

Listen Up

Research Document

Mohammad Nazibul Kabir Khan

4263308

Contents

Introduction	3
Research Questions	4
Problem: Data to be stored.....	4
Major Question: What data should the software store?	4
Minor Questions	4
1: What user data should be stored?	4
2: What details of main component need to be stored?	4
Problem: Quality of the code	5
Major Question: How to determine quality of code?.....	5
Minor Questions	5
1: What is JaCoCo?.....	5
1: What is SonarQube?.....	5
References	6

Introduction

This document is about the research done to get answers of list of questions. These questions are related to the understanding new things and getting new ideas to be used in ListenUp project.

Research Questions

Problem: Data to be stored

Major Question: What data should the software store?

First thing to worry about when creating a software solution is what data it should store. At the beginning of the creation of website, questions like these arises: *'What data should be stored for making a secure environment?'*, *'What data is relevant to the website?'* and more. After doing **competitive analysis** on Spotify[1], some information about the data collection is gathered. For user personal information, ListenUp website will store their email and username. And for good security, encoded version of their password will also be stored. User will be able to login with their email and password.

Other important data should be collected to make website function relevantly. ListenUp is a website about songs. People can make playlist and add songs in their playlists. Again, **competitive analysis** is done on Spotify and **user requirements are explored** to get what data should be stored. ListenUp is storing all song info and the info of playlist that were made.

Exploring user requirements and going through **data analytics** relevancy of data stored is validated.

Minor Questions

1: What user data should be stored?

Methods:

- Competitive analysis
- Explore user requirements
- Security test

When users are told to make an account at the startup of ListenUp website. User are told to put in their username, email, and password. Email and encoded version of the password is stored to make user's environment secure and restricts access to act on user's playlists and liked songs. Conclusion is made after doing **competitive analysis** to any other websites with login requirements. After **security test**, encoded password is stored in database rather than the raw password to protect against hackers.

2: What details of main component need to be stored?

Methods:

- Competitive analysis
- Explore user requirements

Again, after **competitive analysis** on Spotify, right now ListenUp storing all main info of song, artist, and album. Song containing info like name, its artist, released date, genre, and popularity. Artist also has their name and list of songs they have. ListenUp also storing user data like which song they listened the most, list of playlists they have and songs in them.

By **user requirements**, it has been discovered that user want to view their data e.g., playlists and liked songs along with the public list of songs. They can also keep up with other people's playlists and activity if those activity and playlists are set public by those owners.

Problem: Quality of the code

Major Question: How to determine quality of code?

After doing **literature study**, this web application must worry about bugs, errors, syntax, security risk and many other. [2] Research shows good code must have these five traits: Reliability, Maintainability, Testability, Portability and Reusability:

Reliability measures how reliable the software solution for the application is. It checks if the application crashes or have any defects.

Maintainability checks if the software is easy to maintain. This trait also depends on two other traits: testability and understandability.

Testability determines if the logic of the app is what developer wants. There are currently over 150-unit tests.

Portability checks if it is used in different platform. In future, docker will be used in ListenUp.

Reusability checks if code can be used repeatedly on different software solution.

This app is currently using Jacoco plugins. Jacoco gives report on the percentage of the logic covered by unit tests. It also shows if all possible results of if-else statements are covered. Functionalities of Jacoco is limited; it only shows the percentage of coverage not other important parts which leads to good code quality.

Due to Jacoco's limitations, ListenUp is also have SonarQube plugin installed in it. SonarQube have other features which shows if the code has any bugs or how the code can be written in a better way. Moreover, it also reminds developers if code leads to major security risks.

Minor Questions

1: What is JaCoCo?

Method: Literature Study

JaCoCo stands for Java code coverage. This tool main functionality is to find out which part of the code is tested, and which part needs testing and it helps to run or merge[3] multiple different codes. It was developed to replace EMMA[4] for Eclipse. This tool has two different approaches when checking code for coverage. First, like JCov[5] while running the code another being like Cobertura which is before the execution. There are other tools which are using or including JaCoCo as their test coverage tool. Among them there is SonarQube and IntelliJ IDEA[6]. Both SonarQube and IntelliJ IDEA is used to develop ListenUp application.

1: What is SonarQube?

Method: Literature study

SonarQube previously known as Sonar[7] is a software system which supplies with automated testing of bugs, code smell and quality of codes. With SonarQube static analysis ListenUp have one place to measure the Reliability, Security, and Maintainability.

References

1. ["About Spotify"](#). Spotify. Retrieved 10 January 2022
2. [Steve McConnell's Code Complete](#)
3. [Measure Coverage by Integration Tests with Sonar – Updated](#)
4. [JaCoCo Mission](#)
5. Patroklos Papapetrou (19 December 2012) ["Code Coverage Tools \(JaCoCo, Cobertura, Emma\) Comparison in Sonar"](#). Only Software matters. Retrieved 3 March 2013
6. ["Code Coverage"](#). IntelliJ IDEA 12.0 Web Help. JetBrains. Archived from [the original](#) on 26 April 2013. Retrieved 3 March 2013.
7. Freddy Mallet (20 March 2013). ["SONAR is becoming SONARQUBE"](#). SonarQube project mailing list. Archived from the original on 24 July 2013. Retrieved 3 July 2013.