**Project Overview**

Generating Music in different ways has been an active area of research for a long time. Some combinations of instruments that we are familiar with today, evolved over hundreds of years while others were the result of necessity, or an accident, or a crazy idea on the spur of the moment. Solo performances are not any combinations but generally use piano or guitar along with singing. Guitar and Piano are particularly suited for solo accompaniment because they can provide both harmony and rhythm.

We aim to look at novel and highly realistic Music composition for Single Instrument (Piano in our case as of now) when fed with raw audio waveforms as an input which follows all the rules of a particular genre chosen (ex. Jazz, Classical, etc.), using various algorithms (knowledge based approach) and Machine Learning models (learning based approach). The preliminary results in both the cases are very encouraging.

In our project, the user will play a small piece of music as an input and our model will analyze and categorize the short musical piece being played by the user and find out similar pieces in our database to generate pleasant music as an output which will be in continuation to the initial musical piece fed into the machine by the user. We will be using high processing computer systems to train our model. The risk included in our project is that the device will use a large amount of processing power and a bit of time to respond. Another risk factor we have considered is that the model might produce some unpleasant sound in some cases until it attains the desired accuracy.

**Need Analysis**

In earlier days, composing new music was considered a tedious and time consuming task but nowadays, since time is money, if we are able to reduce the time spent on composing new music, we will be able to reduce a lot of human effort, and cost as well. In this project, we aim to look at different approaches to effective Music generation ranging from purely knowledge based approaches to completely learning based systems.

SIMUGE can be used by:

1. Lyricists, who can compose a full-fledged song on their own without much help of music composers. It can thus help in reducing the overall production cost of the song.
2. New and young artists, who don’t have the required amount of knowledge and initial funding to give their carrier an initial start.
3. Existing professional composers as an inspiration to generate new genre of music which is different from their existing productions.
4. In Youth festivals, college fests and events, where students can’t afford professionals for original music generation.

The project makes composing music easy for commoners using the latest Machine Learning techniques and acts as a step forward towards a **New Musical Era**.

**Objectives**

The objectives of our project are described below:

1. To study the existing models, related projects and analyze them to deduce the feasibility and working of our own project.
2. To gather and preprocess the input audio datasets (.wav files, .mp3 files, etc) and apply various data mining operations on them to extract the useful data.
3. To optimize the chosen model to generate highly realistic and pleasant music with desired accuracy.
4. To depict the outcome in a waveform over a web/desktop or mobile application as shown in figure 1.



Figure 1: Expected output waveform

**Literature Survey**

1. **Lejaren Hiller and Leonard Isaacson, MUSICOMP [1]**
2. The earliest instance of computer generated composition is that of **Lejaren Hiller** and **Leonard Isaacson** at the University of Illinois in 1955-56.
3. Using the **Illiac** high-speed digital computer, they succeeded in programming basic material and stylistic parameters which resulted in the *Illiac Suite* (1957).
4. What Hiller and Isaacson had done in the *Illiac Suite* was: generated certain "raw materials" with the computer, modified these musical materials according to various functions, and then selected the best results from these modifications according to various rules.
5. This "generator/modifier/selector" paradigm was also later applied to **MUSICOMP**, one of the first computer systems for automated composition, written in the late 1950s and early 1960s by Hiller and **Robert Baker**.
6. **Iannis Xenakis [2]**
7. Another pioneering use of the computer in algorithmic composition is that of **Iannis Xenakis**, who created a program that would produce data for his "**stochastic**" compositions, which he had written about in great detail in his book *Formalized Music* (1963).
8. Xenakis used the computer's high-speed computations to calculate various probability theories to aid in compositions like *Atrées* (1962) and *Morsima-Amorsima* (1962).
9. The program would "deduce" a score from a "list of note densities and probabilistic weights supplied by the programmer, leaving specific decisions to a random number generator". As in the previous example of the *Illiac Suite*, these scores were performed by live performers on traditional instruments.
10. Genetic Algorithms for Music Generation. Rather than basing its grammar on scores input to the computer as in EMI, genetic programming *generates its own musical materials* as well as forms its own grammar.
11. The composer must also program a "critic" function, therefore, which then *listens* to the numerous automatically produced outputs at various stages of the processing to decide which are "fit" or suitable for final output (the composer having final say, then, as to which of these to discard and which to save).
12. **Automatic Music Generation for Indian Classical Music [3]**

(Paper by Sarika Mohapatra & Ankit Awasthi, Student at the Department of Computer Science and Engineering Indian Institute of Technology Kanpur)

1. In this project they aimed to look at different approaches to effective music generation ranging from purely knowledge based approaches to completely learning based systems.
2. They looked at music composition in the context of Indian Classical Music or the generation of Ragas using algorithms, mainly Hidden Markov models and Genetic Algorithms.
3. They looked at both knowledge based approach using Genetic Algorithms and learning based approach using Hidden Markov Models. Genetic Algorithms could also use be fed with guide pieces instead of Raga information to generate musical pieces making it a learning system as well.
4. **WaveNet: A Generative Model for raw audio [4]**
5. The work introduces WaveNet which is a deep neural network for generating raw audio waveforms. The model is fully probabilistic and autoregressive, with the predictive distribution for each audio sample conditioned on all previous ones.
6. When applied to text-to-speech, it yields state-of- the-art performance, with human listeners rating it as significantly more natural sounding than the best parametric and concatenative systems for both English and Mandarin.
7. A single WaveNet can capture the characteristics of many different speakers with equal fidelity, and can switch between them by conditioning on the speaker identity. When trained to model music, it is found that it generates novel and often highly realistic musical fragments.
8. **Audio synthesis at Jukedeck [5]**
9. Jukedeck is an AI based application that generates original musical audio from scratch.
10. What happens between a user requesting a piece of music and them delivering the audio are two things: **composition**and **synthesis**.
11. Composition means generating a **symbolic score,** similar to traditional music notation — i.e. they generate a representation of the notes and chords that will be played.
12. Up till now, the composition side is the thing they have focused on, and the part in which they have used machine learning — so the selection of notes and chords has been done by neural networks.
13. Synthesis, on the other hand, means turning that score into audio. For the synthesis, more traditional methods have been used — things like virtual instruments, through which notes composed by the neural networks were played.
14. Traditional synthesis methods, though, aren’t very flexible, and AI-driven synthesis has the potential to match the natural sound and variation of real-world instruments more closely.
15. Jukedeck’s goal is to provide the highest quality audio as fast as possible, powered by AI. To attain that goal, they have been working on their own audio synthesis algorithms.

**Methodology**

Figure 2 represents the overall Project Execution Plan in the form of a Flowchart.

1. Planning of a project and documenting of a Synopsis of the project.
2. Collection of useful and required datasets (audio file) from various sources in the form of .wav/.mp3.
3. Preprocessing of the dataset and extraction of the required features from the dataset. [6]
4. Selection of an appropriate machine learning model for our dataset by training different models and then selecting the one with the highest accuracy. [7] [8]
5. Since our machine learning model will be trained based on unsupervised learning approach, we will apply quality testing on some samples generated by our model either by enforcing rules of composition of different genres of music or using a similarity index between the input and output to increase the accuracy of our model so that it produces musical fragments as an output which are pleasant to the ear. Devising methods of testing and increasing our model accuracy also remains an objective of this project.
6. Depicting a waveform of the output music generated over a web/desktop or mobile application.
7. Deployment of our product in the form of a web/desktop/mobile application.

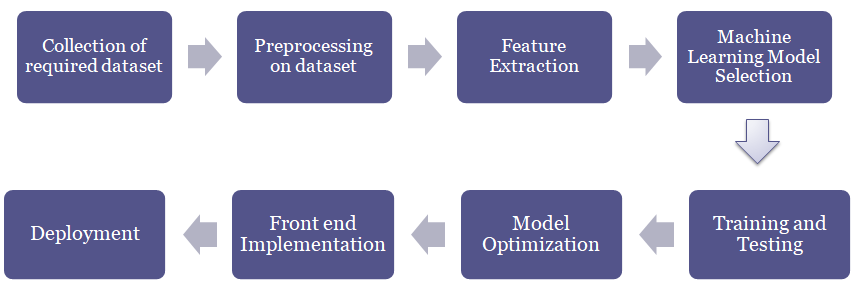


Figure 2: Project Execution Plan

**Requirement Analysis and Design Diagrams**

1. **Use Case Diagram: SIMUGE**

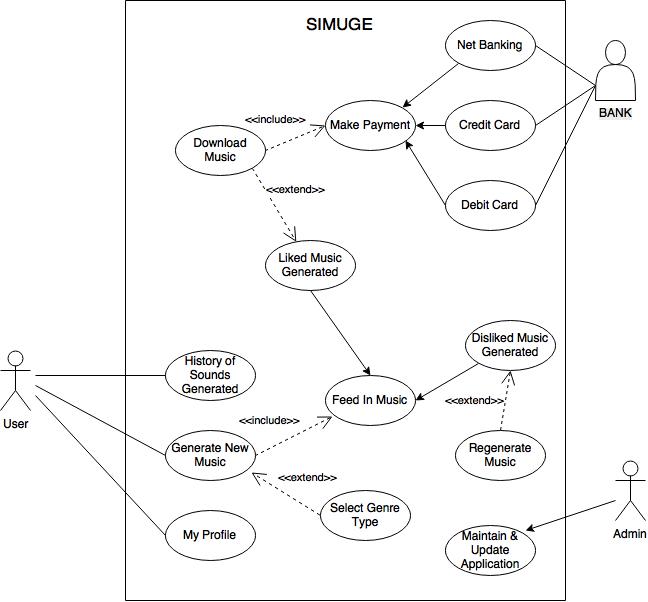


Figure 3: Use Case Diagram: SIMUGE

A user once logged in can do two things: View History of generated sounds previously and Generate New Music. Profile of the user can also be visited. Once Generate New Music is clicked, option to Feed in sample Music will appear which then will undergo manipulations and an output will be generated by the help of the Machine Learning model. Once a new music is being generated, user has an option to either Like or Dislike the Music. Disliking it will re-take the user to Feed in Music segment whereas on Liking the Music, two options will appear: End Session or Download the Music in suitable formats. The Admin maintains the application and Updates it regularly. A particular amount will be charged in case of downloading music whose payment will be managed at the back end by banking facilities.

1. **Use Case Template:**

Use Case: SIMUGE

ID: UCS001

Description: If the user likes the generated Music, he/she can download the same in suitable formats or can end session. Whereas, on disliking the music, New Music needs to be fed in.

Level: User Goal

Primary Actor: User

Supporting Actor: Admin, Bank

Stakeholders and Interests: Developers

Pre-Condition:

* Microphone should be working
* Speakers should be working
* Music needs to be fed in
* Internet Connection must me working

Post-Condition:

* Success End Connection: To Download Music or End Session
* Failure End Connection: To Regenerate Music or End Session

Minimal Condition: An assured Audio Output

Trigger:

* Mic detects Input Audio
* Speaker generates Output

Extension: End Session if Output not generated within specified time with a Message.

Variations: N/A

Frequency: Regular Basis (depends upon user’s interest)

Assumptions:

* Dataset is easily available
* Music generated is Novel and highly realistic
* Likeness of Generated Music depends upon user’s taste

Special Requirements: N/A

Issues:

* Accuracy of model chosen
* Generated Music should be realistic

To Do: N/A

1. **Activity Diagram:**

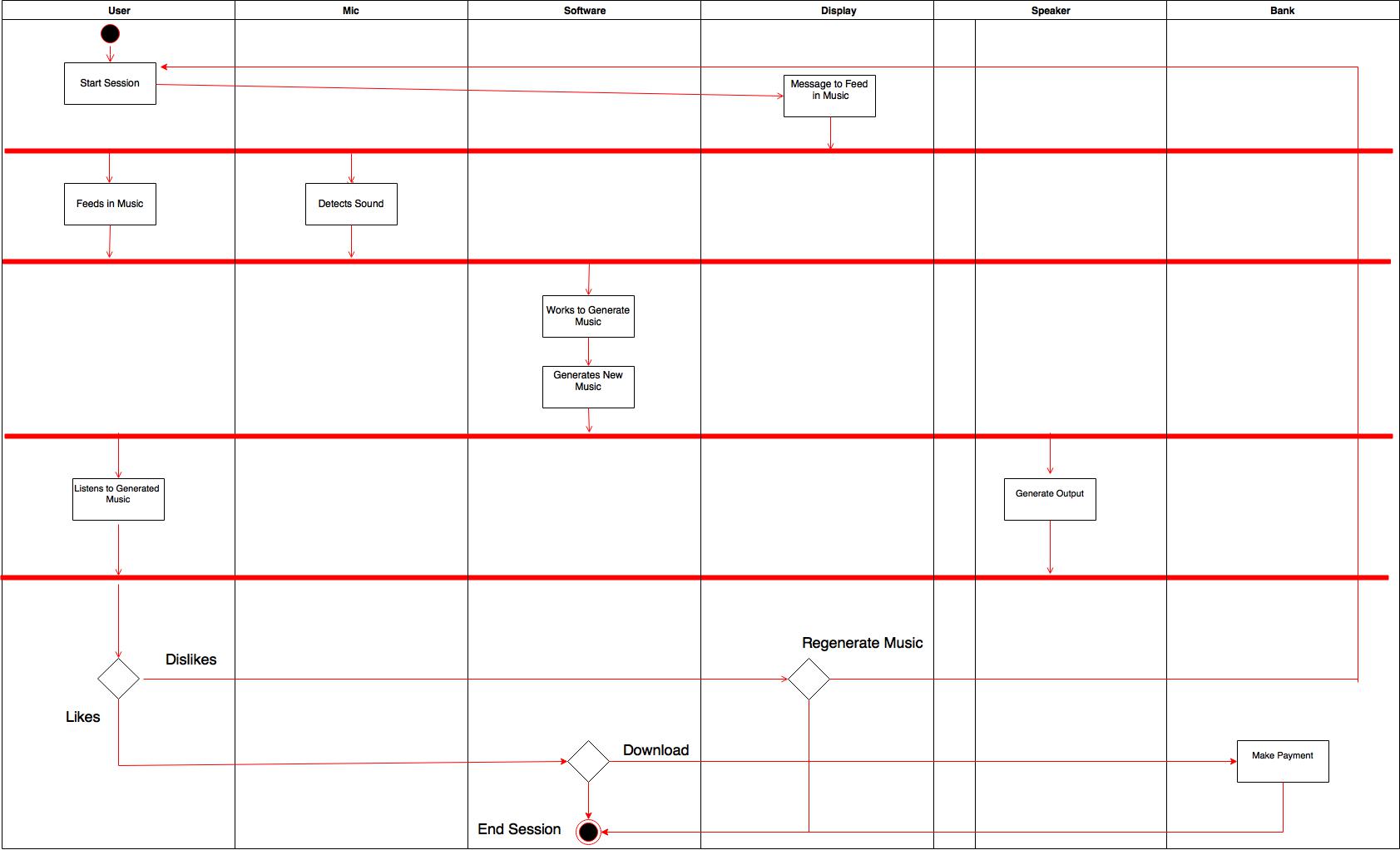


Figure 4: Activity Diagram

When the session starts, it is passed on to Feed in Music segment. Once Mic detects sound, Music is fed in. The Machine Learning Model works to generate New Music which then flows to User as an output through speakers. User Likes or Dislikes the Music further giving options to Regenerate or Download Music respectively or to End Session in both cases.

1. **Sequence Diagram:**

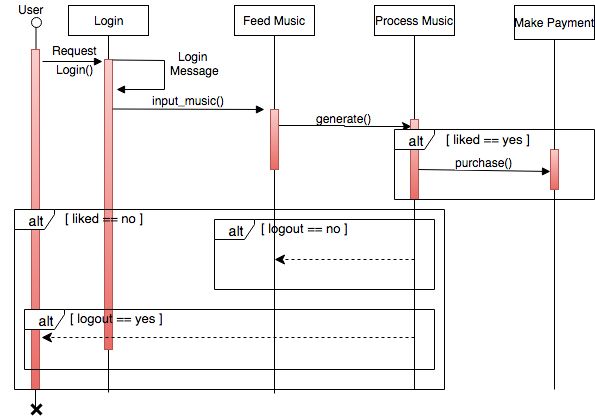


Figure 5: Sequence Diagram

User will request to Login. If Login is successful, he will be directed to Input Music otherwise will be returned to the Main Page. After feeding in music, the system will generate an output music. If the user likes the output generated, the user will have an option to purchase the music or just save and logout.

1. **E-R Diagram:**

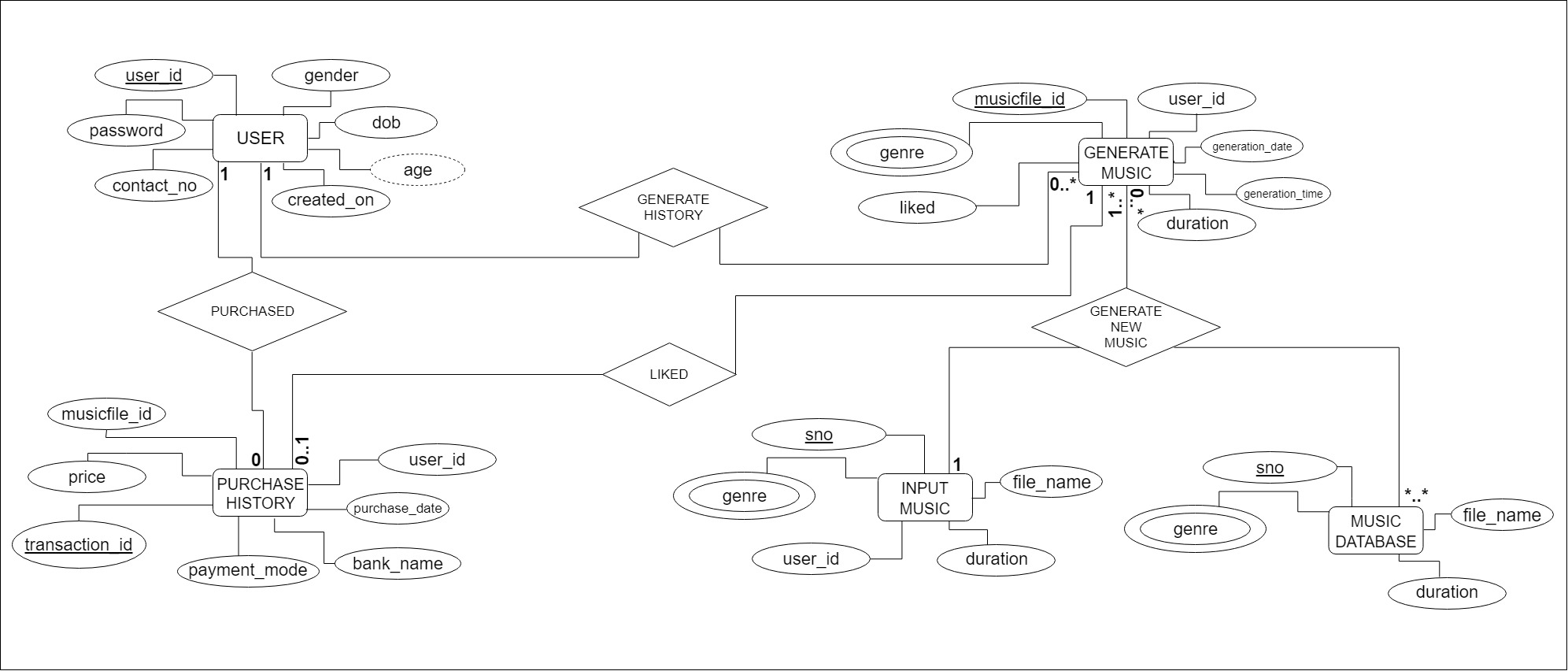
****

Figure 6: E-R Diagram

1. **Data Flow Diagram:**

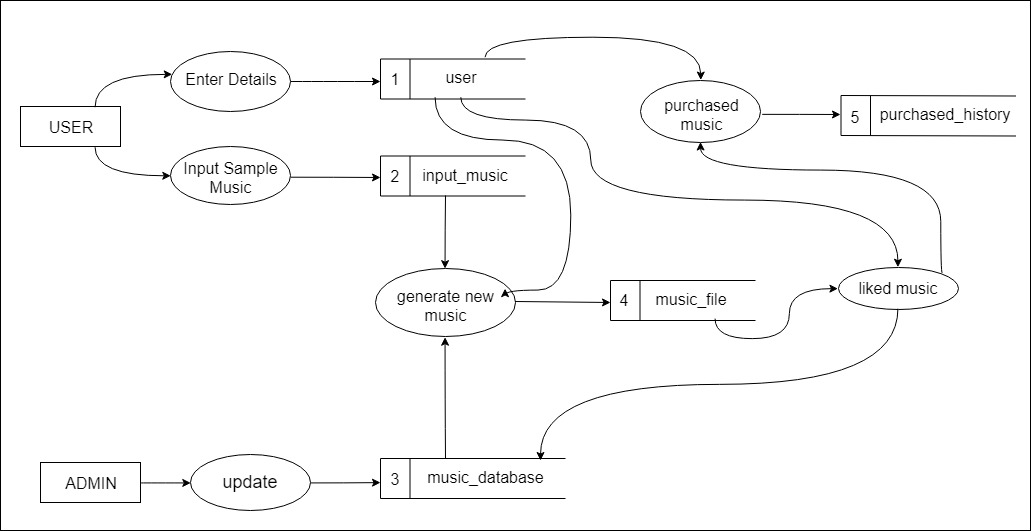


Figure 7: Data Flow Diagram

User and Admin are two entities. There will be various processes included in the whole scenario like Enter Details (where user will input its credentials), Input Sample Music (where user will be asked to feed in music), Generate new Music (where the system will undergo processing to generate an output), Liked Music (where the user will be ask to whether like or dislike the output generated), Purchase Music (where the user can buy the music generated depending upon their likeness), Update (only for admin where he/she can update any data at the backend).

Databases used include:

1. User: This will store details for all the users.
2. Input\_music: This will be the database for all the music samples fed in by the different users.
3. Music\_database: This will store all the music files needed to train the model containing .mp3/.wav files.
4. Music\_file: This will be the database for all the music files generated by our model.
5. Purchased\_History: This will store the transaction details for all the previous transactions.

**Tasks and Subtasks of the Project**

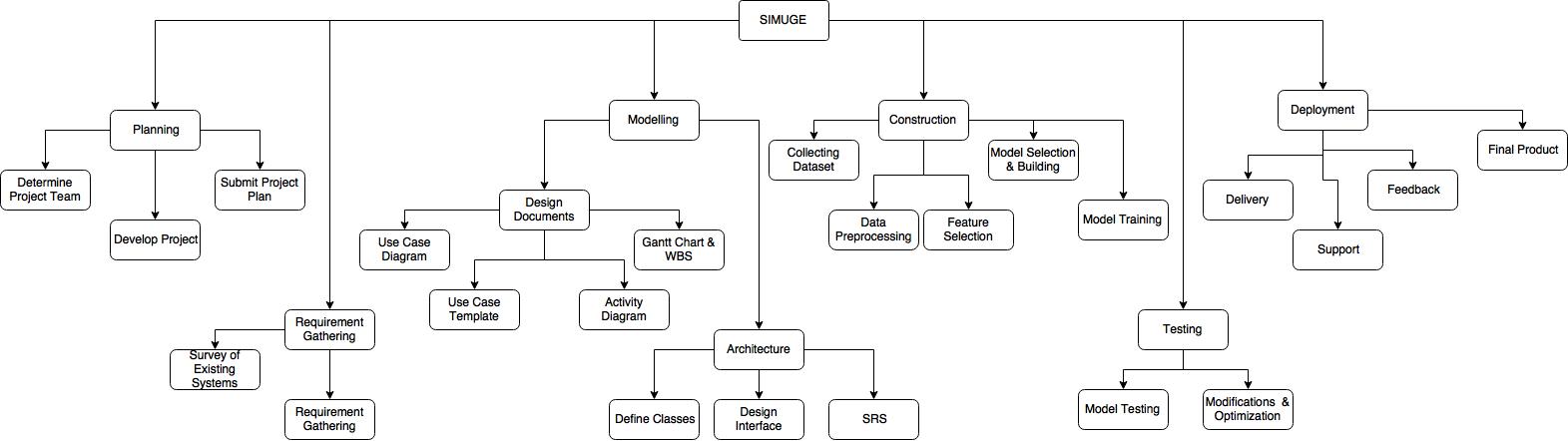
****

Figure 8: Project Schedule

**Assumptions and Constraints:**

**Assumptions:**

1. Dataset is easily available.
2. Music generated is highly realistic and novel.
3. There’s an assured Audio output generated by the system.
4. Likeness of generated music depends upon user’s taste.
5. Assuming that there is no delay in any sub task of the project for in-time completion of the same.

**Constraints:**

1. High computational power of the system is required to train our model with the help of a large database which is not feasible with our general-purpose laptops.
2. Dataset to be downloaded from Web is one of the other constraint because of its size and daily download limit restrictions.

**Analysis and Working Principles**

We explore raw audio generation techniques, inspired by recent advances in neural autoregressive generative models that model complex distributions such as images and text. We plan to use Convolutional Neural Networks to achieve this.

Now what is a CNN?

**Biological Connection**

CNN’s have been widely used for image classification since they take inspiration from the mammalian visual cortex. The visual cortex has small regions of cells that are sensitive to specific regions of the visual field. This idea was expanded upon by a fascinating experiment by Hubel and Wiesel in 1962 where they showed that some individual neuronal cells in the brain responded (or fired) only in the presence of edges of a certain orientation. For example, some neurons fired when exposed to vertical edges and some when shown horizontal or diagonal edges. Hubel and Wiesel found out that all of these neurons were organized in a columnar architecture and that together, they were able to produce visual perception. This idea of specialized components inside of a system having specific tasks (the neuronal cells in the visual cortex looking for specific characteristics) is one that machines use as well and is the basis behind CNNs.

**Structure**

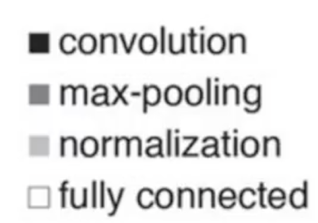
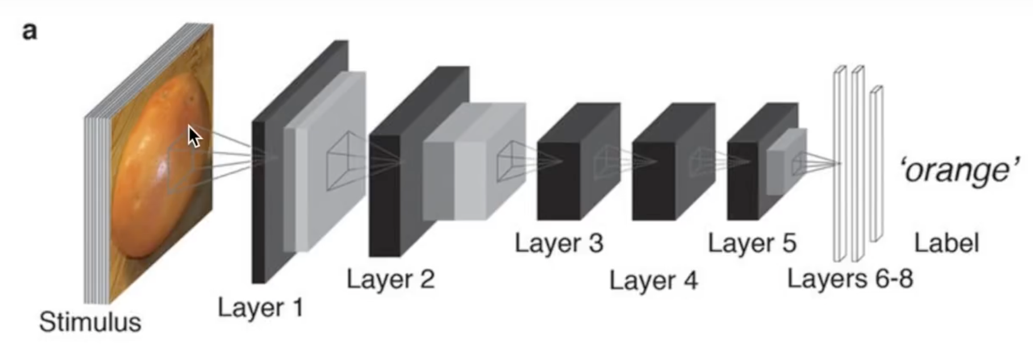
****

Figure 9: CNN Structure

A simple ConvNet is a sequence of layers, and every layer of a ConvNet transforms one volume of activations to another through a differentiable function. We use three main types of layers to build ConvNet architectures: **Convolutional Layer**, **Pooling Layer**, and **Fully-Connected Layer** (exactly as seen in regular Neural Networks). We will stack these layers to form a full ConvNet **architecture**.

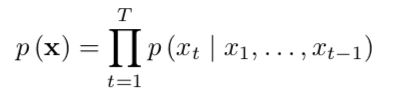
*Example Architecture: Overview*. We will go into more details below, but a simple ConvNet for CIFAR-10 classification could have the architecture [INPUT - CONV - RELU - POOL - FC]. In more detail:

* INPUT [32x32x3] will hold the raw pixel values of the image, in this case an image of width 32, height 32, and with three color channels R, G, B.
* CONV layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume. This may result in volume such as [32x32x12] if we decided to use 12 filters.
* RELU layer will apply an elementwise activation function, such as the max (0, x) this leaves the size of the volume unchanged ([32x32x12]).
* POOL layer will perform a downsampling operation along the spatial dimensions (width, height), resulting in volume such as [16x16x12].
* FC (i.e. fully-connected) layer will compute the class scores, resulting in volume of size [1x1x10], where each of the 10 numbers correspond to a class score, such as among the 10 categories of CIFAR-10. As with ordinary Neural Networks and as the name implies, each neuron in this layer will be connected to all the numbers in the previous volume.

In this way, ConvNets transform the original image layer by layer from the original pixel values to the final class scores. Note that some layers contain parameters and other don’t. In particular, the CONV/FC layers perform transformations that are a function of not only the activations in the input volume, but also of the parameters (the weights and biases of the neurons). On the other hand, the RELU/POOL layers will implement a fixed function. The parameters in the CONV/FC layers will be trained with gradient descent so that the class scores that the ConvNet computes are consistent with the labels in the training set for each image.

**What we are doing**

we use a new generative model operating directly on the raw audio waveform. The joint probability of a waveform x = {x1 ..., xT} is factorized as a product of conditional probabilities as follows:



Each audio sample xt is therefore conditioned on the samples at all previous timesteps. Similarly, to PixelCNN’s conditional probability distribution is modelled by a stack of convolutional layers. There are no pooling layers in the network, and the output of the model has the same time dimensionality as the input. The model outputs a categorical distribution over the next value xt with a SoftMax layer.

The neural network architecture directly generates a raw audio waveform.

The network models the conditional probability to generate the next sample in the audio waveform, given all previous samples and possibly additional parameters.

After an audio pre-processing step, the input waveform is quantized to a fixed integer range. The integer amplitudes are then one-hot encoded to produce a tensor of shape (num\_samples, num\_channels).

A convolutional layer that only accesses the current and previous inputs then reduces the channel dimension.

The core of the network is constructed as a stack of causal dilated layers, each of which is a dilated convolution (convolution with holes), which only accesses the current and past audio samples.

The outputs of all layers are combined and extended back to the original number of channels by a series of dense postprocessing layers, followed by a SoftMax function to transform the outputs into a categorical distribution.

The loss function is the cross-entropy between the output for each timestep and the input at the next timestep.

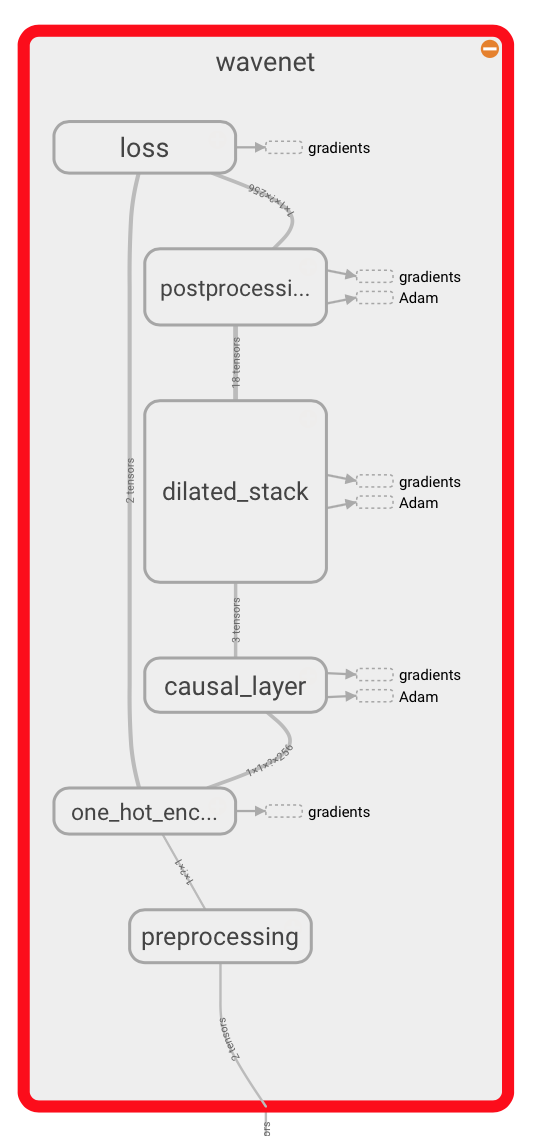
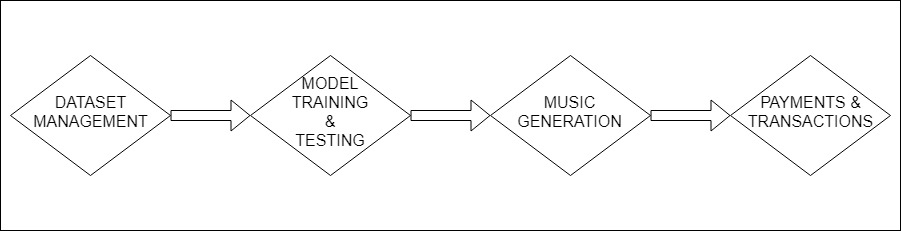


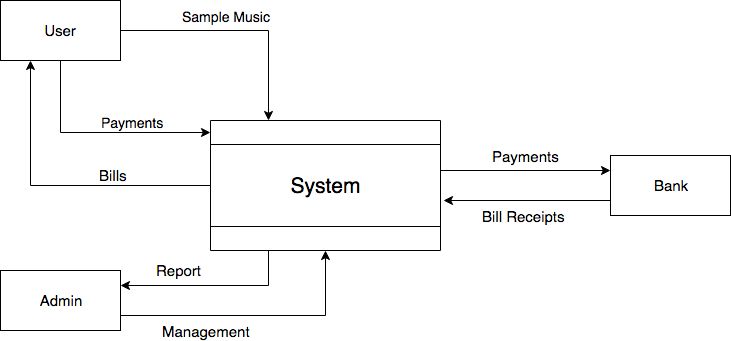
Figure 10: CNN Process Flow

**Design Model**

1. **System Architecture:**

Figure 11: System Architecture

1. Dataset Management: Management of various piano .mp3/.wav files.
2. Model Training and Testing: Selection of the best Machine Learning model and make it undergo Training and Testing.
3. Music Generation: Generation of novel and highly realistic music using sample music fed to the trained machine learning model.
4. Payments and Transactions: Management of various payments and transactions between a bank and customers through a gateway.
5. **Context Diagram:**

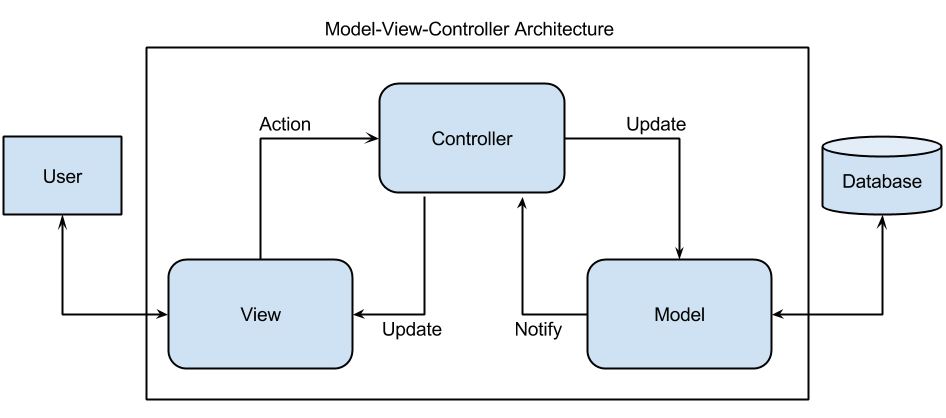


Music Generated

Figure 12: Context Diagram

This diagram shows the requests made by different entities and answers or return outputs being sent to them by the System. For example, User can feed in sample music or can generate a query to pay/buy the generated music. In return, system sends the generated music and bill receipts.

1. **MVC Architecture:**

 Figure 13: MVC Architecture

The three components to the MVC software architecture perform the following:

1. View – provides the interaction that the user sees (typically, a web page).  The view components provide data to the user and send actions to the Controller for manipulating data.
2. Model – defines the data for the application (typically, the data is stored in a database).
3. Controller – basically, provides the interface between the View and the Model.
4. **Tier Architecture:**

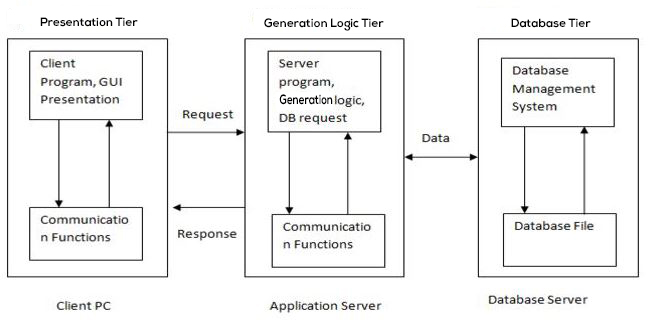
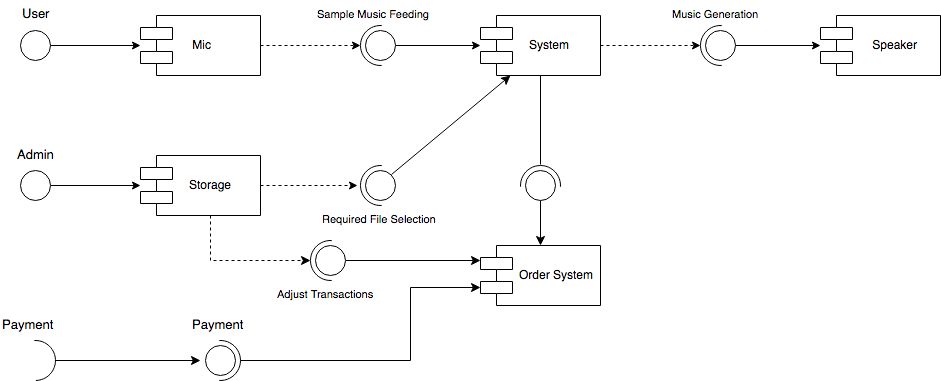


Figure 14: Tier Architecture

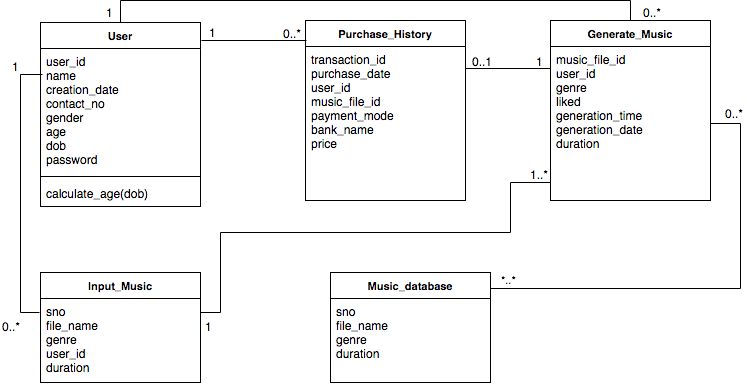
In 3 tier architecture, there are 3 components: Client PC, An Application server and A Database Server.

1. The work of server is distributed among application server and database server.
2. Application server has the required communication functions.
3. The data required by the business logic exists in database server.
4. The required data is returned to public servers and then to client PC.
5. **Component Diagram:**

 Figure 15: Component Diagram

The various components used in our model are:

1. Mic: To feed in sample music by the user.
2. Storage: This contains all the databases (sample music to train model, input music by the user, generated music, User details, Transaction details).
3. System: This will undergo the process of generating music with the help of our chosen model.
4. Order System: This will process all the transactions and manage their details.
5. Speakers: This will play the output music generated by the model.
6. **Data Design Diagram:**

 Figure 16: Data Design Diagram

It determines the data to be stored in the database and the relationships between the different data elements. It then helps in making a logical structure upon the data on the basis of these relationships.

**Work Plan**

The tables in the following page describe the working on different phases of the project in the form of a timeline. The work plan can be described in 6 discrete phases:

1. **Planning of Project** –Surveying Music Industry for easy generation of Music and making of a Synopsis for our finalized project.
2. **Collection of required datasets** –This phase of project includes gathering various audio files of different genres of music compositions being played on Piano only in the form of .wav or .mp3 files.
3. **Pre-processing of data and extraction of required features** –This phase includes pre-processing of the input files to extract required features using various audio libraries and neural network algorithms.
4. **Study of Machine Learning models and appropriate model selection** – Various models will be trained and the one with the best accuracy will be accepted as the most appropriate one.
5. **Modifying and Optimizing the model** – Applying quality testing on some samples either by enforcing rules of composition of different genres of music or using a similarity index between the input and output to increase the accuracy of our model.
6. **Final Report, Documentation and Deployment** – Depicting output on a web/desktop or mobile application in a waveform. The website then can be deployed for proper commercial use in later stages of development.

**Project Schedule through Gantt Chart**

Table 1: Work Plan for Semester 6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Activity** | **January** | | **February** | | | | **March** | | **April** | | | | **May** |
| **3** | **4** | **1** | **2** | **3** | **4** | **1** | **4** | **1** | **2** | **3** | **4** | **1** |
| **1** | **Planning of Project** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** | **Making a Synopsis of the Project** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** | **Drafting of SRS & Flow Diagrams** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** | **Collection of Required datasets** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **5** | **Pre-processing of data and Extraction of required features** |  |  |  |  |  |  |  |  |  |  |  |  |  |

Table 2: Work Plan for Summer and Semester 7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr. No.** | **Activity** | **June & July** | | | | **August** | | | **October** | | | | **November** | | |
| **1** | **2** | **3** | **4** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** |
| **6** | **Study of Machine Learning models and Data Analytics** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **7** | **Appropriate Model Selection** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **8** | **Model Training and Testing** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **9** | **Modifications and Optimization** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **10** | **Final Report and Deployment** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**WBS &** **WBS DICTIONARY**

|  |  |  |
| --- | --- | --- |
| **Project Title** | **SIMUGE** | |
| **Project Manager** | **ANEESH JAIN** | |
| **Document Version  & Update Date** | Version 1.0  25th March 2018 |  |

**TEAM MEMBERS:**

A1: ABHISHEK BANSAL A2: ANEESH JAIN

A3: ANKUR SHARMA A4: ANTRA SHARMA

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.0 | **WBS Name:**  SIMUGE | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *None* | **WBS Description:**  We are trying to build an Automatic Single Instrument Music Generator using Machine Learning approach. The software, for now, targets various events, fests held in college or singers who need single instrument music with a completely new touch. The project can be extended as per requirements in later stages to multiple instruments, and then used by music composers and other artists as well. | |
| **Must-Start Date:**  17th January, 2018 |
| **Must-Finish Date:**  30th November, 2018 |
| **WBS Estimated Time:**  318 days | **Assumptions and Constraints:**  *Assuming there is No delay at every sub task in the project.* | |
| **Assigned to: A1, A2, A3, A4** |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.1 | **WBS Name:**  Planning of Project  (Why did we start this project?) | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *No Predecessor* | **WBS Description:**  We started planning and working in following steps:   1. Building of a proper team with similar ideology to work on the project. 2. Determining a Project Plan | |
| **Must-Start Date:**  17th January, 2018 |
| **Must-Finish Date:**  30th January, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.2 | **WBS Name:**  Making a Synopsis of the Project | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Planning of the Project* | **WBS Description:**  The project was started in order to develop a new tool to generate novel and highly realistic music using the latest technology and concepts of Machine Learning.   1. Preparing a Synopsis of the Project 2. Submitting the final Plan of the Project | |
| **Must-Start Date:**  1st February, 2018 |
| **Must-Finish Date:**  18th February, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.2 | **WBS Name:**  Drafting of SRS & Flow Diagrams | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Making a Synopsis of the Project* | **WBS Description:**  Creating all the Flow Diagrams, documenting SRS followed by their final Correction and Approval. | |
| **Must-Start Date:**  21st February, 2018 |
| **Must-Finish Date:**  4th April, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.2.1 | **WBS Name:**  Drafting of Flow Diagrams | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Drafting of SRS & Flow Diagrams* | **WBS Description:**  Created Use Case Diagram, Use Case Templates, Activity Diagram. | |
| **Must-Start Date:**  21st February, 2018 |
| **Must-Finish Date:**  1st March, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.2.2 | **WBS Name:**  Documenting SRS | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Drafting of Flow Diagrams* | **WBS Description:**  After the User Requirements are gathered and Flow diagrams have been made, SRS is made with detailed information about: Purpose, Scope, User Interface, and Hardware & Software Details of the project. | |
| **Must-Start Date:**  21st March, 2018 |
| **Must-Finish Date:**  28th March, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.2.3 | **WBS Name:**  Corrections and Approval of SRS & Flow | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Documenting SRS* | **WBS Description:**  The documented SRS along with the Flow diagrams is then reviewed by the team, mentor and all the errors (if any) are corrected and then the SRS is approved. | |
| **Must-Start Date:**  28th March, 2018 |
| **Must-Finish Date:**  5th April, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.3 | **WBS Name:**  Collection & Analysis of Data | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Documenting SRS* | **WBS Description:**  The project was started in order to develop a new tool to generate novel and highly realistic music using the latest technology and concepts of Machine Learning for a Single Instrument. | |
| **Must-Start Date:**  5th April, 2018 |
| **Must-Finish Date:**  25th April, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.4 | **WBS Name:**  Pre-processing of data and Extraction of required features | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Corrections and Approval of SRS & Flow* | **WBS Description:**  Once we have enough datasets to work upon, the project moves to construction stage where data is pre-processed and required features are studied and extracted. | |
| **Must-Start Date:**  20th April, 2018 |
| **Must-Finish Date:**  5th May, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.5 | **WBS Name:**  Study of Machine Learning Models & Appropriate Machine Learning Model Selection | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Pre-processing of data and Extraction of required features* | **WBS Description:**  Various Machine Learning Models are studied over time and datasets are being run over in order to select the one with the best accuracy. | |
| **Must-Start Date:**  1st June, 2018 |
| **Must-Finish Date:**  20th July, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.6 | **WBS Name:**  Model Training & Testing | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Study of Machine Learning Models & Appropriate Machine Learning Model Selection* | **WBS Description:**  The chosen Machine learning model is trained over several databases and numerous input music files in order to generate highly realistic and novel music which is the major objective of the project. | |
| **Must-Start Date:**  7th August, 2018 |
| **Must-Finish Date:**  27th August, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.7 | **WBS Name:**  Modifications & Optimization | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Model Training & Testing* | **WBS Description:**  The chosen Machine learning model is run over various input databases until it generates highly novel and realistic music. Various modifications are done in order to achieve Optimization. | |
| **Must-Start Date:**  28th September, 2018 |
| **Must-Finish Date:**  28th October, 2018 |

|  |  |  |
| --- | --- | --- |
| **WBS Number:**  1.8 | **WBS Name:**  Deployment & Future Scope | **Work Package?**  **☐ Yes / ☐ No** |
| **WBS Predecessor(s):**  *Connecting Database with GUI* | **WBS Description:**  The project which we have implemented is for a Single Instrument only i.e. Piano which can be further extended to other music Instruments like Guitar, Flute, etc. and then to multiple instruments at a time. The final product of our project will be released for deployment with proper reports and documentation. | |
| **Must-Start Date:**  1st November, 2018 |
| **Must-Finish Date:**  30th November, 2018 |

**Functional and Non-Functional Requirements of Project**

**Functional Requirements**

A new user will first sign up through an email-id and set a password. After the initial signup the user can login into their accounts to either view history of previously generated sounds or generate new music.

If new music generation option is selected, the user is asked to feed in sample music. The sample music is recorded and hence fed into the system with the help of a mic. This sample music is used to generate a whole musical piece using the basic concepts of Machine Learning.

The newly generated music can be liked or disliked by the user based upon his/her taste. If disliked, the user is asked to again feed the sample music into the system. If liked, the user can either end the session or download the whole musical piece after payment is done through respective gateways maintained by banks.

**Non-functional Requirements**

1. Performance and Efficiency **-** Model selection and optimization is crucial since it will directly affect the response time and quality of music generated. Better quality and highly realistic music will improve the efficiency of the product.
2. Usability **-** The user interface should be friendly and easy to use so that people not having much knowledge about music and technology are also able to use the product easily.
3. Secure **-** Payments should me made securely.

**Project Outcomes**

The project aims at generating novel and highly realistic music as an output by learning from raw audio wave forms as an input which follows all the rules of the genre chosen (ex. Jazz or Classical, etc.) for a single instrument only (piano in our case) and depict the outcome in a waveform on a web/desktop or mobile application.

**Individual Roles**

Table 3: Individual Roles of all Team Members

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Team Member Name** | **Literature Review** | **Data Mining** | **Model Training** | **Testing & Optimization** | **Front-end Development** |
| **Abhishek Bansal 101510006** | **✓** | **✓** |  | **✓** | **✓** |
|
| **Aneesh Jain 101510015** | **✓** | **✓** | **✓** | **✓** |  |
|
| **Ankur Sharma 101510016** | **✓** | **✓** |  | **✓** | **✓** |
|
| **Antra Sharma 101690010** | **✓** | **✓** | **✓** | **✓** |  |
|

**Course Subjects**

The following course subjects come into use during the execution of our project:

1. **Data Analytics -** Data Analytics is the science of examining raw data with the purpose of finding patterns and drawing conclusions about that information by applying an algorithmic or mechanical process to derive insights. Since our data is in the form of raw audio files in .wav/.mp3 format, it is the most crucial step to convert these audio files to machine understandable form such that finding similarities and patterning the data becomes easy.
2. **Machine Learning -** **Machine Learning focuses on the development of computer programs** that can access data and use it to learn for themselves. Once the data is extracted from raw audio files, it is fed into the best suited machine learning model giving the desired accuracy. The model once trained is then tested and optimized either by enforcing rules of composition of different genres of music or using a similarity index between the input and output to increase the accuracy of our model so that it produces musical fragments as an output which are pleasant to the ear. Devising methods of testing and increasing our model accuracy also remains an objective of this project.
3. **Deep Learning –** Deep Learning is a part of broader family of Machine Learning methods based on learning data representations and can be supervised, semi-supervised or unsupervised. Deep learning models are used in Neural Networks and have been applied to the fields including Natural language processing, Audio recognition, speech recognition, computer vision, etc.
4. **Front end Development –** Front end development languages include HTML, CSS and Bootstrap for styling of the web pages whereas Java Script is used for functionality. When the whole model is ready, the output will be shown on a Web/Desktop or Mobile Application in a waveform using various front end languages along with PHP to store files at the back end database.

**REFERENCES**

**[1]** Lejaren A. Hiller and Leonard M. Isaacson (1979). Experimental Music: Composition With an Electronic Computer, second edition (New York: McGraw-Hill, 1959): 5–7. Reprinted, Westport, Conn.: Greenwood Press, 1979. ISBN 978-0-313-22158-3.

**[2]** Xenakis, Iannis (2001). Formalized Music: Thought and Mathematics in Composition (Harmonologia Series No.6). Hillsdale, NY: Pendragon Press. ISBN 1-57647-079-2.

**[3]** Sarika Mohapatra (Y6430), Ankit Awasthi (Y8084) (2010). Automatic Music Generation for Indian Classical Music.

**[4]** Aaron van den Oord, Karen Simonyan, Nal Kalchbrenner, Sander Dieleman, Oriol Vinyals, Andrew Senior, Heiga Zen†, Koray Kavukcuoglu, Alex Graves (2016). "WaveNet: A generative model for raw audio". *arXiv:1609.03499v2, 2016.*

**[5]** Ed Rex (Founder and CEO) (2015). Jukedeck. Available: https://www.jukedeck.com/about

**[6]** Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin (2016), et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

# [7] Siraj Raval (June 2017). Convolutional Neural Networks - The Math of Intelligence [Online]. Available: https://www.youtube.com/watch?v=FTr3n7uBIuE

### [8] Ujjwal Karn (August 2016). An Intuitive Explanation of Convolutional Neural Networks [Online]. Available: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/