# Project Management Part 3
# Project Initial Concept and Design

**Group Name: MCS08**
**Group Member:**
**Yeoh Ming Wei (32205449)**
**Toh Xi Heng (33200548)**
**Yew Yee Perng (32205481)**

# Table of Contents

# Introduction

A short introduction for our team, our group is named MCS08 that consists of 3 members, Yeoh Ming Wei, Yew Yee Perng and Toh Xi Heng. In our previous discussion, we had chosen the project that we are interested in and discussed our project management plan. The project topic that we chose was called “Singing Video Generation with Music Separation” supervised by Dr. Arghya Pal. The main focus of this topic is to create a decomposed music file with a separation of background music and human voice and later generate a music video with a virtual avatar based on the provided music files.

This project proposal aims to provide a breakdown structure of multiple sections in order to complete our final product to the client. A deeper analysis of the topic will allow us to have a better understanding of the requirements that we need in order to successfully produce a working product. Besides that, it also benefits us by having a proper plan on how to start our project as well.

At the beginning of our proposal, we will discuss our project goals and the expected deliverables of our project to the client. This starting point will give a short brief on what is our final product and what deliverables are we providing at the end of our project. Next we dive deeper into representation using diagrams to show our initial software design. In addition, we will provide explanations as well to explain more detail on the representations. To ensure that the software runs smoothly, we also need to list out the software and hardware specifications in need to produce the product for our current project. These will help us in many scenarios such as coding, the efficiency of our software, software that requires higher specifications and many more. Lastly, we will justify our last choice to provide a decision for our project.

Without further ado, we start our discussion with our project goals and expected deliverables of our project.

# Project Goals and Expected Deliverables

The goal of this project is to develop a system that can generate a high-quality human singing face video synchronised with a provided song. This will involve decomposing the input music into human voice and background music components and then generating a human singing face video with accurate lip synchronisation to the vocals. This project aims to deliver a singing video generator with music separation with accurate results given the input of a music audio.

# Representation

## Flowchart

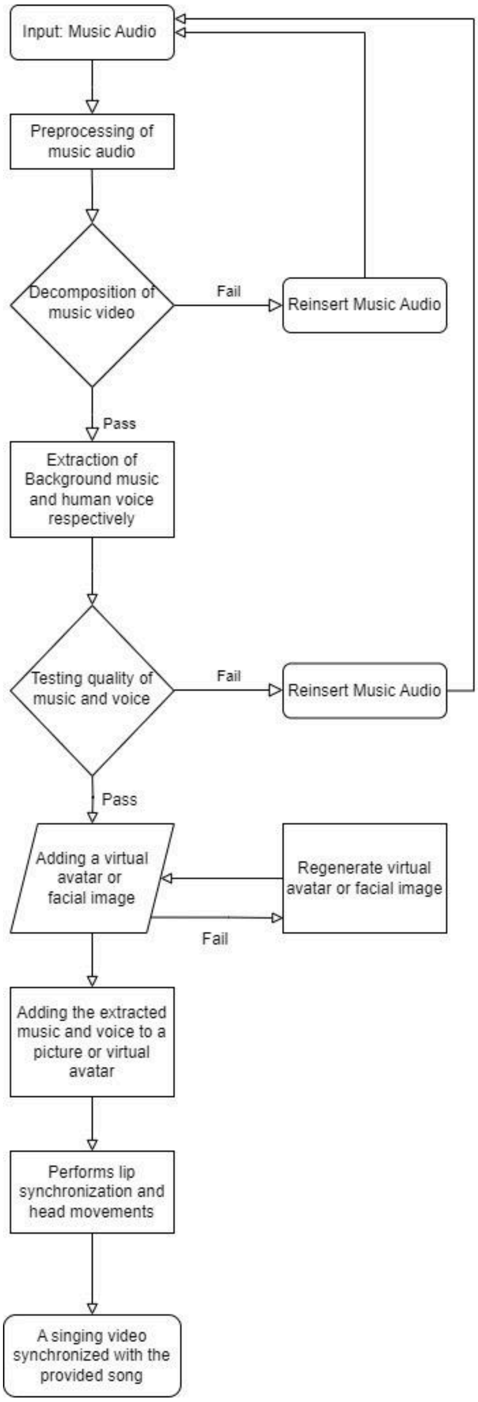
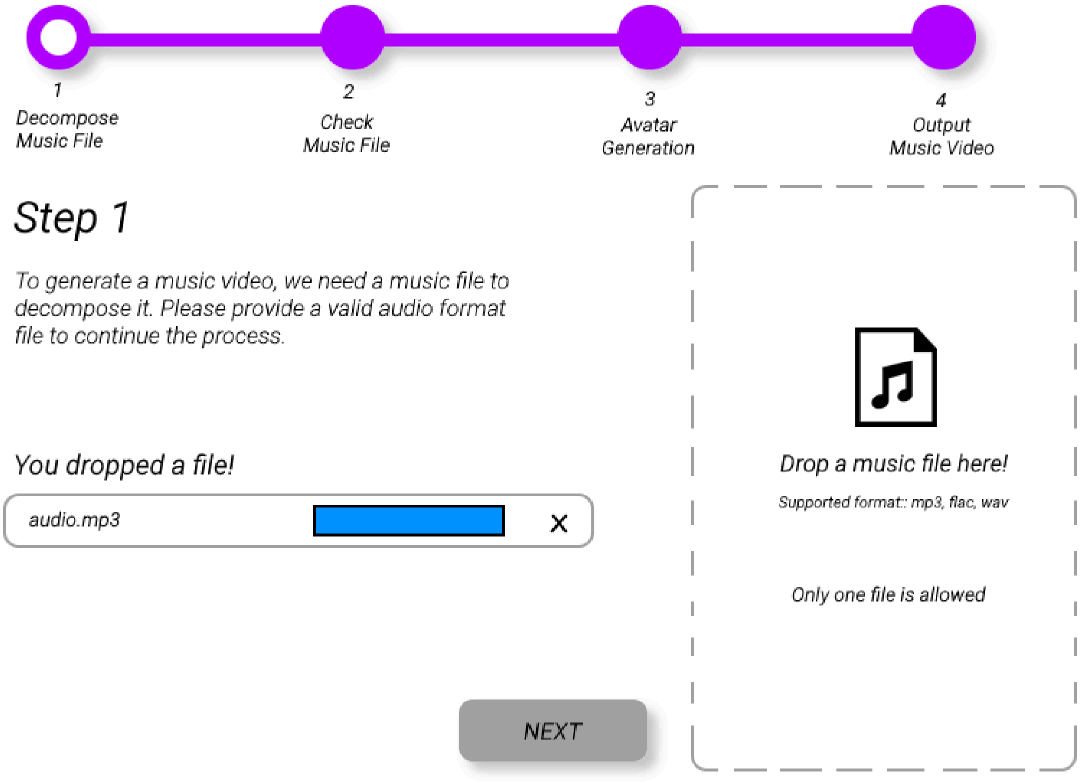

Input: Music Audio
Preprocessing of music audio
Decomposition of music video
Fail
Reinsert Music Audio
Pass
Extraction of Background music and human voice respectively
Testing quality of music and voice
Fail
Reinsert Music Audio
Pass
Adding a virtual avatar or facial image
Regenerate virtual avatar or facial image
Fail
Adding the extracted music and voice to a picture or virtual avatar
Performs lip synchronization and head movements
A singing video synchronized with the provided song

**Figure 1: Flowchart regarding Music video generation**

The flowchart in Diagram 1 displays the process of our final year project research, which begins by inputting a music audio in the application. By doing so, the music audio is preprocessed and decomposed into background music and human voice by using Spleeter, the Deezer source separation library with pretrained models written in Python and using Tensorflow. If the decomposition fails, the user will be redirected to the input phase and will be required to reinsert a valid music audio to be preprocessed. The decomposed background music and human voice will then be tested by allowing the user to listen to the extracted music and voice for checking purposes. Similarly, if the user dislikes the decomposed music and voice, the user will be able to reinsert another desired music audio for decomposition purposes. Then, with the extracted music and voice approved by the user, it will be added to a virtual avatar or facial image given by the user to the application. If the image is blurry or unreadable, the user will be required to reinsert a facial image or regenerate a virtual avatar.

Following, the virtual avatar or image will be added with the extracted music and voice, allowing the virtual avatar or image to perform lip synchronisation and head movements that follows the rhythm of the music. Lastly, a synchronised singing video will be generated with the provided song.

## User Interface Prototype



Step 1

To generate a music video, we need a music file to decompose it. Please provide a valid audio format file to continue the process.

You dropped a file!

audio.mp3

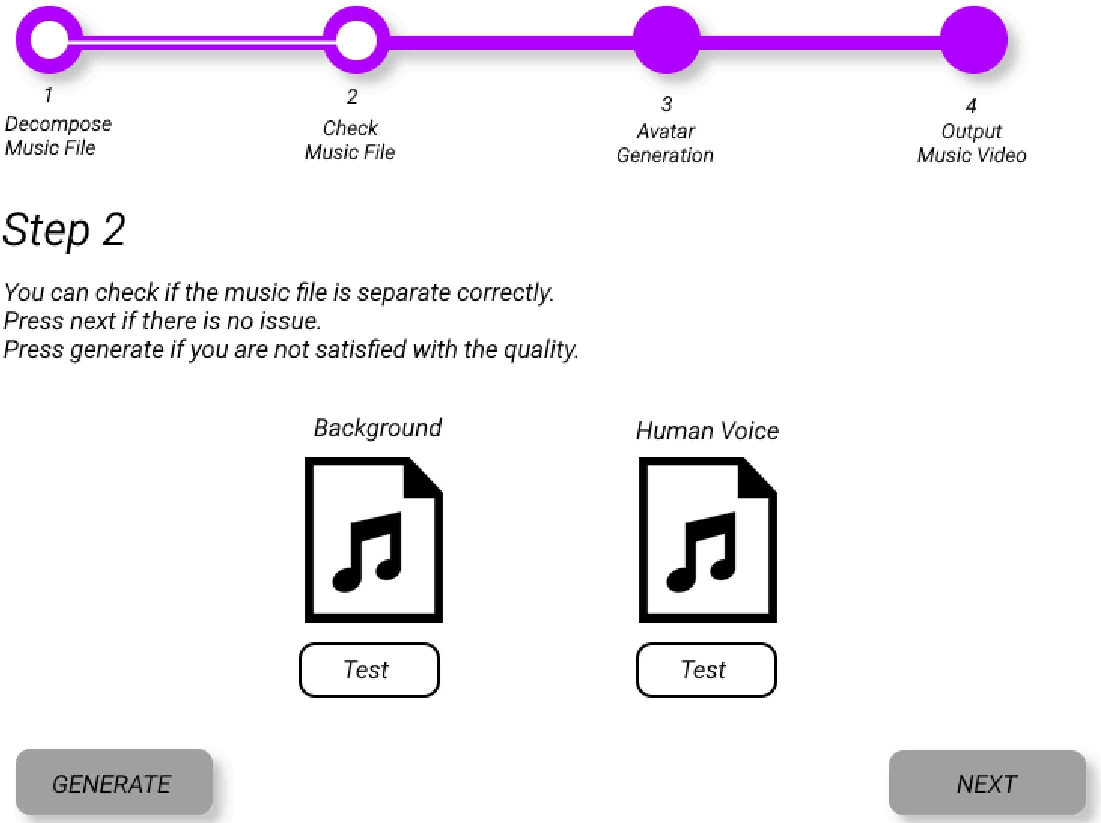
Drop a music file here!

Supported format:: mp3, flac, wav

Only one file is allowed

**Figure 2: Decompose music file through upload**

The first process of the application is to allow users to upload their desired music file into the application. To ensure that no mistakes are made, we only allow users to provide music file formats that are commonly used such as mp3, wav or flac. In addition, since our application only needs one music file, we will limit the number of files to 1. In figure 2, when a user uploads a file, it will display the music file. If you made a mistake by uploading the wrong file, the user can press the cross button and upload it again. After uploading a music file, the user can proceed to the next step by pressing the next button.

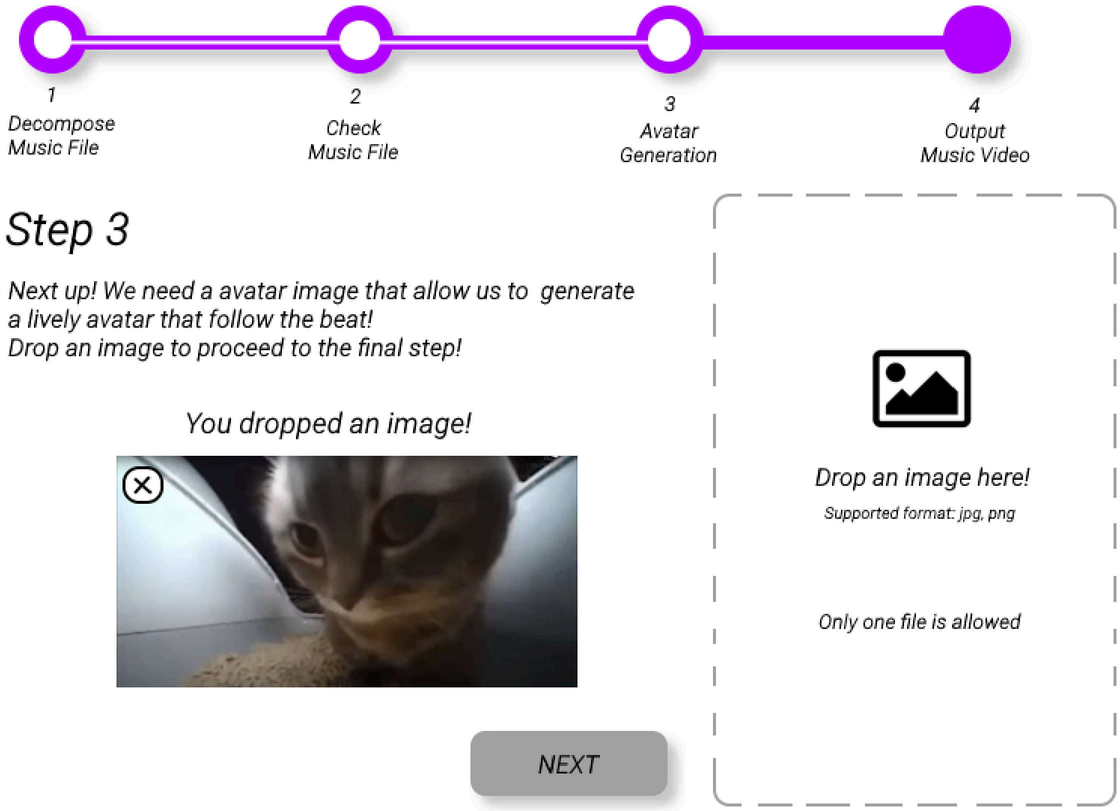


Step 2

*You can check if the music file is separate correctly.*
*Press next if there is no issue.*
*Press generate if you are not satisfied with the quality.*

**Figure 3: Decompose music file through upload**

After the application decomposes the music file through a separation algorithm, the second step involves user testing to ensure that the music file is correctly decomposed. Users can listen to both background music and human voice files by clicking the test button. The user can decompose it as many times until the user is satisfied with it. If the user thinks that there is no issue with the audio files, the user can click the next button to proceed to the next step.



Step 3

Next up! We need a avatar image that allow us to generate a lively avatar that follow the beat!
Drop an image to proceed to the final step!

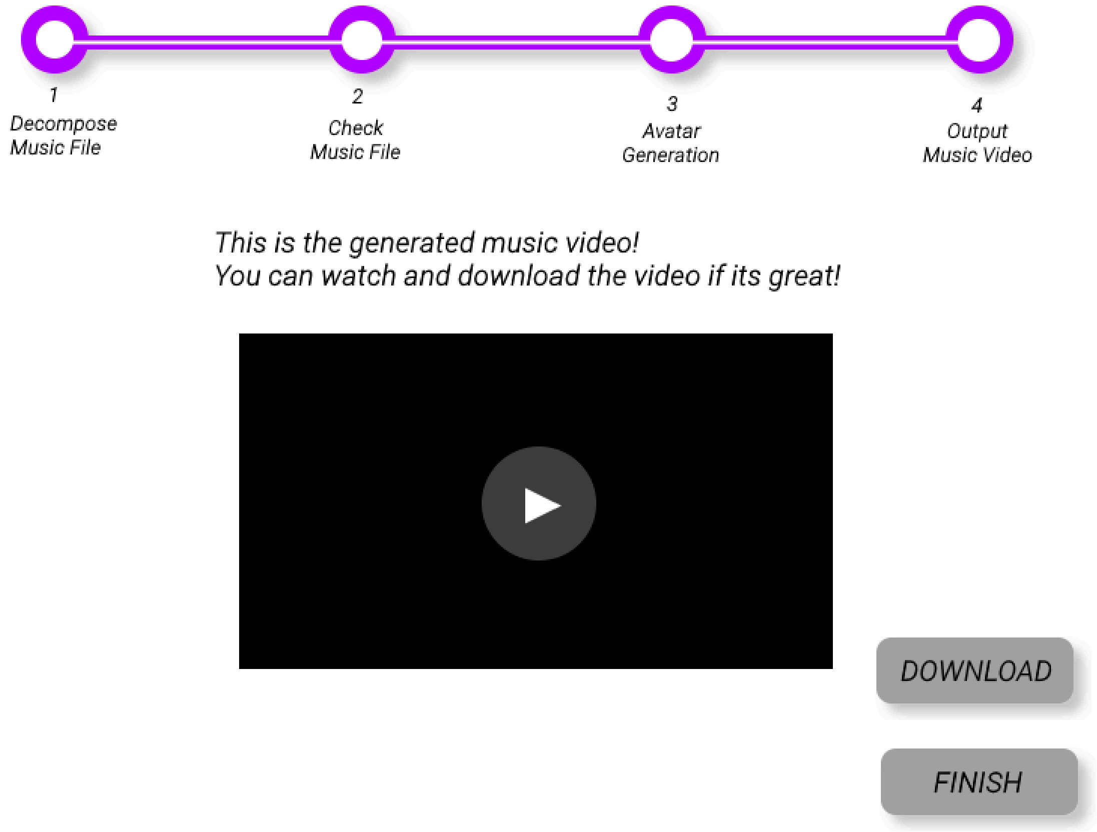
You dropped an image!

Drop an image here!
Supported format: jpg, png

Only one file is allowed

**Figure 4: Upload an image to generate avatar**

In the next step, the user needs to upload a photo of an avatar that allows the application to generate a moveable avatar based on the image given. It is recommended that the user provide a suitable image such that it has a clear front human face so that the application can generate lip and emotion synchronisation. If the user drops the image that the user want, the user can press the next button to the final step.



*This is the generated music video!*
*You can watch and download the video if its great!*

**Figure 5: Music video has been generated!**

Between the process, it will combine the background music along with an avatar that can sing! (The human voice used will remain the same using the human voice file, it will not generate a new voice) The final step will allow the user to watch the preview video and download it if the user is satisfied with the video. Users can end the application if they finish downloading the video.

# Software Specification

| Software | Justification |
|---|---|
| **Operating System**<br>Windows 10 or later | Our team will use the Windows 10 or better version as our operating system(OS) which is the only type of OS our team has. The other reason why we use this OS is because of its popularity with users. Globally, about 71% of users use Windows as their main and also with company. The documentation[1] has given some reasons for using Windows compared to Linux or MacOS such as user-friendly interface and extensive software compatibility. Therefore, designing the project using Windows makes it easier to ensure that users' devices are compatible with our project. |
| **Programming Language**<br>Python | We are going to use Python as our programming language in this project. First of all, it has a vast ecosystem of libraries or deep learning frameworks and will explain further in software libraries. Besides that, Python has been used by our team members in many units in our computer science study career. Furthermore, Python has a large number of developers and researchers. This helps us find out some useful tutorials or pre-trained models to improve the quality and accelerate the speed of designing projects. |
| **Software Libraries**<br>PyTorch, Spleeter | PyTorch is used for irregular input data such as graphs, point clouds and manifolds. PyTorch seamlessly integrates with Python that allows us to use libraries and tools in Python. This streamlines the development process and facilitates interoperability with other technologies used in AI which is our project needed for AI models. Also, PyKale is a library in PyTorch for multimodal learning and transfer learning with deep learning and dimensionality reduction on images and videos. There are few more libraries[2] that can be used and PyTorch is just the main used in our project. Also, |

| | |
|---|---|
| | Splitter can be used to separate human voice and background music. |
| **Programming Language Environment**<br>VSCode | VSCode is now commonly used by our team members for coding. For Python, we can also use PyCharm but we decided to use VSCode. As known, VSCode has lighter weight and faster performance compared to PyCharm. Besides that, VSCode is easier to connect with GitHub which has also been used in this project to store and backup the whole project. One extension that is useful is we can do live coding by using VSCode which makes our team easy to discuss. Furthermore, VS Code provides a Data Viewer that allows us to explore the variables within the code and notebooks, including PyTorch. [3] |
| **Frontend Programming Language**<br>HTML & CSS | These two languages are the standard language used to create a website application. HTML provides web page structure whereas CSS is used to control web page styling. [4] |
| **Backend Programming Language**<br>Python | Because it is mainly using PyTorch, so the backend programming language will use Python. |
| **Cloud Platform Service**<br>Google Colab | Google Colab offers free access to GPU(Graphics Processing Unit) and TPU (Tensor Processing Unit) resources, along with pre-installed libraries and frameworks like TensorFlow and PyTorch. We can directly write code in notebooks in its web browser and help us to avoid needing powerful local hardware. Because the free version has some limitations such as session time limits and restrictions on resource availability, we are suggested by the supervisor to have a pro version of Colab. We would need to pay for RM47.16 to get more units, memory and faster GPUs. |

| | |
|---|---|
| **Cloud Storage**<br>Google Drive | Google Colab has integration with Google Drive which allows us to save and share our Google Colab notebooks easily.[5] |
| **Version Control**<br>GitHub | We use GitHub to control our version as it is easier to clone its repository with VSCode. Besides, we can read the history commit from GitHub which means that we have a backup to our project. |
| **Project Management Tool**<br>Jira | We will use Jira as our project management tool. We chose Jira because the management process we learn is Agile and Jira supports all needed for the Agile process like kanban board, scrum and creating sprint[6]. While the free version has some limitations such as the limit of kanban boards created for each sprint, it still can be used nicely for the process. |

# Hardware Specification

| **Hardware** | **Justification** |
|---|---|
| Student’s Laptop | Although our laptop cannot handle heavy tasks like training large-scale deep learning models, we still can use it to do prototyping and do testing for code. |
| Graphics Processing Unit | GPUs are used to train the deep learning models. Although we are using the GPUs via Google Colab, we can still locally run our code or train models using our own GPUs. If we have enough budget to use, we can buy a NVIDIA GeForce RTX 4090 for local use. |

# References

1. Why use windows - https://www.quora.com/Why-do-so-many-companies-use-Windows-OS
2. Library in PyTorch and Python - https://pytorch.org/ecosystem/
3. How to use PyTorch in VSCode - https://code.visualstudio.com/docs/datascience/pytorch-support
4. Programming language needed for Frontend - https://www.simplilearn.com/tutorials/html-tutorial/html-vs-css#:~:text=HTML%20and%20CSS%20are%20scripting,to%20control%20web%20page%20styling.
5. Access Google Colab and Google Drive - https://towardsdatascience.com/different-ways-to-connect-google-drive-to-a-google-colab-notebook-pt-1-de03433d2f7a
6. Jira -https://www.atlassian.com/software/jira/agile#:~:text=Jira%20is%20an%20agile%20project,projects%20from%20a%20single%20tool.