Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 657-675

AITCS

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

Development of Mobile Learning Application for Chinese Pinyin with Virtual Assistant

Hai Yen Nee[1], Noorhaniza Wahid[1*]

[1]Fakulti Sains Komputer dan Teknologi Maklumat,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.038
Received 16 June 2022; Accepted 26 May 2023; Available online 30 June 2023

Abstract: Hanyu Pinyin is the official romanization of the Mandarin Chinese language. It helps both native and non-native Chinese speakers to learn and understand the Mandarin language. The existing Google Play applications for learning Hanyu Pinyin lack of multimedia elements such as graphics and animations and have yet to be implemented with a virtual assistant such as speech recognition. Therefore, this project is proposed to develop a mobile learning application namely Learn PINYIN that helps the beginner to learn Pinyin. The target users for this project are children aged 7 to 12. Multimedia Mobile Content Development (MMCD) approach is used to develop the Learn PINYIN application. According to user acceptance testing, positive results were obtained with an average of 37.22% of respondents agreeing and 38.70% strongly agreeing with the statements. This application is expected to assist the learners to learn the fundamental of Pinyin, and the addition of a game module and a practice module for speech recognition are suggestions for future work on this project.

Keywords: Mobile Learning Application, Hanyu Pinyin, Virtual Assistant, Speech Recognition

1. Introduction

Across the world, English is the most widely spoken language, followed by Mandarin Chinese language, whether spoken natively or as a second language [1]. Similar to the English phonetic transcription, the Mandarin Chinese language uses Hanyu Pinyin (commonly abbreviated as Pinyin) as the pronunciation specification. Unlike consonants and vowels in English, the basic elements of Pinyin are initials, finals, and tones. Nevertheless, it is difficult for learners to address pronunciation errors in second language class [2]. This is because some components, such as the tones in Pinyin, do not exist in the learners' native language. Thus, the guidance of educators is needed to help Mandarin beginners. Applying the virtual assistant feature is a helpful tool to guide learners in learning pronunciation.

Virtual assistant technology for speech recognition such as Amazon Alexa, Apple's Siri, Google Assistant, and Microsoft Cortana has become popular in recent years. Speech recognition, often known

*Corresponding author: nhaniza@uthm.edu.my

as speech-to-text translation, is one of the most crucial components of such assistants [3]. These virtual assistants are developed with algorithms that interpret spoken language and respond to commands given by the user in spoken language. Speech recognition technology is capable of giving immediate feedback to users [3], allowing learners to practice and test their pronunciation.

Furthermore, existing Chinese Pinyin learning applications that are available in the Google Play Store only use static graphics and audio. However, the study shows that dynamic content, such as animations, is important to attract the user's concentration, especially for children [4]. Meanwhile, the virtual assistant like speech recognition feature is also yet to implement in the existing applications to learn Pinyin. Therefore, the mobile learning application, Learn PINYIN is proposed to be developed.

The objectives of this study are to design the content of Learn PINYIN by implementing a VARK learning style, to develop an interactive learning application by implementing a virtual assistant with speech recognition technology, and to perform functional testing and user acceptance test on the developed application to the target user. The application is focused on users aged 7 to 12 years old. The subject matter expert (SME) participating in this project is Teacher Melissa, who teaches Chinese subjects at Lambano Learning Centre. Furthermore, the VARK model implemented in this project will only focus on visual and auditory learning styles.

The Learn PINYIN application contains 4 learning modules such as History, Initials, Finals, and Tones, 3 Exercise modules including Initials, Finals, and Pinyin, and 1 Virtual Assistant module which provides speech recognition. All interactive buttons in the application are expected to perform well. The sound buttons are provided as part of the learning content. Clear and accurate pronunciation of Pinyin is provided to ease the user to follow the learning content. The exercises should respond to the user based on the chosen answer, as well as record the user performance as the score. Lastly, the Virtual Assistant module's microphone button is expected to recognize the user's pronunciation. The application is then required to respond to the user based on the detected pronunciation.

The rest of the paper is arranged as follows: Section 2 covers the domain of study, the technology used, and the result of the comparative analysis. Section 3 describes the Multimedia Mobile Content Development (MMCD) methodology and the output of each phase of this project. Furthermore, Section 4 presents the results and discussion, and Section 5 states the conclusion of the project.

2. Related Work

This section discusses the study domain, technology used, and result of the comparative analysis.

2.1 Hanyu Pinyin

Hanyu Pinyin is a system that uses Roman alphabet letters and lexical transcriptions of tones to spell out the pronunciation of Chinese words [4]. Pinyin has made the Chinese language easier to learn and understand by non-native Chinese speakers [5]. Nevertheless, simply mastering the 26 Roman letters is insufficient. This is because each element of Pinyin syllables has unique representations of initials, finals, tones, and rules of spelling [6]. Pinyin uses the same letters as the English alphabet, except for the letter 'v' and the addition of 'ü'. However, the sounds of these letters in Mandarin differ from the Roman alphabet sounds in other languages [5]. For instance, the initials 'j', 'q', and 'x' and the finals 'an' and 'ang' represent sounds that do not correlate with any pronunciation in English. Besides, the spelling rules for pinyin also need to be followed. For example, the tone sign is required to be placed following the order of 'a', 'o', 'e', 'i', 'u', and 'ü'. Using the Chinese word '老', which means 'old' as an example, the Pinyin is written as 'lǎo'. The tone sign is placed on the letter 'a' instead of the letter 'o'. Only when the learners truly master the characteristics of Pinyin, they will be able to spell and pronounce it correctly. The next subsection explains the technology applied in the application.

2.2 Technology Used

Three technologies have been applied in this project. Firstly, m-learning technology, which stands for mobile learning. It has become the extension of e-learning (electronic learning) that benefits from mobile technology [7]. The portability of m-learning allows the teaching and learning process to expand beyond the confines of a traditional classroom. This is because m-learning allows users to learn from anywhere and at any time. Compared to the past, mobile technology has made a great contribution to language learning, as it has become more attractive [8]. The activities provided by the learning applications, such as instant navigation, multiple choices, and feedback, provide an engaging learning experience for students. Furthermore, enhancing the interactivity of the applications with audio, animation, and video also attracts the interest of students. A popular application such as Duolingo [18] applied m-learning technology to allow users to learn languages.

Secondly, virtual assistant is a widespread technology on digital devices that can translate human voices into words [9]. It works by implementing speech recognition, which converts human words into electrical signals that are then transformed into coding patterns. Therefore, the spoken words can be used as input to a computer. Nowadays, speech recognition technology has provided numerous benefits in a variety of areas. For example, voice-activated assistance such as Amazon's Alexa and the Google Home device to play music and answer questions [10]. Furthermore, there are mobile entertainment applications applied with speech recognition technology, such as My Talking Tom 2 [19] that repeat the speaker's words. Additionally, the advantage of speech recognition technology is that it provides instant feedback, allowing learners to have an engaging learning experience using mobile devices. For example, an application called "Speak and Translate Languages" [20] allows the user to speak a word, and the application instantly translates the word into the language of the user's choice.

Thirdly, the VARK model of learning style, developed by Neil Fleming, a well-known learning style theory [11]. It consists of visual, auditory, reading or writing, and kinesthetic. Visual and auditory are always the essential elements for mastering a language. The auditory is the beginning point for a learner to recognize and distinguish a language [12]. Meanwhile, the visual provides the information that corresponds to what is heard. This information obtained will then be processed in the mind and thus become the essential process to mastering a language. When the learning process is based solely on textbooks, learners become less engaged in attending to the learning process and attaining less meaningful learning experiences [4]. Thus, a mobile application that can offer visual and auditory learning media such as graphics, sounds, and animations will provide a different learning experience for the students and attract the interest of students. The comparative analysis of existing applications and the proposed application are discussed in the next subsection.

2.3 Comparative Analysis

In this section, a comparison has been made between existing applications, such as Pinyin Drill [13], Chinese PinYin [14], and Pin Yin [15], and the proposed application. Figure 1 shows the main menu interface of the three existing applications. Meanwhile, seven features have been discussed, as shown in Table 1. It includes the operating system, virtual assistant, quiz and exercise modules, content, multimedia elements, metaphor, and advertisement.

Based on Table 1, several strengths and limitations of the proposed application can be concluded. Firstly, a virtual assistant module with speech recognition technology will be applied to the proposed application. Secondly, the proposed application will be improved by covering all components of Pinyin, such as initials, finals, and tones with an English explanation in the learning content. Thirdly, the existing applications only applied various multimedia elements such as text, graphics, and audio. Therefore, to improve users' engagement with the proposed application, animations will be applied to the learning modules. This is because animation improves the application's interaction and attracts users' attention. On the other hand, the proposed application does not support iOS devices and only supports mobile devices with Android version 5.0 and above. Also, it does not contain a game module. Instead, Learn PINYIN only provides the exercise module.

Figure 1(a): Pinyin Drill [13] Figure 1(b): Chinese Pinyin [14] Figure 1(c): Pin Yin [15]

Table 1: Comparison between existing applications and proposed application

Features	Pinyin Drill	Chinese Pinyin	Pin Yin	Learn PINYIN
Operating System	Available for both iOS and Android systems.		Only available for Android systems.	
Virtual Assistant	Do not contain virtual assistant technology.			Contain a virtual assistant module (speech recognition).
Quiz and Exercise Modules	Contains only a quiz module.	Contains both exercise and quiz modules.	Does not contain both modules.	Contain only an exercise module.
Content	Covers 4 tones for all Pinyin syllables without explanation.	Covers all initials, finals, and tones with only Chinese explanation.	Covers only the initials and finals of Pinyin without explanation.	Covers all initials, finals, and tones of Pinyin with an English explanation.
Multimedia Elements	Text and audio.	Text, graphics, and audio.	Text and audio.	Text, graphics, audio, and animation.
Themes and Metaphor	Tools and instruments theme.	A green colour theme.	A green colour theme.	Chinese culture and calligraphy theme.
Advertisement	Applied at the bottom of the application.	Applied at the bottom of the application.	Applied when the application is launching.	No advertisement.

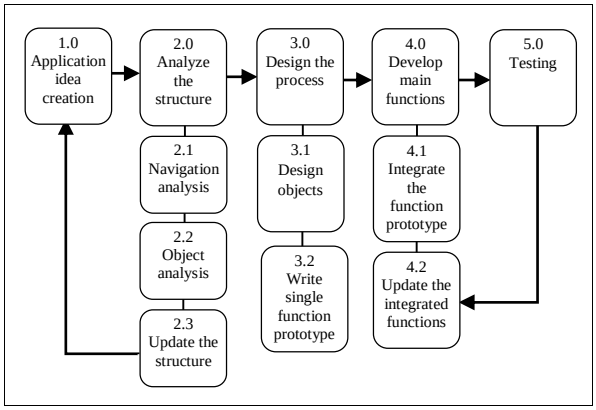


Figure 2(a): MMCD Methodology [16]

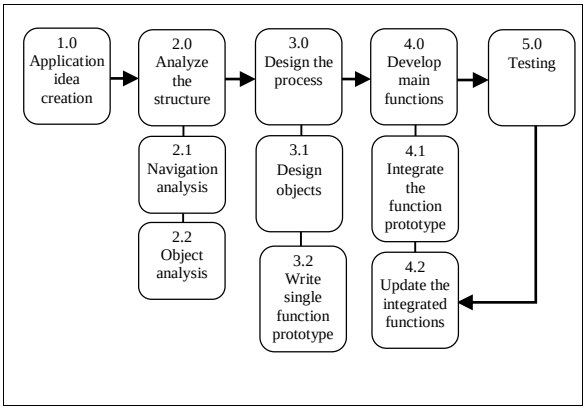


Figure 2(b): Modified MMCD Methodology

3. Methodology

Learn PINYIN is a mobile learning application designed as a tool to learn Pinyin. Thus, the Multimedia Mobile Content Development (MMCD) [16] approach is chosen to develop this application with some modifications as shown in Figure 2(b) [16]. The 5 phases of the MMCD methodology are shown in Figure 2(a). Meanwhile, in Figure 2(b), the substage named "update the structure" has been removed from the MMCD for this project. This is due to this project is not planned to be updated and returned to the application idea creation phase after completing the second phase. Instead, after the completion of the navigation and object analysis, the project will proceed to the process design phase. Next, each phase of the MMCD is discussed in the following subsections.

3.1 Application Idea Creation

Application idea creation is the first phase of MMCD. In this phase, the information required before the design and development of the Learn PINYIN is determined. Two information gathering methods have been used to identify user requirements. Firstly, an interview session was conducted with a Subject Matter Expertise (SME), Teacher Melissa. Additionally, a set of questionnaires was prepared and distributed to target users via the Google Form. A total of 25 responses from children aged between 5 and 12 years have been collected as attached in Appendix A.

For the user analysis, the highlighted issues are user experience analysis and preferable multimedia analysis. The user experience analysis revealed that 11 respondents stated that they have Mandarin basics. At the same time, 13 respondents stated that they know what Hanyu Pinyin is. Nevertheless, the 17 respondents have agreed that Hanyu Pinyin is hard to learn. 15 respondents also agreed that Hanyu Pinyin is hard to understand by reading books. Additionally, 11 respondents expressed interest in learning Hanyu Pinyin, while 11 respondents have a neutral attitude toward learning Hanyu Pinyin. Meanwhile, the preferable multimedia analysis recorded animation is the most preferred multimedia element with a total of 21 respondents (84%). It is followed by the audio with 17 respondents (68%). Meanwhile, the option 'text' recorded the least number of respondents, with only 4 people (16%) selected. To summarize, the majority with 80% of the respondents preferred to use mobile devices and the English language to learn Pinyin, and 56% of the respondents agreed to include an exercise module in the application. The results of the user analysis are tabulated in Table 2. Table 3 shows the application idea creation checklist.

Table 2: User analysis

Stakeholder category	Role in product	Design implication	Action needed
Subject Matter Expertise (SME)	Content consultant expert on Mandarin subject	Based on the interview, simple user interface design	• Use icon-based buttons instead of text buttons. • Use font types that can show the tone sign of Pinyin, such as Now font types. • Navigation buttons should be consistent in terms of shape and size.
		reliable content, use simple words and short sentences	• Add the rules for writing the Pinyin tones to the learning content of the Tones module. • Include 24 Pinyin finals instead of 36 for the Finals module. • Speech recognition shall not be too sensitive. • Avoid complicated words and long sentences.
		easy to navigate	• Use the previous, next, and home buttons to navigate to relative pages.

	multimedia content	• Use audio and animation to attract children's interest.

Table 2: (cont).

Stakeholder category	Role in product	Design implication	Action needed
General User	End-user of the proposed application	Based on the questionnaire, the user preferences	• The application should be developed on mobile devices. • The application should be developed in English. • Audio and animation should be implemented in the learning content.

Table 3: Application idea checklist

Item	Description
Type of application	Mobile learning application.
Target device	Android-based smartphone.
Target users	Children between 7 and 12 years of age.
Graphic User Interface (GUI)	Background of the main menu, History module, Initials module, Finals module, Tones module, Exercise module, Virtual Assistant module.
Images	Icons, buttons, and backgrounds.
Animation	Learning content.
Audio	Pronunciation of Pinyin.
Application synopsis	Learn PINYIN is a mobile learning application that allows users to learn components of Pinyin such as initials, finals, and tones. The application provides audio and animation to deliver the learning content. This application covers only 23 initials, 24 finals, and 5 Pinyin tones. The application also includes a Virtual Assistant module that uses speech recognition technology to allow users to practice the pronunciation.

3.2 Analyze the Structure

In the second phase of the MMCD methodology, the structure of the application to be developed is analyzed. The object and navigation analysis has been conducted. Functional and non-functional requirements are listed in Table 4 and Table 5. Figure 3 shows the navigation structure and Figure 4 shows the system flowcharts. Meanwhile, the remaining flowcharts of the modules are presented in Appendix A. The content structure is shown in Figure 5.

Table 4: Functional requirements

Functional requirements	Description
Autonomous system activities	• The animation should play when users enter the Initials and Finals modules. • When answering the questions in the exercise module, the application shall determine whether the users selected the correct answer. • At the end of the exercise module, the application should calculate the total score of the exercise. • After the user pronounces Chinese words, the application should determine whether the pronunciation is accurate.

User interaction support	• The application should allow users to select a module from the main menu. • The application should allow users to click on the audio button to listen to the pronunciation in the learning modules. • The application should allow users to input by clicking on the answers to the exercise in the exercise modules. • The application shall allow users to give input to pronounce the words by clicking the microphone button in the virtual assistant module.

Table 4: (cont).

Functional requirements	Description
Provide learning content	• The application should allow users to learn a total of 23 initials of Pinyin. • The application should allow users to learn a total of 24 finals of Pinyin. • The application should allow users to learn the 5 tones of Pinyin.

Table 5: Non-functional requirements

Non-functional requirements	Description
Performance	• The application should be able to load all the modules. • The average response time between click and reaction shall be 3 seconds.
Legal	• Users should be able to view, but not modify the content of the application.
Usability	• Users should be able to access the application from anywhere and anytime. • Simple words should be used to deliver the learning content.
Operational	• The application shall be able to operate on Android mobile devices with Android version 5.0 and above.

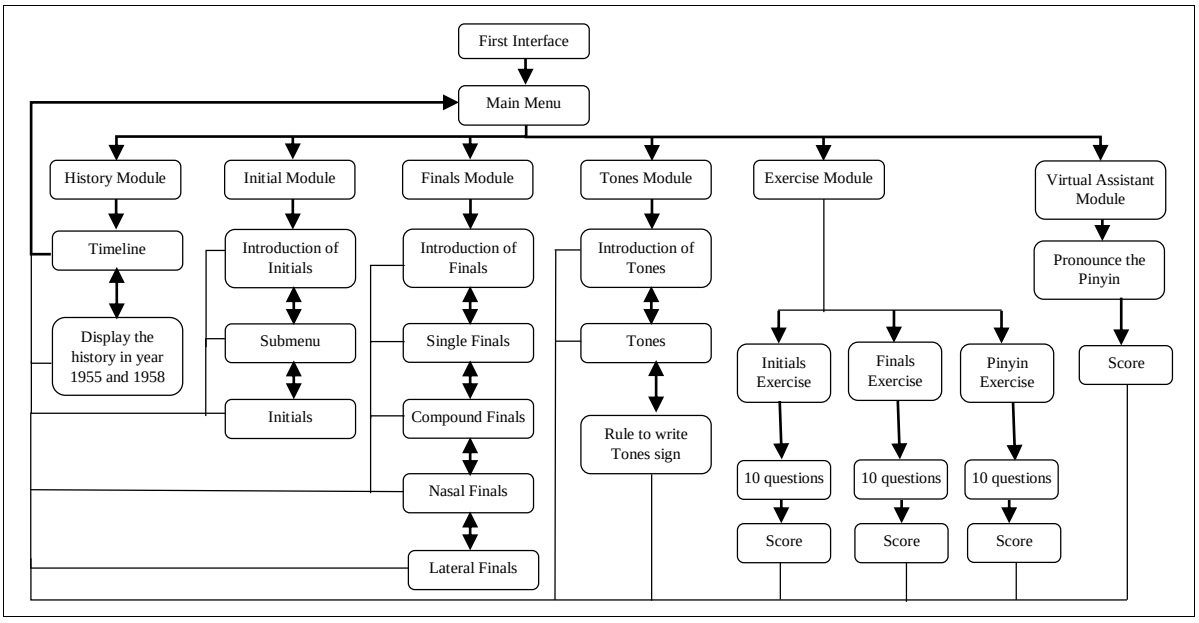


Figure 3: Navigation structure

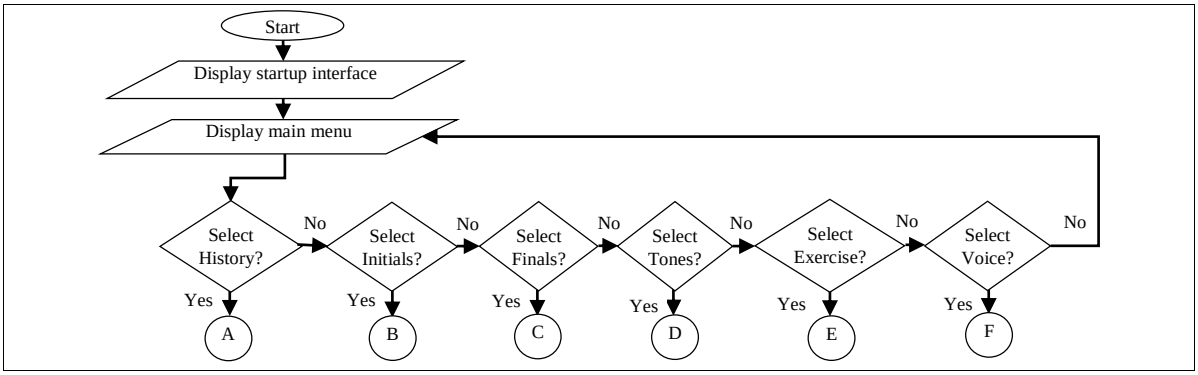


Figure 4: System flowchart

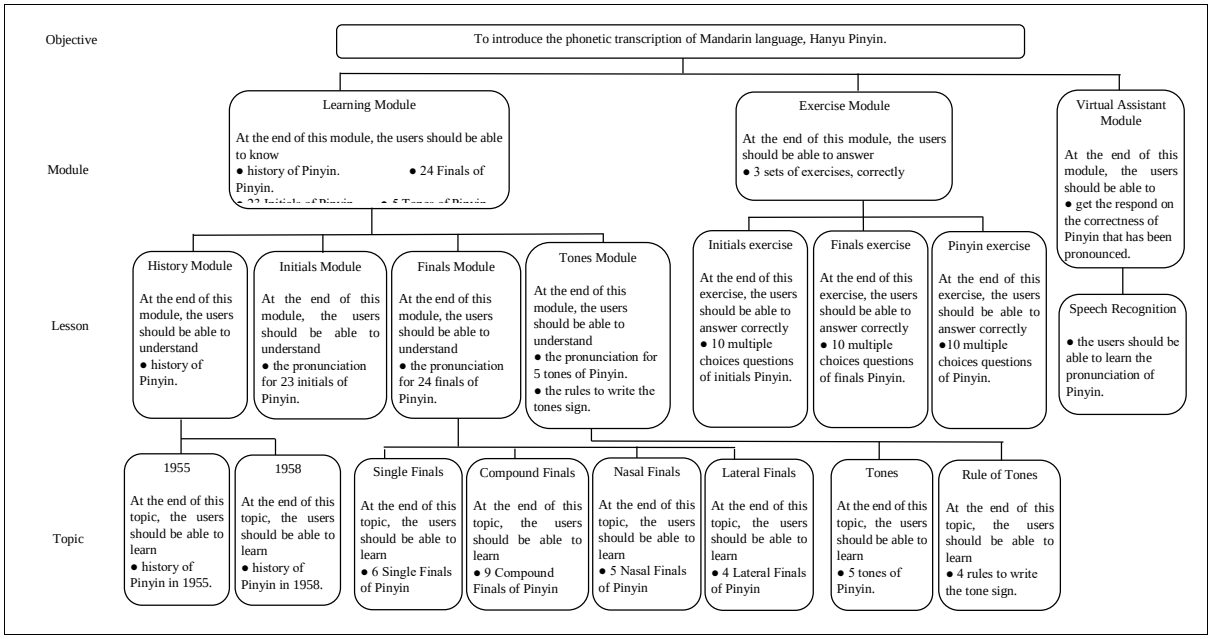


Figure 5: Content structure

3.3 Design the Process

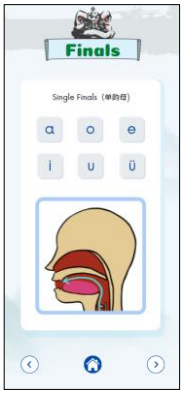
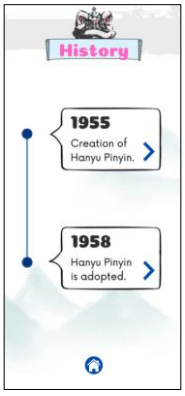
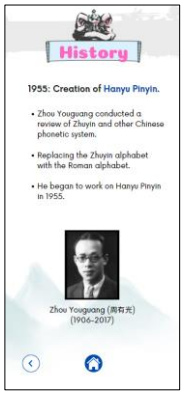
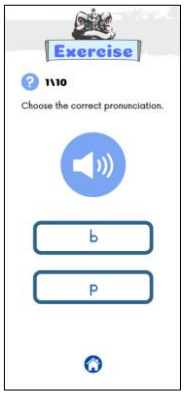
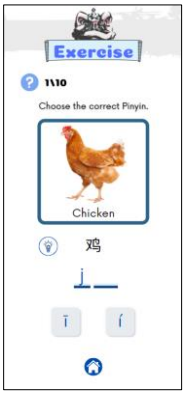
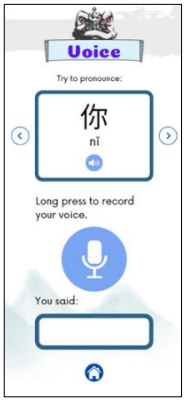
Design the process is the third phase of the MMCD approach. Design objects and write the single function prototype scripting are two subphases in this stage. The prototype for the History module, Initials module, and Finals module will be completed at the end of this process. In this project, authoring tools such as Adobe Photoshop, Adobe Animate, Blender, and Canva are used to create images, 2D and 3D animations, and storyboards. Meanwhile, Unity software is used to compile the assets with scripting. Table 6 shows the button design, while Table 7 shows the interface design.

Table 6: Button design

Button	Description	Button	Description
	History Module button		Tones Module button
	Initials Module button		Exercise Module button
	Finals Module button		Virtual Assistant Module button
	Home button		Hint button

	Next button		Audio button
	Previous button		Microphone button
	Retry button		Close button

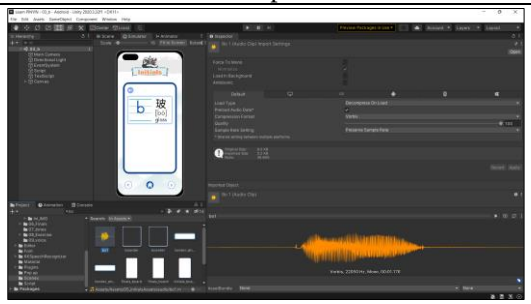
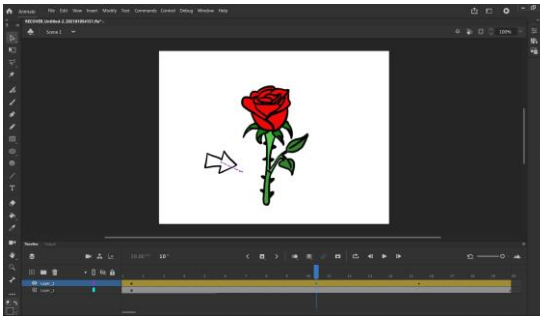
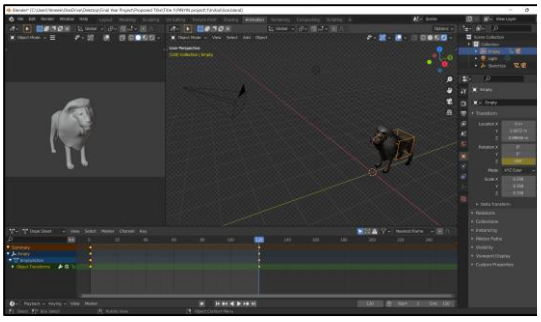
Table 7: Interface design

Interfaces	Description	Interfaces	Description
 	This is the startup interface and scrollable main menu. The main menu contains six buttons, and each button navigates to each module, respectively.	 	The interfaces of the Initials and Finals modules. Finals are classified into 4 types. Each initial and final can be clicked to listen to the audio and view the animation.
 	The interfaces of the History module. The timeline will navigate the user to the content of the history based on the year selected by the user.	 	The interfaces of the Tones module. It explains the five tones of Pinyin as well as the four rules of writing it. Sound buttons are provided.
 	In the exercise module, sound buttons are provided for all exercises. In the Pinyin exercise, a hint button is provided to pronounce the word slower.		The interface of the virtual assistant module. The user can imitate the sound and input via the microphone button. The result will then be shown in the box.

3.4 Develop Main Function

In this phase, the main functions of the proposed application are developed. It involved developing assets for the application and the integration of the assets into Unity software. There are 4 multimedia elements developed as assets of the application. It consists of audio, graphics, 2D animations, and 3D animations, as tabulated in Table 8.

Table 8: Application assets development

Assets	Development	Description
Audio		The audio files in the Learn PINYIN application are in the Moving Picture Experts Group Layer-3 Audio (MP3) format. It is downloaded from the free online database of GitHub. This is to ensure the accuracy of the pronunciation.
Graphics		Canva platform is used to design the storyboard. Designs are then imported into Adobe Photoshop to extract elements such as buttons, header banners, images, and application icons. The background or artboard of the elements is set to transparent.
2D animation		In the Initials module, initials ‘b’ and ‘s’ use free animations from online sources to display real textures of glass and silk objects. Meanwhile, the initials ‘p’, ‘m’, ‘d’, ‘t’, ‘n’, ‘l’, ‘h’, ‘x’, ‘zh’, ‘ch’ and all finals in the Final module are drawn through Adobe Photoshop with different layers. The different layers of drawings are arranged on the timeline to create the 2D animations.
2D animation		Additionally, Adobe Animate was also used to create 2D animations. It includes initials ‘f’, ‘k’, ‘q’, ‘x’, ‘r’, ‘z’, and ‘c’. Instead of drawing the animation layers by layers, Adobe Animate allows the creation of a simple animation that has a fixed animation path. The 2D animations are saved and imported into Unity software in the Moving Picture Experts Group-version 4 (MP4) format.
3D animation		On the other hand, the initials ‘g’, ‘j’, ‘sh’, ‘y’, and ‘w’ are created with 3D animation. First, a free 3D model is imported from CGTrader into the Blender software. The camera's position and rotation are then animated. This is done by connecting the camera to an invisible cube via ‘parent’ and then set the keyframe for camera rotation to 0° initially and to 360° at the final keyframe. The 3D animations are also saved in MP4 format for use in Unity.

Furthermore, the C# scripts are developed to enable the main functions of the application. Such functions including implement animations, randomize scenes, score manager, and speech recognition. These functions are explained in Table 9. Meanwhile, the interfaces of the developed application are presented in Table 10.

Table 9: Integration in Unity

Functions	C# Scripts	Description

Implement animation (Initials and Finals modules)	using UnityEngine;
	using UnityEngine.UI;
	using UnityEngine.Video;
	public class VideoPlayOnAwake: MonoBehaviour {
	public RawImage rawImage;
	public VideoPlayer videoPlayer;
	void Start() {
	videoPlayer.playOnAwake = false;
	rawImage.enabled = false;
	Prepare Video();
	Play();
	}
	IEnumerator Prepare Video() {
	videoPlayer.Prepare();
	while (!videoPlayer.isPrepared)
	yield return new WaitForEndOfFrame();
	videoPlayer.frame = 0;
	}
	void Play() {
	videoPlayer.waitForFirstFrame = true;
	videoPlayer.Play();
	videoPlayer.isLooping = true;
	}
	}
		The animation is controlled by a C# script named VideoPlayOnAwake. The purpose of this script is to ensure the animation will start with the first frame. Disabling the raw image in the Start() function will prevent the animation from being played before it is prepared. The while loop in the IEnumerator will ensure that the animation to be played is in the first frame. If only the animation is in the first frame, the Play function will display the animation and enable the looping by the 'isLooping' command. This script is applied in both Initials and Finals modules.

Randomize scenes (Exercise module)	private int indexId;
	private string sceneToLoadName = "";
	public static List<string> sceneNames =
	new List<string> {"6.1.2", "6.1.3", "6.1.4", "6.1.5",
	"6.1.6", "6.1.7", "6.1.8", "6.1.9", "6.1.10", "6.1.11"};
	
	public void RandomScene() {
	StartCoroutine(WaitForSceneLoad());
	if (sceneNames.Count == 0)
	SceneManager.LoadScene("6.1.12");
	}
	private IEnumerator WaitForSceneLoad() {
	indexId = Random.Range(0,
	sceneNames.Count);
	sceneToLoadName = sceneNames[indexId];
	sceneNames.Remove(sceneToLoadName);
	yield return new WaitForSeconds(0.25f);
	SceneManager.LoadScene(sceneToLoadName);
	}
		In this script, the sceneNames array contains all the scenes from the Initials exercise. A RandomScene() function is created, which is attached to the On-click function of the next button of each question. This code initially calls the WaitForSceneLoad() function, which allows a scene to be loaded randomly depending on its index in the provided array. It will delete the scene from the loaded array item to avoid having a repeating question. Finally, the new scene is loaded using the LoadScene() command. After executing the function named WaitForSceneLoad(), the code will return to the RandomScene() and compare the number of items remaining in the array. If the count of the array items equals zero, the scoreboard page with the scene named 6.1.12 will be loaded.

Table 9: (cont).

Functions	C# Scripts	Description

Score manager (Exercise and Voice module)	public class scoreManager : MonoBehaviour {
	public Text showScore; public Text showHighScore;
	
	void Start() {
	displayScore(); displayHighScore();
	}
	private void display Score() {
	showScore.GetComponent<Text>().text = CorrectAns.score + "%";
	}
	private void displayHighScore() {
	PlayerPrefs.GetInt("Initialhighscore", CorrectAns.highscore);
	if (CorrectAns.score > CorrectAns.highscore) {
	showHighScore.GetComponent<Text>().text = CorrectAns.score + "%";
	PlayerPrefs.SetInt("Initialhighscore", CorrectAns.score);
	PlayerPrefs.Save();
	}
	else
	showHighScore.GetComponent<Text>().text = CorrectAns.highscore + "0%";
	}
	}
		Furthermore, there is a script named scroreManager that handles the exercise score. The text elements for score and high score are declared initially in this script. Then, the displayScore() and displayHighScore() functions are invoked in the first frame of the scene. The displayScore() function displays the user's current score, while the displayHighScore() function displays the user's high score. Additionally, the displayHighScore() function compares the current score with the high score. If the current score is greater than the high score, PlayerPrefs.SetInt() will replace it and store it using the PlayerPrefs.Save() command.

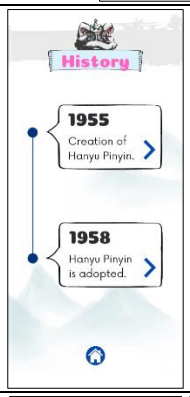
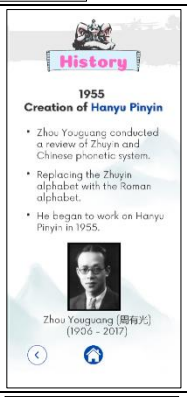
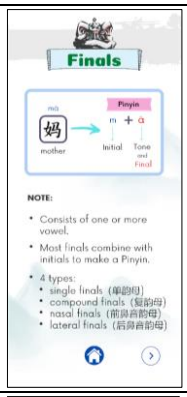
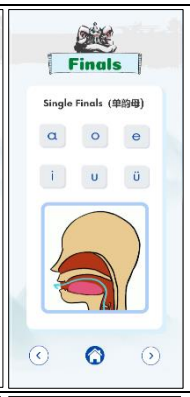
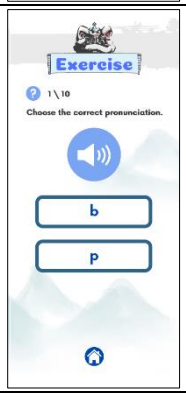
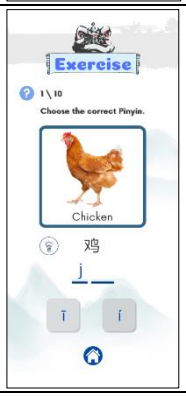
Speech recognition (Voice module)	using KKSpeech;
	public class setLanguage: MonoBehaviour {
	public Text resultText; public Text checkText;
	public Button audioBtn; public Button micBtn;
	public GameObject panell; public GameObject panel2; public GameObject panel3;
	public RecordingCanvas Rcanvas;
	public int attempts = 0;
	public bool isClicked;
	public static int vscore = 0;
	public static int Vhighscore;
	void Start() {
	SpeechRecognizer.SetDetectionLanguage("cmn-Hans-CN");
	panel1.SetActive(false); panel2.SetActive(false);
	panel3.SetActive(false);
	audioBtn.onClick.AddListener(TaskOnClick);
	isClicked = true;
	Vhighscore = PlayerPrefs.GetInt("voicehighscore", Vhighscore);
	}
	private void TaskOnClick() {
	StartCoroutine(DisableMicBtn());
	}
	private IEnumerator DisableMicBtn() {
	micBtn.interactable = false;
	yield return new WaitForSeconds(1.5f);
	micBtn.interactable = true;
	}
	void Update() {
	if (isClicked)
	Check();
	}
		First, 'using KKSpeech' to apply the function from the Mobile Speech Recognizer package. After the declaration of the variables, the SetDetectionLanguage() command is used to implement the voice recognition function from the downloaded asset. Meanwhile, the code "cmn-Hans-CN" indicates that the language used in voice recognition is Mandarin Chinese. Furthermore, the audio button is linked with an on-click function named TaskOnClick. When the audio button is clicked, it invokes the DisableMicBtn function to deactivate the microphone button. The microphone button will be activated again after the audio button has completed its task. Returning to the Start() function, the isClicked variable is set to true, and the Vhighscore is initialized with the PlayerPrefs.GetInt() command. In the Update() function, the Check() function will be invoked if only the variable isClicked is true.

Table 9: (cont).

Functions	C# Scripts	Description

Speech recognition (Voice module)	private bool Check() {
	if (resultText.text.Contains("Sorry") ‖
	resultText.text.Contains("Say") ‖
	resultText.text.Contains("Recognition") ‖
	resultText.text.Contains("pronunciation")) {
	checkText.text = resultText.text;
	return isClicked = true;
	}
	else if (resultText.text.Contains("Try to pronounce")) {
	checkText.text = "Try to pronounce the word :)";
	return isClicked = true;
	}
	else if (resultText.text = "" && checkText.text.Contains("Can't hear clearly")) {
	checkText.text = checkText.text;
	return isClicked = true;
	}
	else if (resultText.text="" ‖ resultText.text == "") {
	checkText.text = "In Progress...";
	StartCoroutine(WaitForResponse());
	return isClicked = true;
	}
	else if (resultText.text.Contains("我")) {
	checkText.text = "Correct pronunciation, well done!"; vscore += 10;
	panel3.SetActive(true); return isClicked = false;
	}
	else if (resultText.text.Contains("偶") ‖
	resultText.text.Contains("哦") ‖
	resultText.text.Contains("窝") ‖
	resultText.text.Contains("握") ‖
	resultText.text.Contains("卧")){
	checkText.text = "Nearly correct pronunciation, try again!";
	Rcanvas.resultText.text = checkText.text;
	attempts++; UserAttempts();
	return isClicked = false;
	}
	else {
	checkText.text = "Wrong pronunciation, \ntry again!"; Rcanvas.resultText.text = checkText.text;
	attempts++; UserAttempts();
	return isClicked = false;
	}
	}
		The if-else statement is used to indicate a distinct output for each input. The input is based on the processed outcome from the voice recognition package. Firstly, if a warning is received, it will be displayed. Secondly, if no warning is received, the statement “Try to pronounce the word” will be displayed. Thirdly, the statement “Can’t hear clearly” will return if an unclear pronunciation is detected. Fourthly, “In Progress” is shown to the user if a long process time is taken. The following condition compares the outcome of the Chinese word to the word pronounced by the user. Consider the Pinyin “wǒ” which means “me”, if the user pronounces it correctly, the statement "Correct pronunciation, well done!" will appear. Additionally, 10 points are assigned, and the pop-up panel is displayed to proceed to the next question. Meanwhile, for the pronunciation that is closed to correct, the message “Nearly correct pronunciation, try again!” will be shown. The isClicked variable is set to false to disable the speech recognition features. On the other hand, for the incorrect pronunciation, the message “Wrong pronunciation, try again!” will be displayed.

Table 10: Interfaces of the developed application

Module	Interfaces	Module	Interfaces
Startup interface		Main menu	
History module	 	Initials module	 
Finals module	 	Tones module	 
Exercise module	 	Voice module	 

3.5 Testing

In the last phase of the MMCD methodology, two types of testing will be performed, including functional testing and user acceptance testing. If bugs are discovered in this phase, the project will return

to the previous phase to update the integrated functions to fix the bugs. The functional testing is presented in Table 11, while the user acceptance testing is discussed in Section 4.

Table 11: Functional testing

Test	Expected Result	Actual Result	Corrective Action
Navigation button	Navigates to each interface.	Works well as expected.	Not needed.
Audio button	Play the audio.		
Close button	Close the pop-up panel.		
Retry button	Retry the exercise.		
Microphone button	Identify pronounced words.		
Animation	Play the animation.	The animation does not play in the first frame.	A command has been added to the code to enable the video only when the video is in the first frame.
Score system	Track the current score and record the highest score of the exercises.	Failed to track the highest score when playing a new exercise.	PlayerPrefs.GetInt() is called in all score managers.
Speech recognition output result	Display the correctness of the pronounced word.	The correctness of the Chinese words pronounced is not recognized.	The scripts have to save manually to Unicode Transformation Format with 8-bit blocks (UTF-8) format.

Table 11 shows that a few errors were spotted. In the Learning module, the waitForFirstFrame command is implemented to avoid the animations that began with a random frame. Furthermore, the PlayerPrefs.GetInt() command is used to overcome the inability of the high score system to record the highest score obtained in the exercises. Additionally, speech recognition was once unable to recognize the correctness of Chinese words according to the script. The Chinese words coded in the if-else statement of the Voice module will turn into undefined characters. To solve this, all scripts in the Voice module must be saved in UTF-8 format. Lastly, an issue occurred in the Gradle file while generating the project's APK. This is due to a file conflict caused by the downloaded voice recognition asset. This is addressed by removing the statement 'android:banner=@drawable/app_banner' from the AndroidManifest.xml file.

4. Results and Discussion

This section presents data and analysis of user acceptance testing. The testing is performed to assess user acceptance of the developed application. In this project, the Technology Acceptance Model (TAM) [17] is adopted and applied. The application was distributed to target users aged 7 to 12 years via Google Drive, along with a Google Form questionnaire. The questionnaire assesses four items: Perceived of Usefulness (PU), Perceived Ease of Use (PEOU), User Satisfaction (US), and Attribute of Usability (AU). Furthermore, a 5-point Likert scale has been applied to the questionnaire with options such as 'strongly disagree', 'disagree', 'neutral', 'agree', and 'strongly agree'. A total of 30 responses were collected and the results were analyzed.

Based on Figure 6(a), an average of 40% of respondents agreed and 36.33% strongly agreed that the information provided in the application is useful to learn Pinyin. However, it also received 1.33% strongly disagreed and 4% disagreed. This is because a very limited set of questions is provided in the application. According to Figure 6(b), an average of 32.22% of respondents agreed and 38.89% strongly agreed with the application's perceived ease of use. The total percentage of agreed and strongly agreed is more than half, therefore, the usability of the Learn PINYIN application is acceptable.

Meanwhile, the statistic in Figure 7(a) shows that an average of 36.67% of respondents agreed and 38.89% strongly agreed that they are satisfied with the overall performance of the Learn PINYIN application. In addition, Figure 7(b) displays that 40% of respondents agreed and 40.67% strongly agreed that the Learn PINYIN application is capable of working effectively. Nevertheless, the results show that 13.33% of respondents disagreed and 3.33% strongly disagreed that the speech recognition features in the Voice module work well. This is due to the performance of speech recognition is easily affected by ambient noise and the quality of the user's phone microphone. In summary, the Learn PINYIN application has completed user acceptance testing with positive results from the target users.

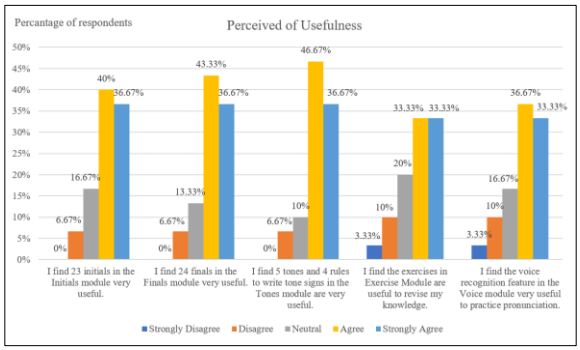


Figure 6(a): Analysis of Perceived of Usefulness

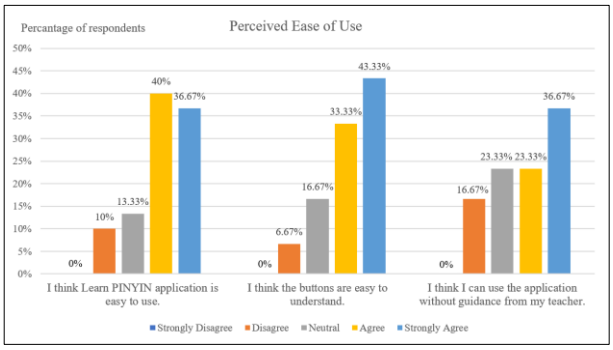


Figure 6(b): Analysis of Perceived Ease of Use

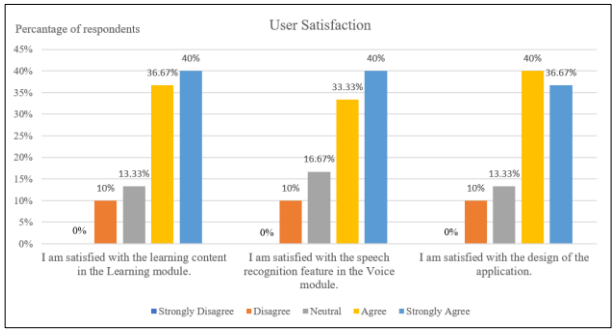


Figure 7(a): Analysis of User Satisfaction

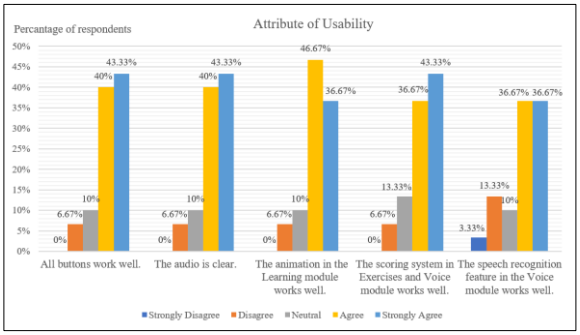


Figure 7(b): Analysis of Attribute of Usability

5. Conclusion

Based on the results analyzed from the testing phase, the Learn PINYIN application is suitable for the target users to learn Pinyin. The three objectives of this project were fully accomplished by first implementing audio and speech recognition features in the application to achieve the visual and auditory learning styles of the VARK system. Second, successfully developing the speech recognition technology that can evaluate the correctness of the user's pronunciation. Lastly, performing the functional and user acceptance testing after the development phase was completed. Multimedia Mobile Content Development (MMCD) methodology assisted this project to be completed on time. Additionally, the advantages and limitations of the Learn PINYIN application are tabulated in Table 12. Last but not least, for future work, it is suggested that all examples of Pinyin's finals can be included in the Finals module. Besides, more sets of questions can be added to the Exercise module and the Voice module, and a Practice submodule can be added for the Voice module. Additionally, a Game module can be added to the application to increase the engagement of the users.

Table 12: Advantages and limitations of the Learn PINYIN application

Advantages	Limitations
• Provides useful information for learning components of Pinyin, such as initials, finals, and tones.	• Does not include all examples of Pinyin's finals.

Table 12: (cont).

Advantages	Limitations
• Introduce Pinyin to beginner Mandarin language users. • Consists of a virtual assistance module that is useful for the user to practice pronunciation. • Compose of a high user acceptance level, where the target users agreed that the application is easy to use.	• Provides only one set of questions in the Exercise and Voice module. • The user could not practice with the application before using the speech recognition features in the Voice module. • There is no game module in the application.

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

Appendix A shows the results of the questionnaires for user analysis and the flowchart of each module.

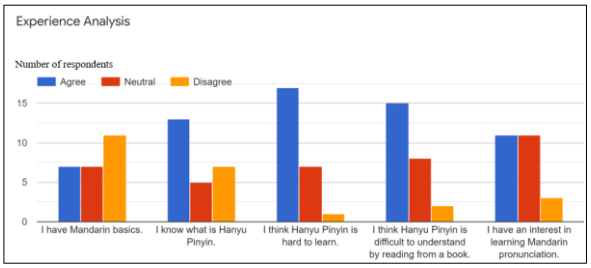


Figure 8(a): User experience analysis

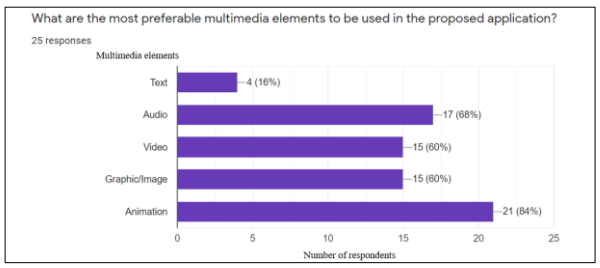


Figure 8(b): Preferable multimedia elements

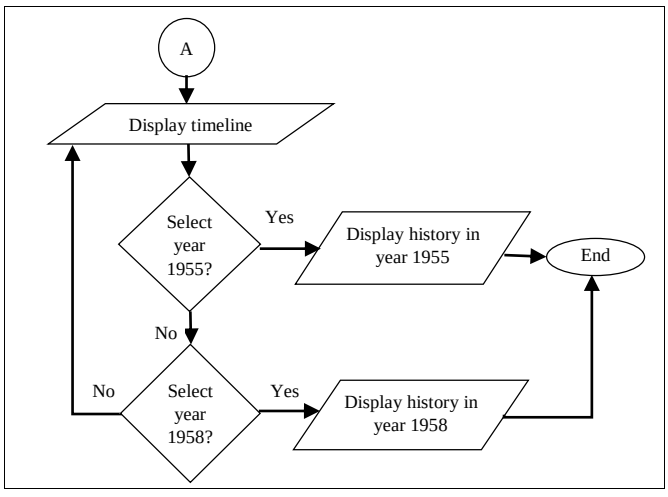


Figure 9(a): History module flowchart

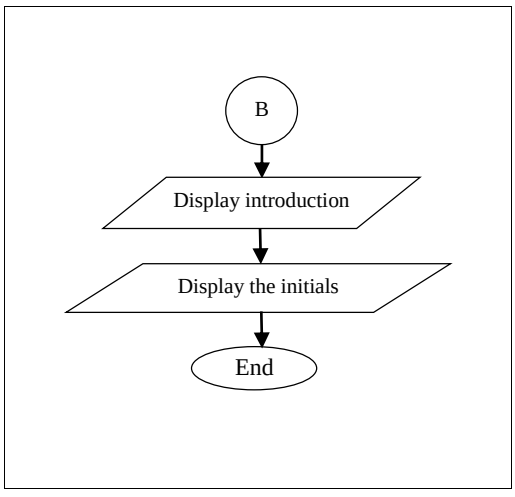


Figure 9(b): Initials module flowchart

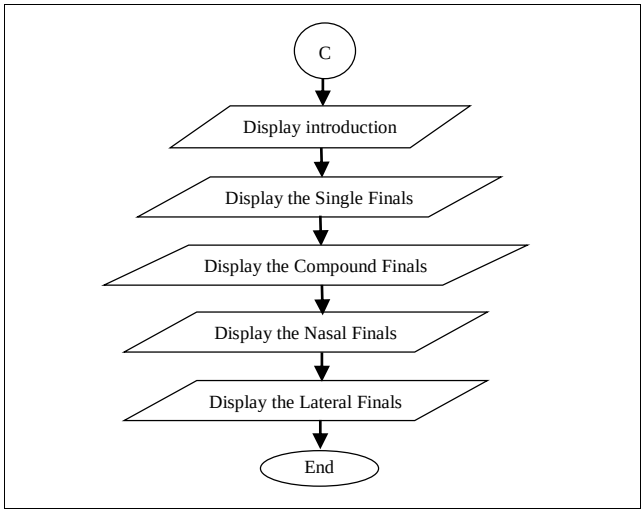


Figure 10(a): Finals module flowchart

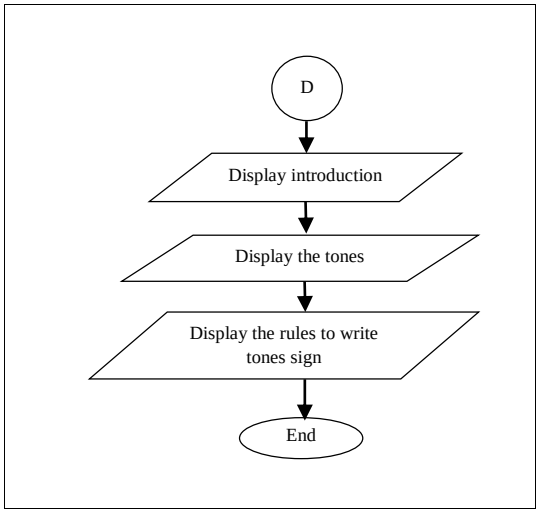


Figure 10(b): Tones module flowchart

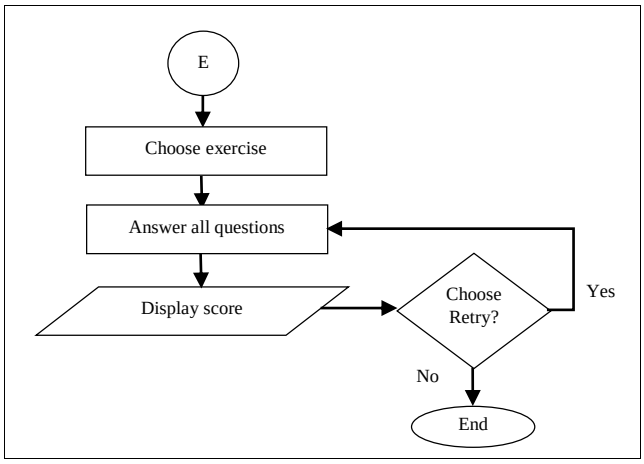


Figure 11(a): Exercise module flowchart

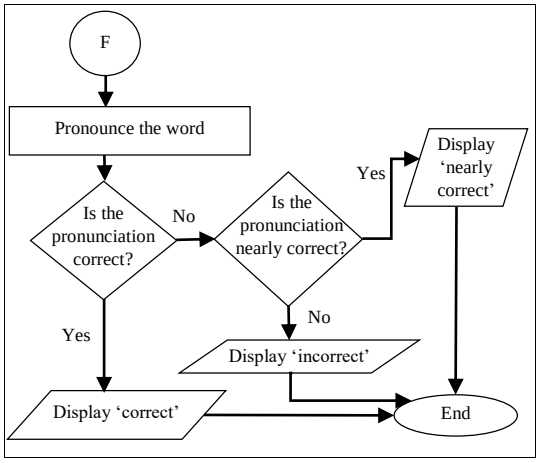


Figure 11(b): Virtual assistant module flowchart

References

[1] I. Ghosh, "Ranked: The 100 most spoken languages around the world," Visual Capitalist, February 15, 2020. [Online] Available: Visual Capitalist, https://www.visualcapitalist.com/100-most-spoken-languages/. [Accessed Jul. 23, 2021]

[2] M. S. Kan and A. Ito, "Language Cognition and Pronunciation Training Using Applications," Future Internet, vol. 12, no. 3, p. 42, 2020, doi: 10.3390/fi12030042

[3] J. Zhao, S. Bhatt, C. Thille, D. Zimmaro, N. Gattani and J. Walker, "Introducing Alexa for E-learning.," in Proceedings of the Seventh ACM Conference on Learning @ Scale. Association for Computing Machinery, pp. 427 - 428, 2020, doi: 10.1145/3386527.3406719

[4] S. Z. Zhang, G. K. Georgiou, T. Inoue, W. W. Zhong, and H. Shu, "Do pinyin and character recognition help each other grow?" Early Childhood Research Quarterly, vol. 53, pp. 476–483, 2020, doi: 10.1016/j.ecresq.2020.06.004

[5] S. I. Odinye, "Mandarin Chinese Pinyin: Pronunciation, Orthography and Tone," The Creative Artist: A Journal of theatre and Media Studies, vol. 11, no. 3, pp. 1-10, 2017.

[6] K. Y. Dennis, "Comparison of Yale and Pinyin Romanizations," in Chinese Romanization Self-Study Guide, Ed. University of Hawaii Press, 2019, pp.1-17.

[7] K. B. Sujit, M. Wotto, and P. Bélanger, "E-learning, M-learning and D-learning: Conceptual definition and comparative analysis," E-Learning and Digital Media, vol. 15, no. 4, pp. 191–216, 2018, doi: 10.1177/2042753018785180

[8] A. Kukulska-Hulme, H. Lee, and L. Norris, "Mobile learning revolution," The Handbook of Technology and Second Language Teaching and Learning, pp. 217–233, 2017, doi: 10.1002/9781118914069.ch15

[9] J. Austerjost, M. Porr, N. Riedel, D. Geier, T. Becker, T. Scheper, D. Marquard, P. Lindner, and S. Beutel, "Introducing a virtual assistant to the lab: A voice user interface for the intuitive control of Laboratory Instruments," SLAS TECHNOLOGY: Translating Life Sciences Innovation, vol. 23, no. 5, pp. 476–482, 2018, doi: 10.1177/2472630318788040

[10] J. Bell, "Voice Recognition," in Machine Learning - Hands-On for Developers and Technical Professionals, 2nd Ed. Hoboken: John Wiley & Sons, 2020, pp. 55-70.

[11] E. Idrizi, S. Filiposka, and V. Trajkovik, "The discourse on learning styles in online education," 2019 27th Telecommunications Forum (TELFOR), 2019, doi: 10.1109/telfor48224.2019.8971204

[12] C. T. Gutierrez, "Auditory, visual and proprioceptive integration as a substrate of language development: Sensory integration and language," Revista de la Facultad de Medicina, pp. 469-475, 2018, doi: 10.1121/1.422788

[13] (2019) Pinyin Drill (Version 1.0.12) [Mobile app]. Retrieve from Google Play Store. https://play.google.com/store/apps/details?id=com.joba.pinyin_drill

[14] (2019) Chinese PinYin. (Version 1.5.7) [Mobile app]. Retrieve from Google Play Store.https://play.google.com/store/apps/details?id=pest.games.ChinesePinYin

[15] (2021) Pin Yin (Version 1.0.1) [Mobile app]. Retrieve from Google Play Store. https://play.google.com/store/apps/details?id=com.tibbytang.android.chinese.pinyin

[16] W. S. N. S. Saifudin, S. Salam, and M. H. L. Abdullah, "Multimedia mobile content development framework and methodology for developing m-learning applications," Journal of Technical Education and Training, vol. 4, no. 1, pp. 15-21, 2012. [Online]. Available: https://publisher.uthm.edu.my/ojs/index.php/JTET/article/view/481

[17] F. D. Davis, "Perceived usefulness, perceived ease of use, and user acceptance of information technology," MIS quarterly, pp. 319-340, 1989.

[18] (2013) Duolingo (Version 5.60.3) [Mobile app]. Retrieve from Google Play Store. https://play.google.com/store/search?q=duolingo&c=apps

[19] (2018) My Talking Tom 2. (Version 3.3.2.2780) [Mobile app]. Retrieve from Google Play Store.
https://play.google.com/store/apps/details?id=com.outfit7.mytalkingtom2&hl=en_US&gl=US

[20] (2018) Speak and Translate Languages (Version 6.6) [Mobile app]. Retrieve from Google Play Store.https://play.google.com/store/apps/details?id=com.speakandtranslate.voicetranslator.allllanguages&hl=en_US&gl