Beat Explorer Content Based Search for MIDI File Format

By

Mohd Hafiz b Wan Omar

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information Technology)
JUNE 2006

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Content Based Search for MIDI File Format

by

Mohd Hafiz b Wan Omar

A project dissertation submitted to the
Information Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION TECHNOLOGY)

Approved by,

(Associate Professor Dr. Abas Md. Said)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
JUNE, 2006

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

MOHD HAFIZ WAN OMAR

ACKNOWLEDGEMENT

I would like to take this opportunity to thank everyone for their help and support during the development of the project until its completion.

First of all, I would like to thank Dr. Abas Md. Said, this dissertation report could not have been written without him. He did not only serve as my supervisor but he had also encouraged and challenged me throughout the program, giving me the ideas as well as suggestions in guiding me.

Lots of appreciation I bid to my friends who assist me a lot with their ideas and their contributions especially to Mohd Hamzali Azizan and Nghiem Anh Tuan.

Last but not least, I bid special thanks to my family for their support and encouragement which motivates me in completing the project.

ABSTRACT

This project aims on a variety of music search methods for users. It allows users from a variety of backgrounds and capabilities to use the system. It provides searching capability using humming, music note and parson code for midi file format. Developers and users require different method in querying and retrieving database information with a rich variety of methods to enhance the capability of the system. For the advance users, especially composers, searching by music note and parson code would be the best choice for them since it would be able to provide more precise output while for other users, humming is the best choice since almost everyone can hum. Since not everyone could remember the song title, humming provide the searching capability based on the melody they hum. The system will record the humming sound and filter the noise before process it to produce the frequency of humming. The frequency value will be converted into music contour in parson code to be compared to the data in the database. For music note, it will also be converted into frequency and music contour for comparison. The system will output the song title together with the similarity ratio of the music contour. This project uses waterfall model approach for the development due to lots of advantages with the appropriate development tools. It is developed using a JAVA technology and open source frameworks together with library provided from SUN Microsystems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background of the study

In the world of technology nowadays, everybody moves towards computerized system almost in all aspect. With the latest technology and invention in hardware and software we demand to have the ability to search through huge quantities of digital document text or office document. For example, finding information on subjects of interest stored in computer. This task would be easier if people provide meta-data in some standard form.

In many ways, the situation for non-textual digitally stored data is different. However, it shares the problem of rarely being tagged and incomplete information attached to the document. Non-textual data takes many forms such as symbolic data (MIDI sound files or vector graphic files). It is also comes with in a simply direct digital data such as analogue data for wav and mp3 file or bit mapped image.

These Non-textual data cannot be easily searched using meta-data of files but it requires access to the source or content of the file based on querying method used by a system or algorithm.

1.2 Problem statement

1.2.1 Problem Identification

With the rapid development of technology, we have come up with a powerful system that is capable of storing and managing different data. People now store lots of documents and files. Imagine that your computer system has to access a large volume of data in many different formats such textual data, still and moving images, sound recordings and drawings for example. Now imagine that you could issue queries such as finding sound of the whistle or humming. Clearly, if the entire file were nicely labeled with meta-data in XML, this would present little challenge. But in reality sounds in midi format especially are not nicely tagged and it is not content based searching. What we're looking for is the development of content-based search techniques for digitally stored data focusing on midi files. This is the key technology that underlies the capability of fulfilling queries like the above by enhancing the capability in searching.

1.2.2 Significance of the Project

It is important to have a searching capability beyond the existing technology which uses meta-data based along with the rapid development of technology. This java content based searching project enhances the capability of query files or document based on the content. Besides saving lots of time, it will help certain people in the music industry such as composers to manage and manipulate their document better for example in midi files. Furthermore, this project gives the opportunity for users to be one step ahead while saving their valuable time.

1.3 Objective and scope of the study

1.3.1 Objectives of the Project

The objectives of the project are as follows:

- To provide searching content for midi format
- Allow user to have more alternatives of searching method
- Improve capability of existing application
- Provide better matching with percentage for more precise output

1.3.2 Scope of the Study

Researches have done in many ways for content based search for common file type such as audio. This project focuses on content based searching for midi files format which provide benefits to users.

As for the time scope, the first semester period will be focused on research of method and latest technology in java programming to improve the application. It will also be used to study and understand the existing application. The second semester of the project will focused on the development portion such as coding part (unit), integration, testing and eventually deliverables of the system.

CHAPTER 2
LITERATURE REVIEW

2.1 Music Content Based Search

According to Music Content Analysis through Models of Audition Journal by Keith D. Martin, Eric D. Scheirer, and Barry L. Vercoe (1998), "The direct application of ideas from music theory and music signal processing has not yet led to successful musical multimedia systems" (p. 1). One of their research goals state that interaction with large databases of musical multimedia could be made simpler by annotating audio data with information that is useful for search and retrieval, by labeling salient events and by identifying sound sources in the recordings. They also state that more intelligent tools could be constructed for composers, whether experienced or novice by allowing rapid access to in-progress compositions, and control over databases of timbre, samples, and other material, a "composers' workbench" incorporating musically intelligent systems would become a powerful tool. They also state that achieving a successful multimedia system requires understanding of the human representations of musical sound and building computer systems with compatible representations. It also requires the interface to be natural, allowing human users to interact in the normal and usual way, perhaps by humming a few bars, articulating similarity judgments, or by performing a "beat-box" interpretation of a song.

Based on M. Anand Raju, Bharat Sundaram and Preeti Rao (2003), there are approaches for query by humming which is using a parson code and edit distance algorithm where parson code representing the music contour and similarity of the string is compared by using edit distance algorithm (p. 3-4). It shows the application of TANSEN, a query-by-humming music indexing and retrieval system based on melody, or the "tune", of the music using the method specified.

Referring to Leon Fu, Xiangyang Xue (2005), "The matching algorithm for the similarity measure must also be considered. In [1], they use an approximate string matching algorithm described by Baesa-Yates and Perleberg. In [8], a new innovative distance metrics between query and songs is proposed based on symbolic data (p. 2). According to this thesis, it states that there are Baesa Yates and Perleberg string matching algorithm that can be used to determine similarities especially in application of humming proving.

2.2 Preprocessing

Referring to Mr. P.A Venkatachalam (2005), he stated that computer processing in low pas, high pass and shareholding frequency is applicable not only to image but also to other application in computer (n.p.).

Noise Removal – Remove noise in sound by removing weak and undesirable artifact in sound.
Low Pass Frequency – Filter Frequency by removing frequency with very high value
High Pass Frequency – Filter Frequency by removing frequency with very low value
Threshold – define the threshold or point where some frequency remains while other is removed. It is also to remove frequency with low energy.
All this methods are applicable for audio and image in frequency domain.

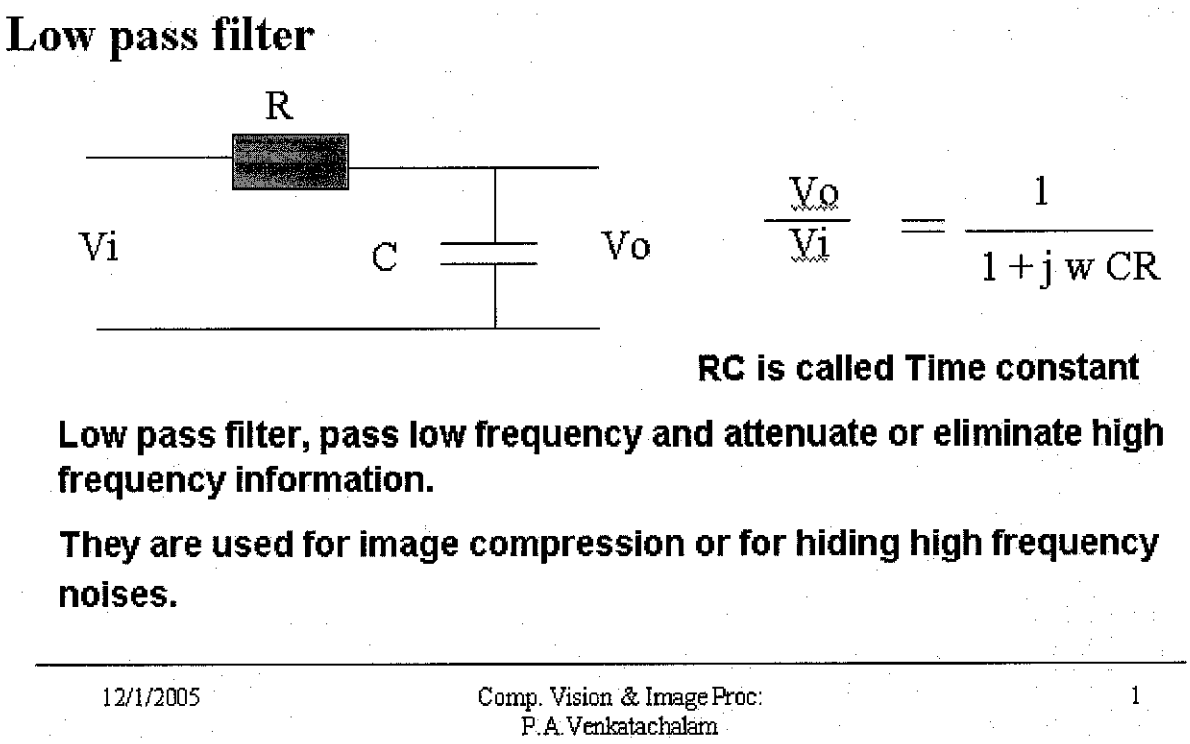


Figure 1Low Pass Filter

2.3 Fast Fourier Transform (FFT)

Referring to Glenda A. Miller (1992), "FFTs are Fast Fourier Transform Algorithms. A Fast Fourier transform (FFT) is an efficient algorithm to compute the Discrete Fourier Transform (DFT) and its inverse. FFTs are of great importance to a wide variety of applications, from digital signal processing to solving partial differential equations to algorithms for quickly multiplying large integers" (p. 5).

FFT transfer sound captured from time domain to frequency domain where the frequency of the sound can be determined. It is an efficient algorithm to compute the discrete Fourier transform (DFT) and it's inverse and widely used in a wide variety of applications especially in digital signal processing. It is calculated using mathematical equation.

Let x0, ..., xn-1 be complex numbers. The DFT is defined by the formula

$$f_j = \sum_{k=0}^{n-1} x_k e^{-\frac{2\pi i}{n} jk} \qquad j = 0, \ldots, n-1.$$

As shown in figure 2, FFT is converted into 2 dimensional array representing amplitude and frequency of a signal.

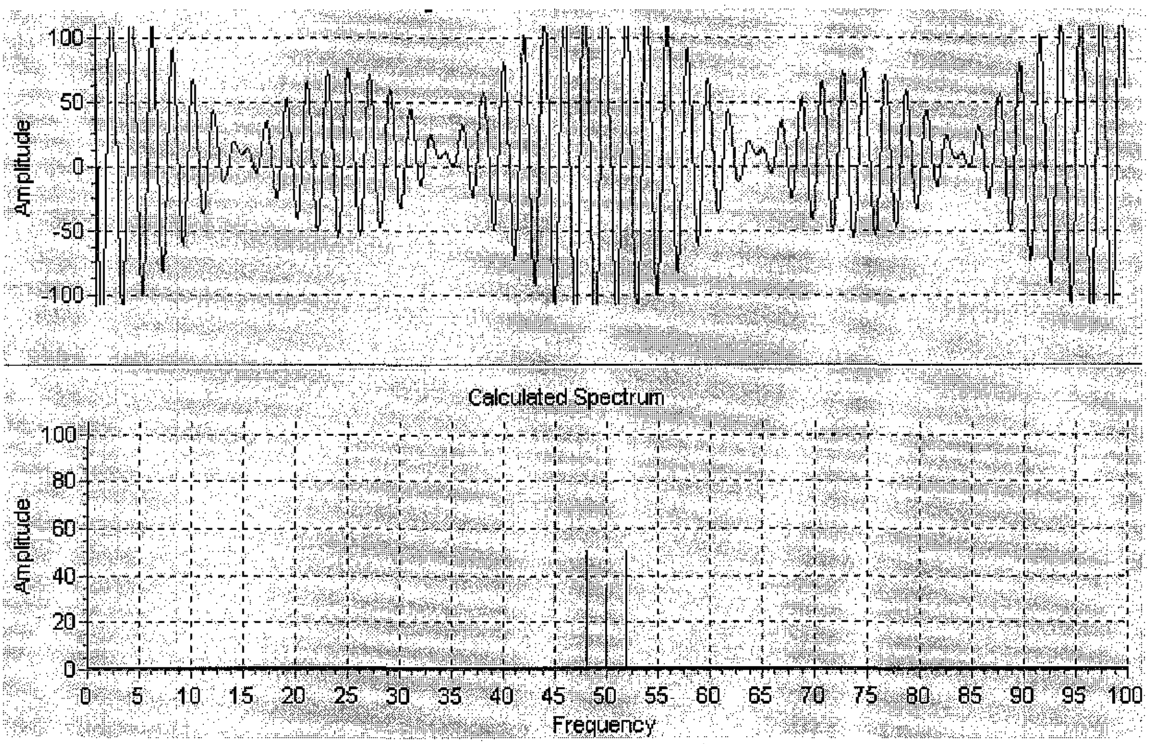


Figure 2 Fast Frequency Transform

2.4 Midi Note and Frequency

Figure 3 below show the relation between Midi Keys, Music Note and Frequency that is used as a base of the project where the relation linked the whole project with addition of parson code to determine the contour of the music. The figure is based on the resource on the internet (http://www.borg.com/~jglatt/tutr/notefreq.htm).

MIDI Note Number to Frequency Conversion Chart

	MIDI Note	Frequency	MIDI Note	Frequency	MIDI Note	Frequency
C	0	8.1757989156	12	16.3515978313	24	32.7031956626
Db	1	8.6619572180	13	17.3239144361	25	34.6478288721
D	2	9.1770239974	14	18.3540479948	26	36.7080959897
Eb	3	9.7227182413	15	19.4454364826	27	38.8908729653
E	4	10.3008611535	16	20.6017223071	28	41.2034446141
F	5	10.9133822323	17	21.8267644646	29	43.6535289291
Gb	6	11.5623257097	18	23.1246514195	30	46.2493028390
G	7	12.2498573744	19	24.4997147489	31	48.9994294977
Ab	8	12.9782717994	20	25.9565435987	32	51.9130871975
A	9	13.7500000000	21	27.5000000000	33	55.0000000000
Bb	10	14.5676175474	22	29.1352350949	34	58.2704701898
B	11	15.4338531643	23	30.8677063285	35	61.7354126570
C	36	65.4063913251	48	130.8127826503	60	261.6255653006
Db	37	69.2956577442	49	138.5913154884	61	277.1826309769
D	38	73.4161919794	50	146.8323839587	62	293.6647679174
Eb	39	77.7817459305	51	155.5634918610	63	311.1269837221
E	40	82.4068892282	52	164.8137784564	64	329.6275569129
F	41	87.3070578583	53	174.6141157165	65	349.2282314330
Gb	42	92.4986056779	54	184.9972113558	66	369.9944227116
G	43	97.9988589954	55	195.9977179909	67	391.9954359817
Ab	44	103.8261743950	56	207.6523487900	68	415.3046975799
A	45	110.0000000000	57	220.0000000000	69	440.0000000000
Bb	46	116.5409403795	58	233.0818807590	70	466.1637615181
B	47	123.4708253140	59	246.9416506281	71	493.8833012561
C	72	523.2511306012	84	1046.5022612024	96	2093.0045224048
Db	73	554.3652619537	85	1108.7305239075	97	2217.4610478150
D	74	587.3295358348	86	1174.6590716696	98	2349.3181433393
Eb	75	622.2539674442	87	1244.5079348883	99	2489.0158697766
E	76	659.2551138257	88	1318.5102276515	100	2637.0204553030
F	77	698.4564628660	89	1396.9129257320	101	2793.8258514640
Gb	78	739.9888454233	90	1479.9776908465	102	2959.9553816931
G	79	783.9908719635	91	1567.9817439270	103	3135.9634878540
Ab	80	830.6093951599	92	1661.2187903198	104	3322.4375806396
A	81	880.0000000000	93	1760.0000000000	105	3520.0000000000
Bb	82	932.3275230362	94	1864.6550460724	106	3729.3100921447
B	83	987.7666025122	95	1975.5332050245	107	3951.0664100490
C	108	4186.0090448096	120	8372.0180896192		
Db	109	4434.9220956300	121	8869.8441912599		
D	110	4698.6362866785	122	9397.2725733570		
Eb	111	4978.0317395533	123	9956.0634791066		
E	112	5274.0409106059	124	10548.0818212118		
F	113	5587.6517029281	125	11175.3034058561		
Gb	114	5919.9107633862	126	11839.8215267723		
G	115	6271.9269757080	127	12543.8539514160		
Ab	116	6644.8751612791				
A	117	7040.0000000000				
Bb	118	7458.6201842894				
B	119	7902.1328200980				

Figure 3 Midi Note and Frequency

2.5 Music Note, Frequency and Midi Note

Figure 4 shown is based on University New South Wales's site showing relation of music note, frequency and music note (http://www.phys.unsw.edu.au/~jw/notes.html).

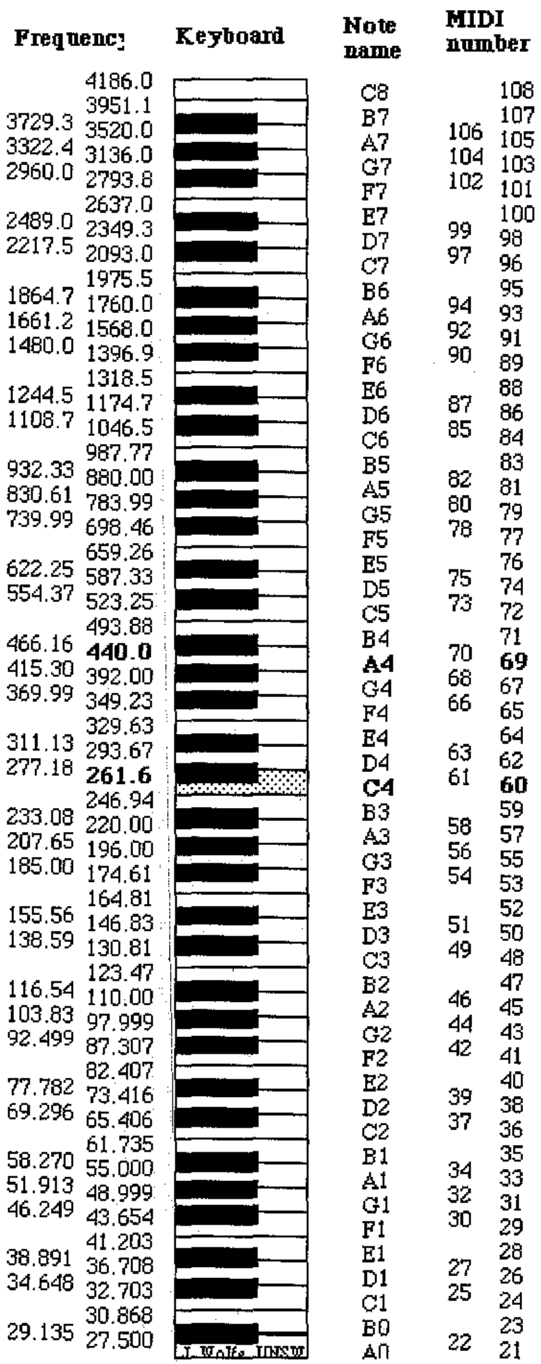


Figure 4 Music Note, Frequency and MIDI music key

2.6 Parson code

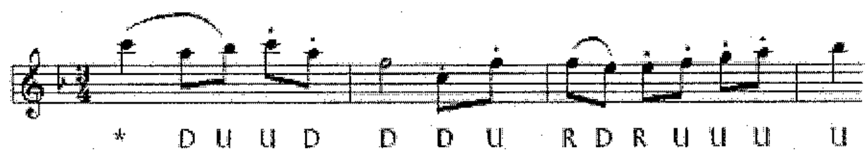


Figure 5 Parson Code

As refer to Musipedia site, figure 5 show the example of parson code of a music note which means that music note can be converted to parson code. Parson code consist of 3 main keys which are U as representing up, R representing remain and D representing down (http://www.musipedia.org/pcnop.0.html). The combination of these 3 keys can create a contour of music and determine the movement of the frequency based on the pitch. Rhythm is completely ignored by using this method.

Figure 5 show the Beethoven's 8th symphony note which is coded into DUUDDDURDRUUUU string in parson code. It comes by comparing in between two note and presenting the movement.

2.7 Edit Distance Algorithm

Based on reference from Faculty of Information Technology, Monash University (1999) " The words `computer' and `commuter' are very similar, and a *change* of just one letter, p->m will change the first word into the second. The word `sport' can be changed into `sort' by the *deletion* of the `p', or equivalently, `sort' can be changed into `sport' by the *insertion* of `p'." (http://www.csse.monash.edu.au/~lloyd/tildeAlgDS/Dynamic/Edit/).

In many applications, it is necessary to determine the similarity of two strings. A widely-used notion of string similarity is the edit distance: the minimum number of insertions, deletions, and substitutions required to transform one string into the other.

Algorithms similar to those for the edit-distance problem are used in some speech recognition systems such as to find a close match between strings.

Based on the example provided by Faculty of Information Technology Monash University in table 1, a word appropriate meaning and approximate matching has a 7 similarity with the word appropriate meaning added – to become appropriate m-eaning to make it correlation with the word approximate matching.

Table 1 Edit Distance

<table>
<tr><td>a</td><td>p</td><td>p</td><td>r</td><td>o</td><td>p</td><td>r</td><td>i</td><td>a</td><td>t</td><td>e</td><td></td><td>m</td><td>-</td><td>e</td><td>a</td><td>n</td><td>i</td><td>n</td><td>g</td></tr>
<tr><td>|</td><td>|</td><td>|</td><td>|</td><td>|</td><td></td><td></td><td></td><td>|</td><td>|</td><td>|</td><td></td><td>|</td><td></td><td></td><td></td><td></td><td>|</td><td>|</td><td>|</td></tr>
<tr><td>a</td><td>p</td><td>p</td><td>r</td><td>o</td><td>x</td><td>i</td><td>m</td><td>a</td><td>t</td><td>e</td><td></td><td>m</td><td>a</td><td>t</td><td>c</td><td>h</td><td>i</td><td>n</td><td>g</td></tr>
</table>

CHAPTER 3
METHODOLOGY AND PROJECT WORK

3.1 Methodology

This project will use the waterfall model for project management and development. It is Most-widely used process model. It allows developer to controls schedules, resources & documentation of the project. It is divided into 5 phases which are requirement definition, system and software design, implementation and user testing, integration and system testing and finally operation and maintenance phase as shown in Figure 4.

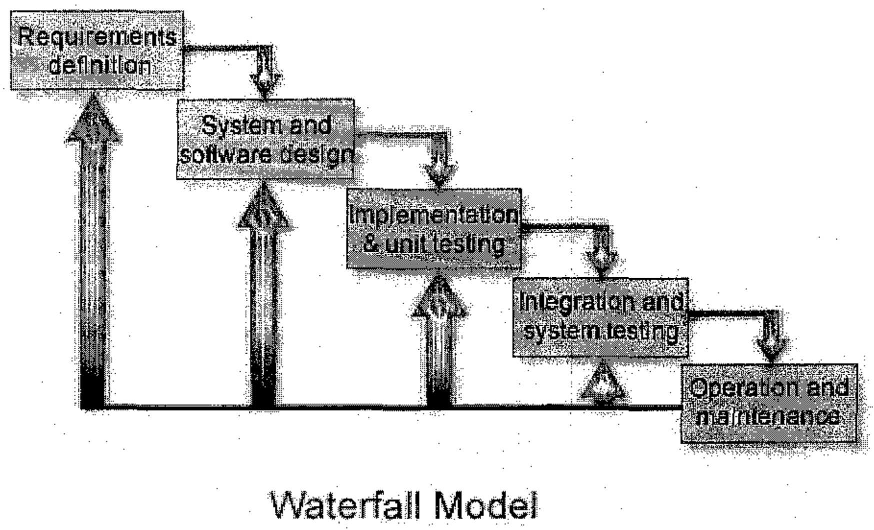


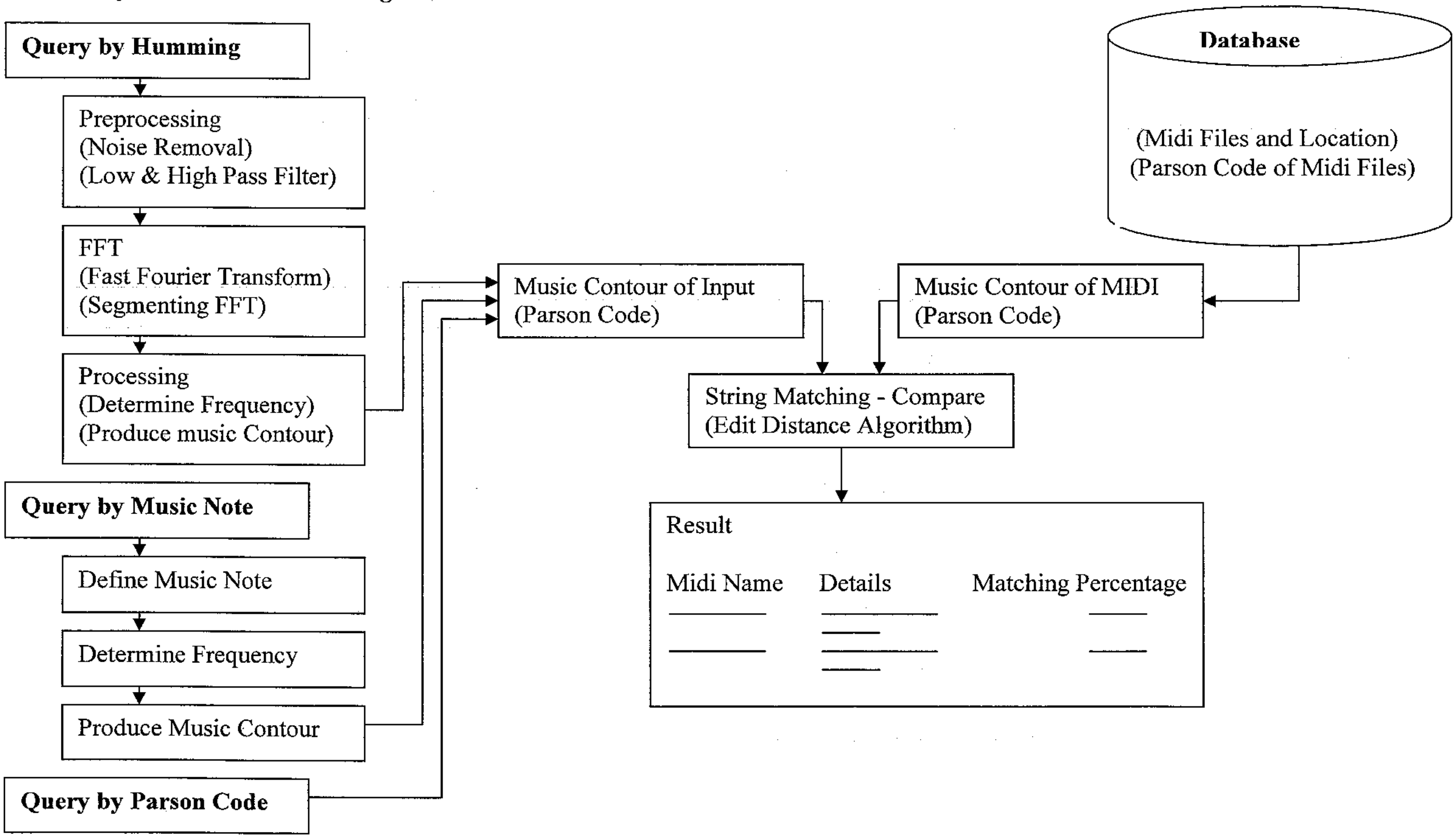
Figure 6 Waterfall Model

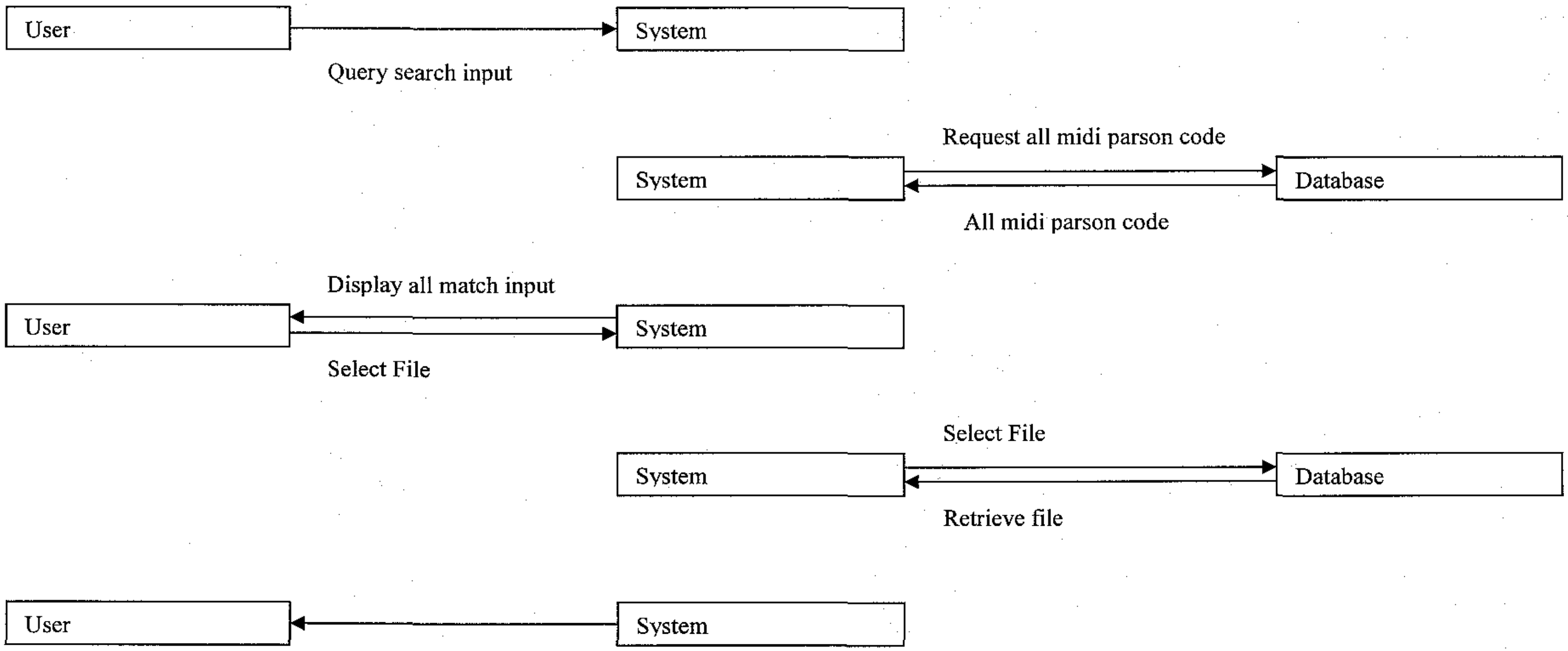
In requirement definition stage, the problems are identified with the desired goals, verified specification of the required functions, interfaces, and performance for the software product. The constraints and limitations of the product are identified. There are also stages where specifications are produced in detail. It is also concerned with the data structure, software architecture, algorithm and the interface. In system and software design, the software is represented by a model. The design then translated into software domain. Implementation and user testing will start the development process unit by unit and tested. The testing will make sure that errors are identified and the software meets its required specification. In the integration and system testing phase, all the program units are integrated and tested to ensure that the complete system meets the software requirements. The deliverable of the product can be made after this stage. In operation and maintenance phase, it may require previous stages to be repeated for maintenance or improvement.

3.1.1 Requirement Definition

Requirements are defined based on problem statement and analysis. It focuses on the objective which is to solve the problem and improve the searching capability for content based search focusing on MIDI file format.

3.1.2 System and Software Design




User
System
Query search input
Request all midi parson code
System
Database
All midi parson code
Display all match input
User
System
Select File
Select File
System
Database
Retrieve file
User
System

3.1.3 Implementation and Unit Testing

In implementing and developing the system for the content based search, 5 main module have been determine which are to capture sound, preprocessing, music note quantization, construct contour of music to parson code and searching.

1. capturing sound

   - Capture sound from audio device from line in or microphone input.

2. pre-processing

   a. convert whistle sound into frequency domain using Fast Fourier Transfer

   b. noise filter – low pass, high pass filter and threshold

      i. the "low pass" filter will filter out all the "too high" frequency

      ii. the "high pass" filter will filter out all the "too low" frequency

      iii. cancel all the frequency which has too weak energy using threshold

3. Music Note Quantization

   -Determined the range of frequency which has the most energy concentrated on to determine the music note

   - determine a reference point: for example: 65 Hz is "C" note for music instrument based on research data.

4. Construct contour of music

   -Applied Parson Code to capture sound and midi

   -midi parson code is stored in database

   -If the next note higher than previous note, mark "u" (mean UP)

   -If the next note equal to previous note, mark "r" (mean REMAIN/REPEAT)

   -If the next note lowers than previous note, mark "d" (mean DOWN)

   -Result in string of "u" "r" "d": for example: druduudrduudrrrdr

5. Searching

   -Compare the parson code of captured sound to the parson code of MIDI

   -Using algorithm called Edit Distance to compare and match the string

   -It runs through the MIDI and finds the portion that closet to the whistle in term of parson code

   -Result of comparing similarity ratio between the whistle and MIDI in term of parson code (ratio return by Edit Distance method) which indicates how close two strings are similar result between 0-1.

   -based on result the list is ranked.

3.1.4 Integration and System Testing

Integration and system testing is focusing on integration between 5 main unit in implementation. The overall system then will be tested to ensure it meets the requirement and error free. It will be tested using white box and black box testing for the software testing. It covers the functionality of the system and also the internal code of the system.

3.1.5 Operation and Maintenance

The system will be improved based on test result and return to any development stage in waterfall methodology for the maintenance and improvement of the system.

3.2 Tools

There are 2 type of tools required for this software enhancement which are software and hardware. For the hardware, a computer that is capable of running and playback midi files is required. Another tool that is required is microphone for the application input for query.

For the software, required application is standard codec for midi playback which is normally support by operating system, java compiler which is j2sdk (Java 2 Software Development

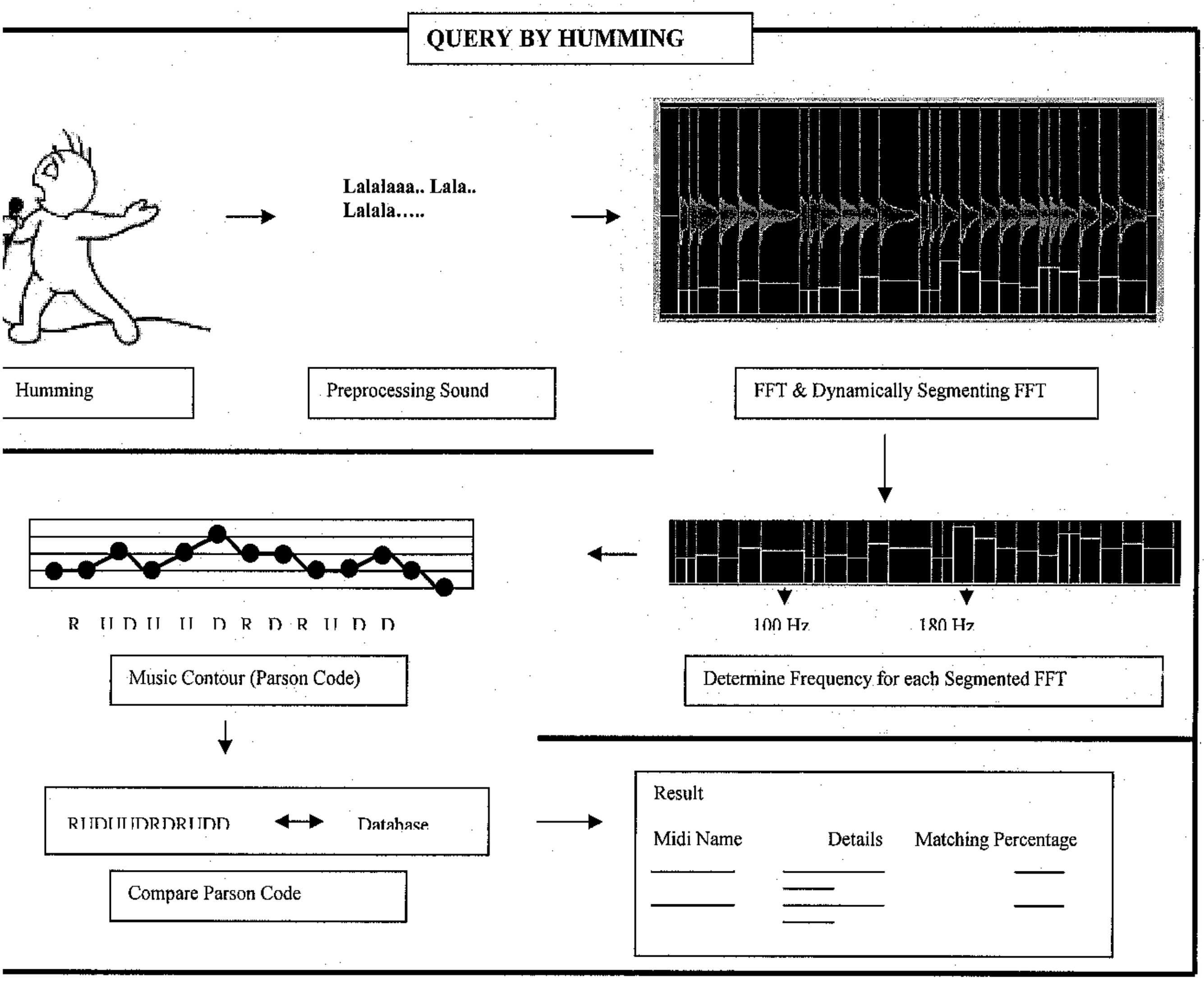
Kit). The version of this kit is v.1.5.04. This is to allow maximum compatibility between different operating systems. The j2sdk 1.5.04 offers the highest performance and better JDK features compared to earlier version such as 1.3 and 1.4. It also offers high level API (Application Programming Interface) which helps developer to save time in certain area by providing built in function comes with the API. For example, Java Media APIs collection consists of many type of API that helps developer to implement and develop multimedia applications.

Other software that will be used as development tools is Eclipse IDE, one of the tools for Java which rich of features such as auto completion, CVS clients, ANT plug-in, and the infamous JAVA unit testing suite, JUnit. All these tools allow developer to speed up development process and also increase reliability of the software.

CHAPTER 4
RESULT AND DISCUSSION

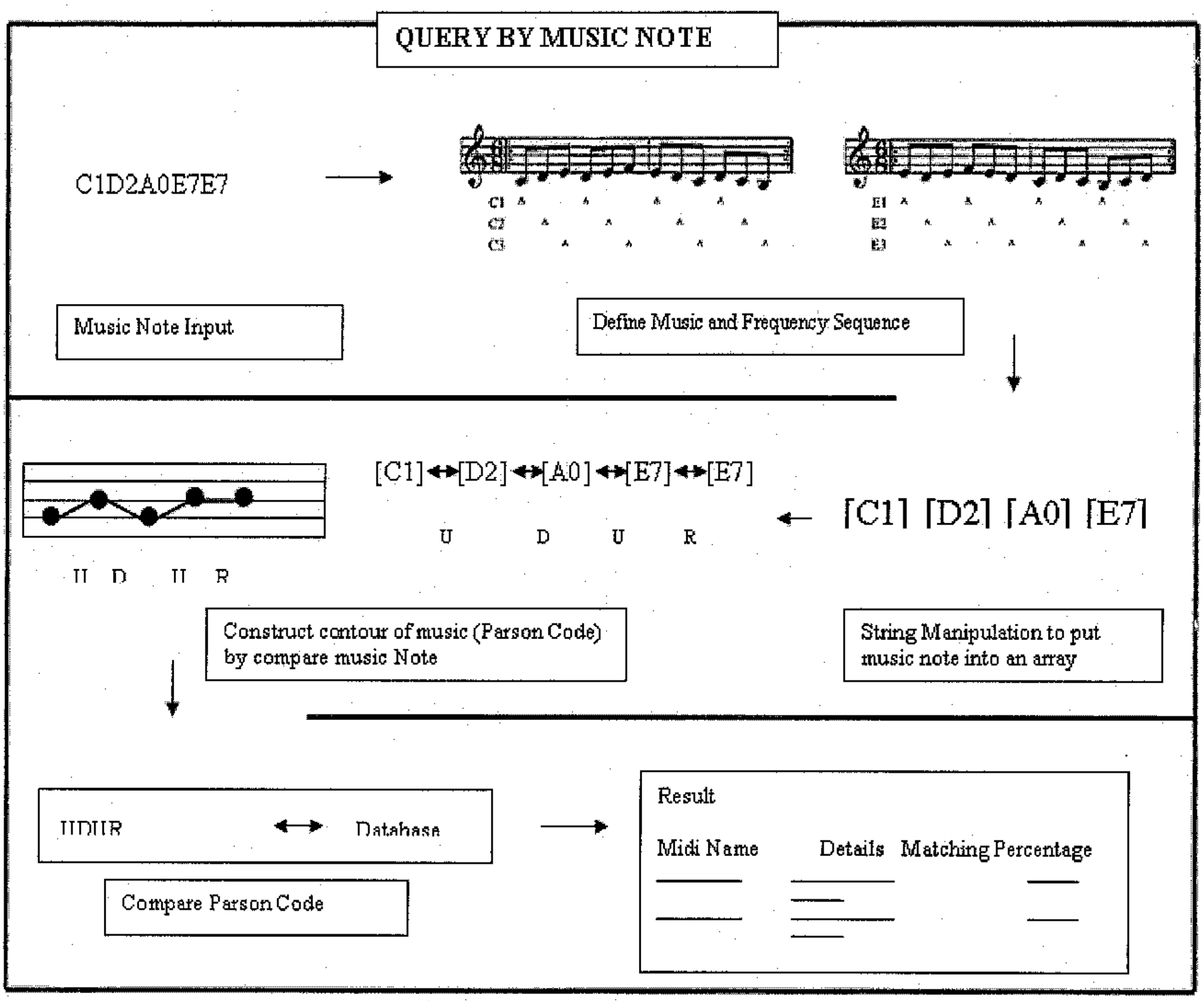
4.1 Result

4.1.1 Query by Humming



For query by humming, I have succeeded in capturing the sound from humming and preprocessing to filter out noise and unwanted signal from the environment. For this section, there are 3 sub modules which are preprocessing to filter noise, segmenting the FFT (Fast Frequency Transform) and produce music contour from segmented FFT and its frequency. The signal gathered after removing the noise by using low pass, high pass and threshold filter than will be analyzed in FFT. For humming, the FFT will be in a form of where we can see the separation between each tone of hum as shown in the figure above. Based on the hum, it will be separated and segmented to produce frequency for each segment. The frequency for each segment than will be constructed into music contours that is in a form of parson code which has the value U, D or R (Up, Down, Remain). Parson code of the input than will be compared to parson code of song stored in the database using edit distance algorithm.

4.1.2 Query by Music Note



For this module, it is tested and working whereby users especially advance users, need to key in the music note based on music or the key from music keyboard. The system allows 58 different music notes to be inserted in system for searching.

Table 2 Music Note and Frequency

C0 16.35	C1 32.70	C2 65.41	C3 130.81	C4 261.63
D0 18.35	D1 36.71	D2 73.42	D3 146.83	D4 293.66
E0 20.60	E1 41.20	E2 82.41	E3 164.81	E4 329.63
F0 21.83	F1 43.65	F2 87.31	F3 174.61	F4 349.23
G0 24.50	G1 49.00	G2 98.00	G3 196.00	G4 392.00
A0 27.50	A1 55.00	A2 110.00	A3 220.00	A4 440.00
B0 30.87	B1 61.74	B2 123.47	B3 246.94	B4 493.88

C5 523.25	C6 1046.50	C7 2093.00	C8 4186.01
D5 587.33	D6 1174.66	D7 2349.32	D8 4698.64
E5 659.26	E6 1318.51	E7 2637.02	
F5 698.46	F6 1396.91	F7 2793.83	
G5 783.99	G6 1567.98	G7 3135.96	
A5 880.00	A6 1864.66	A7 3520.00	
B5 987.77	B6 1975.53	B7 3951.07	

All music notes have their own frequencies which are declared in the system. Then, the input from users will be compared to each previous note to determine the parson code or music contour of the note before comparing with the information in the database.

Database Searching with Approximated String Matching technique: The parson code resulted from precious process is used as the key pattern to search compositions in database. In here, compositions are loaded from database and compared with the key pattern one by one to find the best match. The key pattern is slid along the compositions; the Edit Distance algorithm will calculate the similarity between the key pattern and the corresponding portion of each composition. The system need to slide the key pattern because users may whistle at an arbitrary point in the song. The maximum value of similarities between whistle key patterns and portions of a composition will become similarity value between whistle and the composition. As a result, the system returns a list of compositions which was already put in order based on similarity value.

4.1.3 Query by Parson Code

For this module, it is also tested and working. It received a directly parson code or Music contour. It is another module for advance users where music contour construction is not needed in the system. As the result of this process, the shape of music note pitch is depicted in the form of Parson Code as the input will be directly compared to parson code in the database. Compared to 2 other module, this module is the simplest module. However, it reduces noise and processing problems and provides better and reliable output compared to other method. For the second module, query by music note, it is also reliable where the frequency of music note is define and required conversion into parson code. For query by humming, it is less reliable since there are lots of possible problems such as human error, noise and complexity of processing.

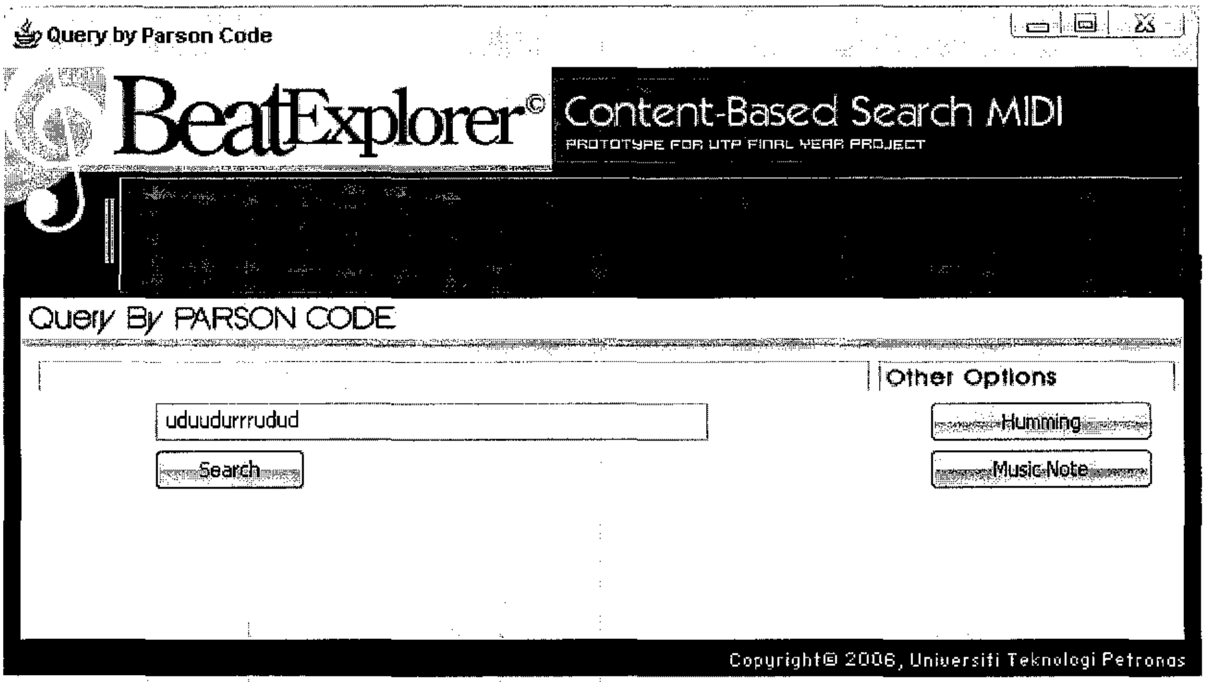
Table 3 Parson Code

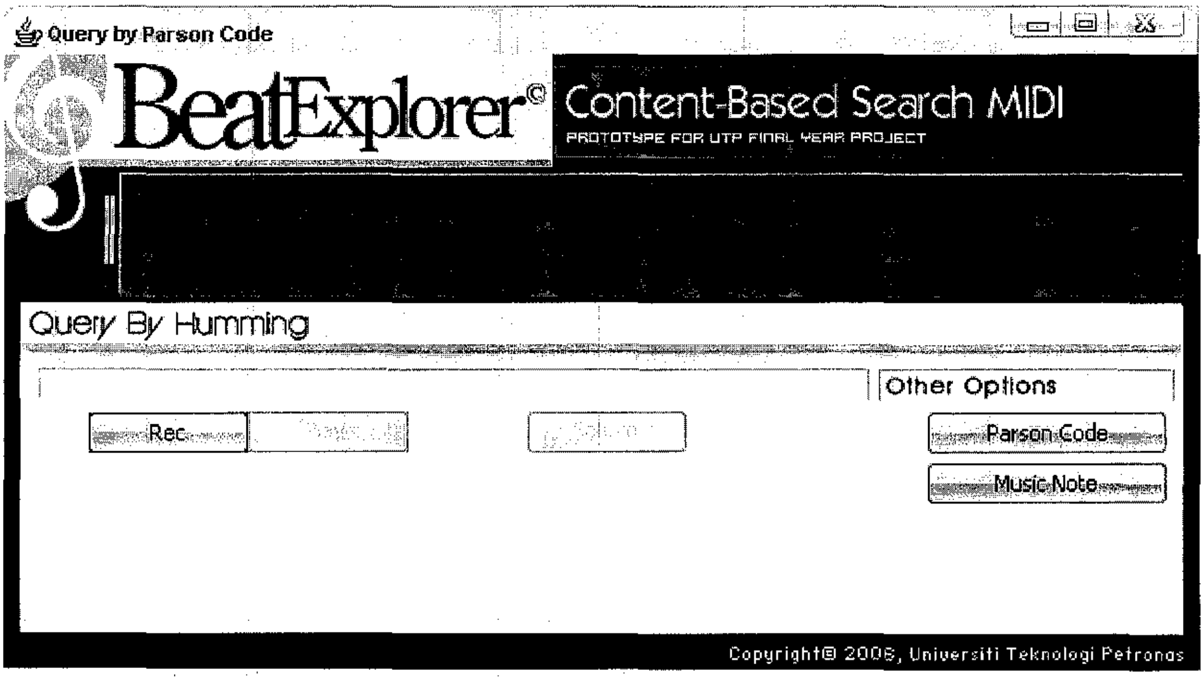
Sign	Meaning
R	If the current note pitch is equal to the previous note pitch (Remain)
U	If the current note pitch is higher than previous note pitch (Up)
D	If the current note pitch is lower than previous note pitch (Down)

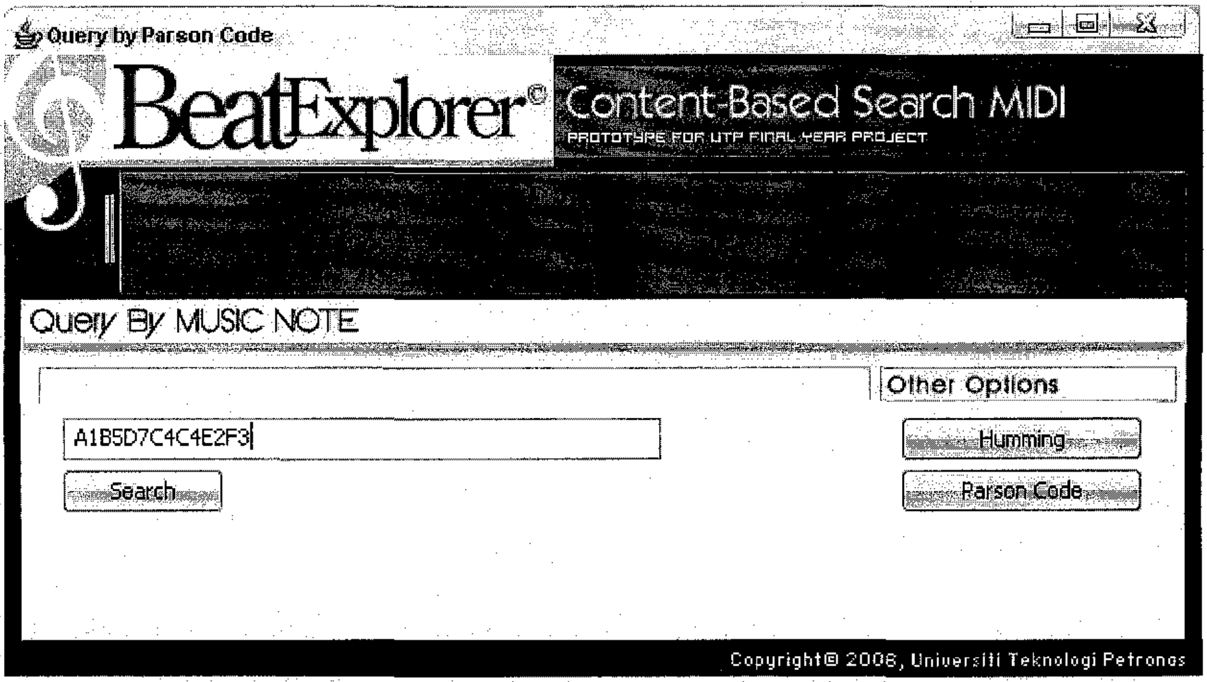
The contour of music is described as a string contains R, D, U letter, for example "DRUUUUUU" (the parson code of beginning of Yesterday- The Beatles).

4.1.4 The System

4.1.4.1 The interface of BeatExplorer






Query by Parson Code
BeatExplorer©
Content-Based Search MIDI
PROTOTYPE FOR UTP FINAL YEAR PROJECT
Query By MUSIC NOTE
Other Options
A1B5D7C4C4E2F3
Humming
Search
Parson Code
Copyright© 2006, Universiti Teknologi Petronas

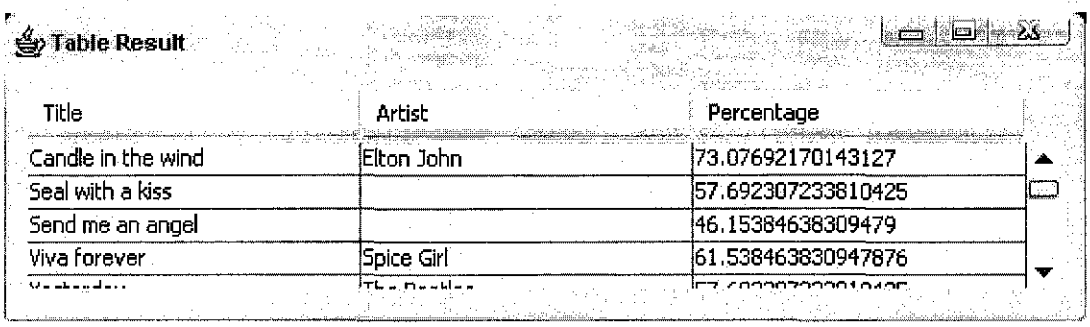


Table Result
Title
Artist
Percentage
Candle in the wind
Elton John
73.07692170143127
Seal with a kiss
57.692307233810425
Send me an angel
46.15384638309479
Viva forever
Spice Girl
61.538463830947876

4.1.4.2 System requirement of Beat Explorer

In order to run the application, the client computer needs to have:

- Java Runtime Environment (JRE) version 1.5
- All library provided

4.2 Discussion

The system is developed based on the basis of music note, humming and parson code relationship where all of them can be related using frequency. For performance, comparing the three modules in the system, parson code and music note is a better way for searching since it will result in more precise output where for music note, the frequency of music note is defined and only comparison and array manipulation is done in processing which result in less error.

The parson code has the best performance since the input is directly compared to the database. However, these two methods will only work with people who have knowledge in the music especially users who work with many music content such as composers. It is less suitable for normal and beginner user.

For query by humming, it is suitable for everyone since almost everyone can hum. In fact people remember music better than names. Although it is easier for users, it requires complex processing for the system and possibility of reliability and precision of the output will be reduced. In humming, the recording sound for searching might have interference from noise and the precision of the user.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

As a conclusion, I've meet the objective of the project which is to provide more variation of search methods using content based and improve capability of Ahn Tuan's existing application which only allow whistle. More users can use this application since it is easier to hum rather than whistle and also there are other methods which uses parson code and music note. The relation between music note, midi key and parson code make the application possible with all related in a link or chain using frequency of each form. This is applicable for music note, parson code and midi key related to frequency, it can be easily convert to each other and compared as the same entity. It also forms the basis of the application. With the usage of FFT to analyze the sound and edit distance to match the string, the application is able to meet its objective.

5.2 Recommendation

There are many ways of enhancing the project since it has lots of weaknesses. It can be improved to increase its marketability. For my recommendation, it can be expanded into more variations of music file other than midi file format. For example, by using it for mp3, wma and wav format which is commonly used nowadays. However, to there are lots of obstacles to achieve this objective such as processing power constrain since the file format is much more complex. A method to match and compare the 2 dimensional arrays stored in the FFT of the music for example and format of storing in the database. Another recommendation is to improve it by using applet and publish it on the internet and accessible by other people.

CHAPTER 6

REFERENCES

Keith D. Martin, Eric D. Scheirer, and Barry L. Vercoe (1998) "Music Content Analysis through Models of Audition". Bristol UK: MIT Media Laboratory Machine Listening Group, Cambridge MA USA.

M. Anand Raju, Bharat Sundaram* and Preeti Rao(2003)" TANSEN: A QUERY-BY-HUMMING BASED MUSIC RETRIEVAL SYSTEM" . National Conference on Communications : Department of Electrical Engineering, Indian Institute of Technology, Bombay.

Leon Fu, Xiangyang Xue (2005) "A New Spectral-based Approach to Query-by-Humming for MP3 Songs Database". Shanghai : Dept. of Computer Science and Engineering, Fudan University, China.

Department of Computer Science, UWA (1984), and (HTML) School of Computer Sci. & SWE, Monash University (1999) "Dynamic Programming Algorithm (DPA) for Edit-Distance". Australia: UWA and Monash University.

Glenda A. Miller (1992) "Music for clarinet and IPSW composed by Cort Lippe".