NoaNodz

Software Design Document (SDD)

Version 2.0

Document Preparation

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Approval (Signature)** | **Approval Date** |
| Amado Lazo | Technical Manager, Developer | Amado Lazo | 11/02/2022 |
| Kevin White | Project Manager, Developer | Kevin White | 11/02/2022 |
|  |  |  |  |
|  |  |  |  |

Document Approvals

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Role** | **Approval (Signature)** | **Approval Date** |
| Kathryn Compton | Documenter, Designer, Tester | Kathryn Compton | 11/03/22 |
| Elizabeth Rubio | Documenter, Designer, Tester | Elizabeth Rubio | 11/03/22 |
| Orlando Gomez | Developer, Designer | Orlando Gomez | 11/03/22 |
| Jennifer Vasquez | Developer, Designer, Tester | Jennifer Vasquez | 11/03/22 |

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 09/21/2022 | 1.0 | Original | Team Toronto Keyboard Maple Leaf Warriors |
| 11/02/2022 | 2.0 | A new format with additional information regarding ideas and designs. | Team Toronto Keyboard Maple Leaf Warriors |

Table of Contents

1. Introduction 5

1.1 Purpose of the Document 5

1.2 Scope of the Document 5

1.3 References 5

1.4 Definitions, Acronyms, and Abbreviations 5

2. Product Scope 5

3. System Design Decisions and Constraints 5

3.1 System Design Decisions 5

3.2 System Design Constraints 5

4. Software Architectural Design 6

5. Software Design - Modules/Classes 6

5.1 Module 1 6

5.2 Module 2 6

5.3 Module 3 6

6. User Interface Design 6

7. Appendix A – 1.4 Definitions, Acronyms, and Abbreviations 7

8. Appendix B – Class Diagram of the Software System 8

9. Appendix C – User Interfaces 9

# Introduction

This software design documents goal is to try and organize our teams thinking so we can get a solid design idea and them down for our website

## Purpose of the Document

The purpose of this document is to document the software design for the project NoaNodz.

## Scope of the Document

This document will represent and be the go-to source for our software engineers when designing and building pages on the website

## References

N/A

## Definitions, Acronyms, and Abbreviations

See Appendix A.

# Product Scope

Within this project's scope, we are trying to be the hub for all music, by creating embedded versions of all other music websites within ours. This is so others can share our website to share and listen to music with their friends. This website is JUST for music, not podcasts or audiobooks.

# System Design Decisions and Constraints

Some users may be at work or home listening to music on their computer and want to share music in a chat with friends or co-workers.

## System Design Decisions

[Present a list of all design decisions.]

|  |  |  |  |
| --- | --- | --- | --- |
| **System Design Decision ID** | **Date** | **Design Decision** | **Reason** |
| DD-1 | 10/8/2022 | Breakdown into 4 main pages. | When searching for music you usually only have 4 criteria, you want: what’s popular, a song, an album, or an artist. |
| DD-2 | 10/17/2022 | 4 streaming site choices. | These where the most popular streaming sites that allowed easy embedding. |
| DD-3 | 11/2/2022 | Use Local Storage to remember users preferred website for ease of use. | This will make it an easier user experience only needing the user to |
| DD-4 | 11/2/2022 | Use a JavaScript custom template in order to show all possible songs instead of adding a few manually | This would be easier for us in the long run, as well as let users use website properly |
| DD-5 | 11/2/20221 | Allow song sharing. | This would allow the user to share songs amongst other users. |

## System Design Constraints

[Present a list of all design constraints.]

|  |  |  |  |
| --- | --- | --- | --- |
| **System Design Constraint ID** | **Date** | **Constraint** | **Reason** |
| DC-1 | 10/30/2022 | To many HTML files in a single folder. | Due to the amount of songs, we implemented, we had to some how keep track of all the songs so the user can click on the song and it will take them to its’ page. |
| DC-2 | 11/2/2022 | Sharing page URL amongst users. | Due to the way we populate the pages currently, it is difficult to implement the URL sharing method. |
| DC-3 | 11/2/2022 | Mobile Responsiveness. | The CSS styling currently works with only Computer screens. |

# Software Architectural Design

[You do not have to fill this section for the course project. You may just write text below and move on to the next section.]

The software architecture of the [project/system name] is described in this section.

# Software Design - Modules/Classes

[This section describes the software modules in the system. The overall software design is presented in Appendix B. Provide a list of modules in the system.]

|  |  |  |
| --- | --- | --- |
| **Module/Class Name** | **Module Short Name (If there is one)** | **Short Description of the Module/Class** |
| Module 1 | Song Template Class | This JavaScript file populates the missing template sections of a song HTML file based on a pre-determined parameter. |
| Module 2 | Drop Down Listen Buttons | When the user clicks on the buttons in the song HTML file, the buttons then show Embeds when clicked on corresponding to the song which is produced by the song template class. |
| Module 3 | LocalStorage System | Gets LocalStorage from user and automatically opens a drop-down menu based on users preferences or does not open any if user does not have preference. |

## Module 1

[Describe Module 1 here.]

In this Module, the JavaScript populates the missing sections of the *song\_template.html* which is pre-determined by parameters that are passed in this file. This creates a responsive html page which the user can view the right song that they have chosen.

## Module 2

[Describe Module 1 here.]

For the 2nd Module there are 4 drop-down listen buttons for each of our streaming sites. These buttons allow the users to pick a preferred streaming site. When these buttons are clicked it will open and load a embedded audio player from the preferred streaming site. This song is based on the song provided by the templet class. Meaning the audio imbed is different based on the song chosen by the user.

## Module 3

[Describe Module 1 here.]

For this module, when the user, uses the drop-down listen module, it will embed information to their LocalStorage, this information will then become their preferred listening platform of choice. When the user interacts with module 1 to produce a song on the template page module 2 will populate along with module 1 to automatically open their preferred streaming platform of choice.

# User Interface Design

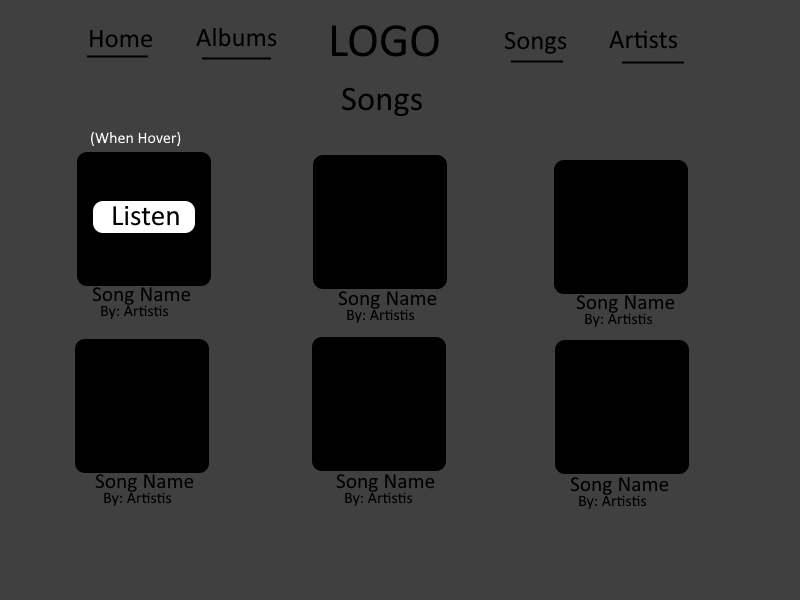
[This section describes the user interface design of the system. Explain the reasoning behind the design of the user interfaces. The sketches/prototypes of user interfaces are presented in Appendix C.]

This is our Homepage when the user first enters the website displaying both Song and Album of the week.

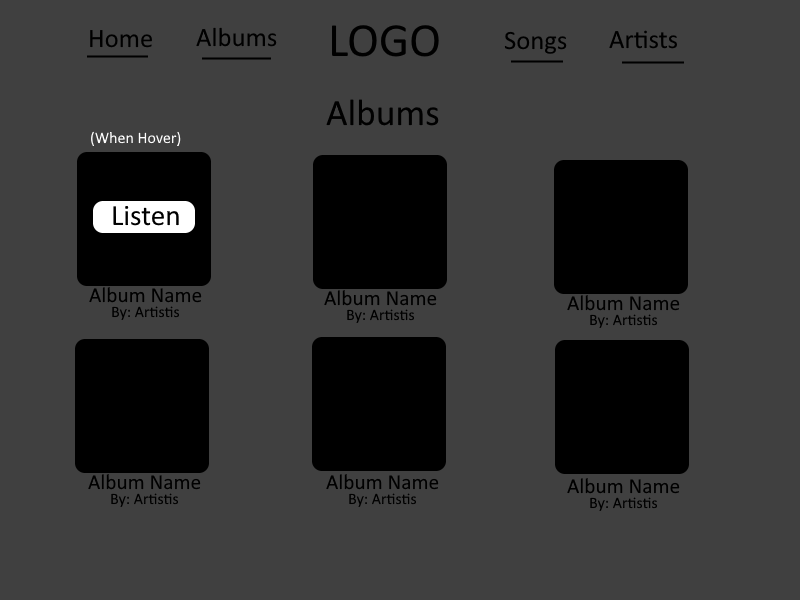
Graphical user interface, application

Description automatically generated

This page is the Song page which is populated with different songs provided by us. To get to this page, it is easily accessible through the Navigation bar of the Home page.



This page is the Album page which is populated with different songs provided by us. To get to this page, it is easily accessible through the Navigation bar of the Home page.



This page is the Artist page which is populated with different songs provided by us. To get to this page, it is easily accessible through the Navigation bar of the Home page.

Table

Description automatically generated

This page is the Song Template page is populated with songs provided by us and is generated through JavaScript. To get to this page, the user must click on the Listen button in the songs page or the song on the artist page.

Graphical user interface, application

Description automatically generated

This page is the Album Template page is populated with an album provided by us and is generated through JavaScript. To get to this page, the user must click on the Listen button in the album page or the album on the artist page.

Graphical user interface, application

Description automatically generated

This page is the Artist Template page is populated with an artist provided by us and is generated through JavaScript. To get to this page, the user must click on the artist on the song/album pages or on the artist page.

Graphical user interface, application

Description automatically generated

# Appendix A – 1.4 Definitions, Acronyms, and Abbreviations

|  |  |  |
| --- | --- | --- |
| **Term** | **Abbreviation / Acronym** | **Definition** |
| HyperText Markup Language | HTML | is the standard markup language for documents designed to be displayed in a web browser. |
| Cascading Styling Sheet | CSS | is a style sheet language used for describing the presentation of a document written in a markup language. |

# Appendix B – Class Diagram of the Software System

[Present the class diagram here. If it is not applicable, write Not Applicable. For some web application, we may not have a class diagram.]

Not Applicable

# Appendix C – User Interfaces

[Present the user interfaces of the system. They may be sketches and even design in powerpoints.]

