Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 614-628

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

iFingerstyle: Development of a Mobile Application to Generate Fingerstyle Tablature

Muhammad Syafiq Syamsuddin[1], Ezak Fadzrin Ahmad Shaubari[1]*

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.035
Received 25 July 2022; Accepted 24 May 2023; Available online 30 June 2023

Abstract: "*iFingerstyle*" is the guitar mobile application that implementing the simple User Interface (UI) and User Experience (UX), including audio recognition technology that can generate fingerstyle tablature, because the existing similar application has inexplicit instructions, and not implement the fingerstyle tablature generator feature. Next, the main research objective is to develop a mobile-based guitar application that provide audio recognition feature, while targeted users are the people who fascinated to learn about the basic of guitar. Furthermore, the methodology applied for this project is Evolutionary Prototyping model, which all the requirements are identified through survey after the planning phase and has the iteration process in the development phase that depends on the testing evaluation. In addition, the project is expected to finish successfully and all the requirements can be fulfilled according to the analysis. Thus, "*iFingerstyle*" will satisfy the target user and refer for the incoming project with similar domain.

Keywords: Audio Recognition, Fingerstyle Tablature, Evolutionary Prototyping

1. Introduction

Music is the art of composing and arranging the various sounds that produced from any physical things through its' main elements which are harmony, melody, tone quality and rhythm [1]. In the term of harmony, guitar can be harmonized into a various type of scales. For example, there has three easy ways to harmonize melodies on guitar for major scale through interval (distance between two notes) which are harmonize in thirds, fourths, and sixths. Next, according to [2] playing guitar is a form of therapy to manage stress, enhancing creativity, boosting confidence that can improve communication skills, and enables to express feelings. In Malaysia, guitarist is assumed as the talented people because majority people do not know how to play the guitar. However, there has some negative perceptions towards the guitarist such as, the guitarist had no steady income and no future [3]. Besides, because of Islam is the official religion in Malaysia, most people thought that playing the guitar is haram because

*Corresponding author: ezak@uthm.edu.my

it can make people become oblivious. However, Islam permits playing musical instruments because it recognizes that human need music to relax [4].

iFingerstyle is an application that guides beginner guitarists or enthusiasts on how to play a guitar, enhance their guitar skills and memorizing guitar chords quickly. Because most individuals can now afford a mobile phone, the application will be available on Android platform. It can also be accessible at any time and from any location. Furthermore, the design of its learning approaches is a self-taught visual and kinesthetics learning style [5], with composite navigation applied to the application's design structure.

Next, there are some issues with the currently existing applications, such as the application not loading or functioning properly when the user starts it [6]. Aside from that, while using the application, users may encounter connection or server errors, causing it to stop responding and crash [7]. The next issue is the application has inexplicit instructions that cannot guide the user properly.

The objects of the project are to: (1) design a mobile-based guitar application; (2) develop a guitar application that could provide audio recognition feature; and (3) evaluate the proposed application with target users via both alpha and beta testing. In general, this proposed application consists of two major elements, which are guitar learning module and sound recognition framework, with no age limitation for its target users so long as they are enthusiasts and are fascinated to learn about how to play guitar, including beginner guitarists.

Based on the project, there are several outcomes to be expected for the final product. The application has three modules that expected to function properly which include: (1) a guitar tuner module for calibrating and tuning user's personal guitar to the right chords and notes; (2) a chord learning module that consists solely of chords and finger indicators for each chord; and (3) a sound or chord recognition and iFingerstyle module, which would indicate what chord or strings of chord that user is currently playing.

The rest of the paper is organized as follows: Section 2 discusses the literature review of the related work and existing applications. Next, the methodology used to develop the application including the analysis and design is described in Section 3. Finally, the last section concludes the current work and highlights the future work to be performed in Final Year Project 2.

2. Related Work

This section will discuss the technology involved in the development of similar type of guitar mobile application that related to the project.

2.1 Audio Sound Recognition

Audio sound recognition is a technology that is based on both traditional pattern recognition theories and methods for analyzing audio signals. Preliminary data processing, feature extraction, and classification algorithms are all included in sound recognition technologies that can classify feature vectors produced by preliminary data processing and linear predictive coding. There are many benefits from the implementation of sound recognition technology such as music and speech recognition, automatic surveillance identification, monitoring systems based on the acoustic environment, assistance to people who are affected in hearing capabilities, and identifying species of animals. There are many algorithms involved in the development by using audio sound recognition technology. But there have two basic algorithms that have to be considered which are Discrete Fourier Transform (DFT) and Fast Fourier Transform (FFT) algorithm [8].

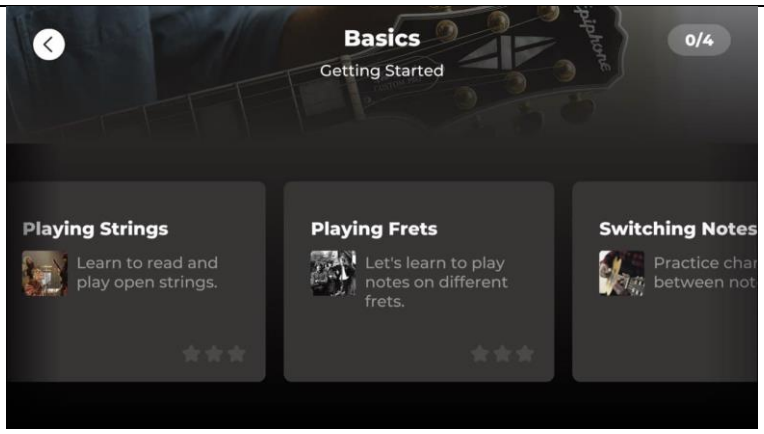
The Discrete Fourier Transform (DFT) in mathematics will transform a finite sequence of equally spaced samples of a function into a same-length sequence of equally spaced samples of the discrete-time Fourier transform (DTFT), which is a complex-valued function of frequency. DFT is a much

slower operation than FFT, but it is not affected by the number of input data. DFT can be very useful to receive a piece of frequency analysis [9].

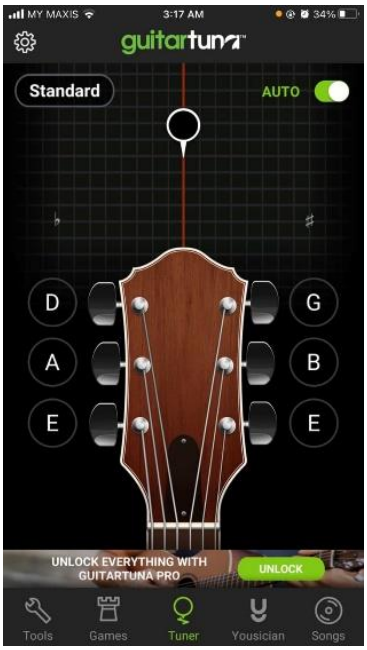
A fast Fourier transform (FFT) is an algorithm that computes a sequence's DFT or its inverse (IDFT). Fourier analysis transforms the input signal from its original domain, which is usually a time or space, to a frequency domain representation. FFT can quickly compute transformation by factorizing the DFT matrix into a scattered product with mostly zero factors [10].

2.2 Similar Guitar Mobile Application

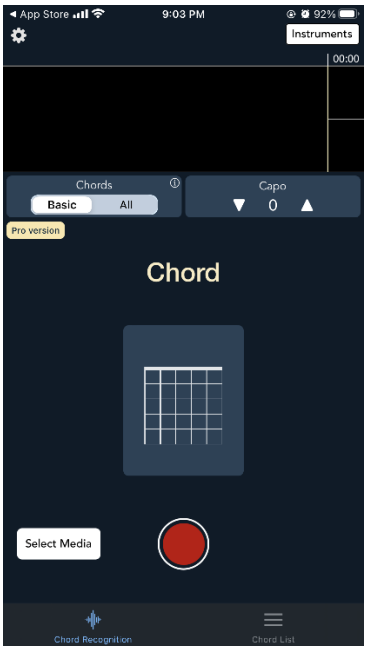
There are three similar guitar mobile application that uses Audio Sound Recognition technology such as Yousician [11], GuitarTuna [12], and Chord ai [13], are reviewed on the features and the functionality of the system. The sample interface of these three similar guitars mobile application is presented in Figure 1.



(a)



(b)



(c)

Figure 1: (a) Yousician, (b) GuitarTuna, (c) Chord ai

Furthermore, the comparison between these three applications and developed application in terms of different features such as visibility, constraints, technology used, consistency, and additional information are tabulated and shown in Table 1.

Table 1: Comparison between reviewed application and proposed application

Features	Yousician	GuitarTuna	Chord ai	iFingerstyle
Visibility	The instructions are in a text form without labels in certain modules and have colored finger indicators for the tab at the guitar's fret picture.	The instructions consist of simple labels in some interfaces and have a finger indicator for the tab at the guitar's fret picture.	The instructions are available in labels and do not have a finger indicator for the guitar's tab.	The instructions have specific labels in each module's interfaces but are not too crowded and have a finger indicator for the tab at the guitar's fret picture.
Constraints	Has different selectable tunes for different types of instruments.	Has different tunes for a guitar string that depends on the user's option and comfortable playstyle.	Recognize and display chords only for guitar, ukulele, and piano either right-handed or left-handed user.	Has only basic tuning for guitar string which is 'E', 'A', 'D', 'G', 'B', 'E' tune from the thickest string to the thinnest string.
Technology Used	Audio signal processing technology.	Audio signal processing technology.	Implementing recent deep learning algorithm which is Artificial Intelligence (AI) technology.	Audio signal processing and Neural network.
Consistency	Each module has a different background and UI. Only labels, indicators, and navigations are available with the same size and position in every submodule.	Each module including the main module has the same background and theme. The interface design for every submodule is consistent depends on its functionality.	Each interface has the same background color and each label has consistent white and peach color.	Every module has the same background and theme. The interface and navigation design for every submodule is consistent depending on its functionality.
Additional Information	The current version of the application is 4.42.1. Supported in iOS and Android version 6.0 and above.	The current and supported version of the application depends on the user's device.	The current version is 1.10.02 and is supported in iOS and Android version 8.0 and above.	The application will be supported in iOS and Android version 6.0 and above.

These guitar mobile applications utilize sound recognition technology that can generate the correct output for the captured guitar sound. These applications can help users calibrating the tuning for their guitar to keep their guitar strings in tune without using physical tuner. Based on Table 1, some limitations exist in the existing applications where they do not provide fingerstyle tablature generator. Also, these applications consist inexplicit instructions and lack of consistency for interfaces design.

Thus, iFingerstyle is developed to overcome those limitations where the application can generate the fingerstyle tablature and the users can have a better understanding for the instructions given.

3. Methodology

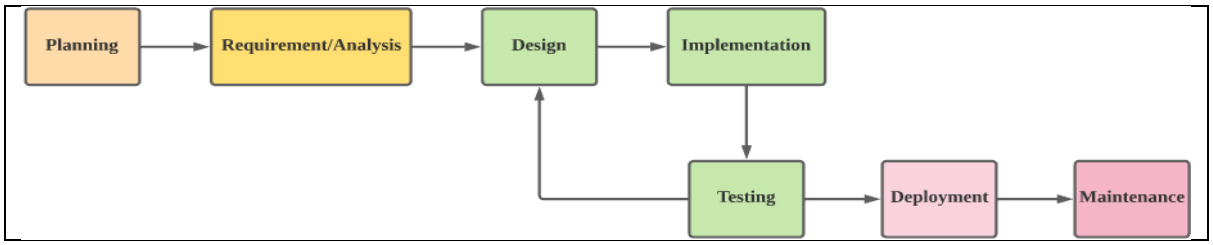


Figure 2: Adapted from Evolutionary Prototyping Software Development Life Cycle Model [14]

The Evolutionary Prototyping SDLC model is a method of software development in which the developer or development team first builds a prototype. There are seven phases involved which are the planning phase, requirement analysis phase, design phase, implementation phase, testing phase, deployment phase, and maintenance phase. Firstly, the project will be started with two initial phases which is planning and requirement analysis phase. These two phases will give more vision on how the project will be conducted and what the element required for the project to run smoothly. Then, each completion of the process flow from the design phase to the testing phase will be considered as one iteration, then the process flow will be repeated in the second, third, and 'n' iteration depends on the user feedback in testing phase. The process flow from the first iteration to the second iteration is called the incremental process. Figure 2 shows the project flow for this methodology.

3.1 Planning phase

In this phase, the initial requirements of the project include problem statements, objectives, project scope, project significance, and the expected result is determined, "see Section 1". The significance of the requirements is to gather as many information as possible about the development of the application. Moreover, the collected initial requirements are analyzed and the approval was required from the supervisor and the panel about the project progress whether the project can be executed or not. This includes the details presentation about the application and the workflow of the project.

3.2 Requirement Analysis phase

In this phase, the application and user requirements were collected by performing a similar application reviewing and user analysis. A review of the similar existing applications was conducted, and a comparison is made in Table 1 to identify strengths and weaknesses of each application. The reviews were considered as references in enhancing the proposed application. Next, questionnaires were created, "see Table 2 in Appendix A" and distributed to 30 target users. This survey helps to gather the target user's opinion and requirements on the proposed application. Besides, functional, and non-functional requirements for the application were identified in Table 3 & Table 4 respectively in Appendix A. Furthermore, suggestions and requirements from the supervisor were also accepted for the development of the proposed application.

3.3 Design phase

In this phase, the system overview and overall system flow was designed. The design of the application overview and the overall application flow is presented in the form of flowchart, in Figure 3 and a storyboard, in Figure 4 which includes the user interfaces, the interaction and navigation buttons. Other than designing the flow of the project and application, this phase also involves in designing the programming flow. In the term of programming flow, it is the process of managing the required algorithm for the project to develop the application. This process can help to avoid all the uncertainty and unwanted error when implementing the coding.

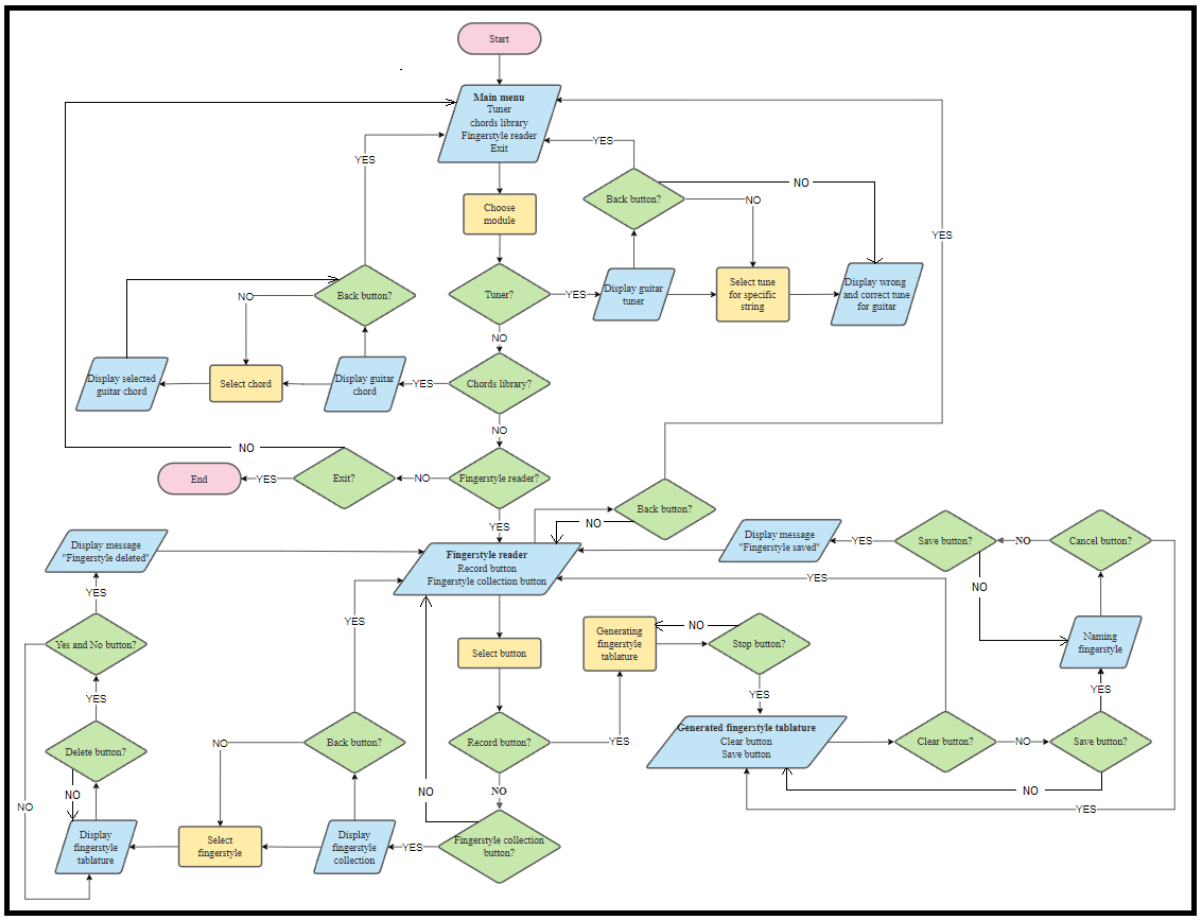


Figure 3: Application flowchart

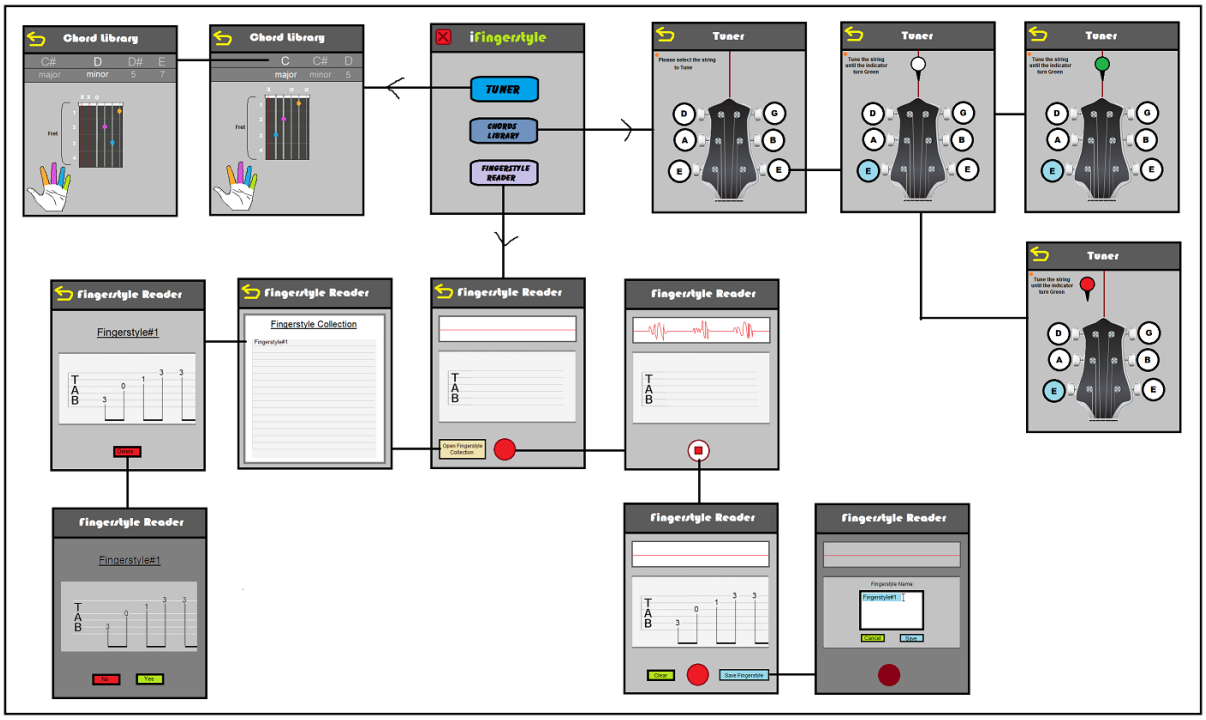


Figure 4: Application storyboard

3.4 Implementation phase

In this phase, the interfaces, functional components, and coding will be applied according to the designated system overview. There are several things to be considered which are the quality of user interfaces, the fully well-function systems and navigations, and the programming without errors or bugs. This phase is the crucial phase compared to others because it is the application development part in the project. Other than building the application, if there have any problems with the application, this phase will be executed iteratively to fix it until the perfect application is produced.

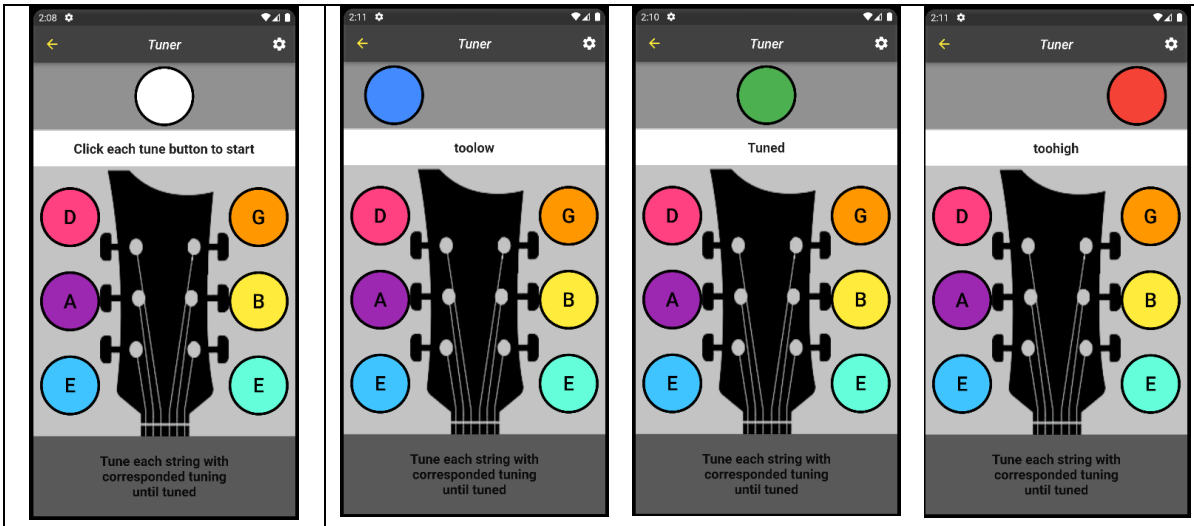


Figure 5: (a) Before start recording, (b) When recording.

Based on Figure 5, the interface of tuner module, the red round-shape icon button is the record button to capture the audio produced by the guitar string to be in tune. Initially, the button's icon will be a microphone icon and the button color will be a soft red color, meaning that the button is not pressed yet like in (a). After the button be pressed, the button's icon will animate to stop icon and the button color will be more reddish, meaning that the button has been pressed and the audio capture function is running like in (b). The button is created using 'OutlinedButton' component and as a child under a widget called 'Sizedbox'. Moreover, the state control between button's icons and colors is created using '_startCapture' component under 'icons' child function. The implementation of the button in a source code can be referred in Figure 6 below.

```
SizedBox(
        height: 100,
        width: 90,
              child: OutlinedButton(
              child: const Text("E", style: TextStyle(color: Colors.black, fontSize: 30),),
              style: OutlinedButton.styleFrom(
                    backgroundColor: Colors.lightBlueAccent,
                    side: const BorderSide(color: Colors.black, width: 4),
                    shape: const CircleBorder()
               ),
                    onPressed: _startCapture,
        ),
),
```

Figure 6: Code segment for icon buttons

Furthermore, the button function to capture audio is declare in 'onPressed' function under 'AnimatedIconButton' child. The function class was created named 'toggleIcon', to control the state either the button is capturing the audio or stopping the capture process. Next, the audio capture algorithm is using API dependencies that has several function classes to make the audio capturing process can be executed. The function class to start or stop the audio capture process is '_startCapture' and '_stopCapture', and the function class to give the result from the audio capturing process is called 'listener'. Figure 7 below showa code segment of the function class stated to execute the audio capture process.

```
Future<void> _startCapture() async {
    await _audioRecorder.start(listener, onError, sampleRate: 44100, bufferSize: 3000);
    setState(() {note = ""; status = "Play something";});}

Future<void> _stopCapture() async {
    await _audioRecorder.stop();
    setState(() {note = ""; status = "Click record button to start";});}

void listener(dynamic obj) {
    var buffer = Float64List.fromList(obj.cast<double>());
    final List<double> audioSample = buffer.toList();
    final result = pitchDetectorDart.getPitch(audioSample);
    if (result.pitched) {
        final handledPitchResult = pitchupDart.handlePitch(result.pitch);
        setState(() {
        note = handledPitchResult.note;
        status = handledPitchResult.tuningStatus.toString();});}
      else if (status == "TuningStatus.tuned"){
      _stopCapture();
      setState((){
        kalerm = Colors.green; Bkalerm = Colors.black;
        kaler1 = Colors.transparent; Bkaler1 = Colors.transparent;
        kaler2 = Colors.transparent; Bkaler2 = Colors.transparent;});
      await Future.delayed(const Duration(seconds: 3));
      setState((){
        kalerm = Colors.white; tunedSize = 8.0; tunedStat = "Click each tune button to start";});}}}
```

Figure 7: Function class to execute the audio capture process

In addition, based on figure 5 (b), the result of audio capture can be display in child 'Container' by calling the 'tunedStat' variable that has been declared in function class '_startCapture', '_stopCapture', and 'listener'. The example of the implementation is shown in code segment in Figure 8.

```
AspectRatio(
          aspectRatio: 22 / 3,
          child: Container(
            color: Colors.white,
            child: Center(
               child: Text(
                 tunedStat,
                 style: const TextStyle(
                    color: Colors.black87, fontSize: 20.0, fontWeight: FontWeight.bold),)),),),
```

Figure 8: Code segment for the implementation of audio capture

Thus, this module also has the tuning process for a left-handed guitarist. All the components in the interface are just being flip to 180 degree and all the functionality is remain the same. Besides, all the tune around the guitar head is a button that the user can press to tune the specific string and it will give the result in red and green color. Red color means the string is off tune and green color means the string

is in tune already. Also, there have setting button for user to switch between right-handed and left-handed playstyle.

3.5 Testing phase

In this phase, the prototype of the application will be given to the target users to testing its' functionality and acceptability. This concludes that this phase will be the determinant for the iteration process between the design phase and implementation phase to be occur or not. Because, if the application has problems like minor bugs or low level of performance, the design phase and implementation phase will keep in process to improve the application. Table 5 below is the alpha test plan result for all the functional and non-functional components in the application.

Table 5: The test plan result

Test activity	Expected output	Actual output
The pop-up permission to allow device's microphone	The microphone permission will successfully be allowed in the device.	The microphone permission is successfully allowed in the device.
The exit button or back button in the device will pop-up option to exit the application.	The pop-up exit message will display and exit the application when choose 'Yes'.	The pop-up exit message display and exit the application when choose 'Yes'.
All the navigation buttons to redirect to each module or interface.	All the navigation buttons will redirect correctly to each specific module interface.	All the navigation buttons can redirect correctly to each specific module interface.
The back button in application and on device to go back to the previous interface.	The back button in application and on device will go back to the previous interface correctly.	The back button in application and on device can go back to the previous interface correctly.
The record button to capture the generated audio.	The button will be animated from microphone icon and stop icon, and successfully execute the sound capture function.	The button can animate from microphone icon and stop icon, and successfully execute the sound capture function.
The container that display the result from captured audio.	The container will display the result accurately from the captured audio.	The container displays the result accurately from the captured audio.
Selecting the Chord's type while display the correct chord's output.	The chord's type list view can be selected and will display the correct result for selected chord.	The chord's type list view can be selected and display the correct result for the selected chord.

3.6 Deployment phase

This is the phase where the users will receive the developed application and using it officially. The application will be deployed at app store for Android and iOS device. Moreover, the only thing to consider in this phase is manage to deploy the application officially through selected platform without any problem.

3.7 Maintenance phase

The Maintenance phase is the last phase in the Evolutionary Prototyping SDLC model. Basically, the function of this phase is to monitor the released application and consider the feedback from the users. Besides, there must have some overlooked error from the released application. Thus, the feedback from the users is important to improve certain part of the application and keep the application in update.

4. Result and Discussion

User acceptance test, or so be called beta testing, is testing the functionality of the developed application on the real device of some target user for the project. The objective of this testing is to get the feedback from the users about the efficiency of the application when running on their own device. The feedback then can be analyzed either the application has bugs or errors to be fixed before deploying the application at application publisher platform like Google Play Store. For this user acceptance test, the application feedback is fully collected by conducting a questionnaire, which has at the very least, 20 respondents of the target user. The feedback result is in Table 6 below to identify the functionality of the developed application.

Table 6: The user feedback

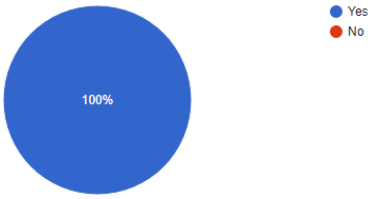
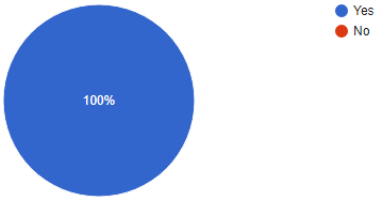
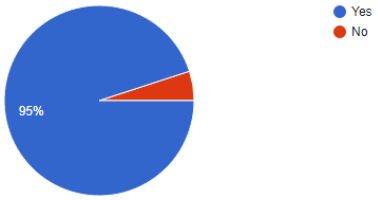
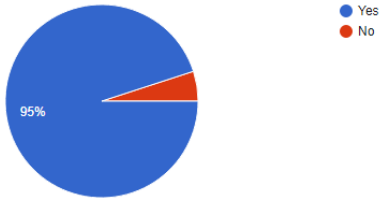
Feedback receive	Analysis
 The microphone can be allowed in the phone. • Yes (100%) • No (0%)	There has no problem with the microphone permission in all Android devices.
 The Exit or back button to exit, displayed the pop up yes/no message and functioning well. • Yes (100%) • No (0%)	There has no problem with the exit function to quit the application.
 All the buttons in the main menu can be interacted and correctly redirected to selected section. • Yes (95%) • No (5%)	There may have some bugs when navigating to any of each module.
 The record button in tuner and fingerstyle reader section can accurately detect the string sound and give the accurate result. • Yes (95%) • No (5%)	There may have some bugs or errors in the codes. Otherwise, there have to be user's microphone problem, which is too sensitive or too insensitive to detect the captured audio.

Table 6: (cont).

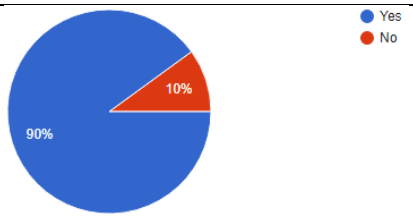
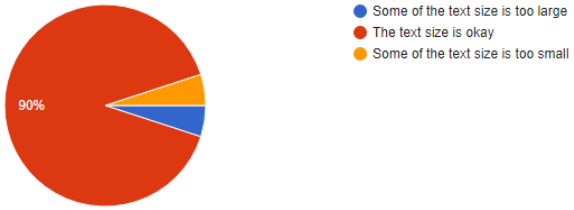
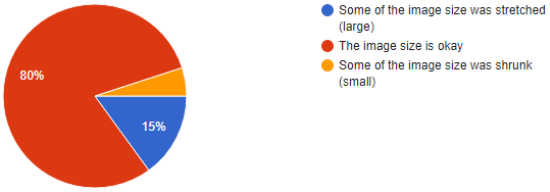
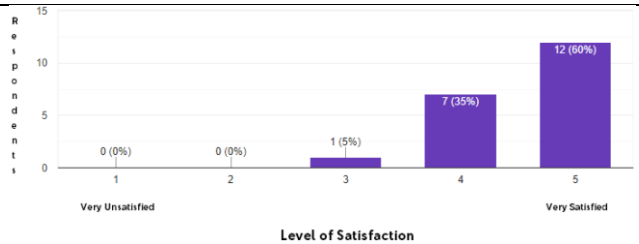
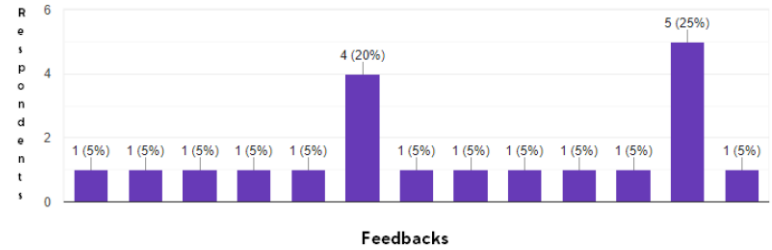
Feedback receive	Analysis
 The chords selection button in chord library section can be pressed and scroll horizontally while display the correct output for the selected chord. • Yes (90%) • No (10%)	There may have some bugs or errors in the codes. Otherwise, there have to be user's screen problem, which is too sensitive or not responsive to detect the finger sensor.
 What is your opinion about the text size in the application? (specifically, for any instructions). • Some of the text size is too large (5%) • The text size is okay (90%) • Some of the text size is too small (5%)	The text size may not synchronize with some of the device that has a small screen or big screen.
 What is your opinion about the image size in the application? • Some of the image size was stretched (large) (15%) • The image is okay (80%) • Some of the image size was shrunk (small) (5%)	The image size may not synchronize with some of the device that has a small screen or big screen.
 In the scale of 1 to 5, what are your level of satisfaction while using the application? • 1 – Very unsatisfied (0%) • 2 – Unsatisfied (0%) • 3 – Average (5%) • 4 – Satisfied (35%) • 5 – Very satisfied (60%)	There have some respondents that not fully satisfied with the application functionality. Maybe due to the text or image sizing, or microphone and screen problems.

Table 6: (cont).

Feedback receive	Analysis
 • Any feedbacks about the application can be written here. If none, type 'None'.	Most of the respondents give feedbacks about the application is easy to use, the application is interesting, and the application is well designed.

5. Conclusion

There are several progresses that has been done in this semester which are developing the application, conduct the functionality testing through alpha testing and beta testing, fixing the bugs and errors by implementing the solution codes, and deploy the application at application publisher platform like Google Play Store.

The project has fulfilled all the significances and objectives that are stated at the early state of the project development. The benefits of the project include gaining new knowledge on how to implement sound recognition features that can generate a programmed result based on the recorded sound are absolutely a huge achievement for the project because it also fulfils the objective of the project, which to develop a guitar application that could provide audio recognition technology. Therefore, the sound recognition features were successfully applied in the project application without any flaw by calling the API's functions strategically. Furthermore, the new simple and understandable interfaces design has provided the best experience and optimal efficiency of self-learning for users to learn from the mobile application, due to a good interaction design between the user and the system. Moreover, the costs of the application are free to download by any user and all features can be accessed, also there is no development costs for the project.

In the future, the developed project may be improved based on the recent technology at that time. For example, maybe there will has the algorithms to detect the guitar fingerstyle tabs through video recording or Augmented Reality (AR). Also, the sound recognition technology can be improved until it can detect all the guitar notes or tabs separately without any errors in storing each frequency individually. Thus, if the stated technology will exist, then the audio capture technology is the best technology to produce any audio recognition system that ever exist at that time.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support.

Appendix A

Table 2: The questionnaire

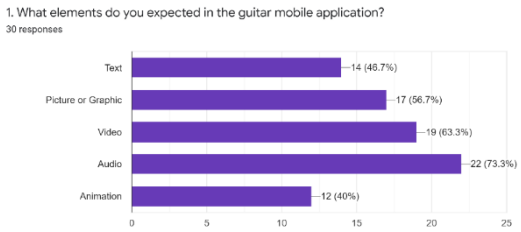
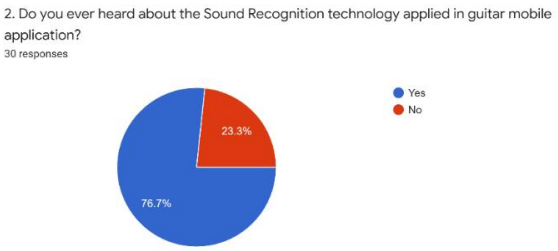
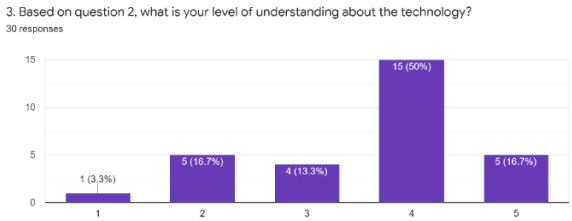
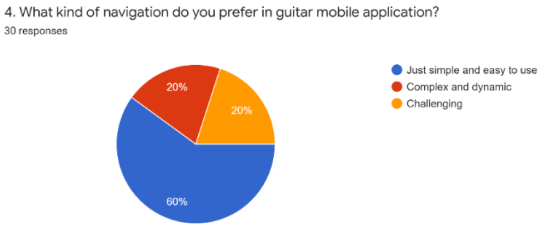
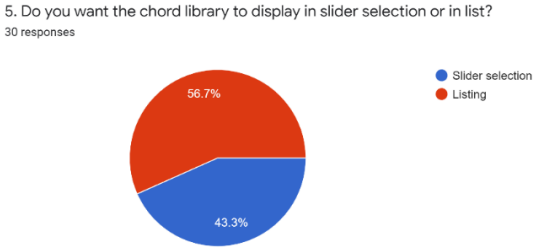
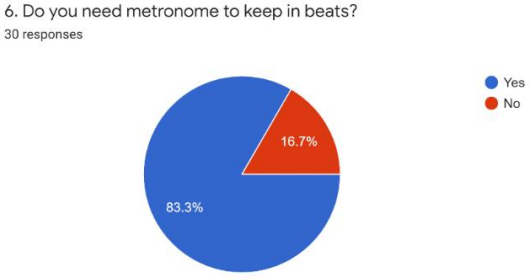
Feedback receive	Feedback receive
 What elements do you expected in the guitar mobile application? • Text (46.7%) • Picture or Graphic (56.7%) • Video (63.3%) • Audio (73.3%) • Animation (40%)	 Do you ever heard about the Sound Recognition technology applied in guitar mobile application? • Yes (76.7%) • No (23.3%)
 Based on question 2, what is your level of understanding about the technology? • Not understand at all (3.3%) • Not understand a bit (16.7%) • Maybe (13.3%) • Understand (50%) • Very understand (16.7%)	 What kind of navigation do you prefer in guitar mobile application? • Just simple and easy to use (60%) • Complex and dynamic (20%) • Challenging (20%)
 Do you want the chord library to display in slider selection or in list? • Slider selection (43.3%) • Listing (56.7%)	 Do you need metronome to keep in beats? • Yes (83.3%) • No (16.7%)

Table 3: Functional requirements

Category	Description
Autonomous System Activities	• While recording each guitar string in guitar tuner, the system shall generate the correct tune for the real-time recorded sound. • While recording the fingerstyle sound in iFingerstyle, the system shall generate the correct tablature for the real-time recorded sound. • While running iFingerstyle for the first time, the system shall ask the user for the microphone permission.
User Interaction	• The system shall provide users with the ability to navigate through the application by using appropriate buttons. • The system shall provide users with the ability to display the guitar chord based on the selected guitar chord. • The system shall provide users the ability to record and generate the correct output for the recorded sound. • The system shall provide users with the ability to give input by touch screen on their Android mobile platform. • The system shall provide users with the ability to manage fingerstyle collection through delete and cancel button.

Table 4: Non-functional requirements

Category	Description
Performance	• The generate time of recording sound should be in real-time. • The response time of the button should not be more than one second for users. • The application shall be able to operate completely offline.
Cultural	• The application will use a simple English language including clear instruction.
Usability	• The application's interface has to be user-friendly and easy to use.
Operational	• The application shall be able to operate on any android device as long as it is Android 7.0 and greater.
Implementation	• The application shall be able to run on any Android mobile device.

References

[1] M.C. (2021, August 10). Music 101: What Is Harmony and How Is It Used in Music? MasterClass. Retrieved December 1, 2021, from https://www.masterclass.com/articles/music-101-what-is-harmony-and-how-is-it-used-in-music

[2] Davies, C. (2020, January 24). 8 reasons why playing guitar is good for your mental health. MusicRadar. https://www.musicradar.com/news/8-reasons-why-playing-guitar-is-good-for-your-wellbeing

[3] Asyraf, F. (2021, December 21). Music of the streets. Malaysiakini. https://www.malaysiakini.com/news/527545

[4] Andrew, B. (2020, January 26). Is Playing A Guitar Haram (Forbidden by Islamic Law)? Guitar Unit. Retrieved December 1, 2021, from https://guitarunit.com/is-playing-a-guitar-haram-forbidden-by-islamic-law/

[5] Phuong Do. 2021, 4 types of learning styles: How do students learn best? Bau. Bay Atlantic University - Washington, D.C., Phuong Do., viewed 3 October 2021, https://bau.edu/News/types-of-learning-styles/

[6] Yousician, Oy. (2021). Guitar sound recognition issues. Yousician. https://support.yousician.com/hc/en-us/articles/201576782-Guitar-sound-recognition-issues

[7] Andoapp, O. 2021, learning to play guitar app not working (down), White Screen / Black (blank) screen, loading problems (2021). 99images, Andoapp, O, viewed 3 October 2021, https://www.99images.com/apps/education/com.playguitar.myandroapps/problems

[8] Sharan, R. V., & Moir, T. J. (2016). An overview of applications and advancements in automatic sound recognition. Neurocomputing, 200, 22-34.

[9] Winograd, S. (1976). On computing the discrete Fourier transform. Proceedings of the National Academy of Sciences, 73(4), 1005-1006.

[10] Cochran, W. T., Cooley, J. W., Favin, D. L., Helms, H. D., Kaenel, R. A., Lang, W. W., ... & Welch, P. D. (1967). What is the fast Fourier transform? Proceedings of the IEEE, 55(10), 1664-1674.

[11] (2021). Yousician (Version 4.42.1) [Mobile app]. Retrieved from Google Play Store. https://play.google.com/store/apps/details?id=com.yousician.yousician&hl=en_US&gl=US

[12] (2021). GuitarTuna (Version 7.0.2) [Mobile app]. Retrieved from Google Play Store. https://play.google.com/store/apps/details?id=com.ovelin.guitartuna

[13] (2021) Chord ai (Version 1.10.02) [Mobile app]. Retrieved from Google Play Store. https://play.google.com/store/apps/details?id=com.chordai&hl=en&gl=US

[14] Sami, M. (2020, December 16). Software Development Life Cycle Models and Methodologies. Mohamed Sami. Retrieved October 3, 2021, from https://melsatar.blog/2012/03/15/software-development-life-cycle-models-and-methodologies/