicity.

As a framework it was decided to use Flask as it is easy to use and it is suitable for a small project like this one.

As a database management system we decided to use MySQL. The reason we chose this over a no-SQL database is due to the (relatively) small amount of data we have to deal with (it is in the thousands). In other words, we are able to get very fast response query responses.

Database Design

The CSV file with songs have many duplicates of information about the same artist for all songs he made. So it was decided to split data into three tables(relations): Artist, Song, Release (the last one does not contain any

useful data for current API but it was left just in case). Song contains foreign key for Artist (id of the artist) and for Release.

Architecture of the project

The Flask application plays the role of a server which receives HTTP requests from the API user. After request is received server forms a SQL query and sends it to the MySQL DBMS which, in turn, responses with needed data. Thereafter server encodes received data in JSON or CSV format and sends it to the user as a response.