EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION

BY

HAYDEN NG

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements
for the degree of
BACHELOR OF BUSINESS INFORMATION SYSTEM (HONS)
Faculty of Information and Communication Technology
(Kampar Campus)

JANUARY 2019

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION

Academic Session: JAN 2019

I ________________HAYDEN NG__________________________________

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

_________________________ (Author's signature)

_________________________ (Supervisor's signature)

Address:
93, Taman Cahaya,__________
27200, Kuala Lipis,__________
Pahang.___________________

MR. SOONG HOONG CHENG
Supervisor's name

Date: __20/4/2020___________

Date: _____20/4/2020____

EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION

BY

HAYDEN NG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF BUSINESS INFORMATION SYSTEM (HONS)

Faculty of Information and Communication Technology

(Kampar Campus)

JANUARY 2019

DECLARATION OF ORIGINALITY

I declare that this report entitled "**EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : __________________________

Name : Hayden Ng________________

Date : 20/4/2020___________________

ACKNOWLEDGEMENTS

I would like to convey my heartfelt thanks and appreciation to my supervisor, Mr. Soong H.C who has given me an astounding opportunity to participate in an Artificial Intelligence related project. It helps a lot in expanding my knowledge in the IT field as well as my future.

Thanks Mr Soong for assisting me and providing me assistance all the way throughout my Final Year Project, it has been a huge help to my journey in this Final Year Project. Lastly, appreciation to my parents and my family should also not be forgotten. Huge thanks to my parents and family that had stood by me and providing me guidance through my Final Year Project. The patience, love and care is all I needed and it is kind of what that motivates me further.

Abstract

This project focuses on delivering a JAVA application named “Evoluthm” with mainly the functions of generating chords progressions and beats. The system aims to aid in inspiring music composer and also generate music based on the requirements or needs of the music composer. For instance, the music’s speed, pitch range, number of layers, number of notes to be generated and different types of music instruments can be selected. Evoluthm utilizes Harmony Search Algorithm and JFUGUE techniques in compliances with artificial intelligence (AI). These techniques are used in success to produce a perfect music or melody and minimizes the number of error melodies. Thus, Evoluthm is a tool to help musician to obtain inspiration from random generated melodies to compose a new piece of music.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
IT	Information Technology
MIDI	Musical Instrument Digital Interface
GUI	Graphical User Interface
HM	Harmony Memory
HS	Harmony Search
HMCR	Harmony Memory Considering Rate
PAR	Pitch Adjustment Rate
NI	New Improvisation
HMS	Harmony Memory Size
HMA	Harmony Search Algorithm
JDK	JAVA Development Kit
JRE	JAVA Runtime Environment
IDE	Integrated Development Environment

Chapter 1: Project Background

1.1 Music

Music a skill, a creative activity that is formed by rhythm, pitch, melody and harmony. It is a composed of interrelated sound or sounds that allow the utterance of sentiments or the creation of a particular atmosphere to convey ideas through harmonic frequencies. The phrase “music” comes from the Greek word, *mousike*, which means the art of the Muses. Music is made by putting tones and sounds together in an organized order. If it is put together correctly, it can be used to express particular circumstances, surroundings and sentiments. Beethoven is one of the world’s finest musician who can make music creatively compose sounds for a correct outcome. Somebody sees music as an entertainment or an interest. Nearly everyone has a practice or creativity of making music. In addition, music brings a huge range of benefits to people. Playing music helps a person relax his or her mind, relief stress and even bring pleasure to the person. In other words, music is a short escape from reality. Music can be interesting in different ways. Some people like to dance to the music while some people prefer playing and singing along with the music.

Music often has four important things, pitch, rhythm, dynamics and timbre which determine the qualities of a piece of music (Biles.AL, 2004). A pitch is notes that can be high or low. Composition of various notes will make up tunes which is the variation of the pitch. Next, rhythm is when silences, vibrations and sounds are grouped together in an order and produces a consistent or inconsistent (off-beat) beat to the music. Essentially, rhythm is the arrangement of sound in time. Additionally, tempo refers to the pace of beats which made it an essential element to rhythm. Furthermore, dynamics basically means when the music should be quiet or loud or even silence. Lastly, the concept of timbre is merely the differentiation of a person’s sound from another person’s sound. Generally, music can be soft, grind, dry, warm or more, like what makes a sound differs. For instance, the timbre of a guitar is different from the timbre of a piano. This is because guitar put music in a sensation of strings while piano has a completely different feeling from that.

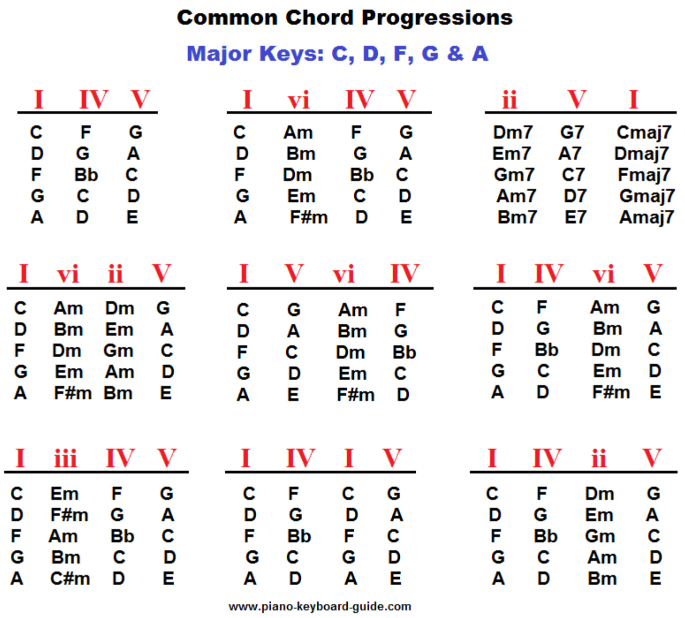
1.2 Musical Instruments

Music can be made by various types of instruments. For instance, preliminary instruments that was invented in the early days are flutes, zithers, drums and so on. The evolution of music is stunning to be acknowledged by the people in this generation. Cultures and traditions had utilized music in cultural ceremonies, religious celebrations and rituals. After that, music began to evolve towards rock and pop. Modern instruments include classical guitar, electric guitar, piano, keyboard, violin, percussion, saxophone, clarinet, trumpet, modern drums and so on. The three most popular instruments that is being widely used in today's generation would be guitar, piano and drum. Guitar and piano are used to create the melody, harmony, and tone whereas drum is used to create beats and rhythms. The combination of these three instruments, if put together correctly, will be adequate to make a marvelous piece of music (Rani. A, 2017).

1.3 Chords and Progressions

In the process of putting tones and sounds in an order, the progression of the chords must be emphasized. Generally, chords are divided into major and minor. The phrase major and minor indicates big and small. Therefore, the interval of a major third is somewhat more spacious than a minor third. Typically, the 2nd, 3rd, 6th and 7th intervals are either major or minor. A major chord is used as the root chord when the composer wants his or her music to sound more peaceful and happier. While on the other hand, a minor chord might sound dingy, sad, and is appropriate with emotional feelings in a certain way. However, others refer to the constructions of the music that consist a major third interval or a minor third interval to choose between major and minor (Ben, N.A).

To choose a chord progression, first thing to do is to choose the key to write in. C major, G major, A minor and E minor are good keys to start with (Ben, N.A). Next, work on the primary chords, which are the first, fourth and the fifth chords to construct the chord progressions. A good practice is to start and end the chord progression on the first chord. There are 4 common progressions that are widely used by musicians to compose music. Progression 1 - I, IV, V; progression 2 - I, V, VI, IV ; progression 3 - I, IV, I, V, I; progression 4 - I, VI, II, V (Ben, N.A).



Common Chord Progressions

Major Keys: C, D, F, G & A

I	IV	V
C	F	G
D	G	A
F	Bb	C
G	C	D
A	D	E

I	vi	IV	V
C	Am	F	G
D	Bm	G	A
F	Dm	Bb	C
G	Em	C	D
A	F#m	D	E

ii	V	I
Dm7	G7	Cmaj7
Em7	A7	Dmaj7
Gm7	C7	Fmaj7
Am7	D7	Gmaj7
Bm7	E7	Amaj7

I	vi	ii	V
C	Am	Dm	G
D	Bm	Em	A
F	Dm	Gm	C
G	Em	Am	D
A	F#m	Bm	E

I	V	vi	IV
C	G	Am	F
D	A	Bm	G
F	C	Dm	Bb
G	D	Em	C
A	E	F#m	D

I	IV	vi	V
C	F	Am	G
D	G	Bm	A
F	Bb	Dm	C
G	C	Em	D
A	D	F#m	E

I	iii	IV	V
C	Em	F	G
D	F#m	G	A
F	Am	Bb	C
G	Bm	C	D
A	C#m	D	E

I	IV	I	V
C	F	C	G
D	G	D	A
F	Bb	F	C
G	C	G	D
A	D	A	E

I	IV	ii	V
C	F	Dm	G
D	G	Em	A
F	Bb	Gm	C
G	C	Am	D
A	D	Bm	E

www.piano-keyboard-guide.com

Figure 1.0 chord progression chart

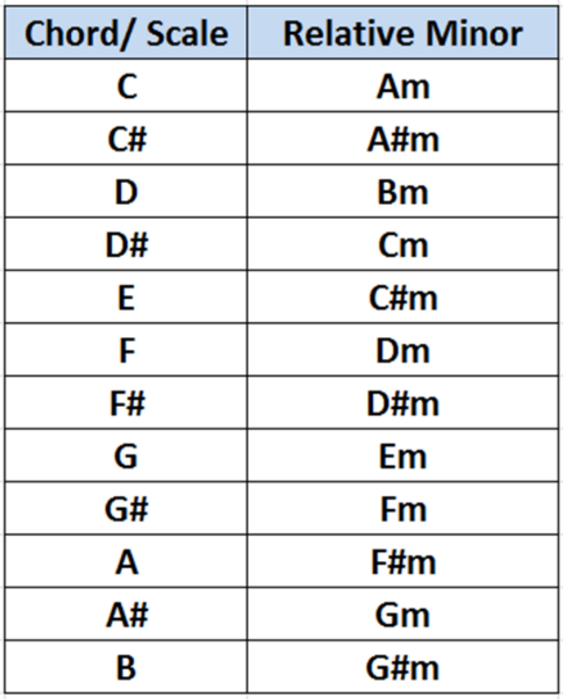
1.4 Scales and Key

A scale is a class of pitches, which is known as scale degrees organized in ascending order. Most of the time, the first and the last note are the steadiest and the easiest to find note. They are known as tonic (Hollis. B, N.A).

In the major scale, there are 7 distinguish pitches. Basically, the 3rd and 4th, and the 7th and 8th notes of the major scale are half steps of its scale levels while the other steps are one whole steps. However, in the minor scale there also exists 7 distinguish pitches (Hollis. B, N.A). Half steps exist between the 2nd and the 3rd, and the 5th and the 6th degrees while the other steps are also one whole steps.

In addition to this, scale patterns can be replicated at the pitch of our choice. The same scale patterns can be rephrased at a distinguish pitch by transposing it (Hollis. B, N.A). Transposition refers to increasing or decreasing a song's key to match certain pitch. Normally, singers would adjust their keys when they sing. The music instrument that is the most convenient to transpose key is the keyboard. People who plays the keyboard can easily transpose the keys to match a certain key. However, guitar on the other hand, when a song is played on capo 0 and the singer wants the song to be transposed to -2, the instrument player has to shift every single chord to lower two half steps.

A song can be in a major or a minor key depending on the scale. If the song is in a major key, then it is based on a major scale whereas if the song is in a minor key, then it must be based on a minor scale (Pouska. A, N.A). For example, a song that is played in the key of G major whirl around the seven notes of the G major scale - G, A, B, C, D, E and F where between B and C, is a half-step. On the other hand, a song that is played in the key of D minor utilizes the notes of the D minor scale - D, E, F, G, A, Bb and C where between E and F is half step. These seven notes are used to make up the chords, melody and also the bassline.



Chord/ Scale	Relative Minor
C	Am
C#	A#m
D	Bm
D#	Cm
E	C#m
F	Dm
F#	D#m
G	Em
G#	Fm
A	F#m
A#	Gm
B	G#m

Figure 2.0 relative minor scale chart

According to figure 2.0 as shown above, the chord of sixth degree of major chord is called the relative minor chord. Normally, musicians compose songs using a relative minor with a major scale. This composing method is well known in the music industry as it is able to generate a pleasing piece of music when played together.

1.5 Problems

Time consuming and expensive

Composing music has been time consuming and expensive since ever. Most of the composers struggles to write music and songs in a quicker and cost-efficient way. Therefore, developers have been doing research on this topic and have been making systems and applications to ease the work of a composer and making it as cost-free as possible. It is considered possible after seeing several apps that are successful in aiding music composers to do their job easier and with a lower cost. However, this issue has become a concern in this project. This project aims to develop a more efficient and quicker system with several new functionalities that can really help and improve the composing efficiency of a music composer.

Confused with complex design

The lack of knowledge or experience in utilizing GUI efficiently and in a balanced manner is critical to the nature of a system. Too simple user interface may cause user to lose satisfaction or desire to user the system, while too complex user interface with too many designs or colors may confuse user when navigating throughout the system. Hence, it is very important that a system adheres a balanced user interface with not much and not less design and navigation, but a balanced user interface.

Lack of creativity of outputs

First of all, the problem faced with previous music applications is the lack of creativity of their outputs. To make this clearer, the applications are incapable of extemporizing or improvising melodies in a more innovative way. With the help of JFUGUE and Harmony Search Algorithm, this project suggests a whole new possible way to improve the issues.

Copyright infringement

Nevertheless, one of the main issues in generating music that must be considered is the copyright infringement of music. Normally, musicians or composers are required to examine whether or not the generated piece of music or melodies have existed before in the music industry. If the generated piece of music or melodies have existed or is similar to other's production in the music industry, the composer needs to change or amend his or her production as to be a piece of unique and distinguished music. This problem was very serious because of the fast-paced advancement of music technology in the music industry. To overcome this problem, Evoluthm helps in generating new piece of melody randomly that can possibly help the composer in getting new ideas for composing a new piece of music.

Dissonance

Moreover, dissonance is also a severe problem in the process of generating a piece of music. Dissonance is created when 2 discordant notes are played together. In other words, it is the inadequate of harmony in music. When the 2 notes are played together, a dissonant sound is produced. Dissonance sound is typically unpleasant to people, but it eventually aids to generate stress and a sense of flow in the process of composing music. Music composers have been facing the conundrum of choosing whether to accept or reject the dissonance for the past few decades.

The problems mentioned above are crucial and are the keys in producing a perfect application. After analyzing and providing solutions for the above problems, the system should be immaculately built with higher efficiency and intelligence as to be more innovative in generating melodies.

1.6 Background

The title of my project is Evolutionary Music; chord progressions and beat generator. It is about making an offline system named Evoluthm that is capable of generating progressions with functions like music loop and beat style. In addition, to explain how to use the system, first of all, the users are required to choose the instruments they want their songs to be played in. Then, they are required to choose the desired chords and the system will automatically generate a random chord progression based on the major or minor scale to produce a music loop. After determining the chords based on the scale, the users can choose to transpose the keys of the songs of their choices.

Nevertheless, after the chord progressions are produced with determined keys, the users can choose the rhythm of the songs. The users can set the rhythm of their songs to 1/2/3/4, which indicates the beats of each chords are being played in a loop. Apart from that, the users can also choose the speed of the songs by adjusting the tempo which is slide-bar adjustable. Lastly, user can add in the drum instrument after setting up the rhythm to finish up the song.

The system will be developed using JAVA programming language and database. Additionally, the JFUGUE API and Harmony Search Algorithm source code for JAVA are also used to develop the system. In order to successfully develop a flawless system, the analysis design phase, development phase, implementation phase and evaluation phase will be effectively gone through. In addition to that, Gantt chart and Timeline are also brought into use in the project as to keep track of the progress.

The output of this project will be a user interactive random chord progression generator with music loop and keys selections. This project utilizes and combines the IT technology with music innovations and as a result of combining these fields together, producing a new era of technology, a technology aid music intellectual that helps musicians in song construction and music tutorial. The system is an innovative system that is capable of generating chord progression based on user requirements. The graphical user interface and functionalities of the system will be properly and adequately implemented to provide a basic yet satisfying experience to the user.

In a nutshell, the system is a JAVA-developed offline system that is capable of generating random chord progressions with supplementary functions like music looping to walk through music composers. Moreover, it also generates chords based on users selected keys with major or minor scales of their predilection with drums mingling to produce rhythms with predefined tempos to the songs.

Chapter 2: Literature Review

2.1 GarageBand

GarageBand is an Apple-developed application directed by Dr. Gerhard Lengeling and it is optimized by macOS and IOS. Dr. Lengeling worked in a German company Emagic makers of Logic Audio before Apple acquired Emagic on July year 2002. The development of GarageBand was arduous and challenging. Soon, Apple realizes the significance of improving GarageBand with more functionalities that can ease the way composer compose music. Thus, over the years, GarageBand has been updating to catch up with other competitors.

At the year 2005, Apple announced GarageBand 2 with added features such as the capabilities to view and edit music in musical notation. Also, to fix timing and pitch of recordings, it is also feasible to record up to eight tracks on one occasion. Apart from that, an automation of track pan position, master volume, and the master pitch was added to the application along with the ability to transpose between audio and MIDI and the import feature that can import MIDI files.

At the year of 2006, GarageBand 3 was introduced with little more functionalities. GarageBand 3 includes a podcast audio with the capability to utilize more than 200 ramifications and incorporation with iChat that enables long-distance meetings.

Not long after that, GarageBand 4 was introduced and it is known as GarageBand ’08, which means that it is a segment of iLife ’08. It integrates the capability to record songs in segments. Besides, the configuration of tempos and instruments is made available, and the formation and transformation to iPhone ringtones. Lastly, a “Magic GarageBand” function, that comprises a three-dimensional view of electric instruments that is aided by an effective jam session.

Next, GarageBand 5 was released, and is segment of the iLife ’09 collection. GarageBand 5 now includes tutorial videos made by artists that are purchasable. Also, new features are released that benefits electric guitar players, the three-dimensional electric guitar track comprising a virtual stomp box pedalboard, and virtual amplifiers with spring reverb and tremolo. Besides, a new GUI and assignment templates that brings more values to the users.

After that, GarageBand 6 was released on 20th of October, 2010, and is segment of iLife '11 collection. New functions like Flex Time, instrument that enables the configuration of beat of a recording, and the capability to synchronize the tempo of song with another immediately, are released. Additionally, amps for guitar and stomp boxes, twenty-two contemporary tutorials for guitar and piano, and "How Did I Play?", which is an instrument to calculate the correctness and advancement of a showing by piano and guitar in a tutorial.

A huge leap occurred after this, Apple introduced GarageBand 10 on October year 2013, that is optimized for the OSX 10.9 Mavericks version. They have decided to forgo Magic GarageBand and the podcast feature. Then, Apple updated GarageBand 10 again to the version 10.0.2 on March 20, 2014. This edition allows the user to export songs in MP3 format and added a new module for drummer. However, since the podcast function is removed, users can still edit them using the old version with the productions established in GarageBand 6. Next, version 10.0.3 was released on October 16, 2014. This update emphasizes on the bugs fixes and some new functions such as a steadfast Bass Amp Designer, the establishment of worldwide track ramifications and energetic song resizing. Lastly, with the final version Garage band 10.2 released on June 5, 2017.

In addition, GarageBand realizes the importance of looping in music application, which is it can solve the tardiness of music composition and enhances the effectiveness of a music composer in the process of building a song. Therefore, they dedicate themselves to add in the feature of audio loop into the application. To do this, GarageBand comprises a large-scale array of pre-defined audio loops to select from along with a choice to import specific audio loops and a supplementary loop package that can be purchased in the App Store. Every loop can be edited with multiple effects choice. The system allows users to develop various tracks with the pre-made MIDI keyboards and loops. Moreover, with the combination of various musical instruments such as keyboard, guitar, strings, drums, bass, world and other external instruments, with appliances like AMP, which stands for amplifier, and audio recorder, enables the tracks to be compiled with multiple effects and as a result a beautiful piece of music perfectly compiled together.

However, Apple did not test the compatibility of their latest MacOS software with the application before completely releasing it to the public (Blache. B, N.A). Many users from around the world had discovered bugs and issues with the application after having their Mac updated to the latest version. Some claimed that they had have issues with dragging, moving, copying and pasting loops within a track (drnbdnrk, 2016). Moreover, after installing the latest software update, some of the loops within a track became locked or glitchy and the users have to restart GarageBand again for it to function correctly (drnbdnrk, 2016). Apparently, Apple does have a habit of releasing buggy software (Blache. B, N.A). Thus, users are doomed to adapt to a temporary way to resolve the issues until Apple release newer version that fixes the bugs.

2.2 Chord Progression Master - By Genres

Chord Progression Master is a free android music application developed and offered by KekStudio on July 31, 2017. This application is still considerably new in the Google Play Store and has already gained many good reviews and ratings. It is an incredible composing tool that aids musicians in writing music. The application prompts musicians with new ideas to develop new music. User can choose keys and their practice and it will generate pleasing chord progressions. This application can provide the user with a new set of ideas of writing their songs with many different varieties of chord progressions. The chord progressions are generated from music foundation. Also, there are different kinds of genres that user can choose from such as Blues, Jazz, Rock, Pop, and many more.

There are several amazing features that this Chord Progression Master offers to their users. First of all, it provides simple chord progressions that are pleasing to the ears. Next, users are allowed to choose the keys and the degree of convolution and generate a chord progression that matches the criteria. Moreover, moods also play a huge part in composing a song. Users can choose the moods that they want their songs to be in, sad, simple or happy. The application is designed for easy-use purpose so that it can delivers the outputs to users at the quickest way. Apart from that, users can also choose the music instrument such as piano and guitar. There are guitar and piano chord charts to make up the chord progressions.

The reviews said it all, there are many good and bad reviews commented on this application. Many users commented on the review section saying that the app is very helpful. The good reviews it got is that it helps to brush up music theories, make songs without expert trainings, good interface design and many more (Jensen. D, 2018). The application is successful in attracting people closer to music and understand more about it (Samuel. O, 2018).

However, some people do not like the recent update of the application. Bernard Ameyaw said that the latest upgrade is lame because of the visuals of the chord notes appeared to be lousier compared to previous version as the previous version helped in finger movements on a static keyboard. After the update, the keyboard would shift in position, which complicates the user. Moreover, the updates also include new chord progressions that requires a pay-to-upgrade pass in order to access them. Now user feels more restricted from better opportunities to write a better song as the previous version was completely free of charge. In other words, this means that people that are poor in financial states cannot afford the charge are restricted from writing a better piece of music from the app. Speaking of revenue, the application is already making revenue based on advertisement.

Nevertheless, a regular user commented that after the update, the progression halts when he closes the app or goes into lock screen. In addition, it became more confusing with the major and minor in the same wall (Tandon. D, 2018). Before the update, the major and minor sorting could fit on one screen. After the update, the user is required to scroll left and right to view everything. Additionally, there is no auto-scroll when the progression is playing, which means that it became less accommodating when playing alongside it and the current active chord that is being played is hindered. For these reasons, the ergonomics of this application are considered bad.

In addition, many disliked the agreements because of its intrusive permissions. The application wants to have access to user's apps, Facebook account, email and contact list (Nicolette. L, 2018). Although there is another reviewer saying that the user can bypass this and still have access to this application, but the fact that there is no notice to the user stating that there is an option to bypass this. Some user claimed that the developer may have good intentions but reasonable thinking the user might still assume that the developer comes with bad intentions and that the data collected might be sold to a spam list (Gaskin. B, 2018). Others even claim that the application is not trustworthy and reliable because of the sensitive data that it is requesting access for.

To resolve the many issues mentioned above, the developer should carefully examine what user likes before taking that away which in turns causing unsatisfactory and unpleasant experience to the user. Moreover, instead of providing an option to bypass this step, the developer should reconsider their option to completely remove this permission. This intrusive permission not only take away the number of users, it also brings down the value and integrity that user has upon the application. Hence, since the data collected is considered unnecessary, the risk is unworthy.

2.3 Suggester

Suggester is a free application that is available to download in the Apple App Store. It is developed and designed by Mathieu Routhier. Suggester is an instrument to aid user in the development of music and chord progressions and search for chords that work well together (Routhier. M, 2017). User can hear how a chord sounds with just a touch on the play button. Moreover, the user can also tune the playback tempo. User can choose either to work forward or backward (Routhier. M, 2017). If the users choose to work forward, they can develop the songs from the suggested chords from the app based on selected scale. This is the most efficient and fastest way to combine matched chords. Whereas if the users choose to work backward, they will be required to select a set of chords that they prefer from the catalog and the app will suggests which scales it is compatible with. After the users have chosen the scale, the app can determine the appropriate chords that matches with the ones they entered before, which is quite helpful when jamming with friends.

After the editing, user can export their production by email using either a MIDI file or plain text. The MIDI file can then be opened using other application. Each sound made is sent to the MIDI output, which can be configured in the options. The notes can be directed to other application, an external MIDI port, and other computer that is located far away using MIDI Network Session. Chords can be stimulated via the MIDI input (Routhier. M, 2017). The music instruments included in Suggester is piano, electric piano, strings and guitar. Additionally, there are a huge amount of scale types and chord types available in the app.

Suggester is praised with its simplicity and ability to draft out chord progressions at any occasions. In other words, it is very convenient to be used. This application emphasizes on creation, neglecting the importance of music theories (Lindsey. C, 2018). Music theory is a foundation that every musician and artist must have to be successful yet lucrative. However, users who are not interested in music theories will find this application more attractive as to its process and features.

Apart from that, the application restricts user from accessing to every features available. User has to pay an amount of $7.99 to reach full functionalities. Moreover, the GUI of this application is rather unattractive and the font used is also unpleasant to the users. The developers are highly suggested to improve on the GUI and add in more functionalities for higher user involvement.

Table below illustrates the comparison among all the similar system with Evoluthm.

Features	Change Instrument	Change rhythm/tempo	Change pitch	Add layer	Useful without instrument	Open/export midi file
Evoluthm	√	√	√	√	√	√
Garage Band	√	√	√	√	√	√
Chord Progression Master	√	√			√	√
Suggester	√	√			√	√

Table 2.0 Features Comparison

Chapter 3: Project Scope and Objectives

The deliverable of the project is a piece of software named Evoluthm, which comes from the project title "Evolutionary Music; chord progressions and beat generator". The title is very clear and specific that it inspired me to name the system "Evoluthm". Evoluthm is a system that combines the best functions available in the five systems or application being reviewed. However, the proposing system would not be as complicated as some of reviewed systems, instead it will be emphasizing on the simplicity, user-friendliness, and the implementation of very useful functionalities only. Upon reviewing the five similar applications, I have gained some useful ideas and general concept on how to design my proposed system in a way that can solve all the disadvantages of the reviewed system. Although these applications are designed to solve the problems musicians and songwriters have been experiencing in the past decades, which are namely the **long duration required to compose a song**, **lack of creativity**, and the **cost of composing a song is rather expensive**, but it also caused other problems with the user interface design, the functionalities, the complexity and the user friendliness. I am looking forward to work on these issues and improve them and implement it in my proposed system.

Free and convenient

Evoluthm would be my proposed system that addresses all the issues mentioned above. First of all, Evoluthm aids its user in conveniently develop songs and music at any occasion in an arguably short duration of time. Also, composing a song no longer requires the assists from actual music assets. The proposed system would be able do almost anything needed to write a song or even more. Evoluthm can replace the old-fashioned way of writing songs, which is considerably time-consuming and expensive.

Attractive and balanced user interface

Moreover, I would propose some ideas of designing the user interface in a way that is pleasant and attractive to the eyes of the users. One of the ideas is using constant colors and designs in the interface so that the user would not feel as they were directed to another unrelated sources or links. Apart from that, the interface must be kept simple. To do so, I would use common user interface elements, input controls, navigational and informational components and containers to avoid user from confusion when navigating and accessing to information in the system (James. J, N.A).

Next, the basic functionalities that a music application compulsory to have must be available to use without additional charges. In Evoluthm, the main functionalities that must be included are namely the music layers, random melody generator, transposition of pitch and selection of various types of music instruments. These are the core functionalities that are crucial in aiding musicians and songwriters to compose their songs. Besides, the complexity is mostly caused by the many functions organized on one interface. For instance, the most complex application goes to GarageBand due to the huge number of functionalities it offers to its user. The navigational control button in GarageBand is represented by symbols or icons. In my opinion, the symbols or icons used is not as representative as to the intention of the navigation button. It is rather confusing to the user when navigating throughout the application.

Inspires Musician

Besides that, Evoluthm can be the solution to the lack of creativity issue. With the random melody generator, user can get inspirations on writing their songs in a more creative manner. User can try generating multiple pieces of melody and choose which one they like to proceed with different music instruments. Lastly, I would propose to make Evoluthm a user-friendly application. Evoluthm will be developed with more interactional components that enhances user experiences. For instance, popping out a notice to guide user on how to use the system upon first use.

Generate new piece of music

Referring to the problems that most of musician encounters when composing music, many musicians tends to include catchy melody produced by other composer. This is a commonly known issue where composers try to include a small part of other composer's work into their own. During the seeking of inspiration of a music, most of musicians tends to unintentionally brain washed by familiar songs with similar chord progressions and rhythm. Thus, in this case, Evoluthm brings into good use by randomly generating unique melody that inspires musicians to write their own original piece.

The main objective of this project is to provide an easier and more convenient access to music to support musicians and songwriters in effectively producing their works. This can be divided into sub-objectives of the project. For instance, user can get to know more knowledge about music and gain interest in music in a whole new way. This project will contribute to the music industry and benefits people who might have hidden talents and hence make a good use of the talent. Moreover, this project also innovates a new way to generate melody and pace that can plays sounds well together.

Table below illustrates which objectives solves the problems mentioned earlier.

OBJECTIVES	PROBLEMS
Free and convenient (compose without music instruments)	• Time consuming • Expensive • Low efficiency
Attractive and balanced user interface	• Confused with complex design • Lose satisfaction and desires to use the system
Inspires Musician	• Lack of creativity • Incapable of improvising
Generates new piece of music	• Copyright Infringement

Table 3.0 Objectives vs Problems

3.1 Diagrams

3.1.1 Use Case Diagram

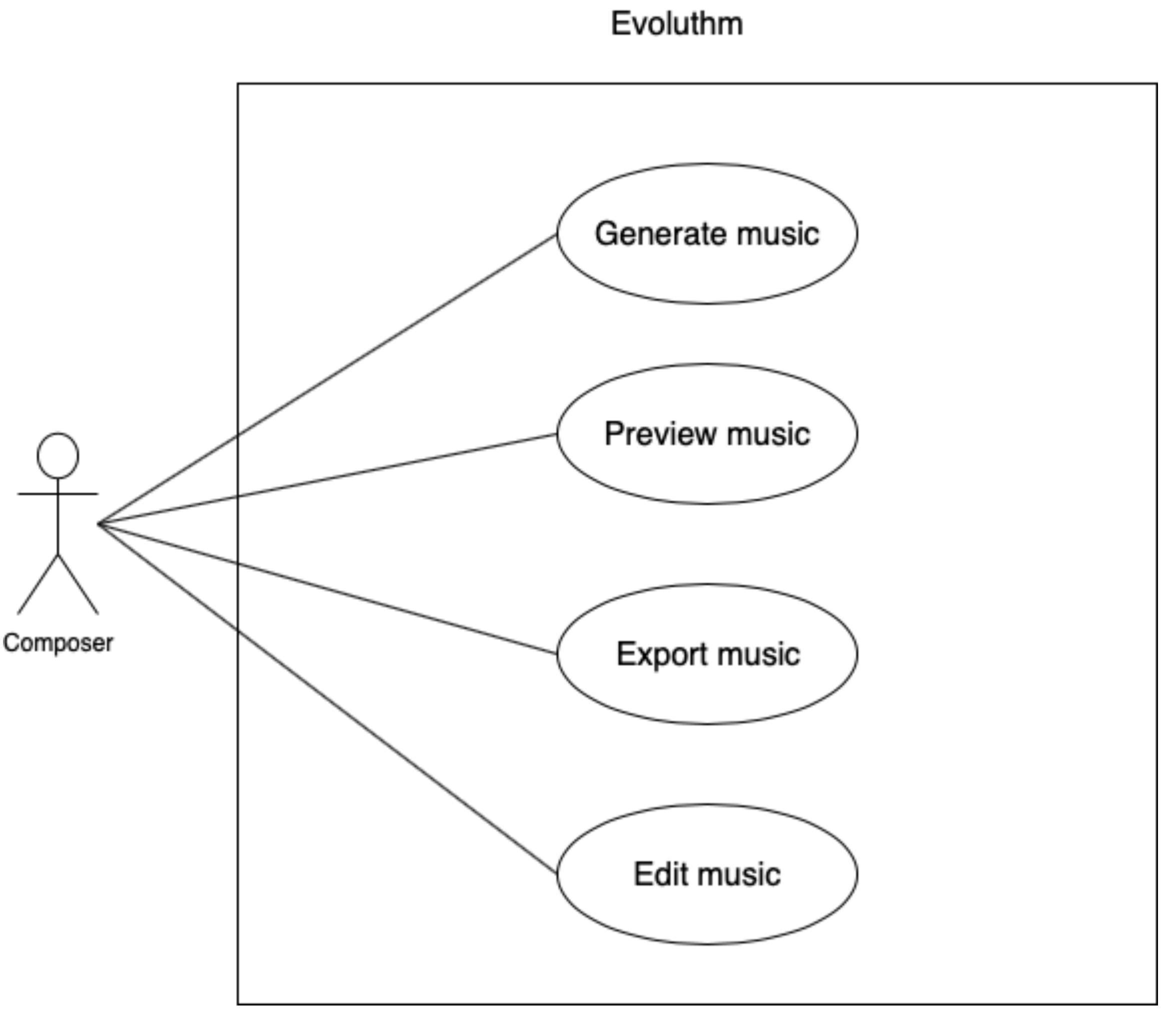


Figure 3.1.1 Use Case Diagram

Figure 3.1.1 above is a use case diagram constructed based on the 4 modules in the project.

3.1.2 Activity Diagram

One activity diagram is constructed for each use case. Hence, there will be 4 activity diagrams constructed to further elaborate on the activities in the use cases.

Use case: Generate Music

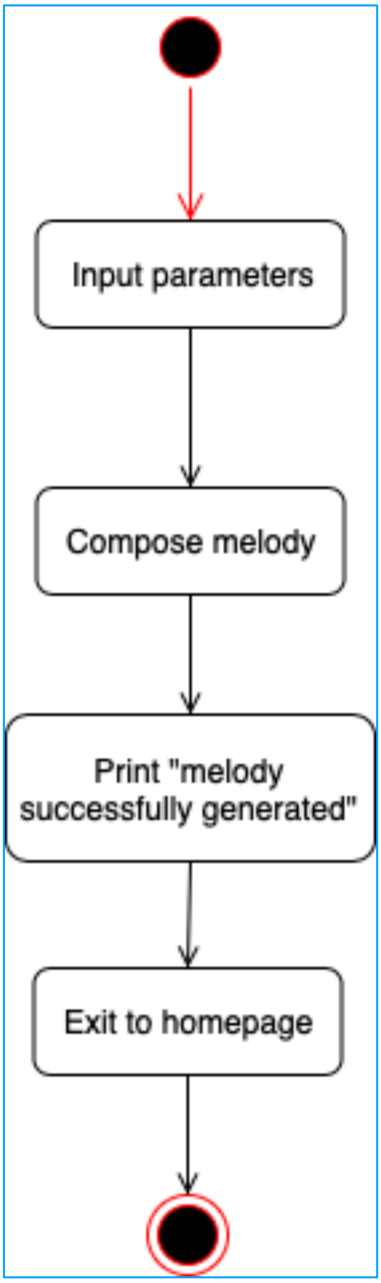


Figure 3.1.2 Activity Diagram for case GenerateMusic

Use case: Preview Music

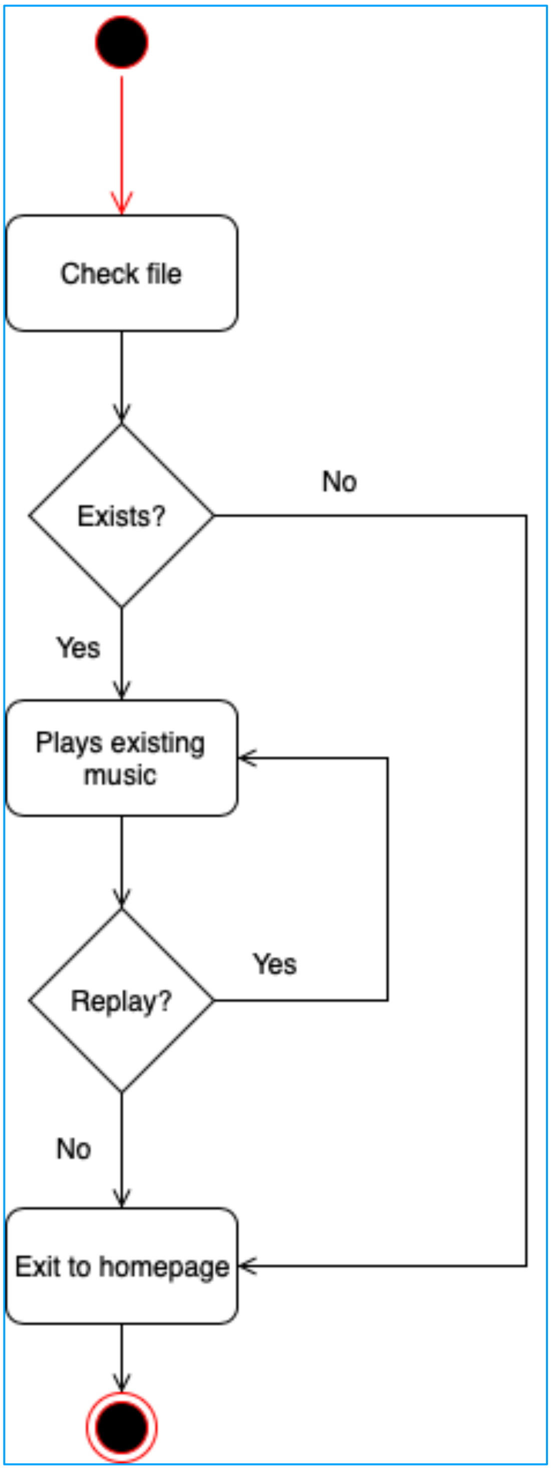


Figure 3.1.3 Activity Diagram for case PreviewMusic

Use case: Export Music

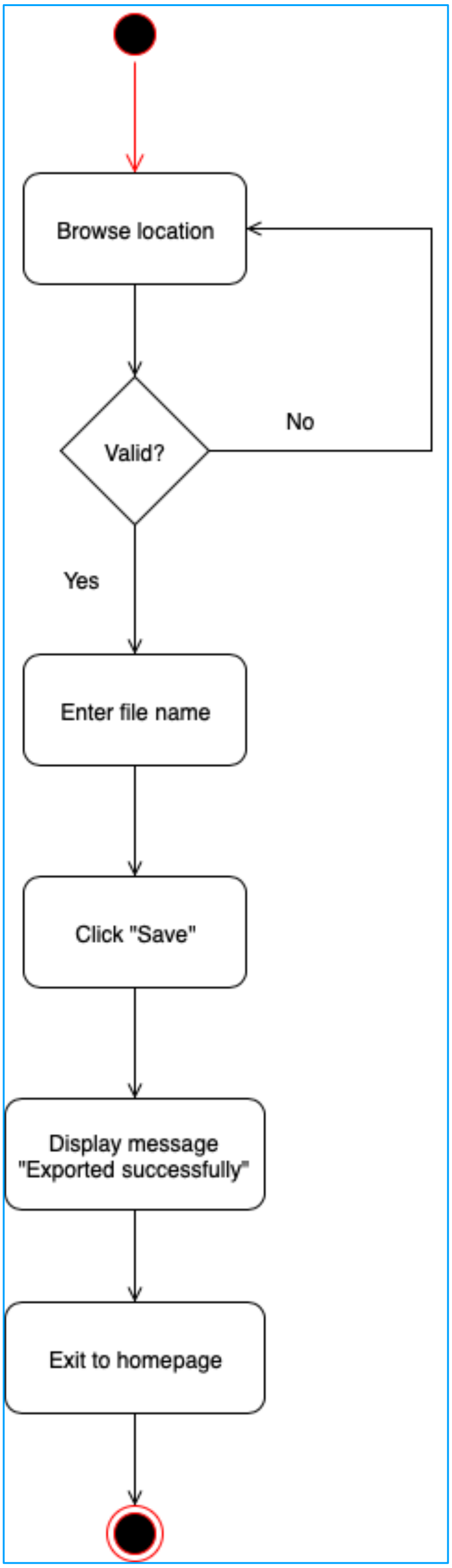


Figure 3.1.4 Activity Diagram for case ExportMusic

Use case: Edit Music

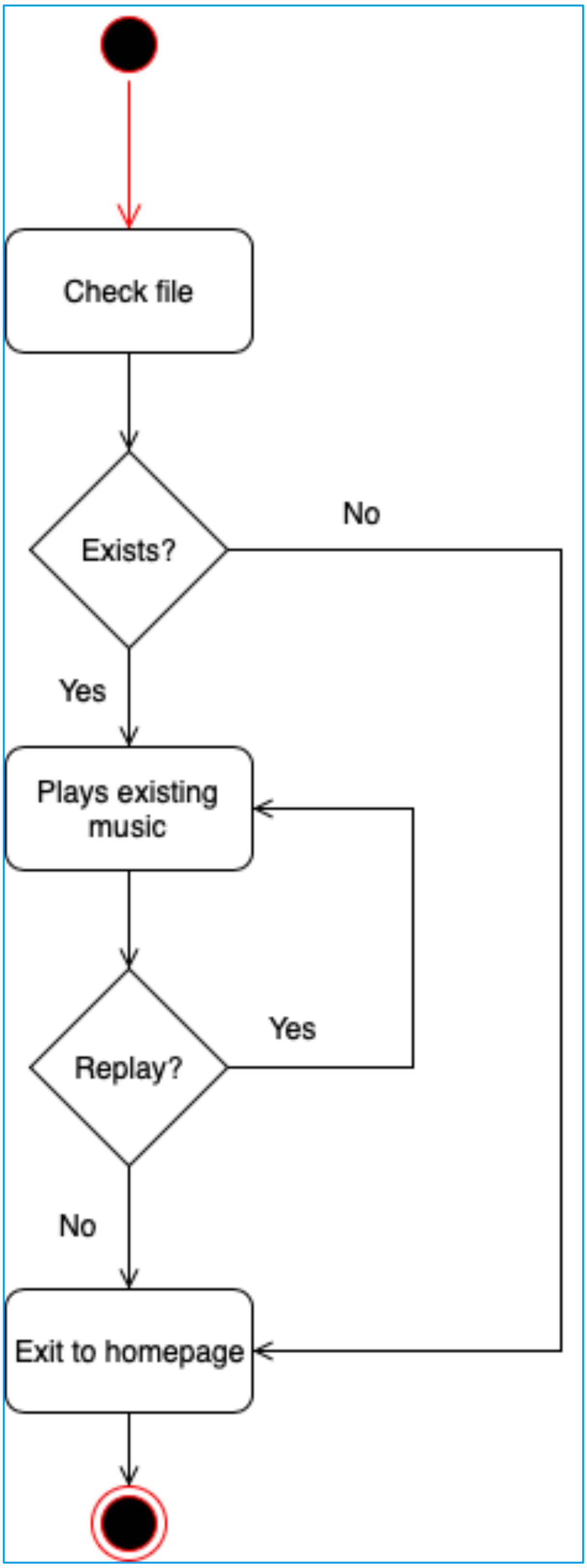


Figure 3.1.5 Activity Diagram for case EditMusic

3.1.3 Sequence Diagram

One sequence diagram is constructed for each use case to clarify the sequences and procedures in each module.

Use case: Generate Music

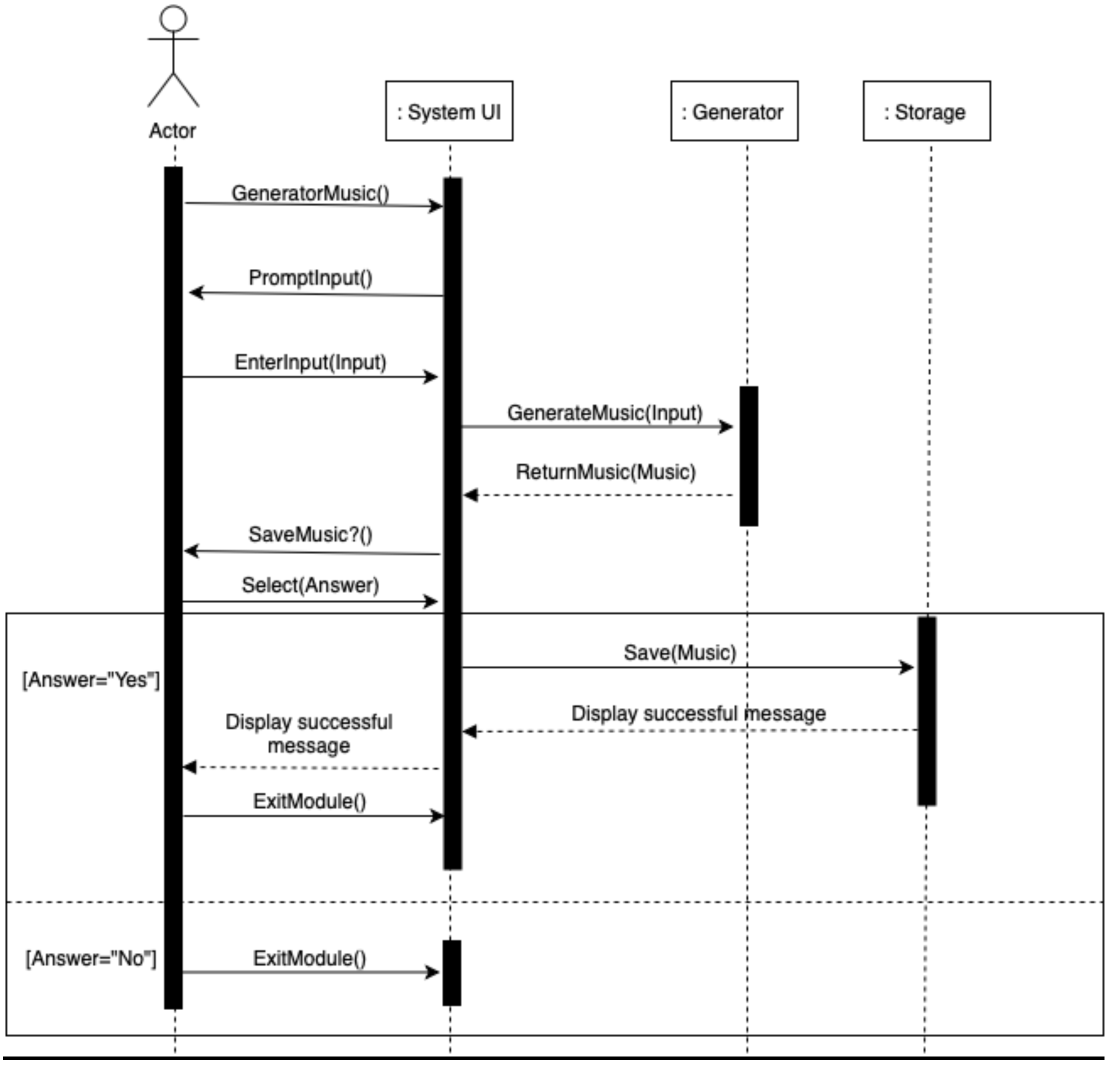


Figure 3.1.6 Sequence Diagram for case GenerateMusic

Use case: Preview Music

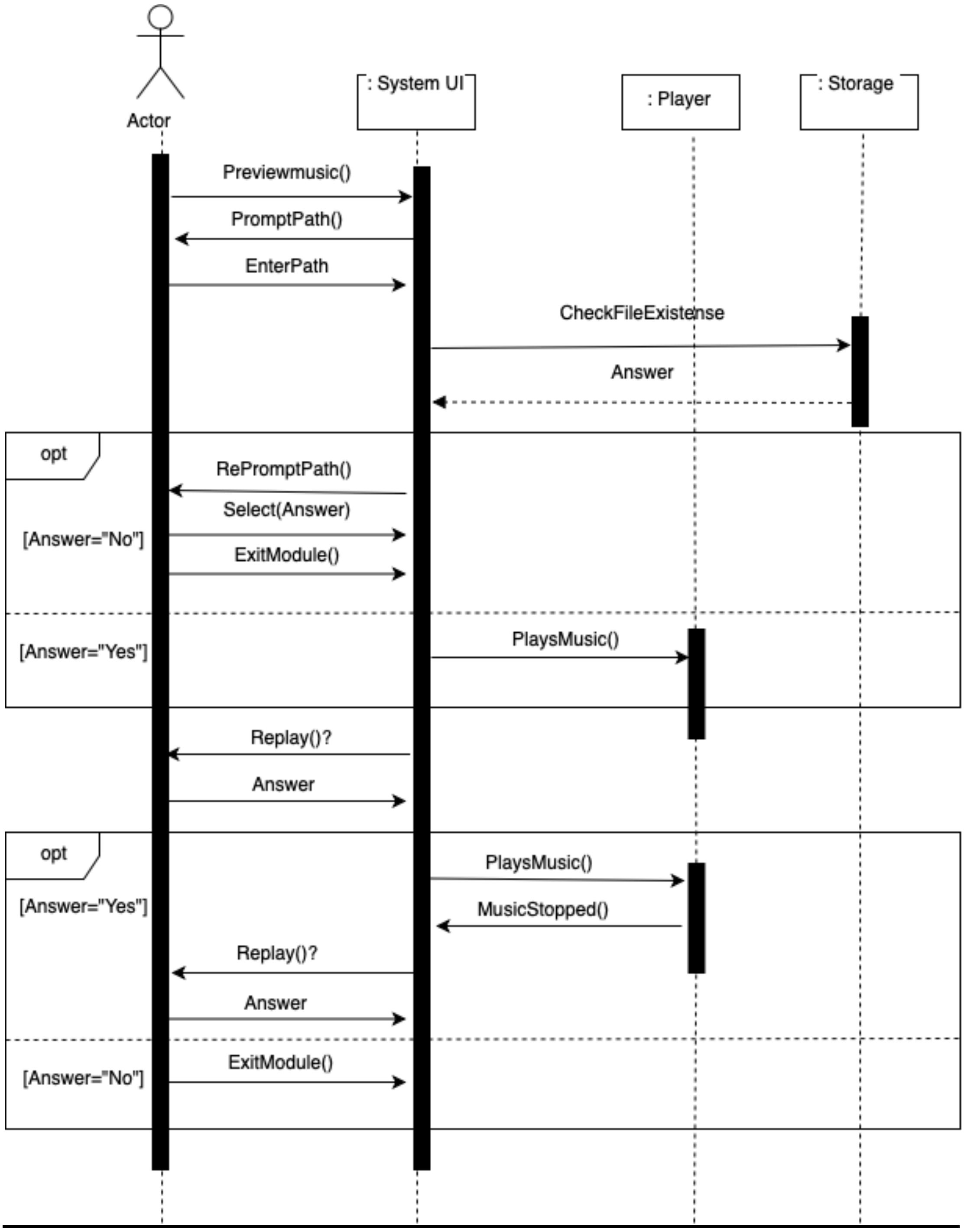


Figure 3.1.7 Sequence Diagram for case PreviewMusic

Use case: Export Music

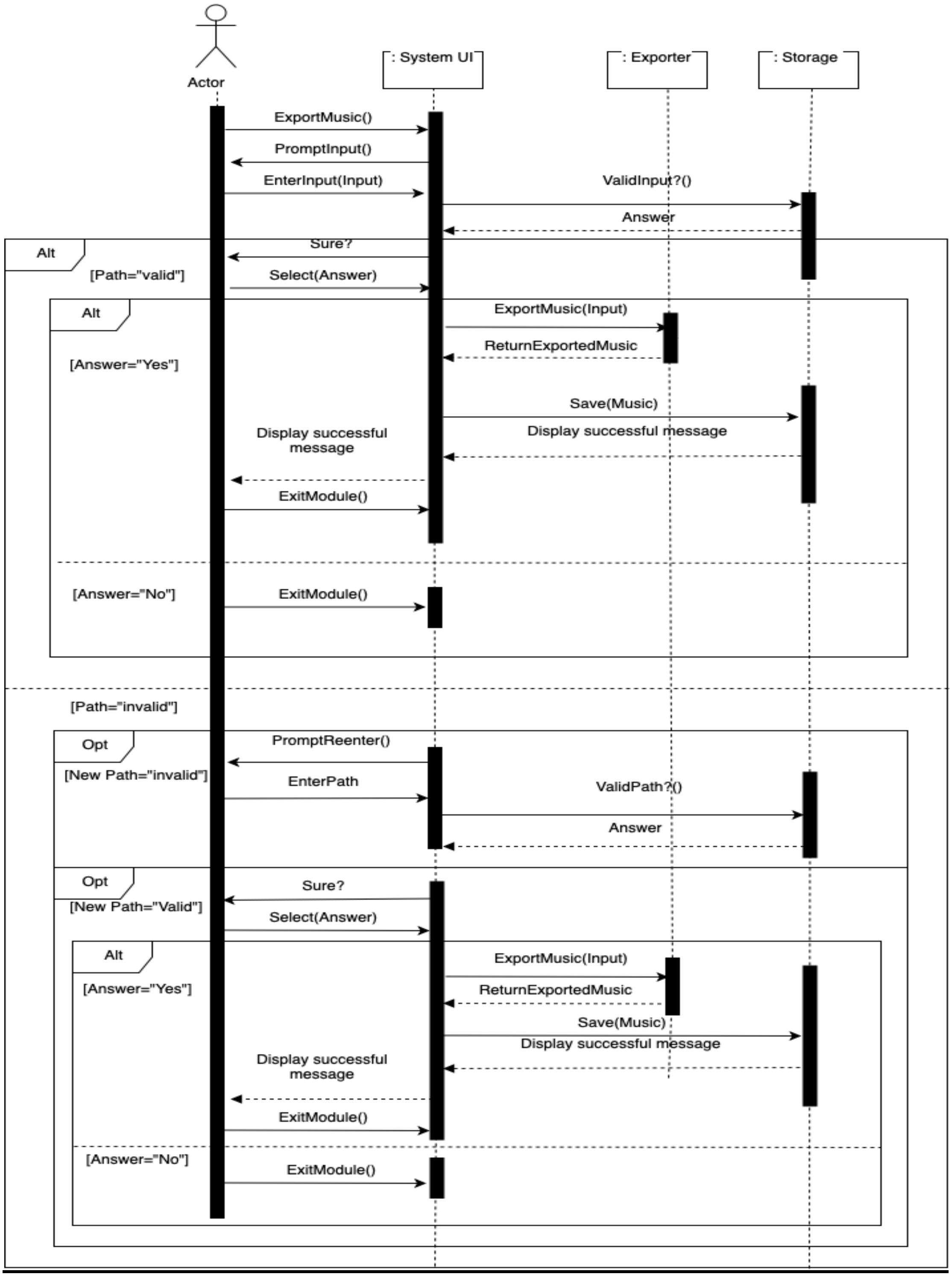


Figure 3.1.8 Sequence Diagram for case ExportMusic

Use case: Edit Music

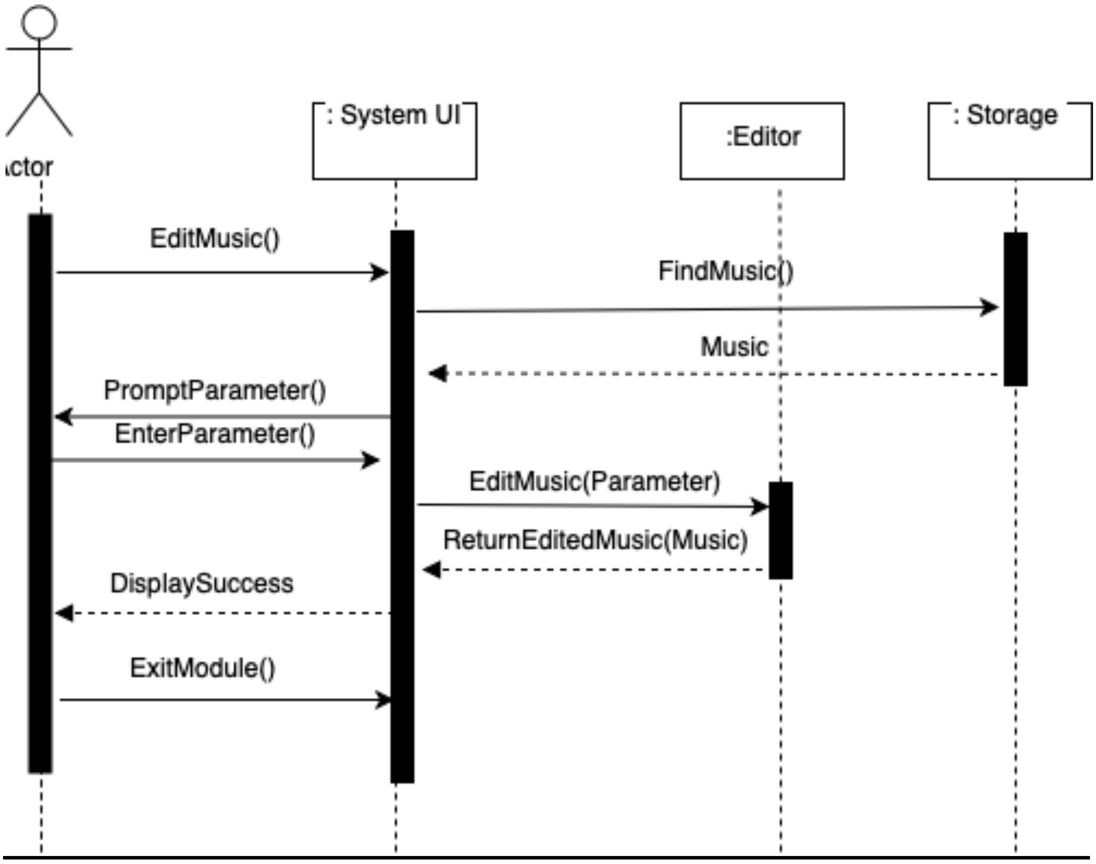


Figure 3.1.9 Sequence Diagram for case EditMusic

Chapter 4: Methods/Technologies Involved

The methodology that will be used to complete this project is by acquiring additional knowledge on JAVA programming language and utilizing API and algorithm such as JFUGUE and Harmony Search Algorithm. These technologies mentioned are the crucial ingredients in developing a perfect music system.

4.1 JFUGUE

JFUGUE is an open source API registered under Apache License that is meant to be implemented in JAVA (Koelle.D, 2008). JFUGUE is how programmers write programs that develop music. Music programming in the past has been harsh and frustrating (Koelle.D, 2008). Playing a middle C requires a huge amount of coding works to accomplish the task. However, with the help of JFUGUE, creating music in a program has never became easier. 20 lines of coding can now be shorten to 2. For instance, a simple coding like player.play("C D E F G A B"); is already adequate to define music.

JFUGUE provides many amazing features that takes programming in music to another level. User can specify classes such as microtonal music, beats, accessing external devices and more. It is also possible to interact with other music equipments and formats. User can easily read or write input from MIDI, MusicXML and more. Almost every music aspects had been thoroughly realized by JFUGUE. JFUGUE makes it possible to create a system capable of generating random chord progressions, beats and keys.

4.2 Harmony Search Algorithm

As for Harmony Search Algorithm, it is created by Geem in year 2001 and it is a population based metaheuristics algorithm motivated from the procedures in music that focuses on seeking for an ideal state of harmony (Ali. A, 2014). During the music extemporaneous activity, every sound must be pitched in a certain range together to develop a harmony. At every loop of Harmony Search, new harmonies are created and saved in Harmony Memory (HM). After that, the possibility of creating a perfect harmony will increase. Same goes with optimization, decision variables within the feasible scope will create the beginning guide indiscriminately. This is the general concept of how Harmony Search Algorithm works. With the perfect harmony search generated by this algorithm, this project can be more effective and accurately built. Figure 4.0 below shows the general process of Harmony Search Algorithm.

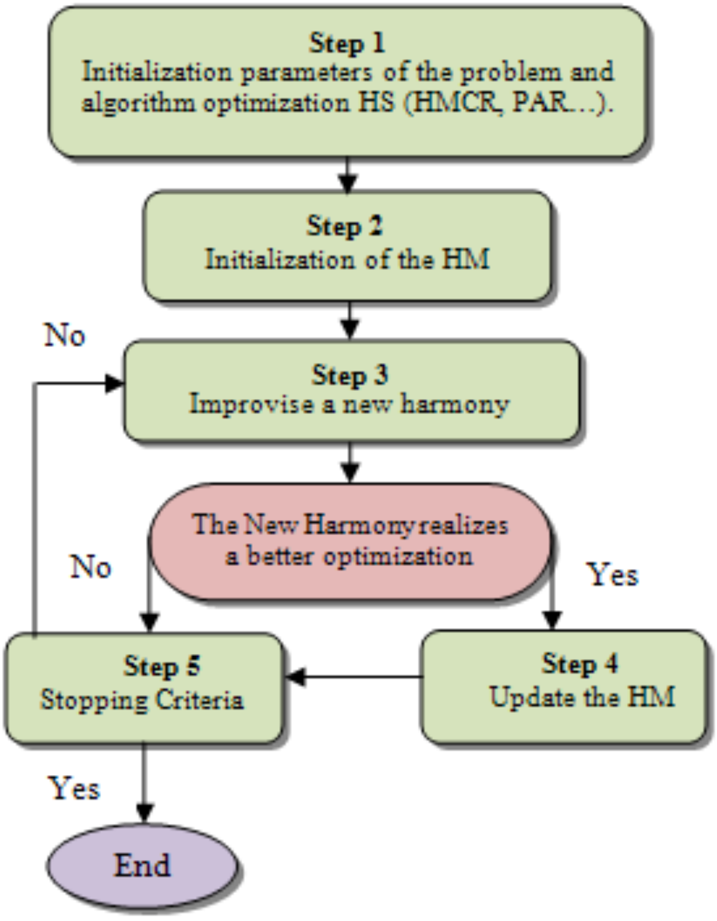

Step 1
Initialization parameters of the problem and algorithm optimization HS (HMCR, PAR…).
Step 2
Initialization of the HM
No
Step 3
Improvise a new harmony
The New Harmony realizes a better optimization
No
Yes
Step 5
Stopping Criteria
Step 4
Update the HM
Yes
End

Figure 4.0 Harmony Search Algorithm

Chapter 5: Implementation and Deployment

5.1 System Implementation

The implementation phase started off by acquiring or downloading all the related development tools. In this case, Netbeans IDE, a package that consists of JDK, JRE and Java Graphical User Interface Builder. Graphical User Interface Builder is significant in developing the system as it is required to improve the efficiency in building the system GUI. Apart from that, Netbeans IDE is also capable of checking compilation or coding error after compilation and suggestions for the related coding error will be given to save the programmer's time.

Next, JFugue and Java Music Programming API was downloaded and loaded into the Netbeans IDE. After the loading has completed, the music generator module revealed and was ready to use. Lastly, the implementation of Harmony Search Algorithm generated the melody generator module.

5.1.1 Harmony Search Algorithm procedures

Step 1	Initialize HSA and the optimization problem parameters. Ex: fitness function $f(x)$, HMCR, PAR, NI, HMS
Step 2	Based on Harmony Memory Size, randomly generate a number of decision variable.
Step 3	Set up uniform number (1 – 100) to determine the need to do Pitch Adjustment or an New Improvisation. Based on the Pitch Adjustment Rate and Harmony Memory Consideration Rate to determine the probability to select Pitch Adjustment or New Improvisation.
Step 4	If the new solution set has a greater fitness value than other solution sets of the harmony memory, the worse harmony memory will be replaced with the new solution set.
Step 5	The loop repeats step 3 to 5 until the harmony memory meet the stop criteria.

Table 5.1 HMA procedures

5.1.2 Implementation Steps

To implement the algorithm into the module, two most critical factors must be defined, which are decision variable and fitness function. The decision variable is the musical string notes whereas the fitness function determines the quality of the decision variables.

After the decision variable and the fitness function are defined, the initialization of harmony memory is involved. Initially, the harmony memory data structure was represented by arrays. However, array was eventually replaced by vector due to the fact that vector provides more flexibility that the algorithm requires. Vector allows new elements or dimension to be appended or rep3laced whereas array does not.
The fitness values for each possible solution set is computed and stored at the end of the last possible solution vector. Each loop will generate and replace the solution set from worse to better basing on the fitness value.

Next, a random vector between 1 – 100 are generated by using random number generator. Based on the Harmony Memory Consideration Rate (HMCR) and Pitch Adjustment Rate (PAR), decision is made between executing the Pitch Adjustment (PA) or creating a new solution set. As an example, the HMCR is set to 80 and PAR to 50, Pitch Adjustment will be executed if the random number falls between 1 – 50 and HMCR will be executed if the random number falls between 51 – 80, else if it falls between 81 – 100, then a new solution set will be generated.

The solution set will then be used to compute the fitness value. If the computed fitness value of one of the solution sets in the harmony memory is lesser than the fitness value of the new solution set, then the worse solution set will be updated with the new solution set. When the loop reaches its maximum loop count, the best solution set will be picked out to be the core music string.

In the final step, JFugue is used to render the final solution as a playable file. The solution is also stored as a text file so that the settings of the solution can later be reviewed and modified. After all the above functionalities is established, user interface was the next thing to work with. The process of designing and making the user interface work and friendly took quite an amount of time. After the finalization of the user interface, the whole system was completed.

5.2 System Installation

The development tools involved in implementing the system is JDK & JRE, Netbeans IDE and JFugue. All these tools are downloaded for free from their official websites. To allow Java program to run, JDK & JRE and Netbeans IDE must be installed. Moreover, Netbeans IDE enables more flexibility in designing the GUI for the system.

Chapter 6: System Testing

After the system is published, several testing steps are done to ensure the system is error-free and is working as intended. Unit testing, functional testing and integration testing are done to find the hidden errors in the system.

6.1 Unit Testing

Unit testing is a form of software testing to test each individual units of the system. The testing is separated in different units so that the errors can be narrowed down and more targeted. Unit testing escalate the confidence in modifying and maintaining code. The system unit testing is conducted and written down each time an unit testing is ran so that errors due to the modification can be captured and be amended to remove the newly introduced errors. In this way, errors can be fixed immediately when it occurs and this helps improves the reliability of the system. The testing results of each unit / component has been established and illustrated below.

Component 1 : Music Compose

Objective : To test if the music compose module works correctly

No	Case	Parameters / Value	Expected Result	Success / Fail
1	Input all correct value in all parameters	Tempo: 100 Pitch Range: 50, 80 Layers: 1 Notes Length: 35	Compose success	Success
2	Input some invalid values in parameters	Tempo: i100 Pitch Range: 50, 80 Layers: 2 Notes Length: 35	Error message telling the user that the data entered is invalid and prompt the user to enter the correct data	Success
3	Some parameters are left blank	Tempo: 100 Pitch Range: NULL, 80 Layers: 1 Notes Length: NULL	Error message telling the user that the data entered is invalid and prompt the user to enter the correct data	Success

Table 6.1.1: Component 1 (Music Compose)

Component 2 : Music Preview

Objective : To test if the music composed from component 1 can be previewed

No	Case	Parameters / Value	Expected Result	Success / Fail
1	Preview music	Music does not exists	Error message informing the user that there is no music exists yet and brings the user to the compose module	Success
2	Preview music	Music exists	The music is played	Success

Table 6.1.2: Component 2 (Music Preview)

Component 3 : Music Edit

Objective : To test if all parameters of the music can be edited

No	Case	Parameters / Value	Expected Result	Success / Fail
1	Tempo	Tempo: 170	Tempo changed	Success
2	Tempo	Tempo: 180jhy	Error message prompting the user to correct the value	Success
3	Tempo	Tempo: NULL	Error message prompting the user to correct the value	Success
4	Tempo	Music does not exists	Error message informing the user that there is no music exist yet	Success
5	Instrument	Instrument Type : Guitar	Instrument will change to Guitar	Success
6	Instrument	Instrument Type : Harmonica	Instrument will change to Harmonica	Success
7	Instrument	Instrument Type : Drawbar_Organ	Instrument will change to Drawbar Organ	Success
8	Instrument	Instrument Type : Church Organ	Instrument will change to Church Organ	Success
9	Instrument	Instrument Type : Violin	Instrument will change to Violin	Success
10	Instrument	Instrument Type : Cello	Instrument will change to Cello	Success

11	Instrument	Instrument Type : Flute	Instrument will change to Flute	Success
12	Instrument	Instrument Type : Piccolo	Instrument will change to Piccolo	Success

Table 6.1.3: Component 3 (Music Edit)

Component 4 : Export Music

Objective : To test if the music composed can be exported.

No	Case	Parameters / Value	Expected Result	Success / Fail
1	Export music	Music exists	Music exported successfully	Success
2	Export music when file does not exists	Music does not exists	Error message prompting the user to compose music before exporting	Success

Table 6.1.4: Component 4 : Music Export

Component 5 : 3rd party software (extra feature)

Objective : To test if the 3rd party software can be launched from the system

No	Case	Parameters / Value	Expected Result	Success / Fail
1	Open 3rd party software	Software exists	Software launch success	Success
2	Open 3rd party software	Software does not exists	Software launch fail	Success

Table 6.1.5: Component 5 (Extra feature)

6.2 Functional Testing

Functional Testing is performed right after Unit Testing is completed. The objective of functional testing is to validate the system against the functional specifications and test each and every function of the system. This is done by providing different inputs/parameters to verify the output of the system if it meets the expected results stated earlier. testing is done to ensure that the system is bug-free before the exact launching of a system. Bugs are usually revealed or exposed in this stage of testing because the system is put into actual use to identify the flaws of it. Therefore, functional testing is done to ensure that the system is bug-free before the exact launching of a system to assure quality.

Functional Testing on Component 1 : Music Composition

Objective : To test if music can be composed and work as intended with different values / parameters inserted into it

No	Case	Parameters / Value	Expected Result	Success / Fail
1	Compose a fast-paced music	Tempo: 120 Pitch Range: 40, 57 Music Layers: 1 Notes Length: 50	A face paced music with single layer is composed	Success
2	Compose a slow music	Tempo: 60 Pitch Range: 40, 57 Music Layers: 1 Notes Length: 15	A slow music with single layer is composed	Success
3	Compose a moderate speed music	Tempo: 90 Pitch Range: 45, 80 Music Layers: 2 Notes Length: 30	A moderate speed music with double layers is composed	Success
4	Compose a music with wide pitch range	Tempo: 80 Pitch Range: 30, 100 Music Layers: 3 Notes Length: 30	A 3 layered music with wide pitch range is composed	Success
5	Compose a music with narrow pitch range	Tempo: 80 Pitch Range: 70, 80 Music Layers: 3 Notes Length: 30	A 3 layered music with narrow pitch range is composed	Success

Table 6.2.1 Functional Testing 1 (Music Composition)

Functional Testing on Component 2 : Music Preview

Objective : To ensure that music can be composed and work as intended with different values / parameters inserted into it

No	Case	Parameters / Value	Expected Result	Success / Fail
1	Preview music after composition	Music file exists	Composed music is played	Success
2	Preview music before composition	Music file does not exists	Error message informing the user that no music file exists yet	Success

Table 6.2.2 Functional Testing 2 (Music Preview)

Functional Testing on Component 3 : Music Edit

Objective : To ensure that music can be edited with all editable parameters

No	Case	Parameters / Value	Expected Result	Success / Fail
1	Edit the music to make it slower	Tempo : 40	The music is slowed down	Success
2	Edit the music to make it faster	Tempo : 140	The music sped up	Success
3	Change the instrument to Guitar	Guitar	Instrument changed to Guitar	Success
4	Change the instrument to violin	Violin	Instrument changed to Violin	Success
5	Change the instrument to Church Organ	Church Organ	Instrument changed to Church Organ	Success
6	Change the instrument to Drawbar Organ	Drawbar Organ	Instrument changed to Drawbar Organ	Success
7	Change the instrument to Harmonica with faster tempo	Tempo: 130 Harmonica	Instrument changed to Harmonica with faster tempo	Success
8	Change the instrument to Piccolo with slower tempo	Tempo: 35 Piccolo	Instrument changed to Piccolo with slower tempo	Success
9	Change the instrument to Cello	Tempo: 85 Cello	Instrument changed to Cello with slower tempo	Success

	with moderate tempo			

Table 6.2.3 Functional Testing 3 (Music Edit)

Functional Testing on Component 4 : Music Export

Objective : To make sure that music can be exported under different circumstances

No	Case	Parameters / Value	Expected Result	Success / Fail
1	Export the music to Desktop	Location: Desktop	The midi file is saved to Desktop	Success
2	Export the music to C:\	Location: C:\	The midi file is saved to C:\	Success

Table 6.2.4 Functional Testing 4 (Music Export)

Functional Testing on Component 5 : 3rd party software

Objective : To ensure that the 3rd party software can be launched and used

No	Case	Parameters / Value	Expected Result	Success / Fail
1	Launch 3rd party software	Software exists	Software launched	Success
2	Open composed midi file with 3rd party software	Launch software	Imported successfully	Success

Table 6.2.5 Functional Testing 5 (extra feature)

6.3 Integration Testing

The objective of integration testing is to test all modules as one group. As there exists multiple modules in the system and each of them are being developed as separate modules, errors may occur during or after the integration of all the modules. According to unit testing and functional testing done previously, it has proven that the system has no integration problem and all modules work well together.

Chapter 7: System Limitations and Future Enhancement

7.1 System Limitation

The objective of this system is to create a random generated melody to aid music composer in their music composition progress and to spark inspiration within musician. Although database can be very useful in aiding the user in organizing and maintaining the composed music, however, up until this stage, there is still no need for database to be implemented into the system with its current functionalities. The system is able to achieve its intended objective for now and further enhancement may be implied in the future.

Moreover, the algorithm is not omnipotence and it holds flaws in it. Different issues may require different algorithms set up and up until this point, there are still no absolute solution to cope with this problem. The only one solution is to slowly enhance the algorithm upon more and more situation or issues faced.

There is also different taste of each musician that may challenge the usage of the system. For example, some people might find that the random generated melodies does not meet their taste or requirement. This is a very general issue where different individual musician and different generation people will face variations in music tastes may have their own unique taste to music. Hence, some people might not prefer to use this system as a method to get inspiration due to their very opinions on their music tastes.

7.2 Future Enhancement

Current system only allows the music to be played with up to 3 layers and with different instruments. The system can be further enhanced with **chords and rhythm to be played along with the melody**, allowing the music to sound more complete and more pleasant to the ear. Also, the system currently can be enhanced to support the **modification of composed melody**. This will allow composer to change, mix or modify the melodies in many other ways manually. Additionally, it is possible to allow the system to be connected to the internet so that it can identify existing melody so that the composed melody does not clash with the existing melody or cause any copyright issues. Lastly, a powerful possible enhancement is that the system can prompt to ask questions to user at the initiation of the system so that the generated melodies can **match the taste** of the particular user.

Chapter 8: Conclusion

Artificial intelligence (AI) is constantly evolving and plays an important role in current generation or even century. The reason behind this is due to the assist that AI is capable of providing to human. In the case of this project, AI aids in composing music by generating melodies in different rhythm, pitch and layers that is analyzed by the algorithm and has passed the acceptance level. Therefore, AI composed melodies can be more pleasant to the ear of the listener as well as the user experience.

If AI is implemented properly, it is possible that it can perform all the tasks that human can performs, including the compose of a great piece of music that is believed by human that cannot be done by Artificial Intelligence or computer that does not has feelings and emotions. But what if AI is capable of comprehending emotional feelings? This is still an uncertainty as the future of artificial intelligence can be very astonishing and out of expectation by only looking at what it is already capable of doing now.

In a nutshell, there are still many possible improvements that can be made to this random melody generator system. In the future, it is possible that even people with zero music knowledge can compose a song. Similar system will emerge with more functionalities and it will impact the music industry in an unimaginable way. IT is constantly evolving and so should we.

References

Komponist, *Major and Minor Chords*. Available from: <http://musictheoryblog.blogspot.com/2016/08/major-and-minor-chords.html>. [2016]

Study.com, *What is Music? - Definition, Terminology & Characteristic.* Available from: <https://study.com/academy/lesson/what-is-music-definition-terminology-characteristics.html>.

Biles, AI. 2004, *Evolutionary Music.* Available from: https://www.cs.york.ac.uk/rts/docs/GECCO_2004/Workshop%20proceedings/gecco04/TUT009.pdf. [2004].

Rani, A. 2018, *Top 15 musical instruments you can easily learn & play.* Available from: <https://medium.com/giglue/top-20-musical-instruments-you-can-easily-play-693f603a2bcf>. [11 April 2018].

Pouska, A n.d., *Keys in Music.* Available from: <https://www.studybass.com/lessons/harmony/keys-in-music/>.

Hollis, B n.d., *The Method Behind The Music.* Available from: <https://method-behind-the-music.com/theory/scalesandkeys/>.

Music Theory Academy n.d., *Chord Progressions.* Music Theory Academy. Available from: <https://www.musictheoryacademy.com/understanding-music/chord-progressions/>.

Axis Music Foundation n.d., *About Us.* Available from: <http://axismusicfoundation.org/about-us/>.

Axis Music Foundation n.d., *Headlines.* Available from: <http://www.axismusic.com/headlines/>.

Garage Band, 2012. Available from: <https://itunes.apple.com/my/app/garageband/id408709785?mt=8>. [2012]

Apple Communities n.d.. Available from: <https://discussions.apple.com/thread/7703445>. [10 October 2016].
KekStudio 2017, *Chord Progression Master for Android.* Available from: <https://download.cnet.com/Chord-Progression-Master/3000-2141_4-77597769.html>. [31 July 2017].

Chord Progression Master for Android - By Genres 2017, Google Play. Available from: <https://play.google.com/store/apps/details?id=com.kekstudio.chordprogressionmaster&hl=en&showAllReviews=true>. [2017]

David, M. 2017, *The Prog Machine - Chord Progressions,* App Store Review. Available from: <https://itunes.apple.com/us/app/the-prog-machine-chord-progressions/id1211139207?mt=8>. [2017]

Ed. 2018, *The Prog Machine - Chord Progressions v2.01.* Available from: <http://a4irebith.weebly.com/blog/the-prog-machine-chord-progressions-v201>. [22 January 2018].

Routhier, M. 2017, *Suggester,* App Store Review. Available from: <https://itunes.apple.com/us/app/suggester/id504740787?mt=8>. [2017]

Blache, B n.d., *Why are Apple releases (IOS and macOS) becoming more buggy?* Available from: <https://www.quora.com/Why-are-Apple-releases-iOS-and-macOS-becoming-more-buggy>.

James, J n.d., *The Elements of User Experience: User-Centered Design for the Web and Beyong (2nd Edition).* Available from: <https://www.usability.gov/what-and-why/user-interface-design.html>.

Koelle, D. 2008, *Jamming with Java Technology: Making Music with JFUGUE and JFRETS.* Available from: <https://www.oracle.com/technetwork/systems/ts-5263-159137.pdf>. [2008].

Ali,.A. 2014, *Harmony Search Algorithm.* Available from: <https://www.slideshare.net/afar1111/harmony-search-algorithm>. [6 November 2014].

Appendices

FINAL YEAR PROJECT WEEKLY REPORT

(Project I [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 1**
Student Name & ID: HAYDEN NG	**16ACB03863**
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Studying on the related topic for system extension

2. WORK TO BE DONE

- Research on HMA and possible extension

3. PROBLEMS ENCOUNTERED

- Lack of libraries or documents for referring purposes

4. SELF EVALUATION OF THE PROGRESS

- Have a notion on what to do next

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 2**
Student Name & ID: HAYDEN NG	**16ACB03863**
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE [Please write the details of the work done in the last fortnight.] - Continued on the study on HMA
2. WORK TO BE DONE - Complete the research and study on HMA
3. PROBLEMS ENCOUNTERED - Lack of libraries or documents for referring purposes
4. SELF EVALUATION OF THE PROGRESS - Have idea on what to do next

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 3**
Student Name & ID: HAYDEN NG 16ACB03863	
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Consulted my supervisor, Mr Soong for advices

2. WORK TO BE DONE

- Add in chapters 5,6,7,8 into the FYP2 report

3. PROBLEMS ENCOUNTERED

- None

4. SELF EVALUATION OF THE PROGRESS

- Had direction on the report

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 4**
Student Name & ID: HAYDEN NG	**16ACB03863**
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Completed chapter 5 (Implementation & Deployment)

2. WORK TO BE DONE

- Double check everything in chapter 5

3. PROBLEMS ENCOUNTERED

- None

4. SELF EVALUATION OF THE PROGRESS

- None

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 5**
Student Name & ID: HAYDEN NG	**16ACB03863**
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Added Preview Music function in the system

2. WORK TO BE DONE

- Add an extra feature

3. PROBLEMS ENCOUNTERED

- JAVA knowledge difficulties

4. SELF EVALUATION OF THE PROGRESS

- The system can now preview music within the application

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 6**
Student Name & ID: HAYDEN NG	**16ACB03863**
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Added extra feature into the system that opens up 3rd party application that can display notes of the generated melody as suggested by Mr Soong.

2. WORK TO BE DONE

- Perform system testing

3. PROBLEMS ENCOUNTERED

- Difficulties in JAVA knowledge

4. SELF EVALUATION OF THE PROGRESS

- This is challenging but I managed to complete it

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 7**
Student Name & ID: HAYDEN NG	**16ACB03863**
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Fix up on the GUI of the system

2. WORK TO BE DONE

- Beautify the system

3. PROBLEMS ENCOUNTERED

- None

4. SELF EVALUATION OF THE PROGRESS

- None

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 8**
Student Name & ID: HAYDEN NG	**16ACB03863**
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Added in Chapter 6 (System Testing) into the report
- Performed unit testing, functional testing and integration testing

2. WORK TO BE DONE

- Test all the components in the system

3. PROBLEMS ENCOUNTERED

- None

4. SELF EVALUATION OF THE PROGRESS

- Understood the importance of testing

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 9**
Student Name & ID: HAYDEN NG	**16ACB03863**
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Added in chapter 7 (System Limitations & Future Enhancements)

2. WORK TO BE DONE

- Study what possible enhancement can be implemented to the current system
- Brainstorm the system limitations

3. PROBLEMS ENCOUNTERED

- None

4. SELF EVALUATION OF THE PROGRESS

- None

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 10**
Student Name & ID: HAYDEN NG 16ACB03863	
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Determined all the possible enhancement that can be made to the system
- Added in chapter 8 (Conclusion)

2. WORK TO BE DONE

- Summarize the whole project

3. PROBLEMS ENCOUNTERED

- None

4. SELF EVALUATION OF THE PROGRESS

- None

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 11**
Student Name & ID: HAYDEN NG	**16ACB03863**
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Completed all the additional chapters suggested by Mr Soong.

2. WORK TO BE DONE

- Arrange the report following the FYP2 guideline

3. PROBLEMS ENCOUNTERED

- Page number sequencing issue

4. SELF EVALUATION OF THE PROGRESS

- Took a long time to learn how to properly use page break

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project I / [✓ Project II])

Trimester, Year: 3, 3	**Study week no.: 12**
Student Name & ID: HAYDEN NG	**16ACB03863**
Supervisor: MR. SOONG	
Project Title: EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Finalizing the FYP report
- Double checked if I have everything in my report

2. WORK TO BE DONE

- Submitting the report for Turnitin checking
- Amend some corrections according to the Turnitin report
- Finizaling and submitting the FYP report

3. PROBLEMS ENCOUNTERED

- Missed several important sections

4. SELF EVALUATION OF THE PROGRESS

- None

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Supervisor's signature

Student's signature

Poster

Evoluthm

Introduction

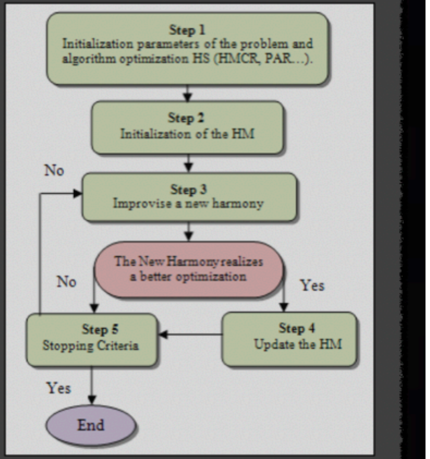
- Evoluthm is a JAVA-developed offline system that is capable of generating random melodies with supplementary functions like adding music layer, selecting pitch, number of notes, tempo and changing music instruments. It helps inspire music composers with new ideas to write new songs.

Innovation

- Evoluthm combines Music and IT, forming a new trend of technology. Music industry will continue to strive to enhance this combinaion and result in the perfect and highest quality application that can help musician from around the world from different aspects.

Methodology

- Evoluthm is developed using Harmony Search Algorithm and JFUGUE.
- With HM, evoluthm is able to generate the perfect piece of melody


Step 1
Initialization parameters of the problem and algorithm optimization HS (HMCR, PAR…).
Step 2
Initialization of the HM
Step 3
Improvise a new harmony
The New Harmony realizes a better optimization
No
Yes
Step 4
Update the HM
Step 5
Stopping Criteria
No
Yes
End

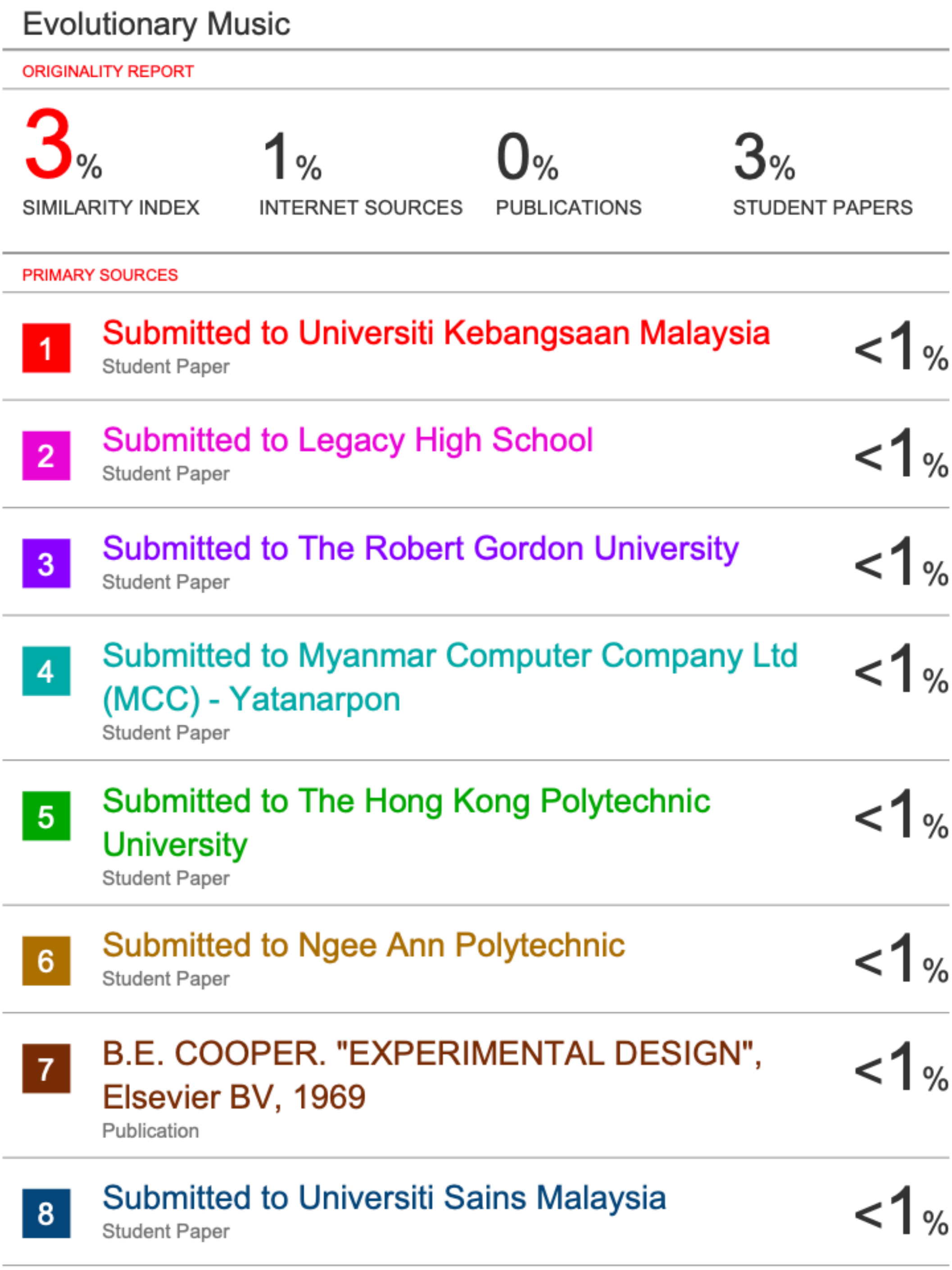
Conclusion

- Evoluthm will be delivering inspiration to allow composition of music.

Technology / Hardware used

- eclipse

Plagiarism Check Result



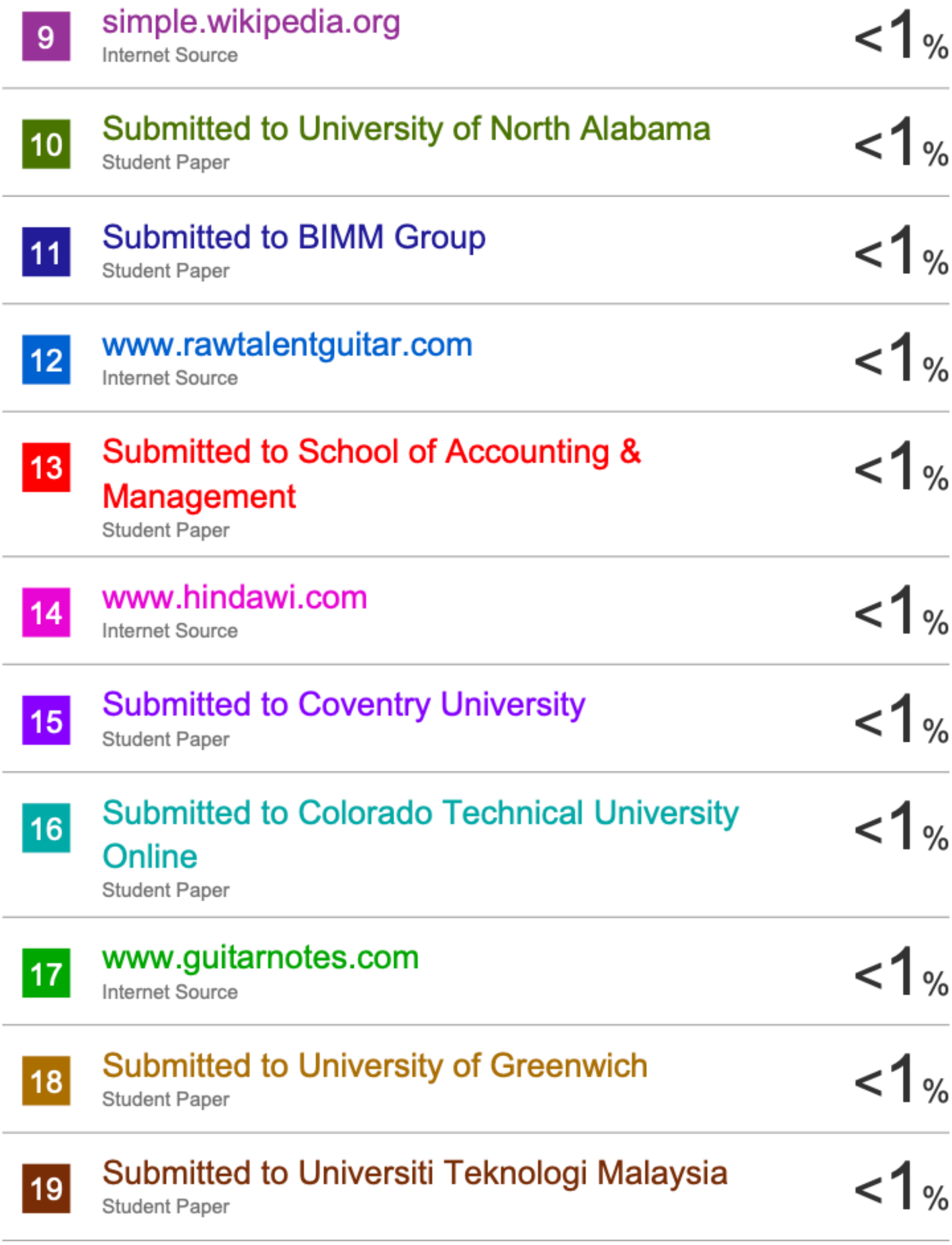
Evolutionary Music

ORIGINALITY REPORT

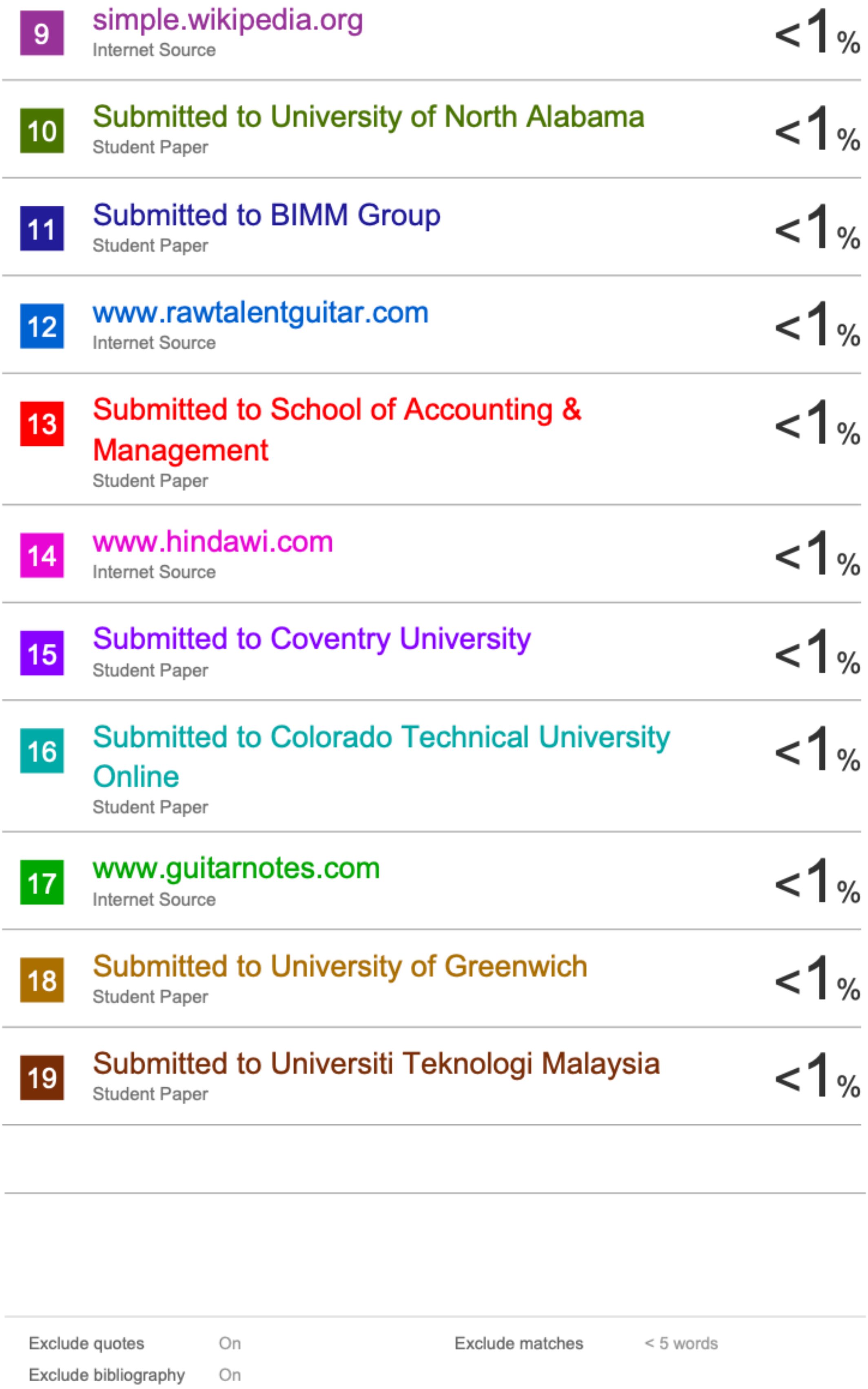
3%	1%	0%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Universiti Kebangsaan Malaysia Student Paper	<1%
2	Submitted to Legacy High School Student Paper	<1%
3	Submitted to The Robert Gordon University Student Paper	<1%
4	Submitted to Myanmar Computer Company Ltd (MCC) - Yatanarpon Student Paper	<1%
5	Submitted to The Hong Kong Polytechnic University Student Paper	<1%
6	Submitted to Ngee Ann Polytechnic Student Paper	<1%
7	B.E. COOPER. "EXPERIMENTAL DESIGN", Elsevier BV, 1969 Publication	<1%
8	Submitted to Universiti Sains Malaysia Student Paper	<1%



9	simple.wikipedia.org Internet Source	<1%
10	Submitted to University of North Alabama Student Paper	<1%
11	Submitted to BIMM Group Student Paper	<1%
12	www.rawtalentguitar.com Internet Source	<1%
13	Submitted to School of Accounting & Management Student Paper	<1%
14	www.hindawi.com Internet Source	<1%
15	Submitted to Coventry University Student Paper	<1%
16	Submitted to Colorado Technical University Online Student Paper	<1%
17	www.guitarnotes.com Internet Source	<1%
18	Submitted to University of Greenwich Student Paper	<1%
19	Submitted to Universiti Teknologi Malaysia Student Paper	<1%



9	simple.wikipedia.org Internet Source	<1%
10	Submitted to University of North Alabama Student Paper	<1%
11	Submitted to BIMM Group Student Paper	<1%
12	www.rawtalentguitar.com Internet Source	<1%
13	Submitted to School of Accounting & Management Student Paper	<1%
14	www.hindawi.com Internet Source	<1%
15	Submitted to Coventry University Student Paper	<1%
16	Submitted to Colorado Technical University Online Student Paper	<1%
17	www.guitarnotes.com Internet Source	<1%
18	Submitted to University of Greenwich Student Paper	<1%
19	Submitted to Universiti Teknologi Malaysia Student Paper	<1%

Exclude quotes	On	Exclude matches	< 5 words
Exclude bibliography	On		

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	HAYDEN NG
ID Number(s)	16ACB03863
Programme / Course	IB
Title of Final Year Project	EVOLUTIONARY MUSIC; COMPOSING BEATS AND CHORDS PROGRESSION

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 3 **%** **Similarity by source** Internet Sources: 1 % Publications: 0 % Student Papers: 3 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required and limits approved by UTAR are as Follows:

- **(i) Overall similarity index is 20% and below, and**
- **(ii) Matching of individual sources listed must be less than 3% each, and**
- **(iii) Matching texts in continuous block must not exceed 8 words**

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Online Signature during COVID-19
SOONG HOONG CHENG (Lecturer)
FACULTY OF ICT
UNIVERSITI TUNKU ABDUL RAHMAN (UTAR)
PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT,
31900 KAMPAR.

Signature of Supervisor	Signature of Co-Supervisor
Name: MR. SOONG HONG CHENG	Name:
Date: 20/4/2020	Date:

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	16ACB03863
Student Name	HAYDEN NG
Supervisor Name	MR. SOONG HOONG CHENG

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Front Cover
√	Signed Report Status Declaration Form
√	Title Page
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction.
	Online Signature during COVID-19 SOONG HOONG CHENG (Lecturer) FACULTY OF ICT UNIVERSITI TUNKU ABDUL RAHMAN (UTAR) PERAK CAMPUS, JLN UNIVERSITI, BANDAR BARAT, 31900 KAMPAR.
(Signature of Student) Date: 20/4/2020	(Signature of Supervisor) Date: 20/4/2020